Christopher M. Rosett
Austin Hagerty

Introducing HR Analytics with Machine Learning

Empowering Practitioners, Psychologists, and Organizations

Springer

Introducing HR Analytics with Machine Learning

Christopher M. Rosett • Austin Hagerty

Introducing HR Analytics with Machine Learning

Empowering Practitioners, Psychologists, and Organizations

Springer

Christopher M. Rosett
Comcast Corporation
Philadelphia, PA, USA

Austin Hagerty
Microsoft Corporation
Austin, TX, USA

ISBN 978-3-030-67625-4 ISBN 978-3-030-67626-1 (eBook)
https://doi.org/10.1007/978-3-030-67626-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Introducing HR Analytics with Machine Learning is a book to demystify machine learning for non-statisticians and non-data scientists as well as to explain why and how using machine learning with employee data (and other workforce data) requires special consideration for all professionals, regardless of technical background. We understand that using data to inform decisions about human capital is paramount to the future of industry, but that very few professionals have all the domain knowledge necessary to use data to make informed decisions across the complex employee lifecycle. This is especially true when the insights to inform these decisions increasingly come from the fields of computer science and mathematics.

This book is split into three parts:

Part I: A Model for Quality Analytics with Workforce Data introduces the book: its purpose, the intended audience, and why this book is relevant given today's workforce and technology. It also introduces a framework for analytics which we will use to contextualize and frame machine learning throughout the text.

Part II: Bringing Science, Machine Learning, and Behavior Together will ground readers in many of the important skills and domains needed to conduct Machine Learning well. We will introduce (or reintroduce) the solid-yet-basic techniques for research methods, statistics, and computing in a way that is accessible to all skill levels and familiarities with these topics. Part II brings these domains together and shows how they create many of the machine learning methodologies in use across industries today.

Part III: Getting Started with Machine Learning will, as the title suggests, help you get started. Once we have shown the need for this work and introduced foundational skills, we need to put the behavioral and workforce lenses on the science and show some practical techniques, tips, and tricks to apply machine learning in your organization.

This part begins at the practical crossroad of using machine learning in an applied corporate setting. As we said, there are many unique considerations when using behavioral and employee data, and there are many great learnings from science and history which apply to this frontier. We will review real cases and events across

psychological, commercial, military, and other landscapes and show how some good (and not so good) decisions have impacted people and business, and what they can teach us about doing machine learning with workforce data well.

Part III will also give you advice on project managing machine learning efforts and will showcase some of the techniques you need to begin using machine learning in partnership with your HR Analytics teams.

We hope you enjoy *Introducing HR Analytics with Machine Learning*. We want your experience with the information in this book to be informative and impactful. And to do that with a topic such as this, we have aimed to make the content engaging and readable with memorable frameworks for you to store your new knowledge and skills. We hope you can use this guide to create insights about yourself, your organization, and your industry and keep them with you as you continue to grow the data-based decision-making culture at your organization. Thanks for taking this step and thanks for taking some time to learn with us—we hope you enjoy reading the book as much as we enjoyed writing it!

Philadelphia, PA, USA — Christopher M. Rosett
Austin, TX, USA — Austin Hagerty

Contents

Part I
A Model for Quality Analytics with Workforce Data

Chapter 1
Introduction

Today's organizations cannot help but see the growing opportunities where HR meets data and where data meets mathematics and computer science. Born from the engineering which brought internet search algorithms telling people which websites they want to visit or which products to buy based on their browsing history, companies are realizing the competitive advantage to be realized if only they could provide the power of machine learning to internal human capital decisions. And while these techniques have the power to fundamentally shift how professionals practice all aspects of human resources, it is imperative that we do not put the cart before the horse. From talent acquisition to learning and development to succession planning, the power of person + machine will change how employees experience their organizations and how organizations manage and interact with their people. And it is our responsibility as conscientious, prudent practitioners to make sure we do it well.

So why not just read a book about machine learning or data science? After all, industries have been using data science for years to optimize supply chains, find inefficiencies in production processes, and forecast financials. Data collection and computing power have gotten so good that cars are driving themselves in California and dermatologists are using apps to augment their diagnostic abilities. The math has been around a while; are we not just applying it in a new space?

Yes and no. The advent of machine learning with employee data brings with it new rules of engagement, new statistical considerations, and new ethical, legal, and functional circumstances. This new realm of application requires its own study, just as the application of any tool in a new environment would. Here are two main considerations to which we will pay considerable attention in this book:

1. Employee data is different

   Data collected, stored, and analyzed about employees is special for many reasons. Making decisions about employees from data gathered about their behavior and descriptive characteristics comes with a whole sub-industry worth of considerations. Whether it is understanding the intricacies of behavioral data, leveraging the

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_1

extensive social theory which underpins the science of human behavior, or considering how business ethics and employment law regulate what types of research methods and statistics can and should be used, data about people, and especially employees, comes with new rules of engagement.

This book will specifically review a great many of those considerations and help the reader understand why they cannot simply throw computers and math at employee problems and expect good things to happen. By the end of this book, the reader will understand what makes behavioral data and data about people's characteristics different from other kinds of data, and why that requires special consideration. It will help them be more prudent when collecting data, analyzing data, and especially when making decisions based on data.

2. Most tech experts do not know people data or systems and most people experts do not speak tech

Most people are not computer scientists or mathematicians. However, data science, statistics, and machine learning are making their way into the public eye and into day-to-day work lives in ways that are entirely unprecedented in history. This means that many people who have been uninvolved in computer science and mathematics (or the power they bring to decision-making) are now having these disciplines thrust upon them in ways that were not anticipated and accounted for during formal schooling or training. Furthermore, these kinds of skills are not easy to build informally or "along the way," like a professional might pick up acumen in financial planning or labor relations across a career. For many, it is a muscle which has never been used but now is expected to be exercised daily.

Conversely, professionals well-versed in mathematics and/or computer science are often placed in, or volunteer for, jobs in this uncharted people data territory without the benefits of a career spent learning the intricacies of people science or how human resources fits into and provides benefits for the businesses they serve. This may produce unintentionally myopic professionals—they are good with the proverbial hammer, and accidentally approach everything in their path as if it were a nail.

For our nontechnical readers who work with employee data or make decisions about employees: many wish they could go back to their 7th-grade selves and plead with them to develop a love for numbers and computers. We recognize that math and code are not likely at the forefront of how you would like to spend your Friday evenings. We also recognize that mathematics is a skillset built and retained through practice, and that HR has not historically provided much opportunity. And while other subjects are not based on the development of skills which build on each other, math skills are built and retained over time. The roots of statistics which feed machine learning penetrate all the way back to the basic arithmetic of childhood.

Good news: you need not become a mathlete. This book will not attempt to turn anyone into a statistician. We will not even try to convince you math is fun. Instead, we will strive to build familiarity in this new discipline by introducing and developing basic skills we never thought we would need as HR practitioners. It is important to note here that HR practitioners are not as behind as they might think—HR

professionals' understanding of Human Resources (i.e., how people come together to get things done) matters quite a bit. Understanding functions like talent acquisition, learning and development, compensation strategy, talent management, and others are indispensable to the integration (and therefore application) of machine learning in HR. At the end of the day, we are integrating new technology and process into human capital decision-making. HR is the framework for that equation. Understanding HR before attempting machine learning is as critical as understanding the soil before planting a crop.

For those readers who are highly technical in math or computer science but are new to the world of people/employee data: we know that this is a fuzzy, newly defined space. Process, data integrity, and data governance are improving, but are often poorly standardized. In an industry which is just beginning to scratch the surface of the power data brings to the table, there is a long way to go. The nature of HR work has resulted in databases being inefficiently structured and organized. It has led to systems designed for basic reporting and compliance auditing, not for advanced data science and analytics. HR has built a stadium for ice hockey and is now being asked to host a soccer tournament—sure there are seats, bathrooms, and a parking lot, but after that the similarities fade fast. Further, the HR industry is excited about the prospects of data science (that is, they are excited to host the soccer tournament), but they are not too familiar with how the game differs from ice hockey. Unlike finance, supply chain, procurement, or consumer insights, most of HR is not steeped in the mathematics of business (beyond budgeting and compensation), nor the complex world of database administration, software development, data governance, and other information technology-related domains. Historically, success in HR has been based in understanding how people work and using that theoretical and experiential knowledge to help make better business decisions. And as we eluded to, HR technology was not sophisticated enough to bring any real math to the table. Arithmetic about headcounts, terminations, and budgets was about as advanced as it got. Using the power of data to make better decisions is a new muscle to build; a new domain to master.

That said, the intimate and nuanced world of applied HR is complex. Organizational structure and design, corporate culture, engagement and retention, compensation strategy, and many others are delicate and interconnected sub-ecosystems which come together to create great, or terrible, places to work. Bringing organizational processes together and applying them to groups of people while meeting financial, inter-company political, and regulatory constraints is not easy work and requires years of experience within company and industry. The world of HR is not something data science can brute-force code its way through. Furthermore, because the goal is to design models which deal with the complexities of decision-making, social groups, emotions, and other complex outcomes, isolated projects often have far-reaching impact that may not be readily apparent. Without an understanding of (or at least the knowledge of which questions to ask), machine learning has the potential to fix one problem but unintentionally create three more. These important contextual considerations must never be underestimated and must be harmonized with, not overshadowed by, the power of data science if we are to create impactful, sustainable solutions for the businesses we serve.

So, if you are a seasoned HR generalist/business partner with decades of experience or an HR specialist who is finding themselves increasingly expected to partner with IT, this book will help marry your pre-existing knowledge with a grounding in the principles needed to build functional competence in the domains necessary to begin your machine learning journey. If you are a technical expert in the realm of computer science, statistics, data science, or related field, this book will help set industrial context and important considerations for working effectively within the many realms and constraints of the employee life cycle. And if you are a student just getting started in HR, this book will introduce both sides of this important partnership.

This book will bring these yin and yang together. We will deliver machine learning out of the realm of black magic and into plain English. It will contextualize the science of machine learning in the fledgling world of HR Analytics as it can be applied in the greater context of overall Human Resources. By the end of this book, the nontechnical reader will be able to explain the basic principles of statistics and machine learning and how their application can help augment decision-making capabilities. The already tech-savvy reader will learn where machine learning fits in the world of HR and what considerations are critical to use it prudently and effectively.

Discussion Questions

1. What makes employee data different from other types of data used for data science?
2. What are the two main types of professionals moving into the HR analytics space today? What skills do they bring and which do they need to develop?

Chapter 2
Analytics About Employees

Before considering machine learning within the context of people data, we must first understand the general framework of overall analytics with people data in an applied setting. Since there is no current standard way to organize people analytics at an organization, the "where" and "how" analytics fits will vary from company to company. As a result, so too will the appropriate placement of machine learning efforts.

However, there are practical considerations about people analytics functions which are company-agnostic and must be handled regardless of how you choose to organize the function. We will review some of these because it will help you to consider (a) where your role sits within the ecosystem at your particular organization and (b) where machine learning with workforce data makes the most sense to exist/be developed in your organization.

2.1 Analytics Versus Digitalization: Three Lenses

Human Resources can use Machine Learning to bring a competitive advantage to organizations in many ways. Before getting deep into these concepts, we would like to call out three major categories in which machine learning adds value in the human capital decision-making and employee experience space. This text is aimed at specific parts of the overall landscape, so we would like to define our focus so readers may follow up with additional topics that may interest them.

1. *Hypothesis Testing*: Using machine learning to test theories of underlying principles driving business outcomes
2. *Forecasting, Prediction, and Simulation*: Using machine learning to simulate future states in order to influence decision-making to optimize future business outcomes

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_2

3. *HR Digital Transformation*: Using machine learning to automate processes and to optimize workflow and infrastructure in an effort to create efficient systems and improved employee experiences

2.1.1 Hypothesis Testing, Forecasting, Prediction, and Simulation

The first two on the list are what will be the primary focus of this book. If you would like to use machine learning for advanced HR analytics, or in any way leverage machine learning as a tool to aid in the investigation of past, present, or future business outcomes, this book is for you. We will review the scientific method, research methods, basic statistics, ethics and legality, introduce social psychology theory applied at work, machine learning techniques, machine learning project management and best practices, and many other topics which will help you on your way. This book is fundamentally about exploring HR data with machine learning and how that can improve investigation and decision-making in your organization.

2.1.2 HR Digital Transformation

Across industries, Digital Transformation has become the term for using digital technology to improve business processes. It is a hugely advantageous endeavor and all companies are invested on some level. Whether a company simply uses laptops and a website or has sophisticated, cloud-based solutions managing everything they do, every company is invested in the digital age and how its technologies enable them to do things faster and better than yesterday. Recently, this idea of Digital Transformation, or digitalization, contains as part of its jurisdiction the use of advanced computing, including (but not limited to) machine learning. In this capacity, machine learning is being used to automate processes, optimize choices for end-users, and just generally grease the gears that make businesses run.

In HR this translates to creating curated, seamless, and thereby exceptional candidate and employee experiences through the use of these new platforms and technologies. And modern HR Information System vendors have gotten on board. Over the last several years, they have been very successful in convincing companies that it is time for them to move from their old, transaction-oriented systems to new systems which have features claiming to be able to automatically find the next role for employees, automate transactions, smooth the headaches of HR transactional work, and produce customized experiences—all while creating an integrated, mobile-enabled experience for everyone who touches the system.

Some of the digitalization behind this is simply better software engineering and design. In the 10–15 years since companies installed their last HR system (let alone

the multiple ancillary systems they purchased and integrated over that time), the ability to store more data, refresh it faster, standardize it, and secure it across more diverse environments has gotten better. Processes like promotions, hiring, terminations, payroll activity, and the like are just easier for the user in these new systems because the IT team is managing more of it behind the scenes.

However, machine learning is also playing a role. Let us take a piece of machine learning we are all familiar with as consumers: predictive purchasing. Every website you buy things from will tell you after a purchase (or right before a purchase) "people who bought this also bought," and then give you a list of things that you might like to buy based on the contents of your cart. They also use that data to advertise. Have you ever noticed that after you buy something, similar product advertisements end up on your social media, video website advertisements, and any other website with banner ads? This is machine learning at work. They correlate buying behavior from their giant datasets and use that to target their marketing to you.

In the future, these same sorts of machine learning algorithms will help employees find their next role, target training that is specific to their desired career path, and automatically remind them when a critical core process is coming due. They will talk to employees via chat or on the phone to help them solve their benefits problems, or answer payroll questions. They will optimize call routing for HR shared services so employees get to the right human with fewer prompts.

Many of these technologies still have a long way to go to be truly effective, but in this way machine learning is being used to drive improved employee experience through higher levels of automation and customization. This is a critical part of how machine learning is impacting the future of work and the future of employees' relationships with their organizations. That said, it is separate and distinct from the major topic of this book, which is how we use machine learning to enhance the maturity of our HR analytics endeavors such that we can investigate and solve workforce problems with greater scale and impact.

2.2 Types of Analytics: Descriptive Versus Prescriptive and Predictive

An important first thing to clarify about analytics, machine learning and otherwise, is that they are not created equally. To the nontechnical person, numbers and data are often homogenized into one type of work which, at worst, helps them get their point across and at best provides evidence which helps them make a better business decision. But even on that spectrum, there are so many ways analytics comes to life that a basic understanding of the types of analytics is important to set context. And while there are entire books written on these categories individually, and numerous different opinions about how to label them, we would like to ground ourselves across three major categories of analytics HR teams are doing (or are being asked to do) at their organizations:

2.2.1 Type 1: Descriptive Analytics

Descriptive analytics[1] are exactly what they sound like: analytics which *describe*. Often this type of analytics is called "reporting" because that is really what it is: reporting information to a user. If you have ever produced a headcount report, span of control analysis, or turnover rates, you are familiar with descriptive analytics. Their purpose is to provide information to someone who needs it, without (and this is important) context, commentary, or other forms of interpretation. If one begins to add *reasons* why the data looks the way it looks, we have moved beyond the world of descriptive analytics.

Describing, not analyzing, is the key feature of this type of work. In fact, many analytics practitioners attempt to separate this type of work from "analytics" entirely, since there is not any "analysis" involved. The reason we label it this way here is that the terms "reporting" and "analytics" are so intermingled in common usage and the work is so closely tied together at most organizations it is simply intuitive to describe it this way. That said, even though the separation does not truly exist (yet), we do advise you to separate this work if you can, at least in name. Calling this type of work "reporting," and even centralizing it organizationally if possible, makes a lot of sense especially if a group exists to handle HR operations, HR transactions, or other standard and central such functions. Typically, 80%+ of this kind of work can be standardized across an organization with an alignment effort that is well worth the time.

Regardless of what it is called, within organizations this function is usually the most mature, and at big organizations it often already sits within the aforementioned centralized group, sometimes called "HR Operations," or "Shared Services." They are teams staffed with reporting experts or people who know how to use the systems to get information into a spreadsheet, pdf, or dashboard format a user can access. They also typically have a close relationship with the HR Information Systems team, which sometimes even shares leadership with the HR Ops team or reports directly into the IT function. Either way, they are the gatekeepers between users and questions like "how many people quit last month" or "how many people are on the sales team in the Atlanta office?"

Though this is the simplest form of working with people data, it is also the most important for three main reasons:

Descriptive Analytics are Critical to the Day-to-Day-Functioning of Your Organization: The reason this is the most mature part of most people analytics teams is because it has been around the longest. And the reason it has been around the longest is because it is necessary to the functioning of every business. Dozens of financial, legal compliance, compensation, talent acquisition, and other decisions

[1] Descriptive Analytics is not the same as Descriptive Statistics, which will be discussed in detail in Chap. 6.

are made using these data every day and so people exist in your organization to deliver said data. If you are in HR, think about the one person in your group you cannot live without. Then ask them the one person *they* cannot live without. Then ask *them*… and so on. You will not get more than two to three names before you run into someone on one of these teams.

These Teams are the Cornerstone of Data Integrity: Whether the data quality in an organization is strong, weak, or somewhere in the middle, it is managed by the people who work with these data every day. To get data into the hands of someone who needs to use it, first the data must come into the system through processes of varying quality and governance, then be calculated, manipulated, and stored in the systems by IT teams, and finally be queried by reporting professionals to get it out of the system and on your desk to answer the question of the day. If data is like water flowing through pipes, these reporting teams are the organization's plumbers. And without good plumbers, nobody gets good water (or water at all).

This is one of the first hallmark points in this book: *quality, accessible data is the most critical aspect of machine learning and every other kind of people analytics anyone will do at any organization*. The more work done to increase data quality and sustainability, the easier of a time they will have getting quality insights when they mature to analytics like machine learning. Ironically, this is also the step most often overlooked because it is unglamorous, expensive, and often a change management nightmare. And to make matters worse, it produces little *direct* return on investment. But when done well, this work pays incredible dividends. It is the equivalent of ensuring to pour a quality foundation for a house before building the frame.

Descriptive Analytics is the Gateway Through Which All Users Pass on Their Way to Understanding more Mature Analytics: Most are likely reading this book because either (1) they already love analytics, (2) they think it matters enough that they must learn to use analytics, or (3) someone is telling them it is necessary to learn analytics. Whatever the case, when someone first gets into analytics, descriptive data is where they start. In fact, one of the key differences between an average user of analytics and analytics professionals is when they learned (and how well they understand) descriptive analytics. We will discuss later how descriptive statistics is a cornerstone of statistics savvy, but here it makes sense to mention that in the same way, descriptive analytics is a cornerstone for general data literacy.

Descriptive analytics is the foundation for all other forms of analytics because it demands good answers to simple questions. If an organization cannot agree on how many people quit last month because they have different definitions of turnover or different systems of record, how can they possibly hope to create quality data to feed into a predictive algorithm? The infrastructure and governance which are demanded to have solid descriptive analytics is an indispensable first step to machine learning with any kind of reasonable quality and scale.

2.2.2 Types 2 and 3: Predictive and Prescriptive

Once an organization has good data to use, they can open themselves to more advanced forms of analytics: namely predictive and prescriptive. They are related, but distinct ways to use more advanced research methods and statistics to aid decision-making. Note: one does not necessarily come before the other and are often somewhat interdependent.

Predictive analytics are exactly that: attempts to predict what is going to happen before it happens. Whereas descriptive analytics is concerned with what *is* and what *was*, predictive analytics is concerned with what *will be*. Essentially, predictive analytics seeks to (1) extrapolate from what is known about the past and (2) integrate it with what is known about the future to create an inference the research team has confidence standing behind. Common places organizations might see predictive analytics with employee data are areas like turnover or sales performance.

Prescriptive analytics, on the other hand, is more opinionated than descriptive and predictive. Whereas descriptive and predictive talk about states of being (what "was," "is," and "will be"), prescriptive analytics asks, *so what?* Descriptive and predictive might tell a team that they lost 2.3% of their engineers last month (descriptive), and even worse that they will lose 2.5% next month (predictive), but *what are they supposed to do about it*? This is where prescriptive analytics adds value.

Prescriptive analytics attempt to do two things. First, they attempt to explain *why* something is happening. This is where the true nature of science comes into analytics because it is concerned with cause and effect. We will talk more about the scientific method in Chap. 4, but for now think of prescriptive analytics as the methods used to explain why something happened (or will happen) and, more importantly, what can be done to influence that outcome.

Second, a big part of prescriptive analytics is the art/science of simulation. Simulations can be very simple: "If Kim has 100 people, then she hires 5 and loses 3, she will have 102 people at the end of the month. What would happen if Kim hired 10 and lost 8, or hired 4 and lost 7…" Simulations can also be complex: " what will the impact of this multi-million-dollar acquisition have on our frontline retention rates and how will that affect the stock price?" And everything in between. What these examples have in common is that they provide a testing ground for premises you assume to be true. If Kim can set up a particular set of conditions (headcount equals 100), and then simulate some set of circumstances (hire 5, lose 3), what will the resulting set of conditions be (headcount = 102)? Then Kim and her business get to decide whether they like that outcome, whether they think the circumstances are accurate, and how they want to run the test again. This example is oversimplified, but the idea is that one of the huge advantages of more advanced analytics (of which machine learning is a piece) is this ability to iterate on potential futures.

This is where it makes sense to explain why predictive and prescriptive analytics are so intertwined (and where machine learning fits). Techniques like forecast mod-

eling, which are fundamentally predictive, are often done well when you understand why your outcomes are the way they are, which is more prescriptive in nature. Conversely, techniques like root cause analysis, which are more prescriptive, typically need a predict-then-test aspect to them if they are to be fully validated (which starts with good predictions). In many ways, prediction and prescription are two sides of the same complicated coin.

Practically speaking, using predictive and prescriptive analytics is difficult in different ways than descriptive analytics. This is important when considering how and where to put different types of analytics teams in an organization. The challenge in descriptive analytics is usually in building quality infrastructure and governance around data ingestion, storage, manipulation, and usage. This is not mathematically or methodologically complicated. If an organization can pair a good IT team who has quality computer science and database administration skills with a solid project and change management team who knows the business, then getting to quality descriptive data is a matter of prioritizing it and getting it done. In fact, most organizations are staffed to do it today. What they typically lack are (a) making it a true priority (i.e., true top-down sponsorship and accountability for progress), (b) funding the changes in tools and infrastructure, and (c) holding end-users accountable for assimilating to the new norms once the changes are made.

On the other hand, predictive and prescriptive analytics requires qualitatively different skills that are not often present in HR or IT organizations today. A team with an understanding of advanced statistics and research methods, paired with relevant business acumen and/or computer science skills is not easily found within the ranks of most companies. And if it is, it is usually siloed: Business Intelligence may have the stats and computer science, but not the HR business skills. HR may have the HR business skills and people-science acumen, but not understand the realities of IT. We will return to this concept several times in later chapters since it is critical to the success of building a team that can create and execute machine learning projects. Suffice it to say here that the organizational capabilities for basic versus advanced analytics vary a great deal, but are both critical in their own ways to overall success.

2.3 The Employee Lifecycle and Where Its Data Lives

In HR, the "Employee Lifecycle" is a familiar term. If it is a new concept, then think about any lifecycle. "Lifecycle" insinuates the creation, maintenance, and end of something: customers have lifecycles, products have lifecycles, sales processes have lifecycles, budgets have lifecycles—and so do employees. There are many ways to bucket the employee lifecycle, but here is the one we will use for this book, along with some brief definitions:

- *Attract and Select*: Activity related to the attraction of talent to the organization, or specific groups within the organization, and selecting talent to fill particular roles
- *Onboard and Assimilate*: After selection, activity related to bringing talent into a team and ensuring their effective connection to the people, tools, and processes they need to succeed in their role
- *Engage and Reward*: Activity related to helping managers and individuals monitor and build positive affect of employees toward their coworkers, job, career, and organization
- *Develop*: Activity related to helping employees build their capability to help them realize their fullest potential
- *Advance*: Activity related to moving talent around an organization for the betterment of their careers, the teams they are on, and the organization overall
- *Separate*: Activity related to leaving a team or the organization overall due to employee volition, involuntary termination, retirement, or other events causing separation between company and employee

Across these parts of the lifecycle, organizations often create groups of employees focused on these areas specifically. For example, companies may have a "Talent Acquisition" or "Recruiting" team, dedicated to the attraction and selection of talent. If an organization is big, it may even have a group dedicated to the onboarding and assimilation of people once they are hired (though often this part is jointly managed between recruiting and the hiring manager). Most organizations also have teams (or single employees in small organizations) dedicated to employee engagement, talent management, and learning and development. All these groups are dedicated to their special area of the employee lifecycle. It is an important ecosystem which exists in every organization, even though the extent to which it is cared for varies by organizational size and strategic prioritization of the employee experience.

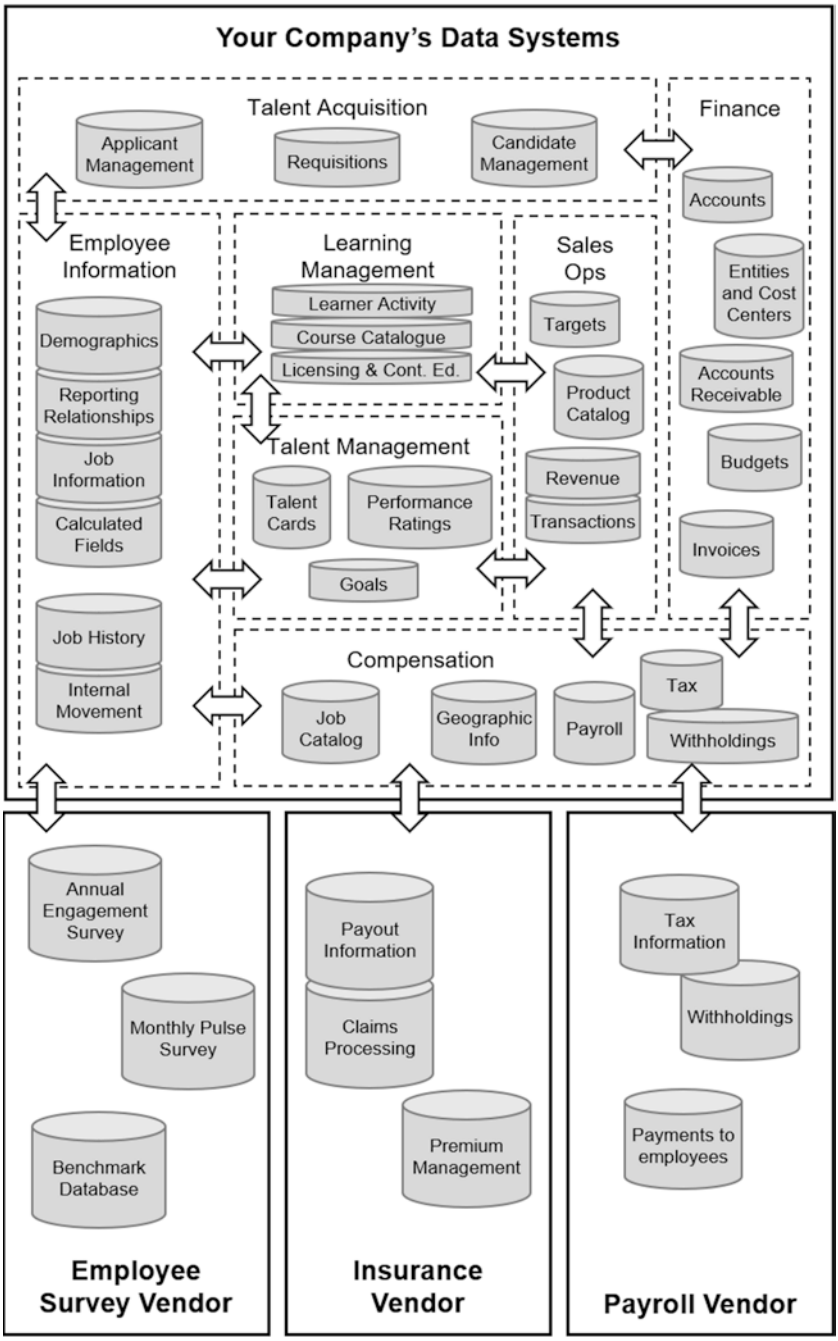
Across these different parts of the employee lifecycle, organizations have many different kinds of data which is created and stored. In its simplest form, these differing kinds of data are information which has been input into a system by an HR employee (e.g., hire date), a business leader (e.g., performance rating), or created/calculated by a system (e.g., employee tenure). After entry, the data are organized and stored for later use. Not unlike spreadsheet programs, the organization of these data is done in big tables that can be accessed, combined, and transformed in many ways. When a large group of these tables is together, it is called a database or a data warehouse. These warehouses are usually big groups of data which are stored, related, and secured by category (i.e., based on what kind of data they store).

The employee lifecycle is a good way to think about those categories and how these data are divided up and stored. Applicant Tracking Systems, Candidate Management Systems, Performance Management Systems, Learning Management Systems, Human Resources Information Systems (HRIS) are examples of such interfaces, each with its own specialized design to organize the data it contains. But at the end of the day, it is helpful to think about these "systems" as just data sources for a particular part of the employee lifecycle.

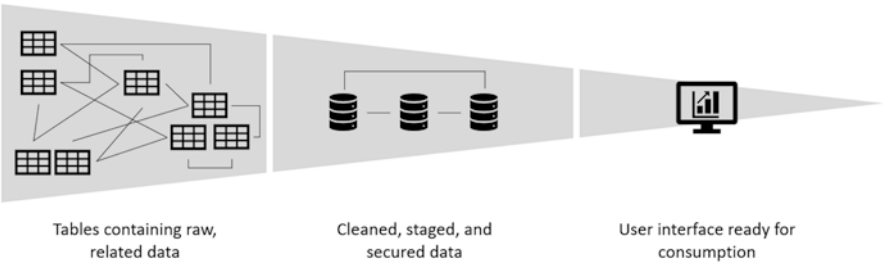
Organizations maintain these databases because they need to store information about their applicants, candidates, and employees, so they can monitor the lifecycle and make it better. They track who is applying, who gets hired, and once they are hired how their employees change over time. In an applicant tracking system this often looks like tables organized by applicant ID (who is applying) and by requisition ID (what job they are applying for). In an employee data system, they track employees over time and how they "change": a new job code from a promotion, a change of address, or a new boss. Other databases organize data differently, like learning management systems where training data is kept based on what courses are available and who is registered, in progress, and complete within them. Or talent management systems where information about employee potential and succession planning is housed; compensation systems where specific information about payroll and taxes are kept; or benefits systems where information about insurance plans, claims, and other data are stored.

Organizations also store data externally with vendors. For example, the employee engagement part of the lifecycle concerns data with high levels of confidentiality. That is, if a team wants to track employee sentiment data from an annual employee opinion survey or a monthly pulse survey, many companies opt to store it with a third party, so employees may feel extra confident that their responses are being kept confidential. The tables with responses to surveys are housed on servers that the organization does not have direct access to. They use contracts and service level agreements to maintain the flow of that information into the organization.

Across all these systems, the degree to which the data is validated, standardized, and can talk to other systems varies greatly from organization to organization. But to illustrate, here is what it might look like at a generic organization:


Your Company's Data Systems
Talent Acquisition
Applicant Management
Requisitions
Candidate Management
Finance
Accounts
Employee Information
Demographics
Reporting Relationships
Job Information
Calculated Fields
Learning Management
Learner Activity
Course Catalogue
Licensing & Cont. Ed.
Talent Management
Talent Cards
Performance Ratings
Goals
Sales Ops
Targets
Product Catalog
Revenue
Transactions
Entities and Cost Centers
Accounts Receivable
Budgets
Invoices
Job History
Internal Movement
Compensation
Job Catalog
Geographic Info
Payroll
Tax
Withholdings
Annual Engagement Survey
Monthly Pulse Survey
Benchmark Database
Employee Survey Vendor
Payout Information
Claims Processing
Premium Management
Insurance Vendor
Tax Information
Withholdings
Payments to employees
Payroll Vendor

There is an additional layer to these data systems we would like to introduce as well. Most people do not access the data from these systems in their purest form. In fact, unless an employee is on an IT or HRIS team, they have probably never seen the view of these systems which look much like a database at all. Most employees who access these data do so using a web-based application where they can use their mouse to leverage drop-down menus, dragging, and radio buttons to select which data they would like to see and have spreadsheets, visuals, or a report generated for them. This is called the "presentation layer" and is what systems use to make their information more easily accessible to nontechnical users. Think about it like this: when someone goes into their favorite department store to shop for clothes, do they walk right into the back rooms and rummage around in the boxes and shelves to find what they want? Of course not. There are teams of people (merchandisers) who's entire job is to take things from the back room and set up the clothing to be browsed and selected easily. That is the purpose these presentation layer interfaces serve. In fact, there are often multiple layers of code, security, and transformation between the data users see, and how it comes into the system. A simple way to think about it could look like this:



The point is, when thinking about employee data it is best to always consider it in these three main ways:

1. The employee lifecycle: Most data created about employees is intended to quantify some aspect of the employee's journey through their time at the company.
2. Many warehouses with varying levels of connection: These data sit in many different places and have varying amounts of connection to other HR and business data.
3. Many layers between true data and data access: Within the warehouses, there is a lot of work done to maintain access for nontechnical users.

Section Breakout: Illustrating Common Challenges

Diverse storage according to the employee lifecycle and varying connectivity between warehouses are concepts aimed at introducing you to some of the complexities of how analytics can help solve problems, but also the environment in which analytics professionals must operate to bring solutions to the table. It can be challenging to create and maintain data within a single system, let alone across the five or more most organizations have. Data input standards,

refresh rates, reporting system limitations, and many other factors can impact how difficult it is to get good data out of a system. But how about some examples of real situations employees in real organizations may have experienced? Here are just four simple examples of issues that could arise within employee data from different systems:

Applicant Tracking System:
John just got approval to hire a new analyst. He is very excited and opens a new requisition (official job opening) and routes it to the proper business and finance leaders.

Unfortunately, before the hire can happen, the budget needs to be allocated elsewhere and the hire does not occur. That requisition is left open and lives on in the system as a job opening which is not really open anymore.

Compensation System:
The research and development department has decided that they no longer need to hire scientists who are experts in 1.0 Widgets. They have a whole department of Widget 1.0 experts with the job code W10ABC. However, the work on the team has advanced such that they all now work on Widget 1.1.

The leaders decide that they need a new job so that when they hire people into this department, they hire Widget 1.1 experts. Widget 1.1 experts have slightly different skills and need to get paid more (Widget 1.1 experts are rarer and harder to hire). They hire a few new folks with the new job code W11ABC, even though they are doing the same job in the same department as people with the W10ABC job code.

Talent Management System:
Mary is a tough grader. She leads a group of 30 people and makes sure that her whole organization is stack ranked every year. She instructs her managers to use their 5-point rating system so that 5% of people get a 5 (the best), 5% of people get a 1 (the worst), 50% of people get a 3 (the middle), and the remaining 40% get split between 2 and 4.

Joe, her peer who also leads a group of 30, thinks that his managers should rate his group relative to the whole organization, not just his team. He instructs his managers to "give them a 5 if they deserve a 5, and a 1 if they deserve a 1." These two different philosophies make the data very different between groups.

Employee Data System:
Tony, an HR partner who processes employee transactions for the group of 400 employees he supports ensures that when someone is promoted, he codes the movement as "Promotion" in the "Action Reason Description" field of the system. Last year, the team that maintains the system created a new action reason called "In-Line Promotion" for people who get a promotion linked to an expanded role within their current job, rather than moving to a new job. The communications were not clear as to when something should be "Promotion" versus "In-line Promotion," so Tony does not really use the new indicator. In addition to the lack of understanding, changing to the new coding would mess up all his saved reports.

Through just these few examples (and there are an infinite number of others) it can be seen that good, consistent data is not easy to obtain. What do organizations do? How do they handle and mitigate all the problems with employee data?

Because it is so challenging to manage (and many other reasons), it is easy to see why most organizations focus on simply maintaining descriptive analytics. This typically includes creating aggregated subtotals in the form of sums, counts, and averages and then reporting those data over time (i.e., "trending"). As we said before, doing analytics this way is indeed a place to start and can help an organization answer all kinds of questions like:

- How many people did we hire last month?
- How many job openings do we have?
- How much do we pay this group of engineers?
- How many hi-potentials do we have?
- Where do we have the most employees?
- How many employees quit last quarter?
- How many hours of training did we complete last year?

And while these questions are simple (and again, there are an infinite number of other questions), they are fundamental and foundational to understanding the human capital within an organization. Beyond getting the right data and cutting it up every which way (which we have already called descriptive analytics), organizations can use this type of investigation as a springboard to learn advanced ways to describe an organization through application of descriptive *statistics*. This branch of mathematics is critical to doing basic analytics well and must be handled before more advanced analytics like machine learning can be effective. It is nonnegotiable groundwork because the skills are necessary to execute advanced analytics well, and also because there is no substitute for knowing an organization's data with an intimacy only descriptive statistics can provide. Every dataset came from somewhere and its history, trials and tribulations, and evolution help the data scientist understand what can and cannot be done with it. We will introduce and discuss descriptive statistics in Chap. 6.

2.4 Where Analytics (And Therefore Machine Learning) Lives in Organizations

Now that we understand which systems house data, and how, at a high level, types of analytics exist, the next question is: where in the organization do analytics teams sit? Who owns analytics about employee data and where does it live on the org chart? Across the industry, it is easy to see self-proclaimed "people scientists," "people analytics leaders," and many other permutations generalizing the traditional realm of employee data analytics into the modernized category of "people analytics." This question does not have a correct answer, because where this type of analytics sits is usually up to the discretion of the senior leader creating or sponsoring

the team. This means the team can sit anywhere from business intelligence under a Chief Operating Officer, to finance under a CFO, to human resources under a CHRO, or even be entirely decentralized and live within the business units they support.

Dubbing the work "people analytics" has other advantages as well. In addition to putting the work in different places organizationally, thinking about workers as "people" is always a good thing, and it has worked wonders to help get non-HR people interested in employees. Similarly, using the unsullied term "people analytics" or "people leader" has really helped general business folks realize that taking care of employees is not just a feel-good exercise. This approach figuratively separates the work from the less-beloved concepts of "performance development," "talent management," and other terms which have traditionally made managers roll their eyes and sigh about "the stuff they have to do instead of their *actual* job."

And though this rebranding has provided a new light in which non-HR businesspeople can view employee data, it has in some ways divorced the exploration of human capital from the human resources department (i.e., the department *fundamentally focused on employees*). For decades HR has been widely unloved by organizations who maintain the existence of their department for legal compliance reasons. Look no further than the media portrayals of HR in productions like The Office, Up in the Air, or Office Space to see how society has typecast the Human Resources function. "HR people" do hiring and firing. They hand out paychecks, enforce policies, and are the only people on earth who actually understand how benefits work. Also, they keep the team out of court.

To HR practitioners who know that doing HR well provides a competitive advantage, this administrative and transactional view has always been a headwind. They know that when professionals use information about how people work, what people respond to, and how to set up teams, processes, and organizations to facilitate success, things get better. When organizations pay attention not only to how business processes drive revenue and operating costs, but also how they drive people outcomes like development, performance, and turnover, things get better. At the end of the day, organizations are just groups of people aimed at a common objective. And those groups of *human resources* are the only way work gets done. The name is not a coincidence.

Therefore, data analytics which improves the employee experience, the employee–employer relationship, job satisfaction, turnover, or any other human capital-oriented outcome should be considered HR territory, or at the very least closely aligned to it. Organizations with mature HR teams have practitioners who understand how people work, how to capture those emotions and behaviors in data, and then turn that data into information and competitive advantage.

This call out of where HR Analytics belongs is not ceremonial. Over the last several years, companies have started to realize and get on board with the fact that HR done well provides a competitive advantage, and HR Analytics is a big part of that realization. This has led to jobs, teams, and indeed entire sub-functions dedicated to the concept of "HR Analytics." But it has also led to investment in the development of better, more strategic Human Resources functions in general.

Upgrading Talent Management to get more from an internal talent marketplace, increasing efficiencies and strategies in Talent Acquisition, and taking the measurement of employee sentiment more seriously are just a few examples of industry trends which speak to this shift. Additionally, we would be remiss to exclude how many organizations are totally overhauling their HRIS systems as well as hiring the dozens of new HR technology vendors to this common end: "If we do HR better, we can get tangible business value from it."

In this light, we are going to call analytics which are specific to the domain of workers and the employee lifecycle "HR Analytics." And while organizations' design choices, industrial context, and leadership philosophies will continue to cause HR Analytics to live in a variety of organizational locations, we *can* say that this work should be closely connected to, if not embedded within, the Human Resources department. That said, if an organization's HR department is totally separate from its organizational design and development work, separate from its employee lifecycle specializations, or separate from its HR Analytics, then those teams may have a larger conversation brewing around how HR adds value to the organization and where does its maturation fall on the organization's priority list.

Discussion Questions

1. What is digital transformation and how is it different than HR Analytics? How does machine learning stand to impact both?
2. What are the main differences between descriptive, predictive, and prescriptive analytics?
3. Why is descriptive analytics the most important form of analytics in HR?
4. What is the employee lifecycle and how does it impact the data ecosystem at most organizations?

Chapter 3
HR Analytics Ikigai

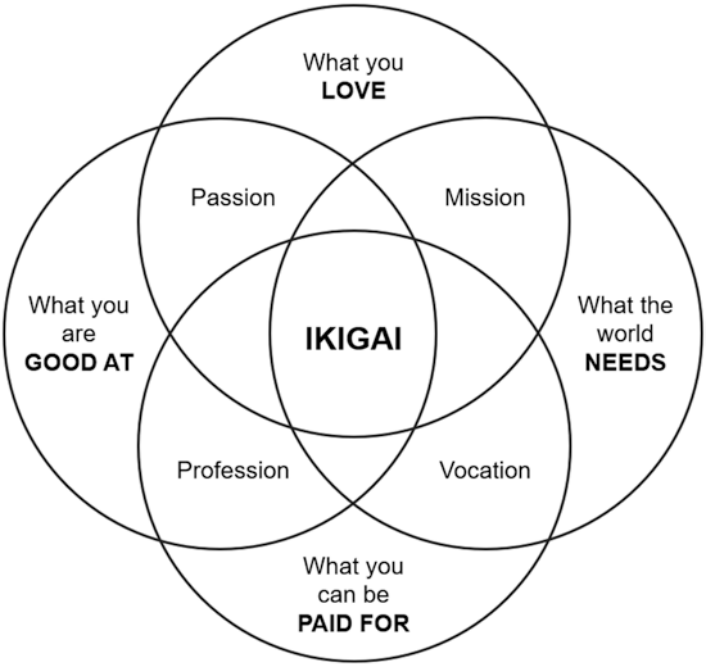
At this point, we would like to zoom into the world of analytics and further explore how machine learning fits into the space. To do that, let us start with the question, "what does it take to do analytics well?"

We will begin with the concept of Ikigai. Ikigai is a Japanese philosophy that roughly translates to "reason for being." In Japanese culture, and increasingly in other parts of the world, people use Ikigai to help guide them toward what they should do with their lives. It combines four distinct, but equivalently important ways to describe an activity:

1. I am good at this
2. Doing this earns me a living (monetarily)
3. I love doing this
4. The world needs this

Overlay these concepts in a four-way Venn diagram and it is easy to see an interesting rubric for thinking about how we spend our time. Almost magically, it casts a new light on everything we do from our job to our favorite hobby:

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_3



As an example, think about the prototypical musician. Aspiring artists love writing and performing their music. They are often very talented, and the world certainly needs artistic expression. That said, the vast majority cannot get paid to perform consistently enough to make a living at it.

Conversely, we might consider an example of a person who is good with numbers and has a knack for understanding tax law. And while there is certainly a need for accounting professionals, this person might not truly love the day-to-day job of being a certified public accountant.

The musician and the accountant bring a great deal to their lives and to society with their chosen work, but we can see with Ikigai how they both lack total fulfillment—the musician has trouble making ends meet, which can cause uncertainty and stress while the accountant does not feel emotionally fulfilled, despite a comfortable and stable lifestyle. To create this balance and fulfillment in their lives, the accountant may play in a band on the weekends, while the musician may have a second job with hours allowing them to rehearse and make their gigs.

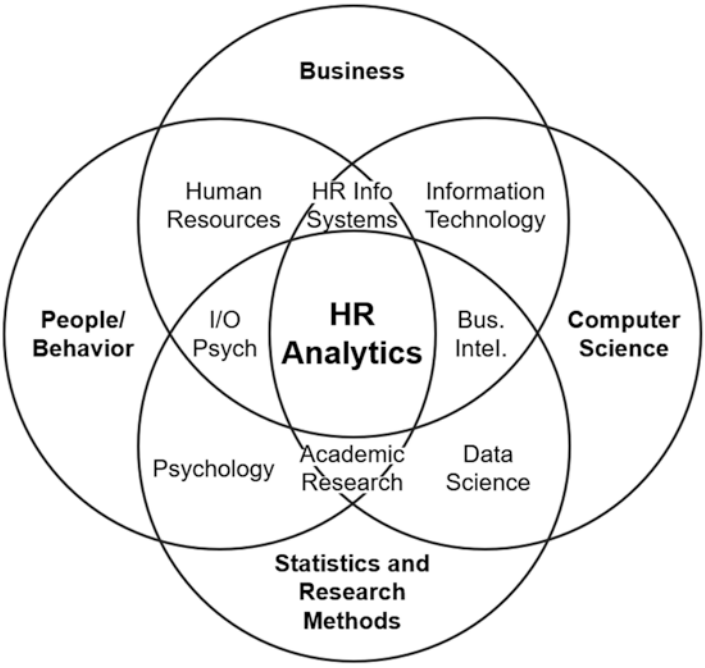
It is important to note that neither path is "good" or "bad." In fact, being happy with either of these paths depends more on personality than some quantifiable rating system. Each circle is distinct and very different, yet critical to overall balance in how someone feels about their life.

In the same way that Ikigai illuminates how activities bring balanced value to our lives, HR Analytics done well is the confluence of four overlapping disciplines.

Making decisions based on behavioral data requires a unique combination of experience across these four major domains:

1. **Computing:** How hardware and software work together to ingest, store, manipulate, and output data.
2. **Human Behavior:** The science of behavior is commonly known as psychology and has numerous subcategories and specialties. In HR analytics, a subfield known as social psychology (and often one of *its* subdisciplines, Industrial-Organizational Psychology) is the primary branch of behavioral science which applies. Behavior is unique to measure and predict because it is more abstract than traditional objects of statistics as well as more variable. Psychology is the science of how this is done.
3. **Statistics and Research Methods:** Statistics is the group of mathematical principles concerned with analysis and interpretation of data. Research Methods are the techniques and strategies leveraged to gather data for such analysis.
4. **Business Acumen:** Though business acumen is often a very generic term used to describe ambiguous competencies, in this application business acumen is understanding the practical business applications of the insights gained and the context in which you are operating.

In the same way that Ikigai comes together to create balance across an individual's life, these four areas of expertise hold their own critical worth when doing HR Analytics well:



3.1 The Data Chef

To illustrate Ikigai in the realm of HR Analytics, we offer the analogy of the Data Chef. A data scientist working with employee data very much resembles a chef in a kitchen. They are hired to prepare something delicious. But deliciousness is a relative term—it will necessarily balance many attributes ranging from taste to nutrition to aroma to presentation. In HR Analytics, deliciousness is a balance of the priorities of the organization, which spans categories like market share, financials, diversity, efficiency, productivity, employee satisfaction, and others. To be delicious in a business sense, insights must lead to a positive impact on the right priorities in the right way. Ikigai will serve as a guide for how it is done.

Skillset 1: Computing. Computing for our data chef is like her kitchen and everything in it. The knives, pots, pans, blenders, spatulas, plates, cutlery, and anything else she needs. The stove, the refrigerator, the cupboards, and the countertops. Anything that the chef needs to store, access, prepare, and present is in the kitchen. Computing is the foundational canvas on which all HR Analytics is painted. From hardware like servers and memory and hard drive space to software, managing databases, statistical processes, or visualizations, computing is how analysts interface with data. A chef cannot chop carrots with her bare hands, and an analyst cannot create an algorithm to predict turnover without a computer which can access the right hardware and software to do the job.

Skillset 2: Behavior. Understanding behavior for the HR Analytics practitioner is like understanding ingredients for our chef. Does thyme go better with rosemary or oregano? What happens to the flavor of garlic when it is roasted? Note that *this is not an understanding of which ingredients to use*, but rather *understanding the nature of the ingredients and how they interact*. When someone says, "Italian food," the chef's mind goes to tomatoes, olive oil, and garlic and when they say, "Indian food," she thinks curry, coriander, and cumin. Using data to improve workplaces and workforce outcomes requires an understanding of behavioral theory because workplaces and workforce outcomes are all *created by people*. This is much more complicated than brute force programming can solve. Brute force methodology says if every piece of data is examined from every angle and in every combination, then the answer will eventually arrive. This a-theoretical, context-agnostic data science in the space of behavioral prediction is the culinary equivalent of adding ingredients at random to a pot and hoping the results taste good—the chances of it working are virtually nonexistent. The data chef must understand her ingredients, so she knows *how* to work with them.

Skillset 3: Statistics and Research Methods. For our data chef, statistics and research methods are her cooking methods. Boil or bake? Emulsion or mixture? Dice, chop, or mince? Similar to behavior, the chef is not talking about *which* ingredients, but rather how we put them together to make something delicious. Roasting an avocado would taste just as bad as serving a pumpkin raw, but they both taste fantastic when prepared properly. Different forms of data have many

types of considerations to make before collecting them and choosing how to analyze them. Statistics and research methods help us understand how to do this well.

Skillset 4: Business Acumen. The final pillar in the chef analogy is business acumen. This equates to understanding the people who are going to eat the food and where and when they are going to eat it. Serving food at a football tailgate is much different than serving it in a 3-star Michelin restaurant. What to prepare, how people are going to eat it, and the setting are all important considerations for what to cook. It is easy to see how an intricate 5-course meal would be tough to execute out the back of a pickup truck before the big game. The point is, business needs matter. Industrial context matters. The appetite and aptitude of the end-users matter. Timelines and resources matter. How data is going to be used, what decisions are going to come from it, and what skills and abilities the consumers of the data have are all important when considering how to develop and use data.

Like Ikigai, if analytics (or culinary) endeavors significantly lack any of these four disciplines, they will not reach fulfillment; they will not create the competitive advantage sought by the business. Teams may use statistics and computers to ruthlessly drive efficiency in operations only to see the business crumble because they burned out all their employees. They may seek to understand the people so well that they collect an impossible amount of data, cannot hope to organize or manage it, and then have to deal with inflated IT costs and analysis paralysis. Or maybe the team knows how the business works, but consistently make impossible demands of the people who own and work with the data, so none of the requests seem to create the impact originally envisioned.

Teams need all four. When an HR Analytics team has HR Analytics Ikigai, they know the tools they have at their disposal, what ingredients are available, how to prepare those ingredients, and for whom they are preparing the feast. Only then are they well equipped to make something that delights their consumer.

3.2 Adding Machine Learning to the Mix

The chef analogy is an important place to begin the conversation because HR analytics teams must ensure they have well-rounded resources, so they are able to handle the critical considerations from all angles. This is largely because the disciplines of computing, behavior, statistics and research methods, and business acumen are quite different and they were developed independently, without the other disciplines in mind. This differs from kitchens and cooking methods because they *were* designed with each other in mind. Stoves are designed to heat food up and refrigerators keep things cold. There are plates to hold solid food and bowls for soups and salads. Small paring knives take apart veggies and fruits while big butcher knives help separate cuts of meat. Ingredients move through the process of pantry-to-table seamlessly because each piece of the process was built to compliment and work in harmony with the others.

HR Analytics does not have the same luxury—the four disciplines did not develop at the same time in human history, nor are their major theories and applications subject to each other. They have four disparate schools of thought and have only recently been expected to work together to help make better decisions about human capital. This significantly complicates keeping balance in HR Analytics Ikigai.

It also means the skillsets across these four groups are rarely present within an individual, or even a team, and practitioners are most often underqualified in at least two of the above major categories. For example, today's modern data scientist is usually unversed in the aspects of behavioral theory and how the unique characteristics of behavioral data require special consideration. Conversely, today's seasoned[1] organizational psychologist may understand behavior, but is most often unprepared to deal with the realities of IT systems and data management. Finally, the prototypical HR IT expert often lacks the background to make sound decisions regarding the statistical implications of her work.

Nevertheless, the competitive advantage to be realized from these data has launched an entire subindustry predicated on having expertise in all four domains. In a way, it is what it is and leaders must navigate the challenge. This means it is the task of the HR Analytics leader to ensure the skillsets of a team are diverse enough to balance considerations across the disciplines. And while advanced analytics is powerful, it threatens this balance by shifting the focus of the model.

Combining this idea with recent advances in computing power and technology has unleashed the powerful field of machine learning within HR. And despite its presence in other organizational disciplines for years, machine learning is a new frontier for most teams working with data about employees.

Among other impacts, within our HR Analytics Ikigai Model machine learning shifts insights creation toward data science:

[1] Data and computer science are quickly becoming part of most scientifically based post-baccalaureate degrees, though tend to be only part of the skillset needed when compared with true data science and engineering skills.

Machine learning has the power to make an extraordinary impact on HR Analytics. Indeed, the boon of HR Analytics on its own, let alone the advances in machine learning, has brought skills to the HR department which have never been seen before. Most practitioners in the world of data science have advanced degrees in statistics, computer science, information technology, management information systems, or other related fields like physics or economics. In fact, many schools now offer accredited degrees in "data science" where they combine the traditional coursework of a statistics degree with a computer science degree to manufacture these sorts of professionals.

And this is a wonderful thing. As technology advances and professionals can increasingly apply statistics in digital settings, the ability to discover and reap the benefits of data science are growing. The advances made in supply chain, finance, and consumer science are nothing short of amazing when considering where data science was just 10 years ago.

That said, although machine learning is powerful it implicitly unbalances HR Analytics Ikigai. In this way, machine learning threatens HR Analytics by unintentionally ignoring (or at least minimizing) many critical aspects of sound behavioral science and business context. Unbridled over-application of its principles will lead to, at best, invalid insights, and at worst, legal and ethical violations. As prudent practitioners of HR Analytics, we must ensure that the application of these advanced analytics does not jeopardize the integrity or applicability of the science.

To this end, some in the organizational psychology community have called for the dismissal of machine learning and other applications of "big data" because they are "unscientific" and threaten the integrity of the theory which drives good decisions about employees. We think simply ignoring machine learning is unwise. Unbalancing the Ikigai in the other direction (i.e., in the direction of HR, business, or I/O) unnecessarily discounts a very powerful tool which is now in the toolbox. Abstention because machine learning is too a-theoretical or too complex is akin to the sixteenth century biologist ignoring the invention of the microscope because it does not fit within the current paradigms of science.

People scientists and practitioners must learn the basics of machine learning, so they can partner with experts and realize value from these powerful methodologies. Like any technological innovation, machine learning has its perils. But this does not mean we should ignore the innovation. Rather, we need to develop a healthy understanding across all domains, so we can facilitate prudent adoption. Much like the microscope, these new approaches allow us to observe the world in new ways. When the community of social scientists can effectively partner with the world of data scientists, we will drive innovation and insights which will push the boundaries of behavioral prediction beyond what either side could achieve on their own. Data science, advanced analytics, and machine learning have so much to give to the world of behavioral prediction and it is important that we fold them into people science without unbalancing the Ikigai.

At this point, we have reviewed the industrial need for this text. We have discussed some of the considerations of leveraging advanced analytics with employee data. Most importantly, we have outlined our model for how four disparate domains of expertise must work together in order for HR Analytics (and machine learning) to operate effectively within an organization. The following chapters will begin by exploring the foundational domains needed for quality machine learning: Research Methods, Statistics, and Computing. Then, we will dive into how they come together to create high-quality machine learning methodologies for the HR practitioner.

Discussion Questions

1. What are the four parts of HR Analytics Ikigai? Why is each important?
2. How does machine learning stand to influence HR Analytics Ikigai?
3. Discuss some pros and cons you see of integrating machine learning into an HR Analytics strategy.

Part II
Bringing Science, Machine Learning, and Behavior Together

Chapter 4
Thinking About Your Problem-Solving Strategies

One of the most important, yet often undiscussed prerequisites for machine learning is that it must be used as a tool within the bounds of logic. That might sound like a given; we often assume that if we are using data to make decisions we are automatically acting in a well-reasoned way. But does using data, by definition, mean we are acting logically? Have scientists always removed the subjectivity and anecdotal nature of decision-making by leveraging data and statistics?

In short, no. Using data does not automatically equate to logic. Machine learning, or any data-based analytics tool, is subject to the will and methods of the team using it. This means that if not done well, they are at just as much risk for bias or invalidity as any other method for decision-making.

This goes for any tool: when used improperly or for the wrong reason, the outcomes are often bad because we are not using the tool as designed. Envision using a hammer to cut a piece of wood or a belt sander to polish shoes. Those seem obvious, but how do these analogies carry over when talking about advanced analytics?

Some common arguments you may hear about using these methods poorly are "if we are just using algorithms to find patterns, then we are no longer using 'science'." Some may call it "dustbowl empiricism." Other ways this concept is phrased is that machine learning is "a-theoretical" (without theory), or that it lacks business context.

What do these objections mean? Where does the term "dustbowl empiricism" come from? And what does it mean to hear analysts are "no longer using science"—everything about machine learning sounds very scientific. Why would anyone balk at such an objective way to approach pattern recognition, discovery, and learning?

Again, just because a method or process uses data or is supported by numbers does not necessarily make it scientific or objective. To our previous point, machine learning (or any method of investigation) is not, by itself, science. Science is a method. Science is a process to observe, hypothesize, test, reobserve, update opinion, and influence future action. To do science, the scientist has many tools at her disposal. Observation, experimentation, creating hypotheses, defining variables,

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_4

machine learning, statistics, data mining, and many others are all concepts which on their own are just potential parts of the process that is science. They are tools in the scientist's toolbox. And a good scientist chooses the right tools to get the job done (not every tool, and not the same tool every time).

This is a critical point: simply putting mathematics inside a computer and aiming them at business problems will not likely solve anything and may make things worse, all while giving HR analytics a bad name. Without sound research methodology, machine learning is as useless as a scalpel to a carpenter or a hammer to a surgeon. Like these tools, machine learning is good for some types of problems but, just as importantly, poor for other types of problems. Would you hire a home contractor who said he could build an entire house with only a hammer? Probably not.

That said, extremists on the other side of the argument essentially say that science should never use machine learning in HR. That, because some people have used their hammers too often or have used them incorrectly, we should abandon the use of hammers altogether. Dustbowl empiricism is a critique which states that making observations in data without establishing a theoretical framework first is not a legitimate basis for scientific inquiry.

This premise is equally incorrect. While the conservative approach may reject the use of machine learning, the challenges of integrating machine learning into the world of workforce data do not outweigh the benefits. Rather, they are merely a collection of important considerations to keep in mind when embarking on machine learning endeavors. Later in this book, we will look into history as a guide—there are many positive and negative examples in our past of how technological, scientific, and statistical breakthroughs have helped, and hurt, humanity. When attention is paid to the challenges in the industry today, and practitioners learn from errors of the past, they can use machine learning with the prudence required to create a positive impact.

Machine learning is not science any more than a hammer is carpentry. Machine learning is a tool to use to apply science. Part of how a carpenter learns carpentry is by exploring and understanding all the tools in his toolbox—he must learn to wield his tools effectively and in harmony with all the other tools at his disposal. Later in this chapter, we will talk about two schools of thought which are foundational to all tools in the scientific toolbox and how machine learning fits in. But before that, we want to introduce (or reintroduce) the scientific method.

4.1 (Re)Introducing the Scientific Method

Science is not a subject. Biology is a subject. Geography, astronomy, mathematics, and psychology are subjects. They are topically oriented groups of facts and theories that are collected, refined, and studied. Science is typically thought of as the grouping of many subjects across domains like natural science, social science, and formal science which all use research and the scientific method to advance. In fact, we often group subjects like these together into colleges at universities like "The College of Arts and Sciences." This differentiates the subjects from other fields of study, like

"The School of Management" or the "College of Professional Studies" because scientific subjects, though disparate in topic, have the common bond that they all use the scientific method to build, grow, and advance their subjects. Practically speaking, science can be thought of as a way of thinking, and its goal is to create certainty.

In fact, the whole purpose of the scientific method is to figure things out for certain. It is a method to deduce facts—this means that the fact must be true given the evidence available. Once the facts are put together, one can see a bigger picture that is called a theory.

All scientific endeavors go something like this: Observe and make a hypothesis about why something happened. Then, test that hypothesis and based on the results there is evidence for or against the hypothesis.

At its most basic level, this is the scientific method. Here is a relatable example:

#	Step	Example
1	Observe something	The sun and planets move across the sky
2	Formulate a hypothesis	The sun and planets must go around the earth
3	Make a testable prediction	If the earth is at the center, then the distances between the earth and sun/planets will remain the same when they are at the same point in the sky
4	Test the prediction	Incorrect. It looks like the distance from the sun is (reasonably) static, but the distance between the earth and the planets changes a lot. Also, planets appear to move in strange, noncircular ways sometimes
5	Go back to step two (formulate a new hypothesis)	All the planets go around the sun
6	Make a testable prediction	Apply advanced physics and geometry to predict patterns of movement around a common center
7	Test the prediction	It works!
8	Make a theory	The earth and planets travel around the sun in big circles
9	Make more observations	…well maybe not perfect circles
10	Refine the theory with new evidence	Ellipses work better

This concept makes intuitive sense because it is a basic flow of logic for cause and effect. If you understand the cause of something (the planets go around the sun), then you can predict some outcome based on that (the location of the planets relative to one another). In fact, this is what we all do every day, albeit more informally. We have existing ideas about how the world works, and every day we move through life updating those ideas based on what happens to us. When you open the refrigerator at home, you have already made a prediction about what will be in there based on what you have seen in the past. This creates an expectation of what will be there (i.e., hypothesis). If what is there does not meet your expectations, you must update your hypotheses by investigating what happened (who finished the milk?!).

Another great example is to watch the behavior of children. The famous cognitive psychologist Jean Piaget went so far as to call young children "little scientists" because of their tendency for experimentation. They spend their earliest days learning—picking things up, putting things in their mouth, calling something by its name, and then looking to an adult for validation. They are constantly trying to figure out the world they live in. Science seeks to take that idea—observe, test, update ideas, repeat—and formalize it.

Science also wants to take it a step further. In the above examples, scientists were observers—they look at the system, noticed something about it, and then used science to figure out how it worked so they could make a prediction. Science as a method also allows us to make changes in the system and see what happens. We might make a hypothesis about how a certain aspect of the system works, change that part of the system, and then observe the effect. Have you ever looked at a new process in your office and thought, "if we do X, then Y is going to happen?" Then the new process is implemented, and exactly what you expected to happen comes true. For better or worse, that prediction is your own application of science.

When practitioners are proactive with this idea, they can design interventions which drive positive changes in their workplaces. This is the essence of experimentation and when done well, can very effectively show how changes in business process improve everything from business operations to engagement to turnover.

Indeed, there are entire textbooks and college courses dedicated to research methods, and if this topic is very interesting to you, we encourage you to explore it in depth. For contextualizing how science enables better machine learning practices, we would like to introduce a few practical topics in the space of sound research methodology to keep in mind when exploring solutions at your organization (using machine learning or otherwise). But first, we would like to introduce how machine learning can often look decidedly *un*-scientific.

4.2 Deductive and Inductive Reasoning

The scientific method introduces the domain of thinking called "deductive reasoning," which is an important type of logic used for empirical research and critical thinking. However, machine learning brings a different kind of logic to insight-generation when compared with this traditional approach to cause and effect.

To begin, we must revisit that an important foundational aspect of the scientific method is that the formulation of theory in science is deductive. Said differently, science is a top-down way of thinking. The thinker starts with the observation, then infers the rules that they think exist (hypothesis). Then they subject the rules to examination by making predictions and testing them. If correct, then the results of the test will align with what was predicted before the test was conducted. Essentially, if the thinker can guess the outcome of the test before the test, then they can say there is evidence supporting the hypothesis.

When scientists take (1) many observations, (2) hypotheses about them, and (3) tests of the hypotheses which provide evidence, they arrive at a theory. Scientists do these many times over in many different, but related experiments. They then put the rules which have been discovered together to make a story, and that is what is called a theory. When outlining the overall progression from earth-as-center of the solar system to sun-as-center of the solar system, we were summarizing the progression from observation to theory. In reality, this was the result of hundreds of scientists conducting thousands of experiments, collecting hundreds of thousands of data points across many generations. It was only when scientists took all those measurements which led to all those conclusions and independent storylines and put them together in a way that all the measurements worked together to tell one story (i.e., under one set of rules) that they arrived at what we now know as the truth about how the planets and sun are physically aligned in space.

This is what makes the scientific method so powerful—constant collection of more information about theories and using the new information to influence the "rules" that govern those theories. And because it is deductive, scientists know that what is contained within the theories is true. They have started with many observations and reduced the possible causes down to contain the least amount of assumptions until they are left with as close to objective facts as they can create. But there is another way to use reasoning to create logical assumptions, and machine learning opens our investigations up to its possibilities in ways deductive reasoning cannot.

4.2.1 Inductive Reasoning (And Why It Matters)

Inductive reasoning approaches logical conclusions in a different way. Where deductive reasoning is top-down, using progressively more specific logic until reaching a conclusion, inductive reasoning is a bottom-up approach. It starts with the specific data or patterns and then extrapolates outward to conclusions which fit what can be seen.

Here are two examples which illustrate the difference:

Deductive:

1. All finance employees in the Tacoma office are hired at the midpoint of their salary band.
2. Maryanne, Brad, and Danielle were just hired to work in Finance in the Tacoma office.
3. Therefore, Maryanne, Brad, and Danielle were hired at the midpoint of their salary bands.

You can see here that we start with a general premise that we know is true (how we pay newly hired finance employees in Tacoma), and then work down to a logical conclusion. This is very important because as long as premises one and two are true, then number three *has to be true*. This is deductive reasoning and is how the scientific method works. Let us try it a different way:

Inductive:

1. Maryanne, Brad, and Danielle were hired at the midpoint of their salary bands.
2. Maryanne, Brad, and Danielle were just hired to work in Finance in the Tacoma office.
3. Therefore, all finance employees in the Tacoma Office are hired at the midpoint of their salary band.

Did this one feel different? It should. In the first example, the exercise started with the general truth that was known and then got more specific until reaching a conclusion which fit. Here, with essentially the same facts, it did not work. Why not?

In the second example, the exercise starts with a specific observation and moves outward to more general ideas which fit what was observed. However, the dependencies between facts do not exist in the same way in the second example. Maryanne, Brad, and Danielle may not be the only newly hired finance employees in Tacoma. And even if they are, it could just be a coincidence that they were all hired at the midpoint. There are many holes in the logic and so even though the premise is reasonable, *inductive reasoning does not give a for-sure conclusion the way deductive reasoning does*. The difference between deductive and inductive is that one deduces what *must* be true, while the other infers what *could, should, or might* be true.

Despite not being as absolute nor as causal as deductive reasoning, inductive reasoning has a great many merits. Inductive reasoning is substantially more feasible than deductive reasoning and this provides many advantages when working within the realities of the applied human performance space:

Speed (for when the cost of waiting is high): Most applied HR research is occurring in the for-profit space. This means that HR is embedded in the realities of financial goals, sales targets, operating costs, stock prices, investor pressures, and the like. Due to this, timelines are often compressed because the difference between understanding results today versus next month can be monumental. The realities of truly empirical research typically require careful design, delicate methods, and exact execution, and as such are not often well-aligned with the speed most businesses are moving.

The goal of science is to know for sure. It values validity (correctness) and reliability (repeatability) above all else; the amount of time it takes is secondary. In business, an idea is usually only useful if it is delivered in time. This brings in the economic concept of diminishing returns: using inductive reasoning to get 70% correct in a few days is often far more valuable than a deductive approach to get 99% right but takes a few weeks or months.

Iteration (for when the risk of trying is low): The speed advantage is added to because often HR Analytics teams are operating in environments where initial results can be iterated and improved over time. Inductive reasoning lends itself to the adage of "build the bridge while you cross it." A researcher can use inductive reasoning to set off down a path and then course-correct over time. Deductive reasoning tends to be more positive about its assertions, but it does not lend itself to iteration as smoothly.

In addition to feasibility, inductive reasoning has other merits as well. It tolerates more uncertainty than deductive reasoning. However, this uncertainty opens us up to concepts like forecasting. No one can technically know the future, so any prediction is inductive by nature. One can use deductive reasoning to inform forecasting tools and even apply confidence intervals which outline expected accuracy, but at the end of the day any predictive method is extrapolating about the future based on what it knows about the past. And especially when dealing with extraordinarily complex or poorly understood systems like the economy or the human psyche, often the most reasonable approach is to use the data at our disposal to infer patterns, and then test them out. And that is fundamentally inductive.

4.3 Ikigai Leaning Too Far Right

Despite the advantages of inductive reasoning, practitioners must be sure that it does not unbalance HR Analytics Ikigai. A good rule to keep in mind is that inductive reasoning should not be used without domain expertise. In order to do inductive reasoning well, a practitioner must know a significant amount about the topic area they are working in. Obviously, expertise is always an advantage in research, but especially in inductive reasoning where logic and method design are not as airtight, the need for expertise becomes even more important.

This is where those telling the industry to slow down with the application of machine learning have a strong point: when it comes to using machine learning with employee data (and behavioral data in general), the analyst must understand the underlying social theory that drives behavior, otherwise one cannot effectively observe patterns in big data sets. The data chef analogy serves nicely here: a gourmet kitchen filled with the highest quality ingredients will not by itself produce good food. The chef needs two critical things: (1) how to use the equipment in the kitchen (Statistics and Research Methods) and (2) how to put the ingredients together in a way that tastes good (People/Behavior).

In the haste to use the machine learning kitchen, many self-proclaimed people scientists have forgotten the extremely important second premise. They must understand the ingredients, not just the stove and the knives and the cutting board. In some cases, they are adding ingredients to a pot at random and hoping the results taste good. If HR Analytics Ikigai leans too far right, we have extraordinary prep-chefs getting the ingredients ready, but then do not know the first thing about the difference between an avocado and a pumpkin. Here is a simple example:

The sales order processing team has a very clunky set of steps for intaking, processing, quoting, and returning information to the sales teams. The company started small, so the process used to be very easy; when there were only a handful of sales reps, the whole process was handled by two people (the finance person and the COO). And while it is good that the company has been growing so fast, there are now almost 50 sales reps, a team of 5 finance people, and the COO does not have

the bandwidth to review any except the biggest orders anymore, so a few project managers have taken her place.

The resulting process is very inefficient and frustrating. Jim, being a motivated and conscientious analyst, decides to take matters into his own hands. Jim designs a survey to ask folks from the sales teams and order processing what works well and what can be improved. With the resulting data, he believes they can objectively show where the pain points are and help drive positive change.

Everyone loves the idea! Jim creates a 30-item questionnaire to analyze the whole process. A resulting sample of the dataset looks like this:

ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q 10	Q 11	Q 12	Q 13	Q 14	Q 15
45	4	5	4	3	4	2		4	4	2	4	5		4	3
46	2	2	2		3	1	1	3	2			3		4	1
47	3	2	5	5	1		5		5	5	4		5		
48	2	4	3	4	1	5	2	4	4	4	4		4	4	4
49	5	4	2	4		4		3		2	2		2	2	2
50	1		2	3	1	1	4	5	3						
51	3	3	5	4	1		4	2	5	2	3	2	3		5
52	2	4	1	1	4	5	5	4		2	1	1	4		4
53	1	2	5	1	2	2	1	5	1	4			5	2	2
54	4	5	2	3	4	3	3		1	3	3	4	4		4
55	4	4		1	4	2	4	2	4				1	5	
56	1	3	4	4	1	2		3	3	3	3			3	3
57	4	5	3	5	4	3	3	4	5	4	4	3		4	
58	1	3	2	4	3		2	2							
59	3	1	1	3	1	5	1	3	2	3	4	3	4	5	1
60	1	1	5	4	4	5	2	5	3	3	3	3	3	3	3

At first glance, Jim's data looks good. There are some holes, but every real dataset has that. There is good distribution in scores for every question and for every participant. Ready to ship these data?

There are two red flags about these data that Jim might miss if he is not seasoned in the research methodology and psychology of survey taking. The first is something called survey fatigue. That is, the longer a survey is, the less likely is to get thoughtful responses (or even responses at all) near the end. You can see that the first 5 questions only have 4 missing datapoints across all 15 participants, whereas the last 5 questions have 29 missing points! Second, if you look closely near the end of the survey, participants stop varying their answers. There are strings of 3's and 4's instead of the question to question differences we see in the early questions. This is called center-lining and is an indication that your participants are not thinking critically about the questions.

A simple way to combat these issues is with item randomization. Simply put, if every participant received questions in a different order, then the fatigue would be randomly distributed among the questions instead of just the last 5 questions suffering.

And it is not just research theory that is at stake here. Another example shows us business acumen is often another skillset that is compromised in our haste to apply advanced analytics. Sometimes practitioners need to understand the business process or regulatory considerations that machine learning insights may influence or compromise.

A machine learning consulting firm was once hired to build a selection algorithm for a business. They wanted to save time and money by not having to manually screen every single resume. This makes sense and is a great application of machine learning—if the algorithm can pick out the important patterns in applicant data, then talent acquisition can save a lot of time by not having to review every single resume.

The consulting firm built a neural network, which is a machine learning technique which examines at all the data points you input and all their relationships, much like how brain neurons are often connected to many other neurons (more on neural networks in Chap. 9). With some advanced math, it can work through all the possible connections in a dataset and create predictions about how different attributes contribute to an outcome.

The algorithm worked and the company implemented their new selection algorithm. Unfortunately, upon further review, the algorithm did not stand up to adverse impact tests, which is the validation procedure used to make sure that a test does not systematically exclude any protected classes like gender or race. This is a federal requirement known commonly as "Title VII" in the United States and applies to all selection practices for all companies.

When HR professionals design these tests, there are many ways to combat adverse impact to make sure that the tests are both predictive AND fair. Concepts like score banding, for example, randomize selection within a range of scores and quiets adverse impact when it is seen in a nonmeaningful way.

Unfortunately for this company, the data scientists were not familiar with the Title VII regulations required when designing selection criteria and simply input all the data they could get into the algorithm. "Brute force" techniques like this are common in data science and are great inductive technique when there is a high volume of data and the researcher wants to examine everything but is not sure where to start. They are not great when the data has considerations like adverse impact which must be handled during model design. And since the researcher cannot explain how a neural network functions (known as "opaque;" see Chap. 8), the algorithm was neither legally defensible nor reasonably editable, so the project had to be scrapped.

4.4 Good Balance Drives Results

However, machine learning also opens huge opportunities for well-informed practitioners to realize huge gains for their organizations. And while the approaches may sometimes seem fundamentally in conflict with the scientific method, the inductive approach can augment and enhance the deductive mindset because the new worlds

of big data and data science simply allow practitioners to observe workplaces in new ways.

The beginning of science relied on naturalistic observation. The proverbial apple hit Newton on the head, it made him wonder, and then physics is born. Next, there were theoretical observations. Einstein notices that Newton's physics do not agree well with Maxwell's equations, and by applying some brilliant brain to some unconventional thinking, general and special relativity are born. In today's world, practitioners do not just have labs and theories like Newton and Einstein, they also have terabytes upon terabytes of unexplored data. It is metaphorically the new frontier for science. And while it might be easy to dismiss digging into all this information as "brute force coding which will never find anything," it is much more advantageous to the science of human performance to jump in and investigate. When a practitioner is armed with the knowledge of people and stats and systems and they are grounded in the context of their business and working populations, a practitioner can navigate this sea of meaningless data and begin to find interesting things.

4.4.1 Building a Business Case for more Heads more Often

In the world of frontline sales, the art of staffing is difficult to master. Despite its complexity, the problem is actually very easy to articulate: if a company has too few associates in their locations, they do not sell as much—long wait times for customers and over-taxed employees make it difficult to optimize every customer interaction. Additionally, the same company will likely have to spend extra money for overtime because they do not have enough employees to meet the shift demands.

The other side of this coin is having too many on staff. In this case, the company is wasting salary dollars on employees without full workloads. And worse, when there are more employees than work, the employees are not having a great experience either. They can be disengaged, bored, and the environment may become too competitive because there are only so many commission dollars to go around. These tax the culture of the workplace and can even impact the customer experience in extreme cases.

The equation gets even more complicated when considering (a) industries with variable sales volume and (b) high turnover (like frontline sales usually has). As a real example, a wireless retailer had a philosophy of being very staff heavy at the times they launched new products and during the holidays, but relatively light at other times of the year. So how should they ebb and flow the number of employees they keep in order to have the optimal amount present and trained?

At this company, there were two sides to the debate: The Efficient Dollars Philosophy versus the Smooth Operations Philosophy. The Efficient Dollars Philosophy says to hire people just before they are needed. That is, increase hiring just ahead of demand to minimize extra labor costs. This is an easy philosophy to see the benefits of: pay people when they are needed, and do not pay them when they are not needed. And if the guess is a little low (and it usually is), then simply

augment the ranks with contingent labor and some overtime. This extra cost is usually far lower than guessing too high.

Smooth Operations Philosophy says that there is lot more lost than salary dollars: more experienced sales employees perform better, well-staffed teams suffer less burnout, hiring booms drive significant stress on Talent Acquisition, and other factors make the Efficient Dollars Philosophy more expensive than it looks. They say it is better to carry a few extra people so that the hiring peaks are lower and the slow-sales valleys are higher. In short, maintaining smooth operations is better for business than massive swings in headcount which are disruptive to the day-to-day operations of the frontline.

How might one go about analyzing this? There are data in many places: turnover data, performance data, talent acquisition costs, training data, quality of hire data, and profit and loss statements to name a few. And each of those categories has mountains of data within them. And each of those mountains has near infinite caves and caverns to explore. This is where the researchers came to the dustbowl empiricism crossroads:

The strictly deductive approach: Use social theory to isolate potential effects and create hypotheses. Then go research-question by research-question, creating and testing those hypotheses until something significant appears.

The inductive plus deductive approach: Knowing what the researchers know about social theory and the current business states, put a huge amount of data together and see what can be found.

The researchers knew what the Efficient Dollar Philosophy said because it has predominated the staffing approach used for years and was widely accepted as the best way to do things. But when the researchers talked to the professionals in the field doing the work, they learned two intuitive, but very important, premises: "People with more experience perform better" and "Talent Acquisition costs get very high and their performance suffers when there is a large hiring demand."

So the researchers dug. They took performance data and chopped it up by tenure. They took hiring data and examined it by volume. They took tenure data and clustered it. They took overtime data and put it up against performance data. They lagged and binned and segmented and did every sort of thing they could imagine to make sense of the patterns that their business partners had seen play out every time a new product hit the market or the US entered the winter holiday season.

They found that when hiring volume gets high, TA performance does not suffer that month, but up to 2 months later. They found that where they were drawing the line for "New Hire" was actually not supported in the data—there were more new hire groups who could be segmented, and they could see differentiation in their performance. And carrying some extra salaries for an extra few months was outweighed by their better performance when busy season hit. Overall, they were able to show that if done properly, the Smooth Operations Philosophy beats the Efficient Dollars Philosophy.

This example illustrates that neither traditional research philosophy would have worked on its own. The traditional, deductive, scientific method would have fallen down: if the researchers could only start with what they already observed, they

would not have had the exploratory latitude to find the effects they found. That said, if they had taken a strictly inductive approach, the exploration would not have been delicate and nuanced enough to find the minor tweaking required to find the 2-month lag effect. It was only by combining a little inductive with a little deductive that they were able to zero-in on the effects and quantify evidence to support the Smooth Operations Philosophy.

Discussion Questions

1. Create two examples of how you have (or might) use the scientific method at a company. Specifically, focus on what makes the approach scientific.
2. What is the difference between deductive and inductive reasoning? What is the value of each?

Chapter 5
Great Results Come from Great Questions

Now that we are grounded in the opposing logical approaches of analytics and can appreciate what each brings to the table, we must aim our problem-solving at actual business problems. The first and most critical step in this process is asking questions well. This may seem intuitive, but the art of understanding the real problem underneath a question and defining the component pieces of a question can be quite challenging. This step is paramount for all consultative activities, but in the world of HR Analytics and machine learning, the answers to these questions drive the definitions which become the foundation of the data models driving all forms of advanced analytics.

5.1 Defining the Problem in a Testable Way

Think about a people-oriented challenge you would really like to solve at work. Maybe it is leadership-related, corporate culture related, or an outcome like turnover or engagement. Now ask yourself: how well-defined is the problem? Defining the problem sounds deceptively simple. Business and HR leaders alike often understand their challenges intuitively as a result of their extensive experience. And in fact, due to this expertise, they are almost always on the right track. Maybe the problem at your organization sounds something like this:

"Our compensation strategy is not working. We are not getting the people we need in the door because they are accepting better offers. And we cannot keep our best people! They are leaving left and right for better paying positions."

Let us unpack this. While this is a well-articulated problem, it is not a well-defined problem. The leader is bringing forth both the symptoms (poor offer acceptance and high turnover) and the diagnosis (poor compensation). They are also making assumptions about the quality of hires ("the people we need") and the quality of employees we are losing ("best" people).

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_5

This means the leader could be in a number of different places about which pain(s) they are experiencing in their business. They have started with observations and assumptions, but the leader has articulated them in a way which is not well-defined because the problem is probably multifaceted and therefore must be parsed out.

An analyst who is going to use machine learning, or any other type of investigation method, must take the time here to figure out what their leader is really saying. The analytical parts of HR analytics are dependent on having a highly specific understanding of the problem. If you went to the dentist and said, "my mouth hurts" would they start randomly drilling your teeth? Of course not. They would say, "show me where." As researchers, analytics professionals must get better at probing and understanding where the pain truly is before taking out the tools.

The first step in research methods in an applied setting is about isolating what you are trying to understand so that you can define it, measure it, and then see if your hypotheses hold up. Here is a short list of what this leader may have brought to us in just their five short sentences:

What the leader said	What the leader might be trying to say...	OR	OR
Our compensation strategy is not working	We do not pay enough	Our employee value proposition[a] is poor	Something has changed which has rendered our comp strategy ineffective
We are not getting the people we need in the door	We have low acceptance rates	We have low quality of hire	We are not hiring people with the right skills or fit
Because they are accepting better offers	We do not pay enough when compared with our competition	Other companies have a more compelling employee value proposition	Some aspect of our employee value proposition is poor (e.g., benefits, location, etc.)
We cannot keep our best people	Voluntary turnover is too high	We are losing important people	
They are leaving left and right for better paying positions	Pay is a major cause for turnover	Other companies pay more for the same jobs	Other companies are offering promotions to our people that we are not

[a]Employee Value Proposition is how the labor market and employees perceive the value employees gain by working in an organization. Employee Value Proposition can be categorized across five main categories: opportunity (career development and organizational growth), people (culture, manager, and senior leader quality), organization (market position, product/service quality, social responsibility), work (job interest alignment and work-life balance), and rewards (compensation, benefits, and perquisites).

The leader came to the analyst with one problem, comprised of five statements, which turned into 14 possibilities.

Additionally, researchers must be sensitive to whether or not the way their partner has articulated the problem is testable. When the sun and planets moved across the sky in the earlier example, the scientists made an assumption: that the earth must be at the center. However, that idea about the earth is not testable on its own. It was only testable

after the assumption was restated like this: "if the earth is at the center, then the sun and planets will have a constant distance from earth." By adding this second piece, the scientists have taken an idea and put conditions around it which can be tested. As researchers, we must help our business leaders define their problems this way so we can help them support or refute their ideas about the challenges they face. So instead of, "we are not getting the people we need in the door because they are accepting better offers. And we cannot keep our best people! They are leaving left and right for better paying positions," we might restate as follows:

What the leader said	Restating in a testable way	OR	OR
We are not getting the people we need in the door	Our acceptance rates are lower than what we expect	The performance of our employees is below market norms	
Because they are accepting better offers	Our hiring salaries are lower than those we compete with for talent	Employees do not view our company as a great place to work.	Our benefits are not competitive when compared with others in the industry
We cannot keep our best people	Voluntary turnover is higher than industry standard	Turnover of business-critical roles is higher than in other parts of the organization	
They are leaving left and right for better paying positions	Pay is a major cause for turnover	Other companies pay more for the same jobs	People are not being moved internally fast enough, so they are taking promotions at other companies

These restatements on their own are not hypotheses, but they are more objectively testable ideas. This applies to machine learning because an understanding of how to break down the problem into its component parts establishes a foundation on which the consultative process begins. It is here where researchers begin to think about what sorts of analyses they would use to test their assumptions and theories. When the home contractor from earlier is building a house, he might use different tools to solve different problems—a leaky pipe compared to a powerless electrical outlet would involve different diagnostic steps, different tools to remedy the problem, and different criteria for success. When an analyst is digging into a leader's problems, this is where they as a researcher should be thinking, "is machine learning a viable option to fix this type of problem?"

In later chapters, we will discuss machine learning techniques and how they may be helpful in solving different types of problems. At this point, we just want to articulate how important it is to take these important initial steps to probe and understand what problem is at hand. Because once we do, the next step is to understand how well established the information and data about the problem is.

5.2 Researching Your Research

The first step after having a testable idea is digging to understand what data is available. In the house-building example, these data come from observation: If the problem is a leaky pipe, the contractor can tell because the pipe is wet or because he can see water or mold somewhere. The contractor then seeks other data to support his theory. Can he find a spray or drip? Can he physically follow the wetness back to the source? The data the contractor needs to move from testable theory (there is a leak somewhere) to diagnosis and fix requires understanding how water and pipes work. If he found wetness on the main floor, he would check upstairs before the basement because the water often follows gravity to the lowest point.

In the world of HR analytics, it is rarely this straightforward. Back to the compensation example: if the researchers find that this leader's compensation complaint boils down to "other companies are getting the best people because our employee value proposition is poor" then the researcher needs to ask themselves, "how do we measure our employee value proposition?" The problem (and theory about the problem) is now well-defined: "People are not accepting our offers and are turning over because our employee value proposition is weak." But if the data does not exist to quantify employee experience or does not exist in a way in which it can be compared to other organizations, then the researcher cannot test their theory (yet).

This is where different types of research become useful. Depending on how well established the data sources are about the problem at hand, a researcher may need to employ different types of approaches. Understanding what is already known, accepted, and accessible about the problem being solved is foundational to the development of how one goes about ultimately collecting data, testing hypotheses, and even the legal and ethical implications of the work. To summarize options, we review research's three main types: Exploratory Research, Constructive Research, and Empirical Research.

Exploratory Research is research conducted for a problem that has not yet been studied well. It is intended to develop operational definitions, illuminate and decide on main areas of focus, and improve the future research design. Essentially, exploratory research are the questions to answer before the big question can be answered. This type of research is often necessary when researching something never studied in an organization before. Often in organizations, the most impactful thing a researcher can do is investigate questions nobody has asked before. But when leaving well-defined paths, a researcher must explore the little things before they can answer the big things.

This is a challenge because innovation and provocative thinking typically go along with a lack of processes or data aligned to answering the question. What if in the above example the researcher had the idea that the weak spot in the employee value proposition was not compensation, but instead was something more nuanced and complex, like low manager quality negatively impacting corporate culture? However, the company has never tried to quantify manager quality well before, let alone its effect on culture. How would a researcher begin to measure it? Maybe it is

touched on it in the annual engagement survey, but how good are those questions, have they been asked consistently over time, and are they a good overall representation of manager quality? And if so, have there been other factors which might be influencing the answers to those questions (like positive or negative trends in the overall survey driven by company performance or economic factors)?

Some examples of exploratory research you may have seen before are focus groups, where a leader or team brings employees together and asks them about their experiences and feelings on a topic. Another is the literature review, where a researcher reads about others doing work and having experiences on the topic they are interested in, so they can learn how others are approaching and thinking about the problem. A third form of exploratory research are some types of pilot studies[1] —where a researcher designs an intervention, but only applies it to a small group. These approaches are all different ways to gather data *about* the problem to solve, rather than trying to solve it right off the bat.

To analogize, if a company were going to build a road through a forest, they would not simply arrive with a paving machine. First, they would walk the path and learn about where the road could go. Then they would clear trees. Then they would have to get the stumps and large rocks out of the way. And only then would paving the road actually work. Exploratory Research is gathering evidence to clear this kind of metaphorical path. Once cleared, other types of research and interventions will get results and have impact.

Constructive Research is comparatively more pragmatic. It assumes the researcher knows a good amount about their problem area and helps them create testable solutions or interventions. Its goal is the same: expand the knowledge base about a business problem but does so with a very practical approach. This type of research is often used for process improvement, where researchers build, test, and then the results give a better grasp of the problem. The reevaluation of the problem improves the quality of the design process. This loop between build and test iterates until a sufficient model is created.

This is the research method behind intervention design. If a company has a process that works poorly (e.g., "promotions take too long") or an outcome they do not like (e.g., "turnover is too high"), constructive research is the method that says, "design something, see if it works, and then iterate on it until it does."

This is important to distinguish from empirical research (discussed next) because it is concerned with *functionality*, not necessarily *causality*. Sometimes a business needs causality to impact functionality, but not always. Business Process Reengineering is a good example of where HR professionals might encounter constructive research.

Finally, **Empirical Research** is the most renowned of the group. Empirical Research deals in evidence, which it uses to back up its claims. The word empirical comes from the Greek word *empeiría* which means experience—and that is where

[1] We say "some types" because pilot studies can also be a type of quasi-experimental design. If you are using a pilot to explore a problem it is exploratory, but if you are using it to test a hypothesis, it is more of an Empirical Research approach.

Empirical Research grounds itself. Empirical Research collects evidence which can be observed, experienced, and documented. In this way, Empirical Research is the most scientific because it begins with an idea based on an observation and then searches for evidence to support or refute the idea.

This is critically different than the previous two types of research and is where the biggest impacts stand to come from in research. This is because it is only in empirical research where we finally understand *why* something is happening. Knowing *that* something is broken is exponentially simpler than knowing *why* something is broken. Often times we jump from the observation *that* something is broken, right into constructive research—just fix it! In some cases, this is a sufficient approach. However, knowing *why* something is occurring is sometimes critical to remedying the problem. This is where empirical research helps.

So when you are about to embark on answering a research question, first ask yourself: is my question about (a) a new problem which is still poorly understood, and I need to explore it before diving into a solution, (b) a well-understood problem which requires a solution to be iterated and engineered, or (c) a premise which needs to be tested to see if we have it right, or if we need to modify our beliefs? Going through this easy first step will help you build the appropriate foundation of knowledge before getting into solutioning.

5.3 Operational Definitions

Now that we understand research types and reasoning models, it is time to review how to define challenges in your organizations effectively. We have already established that problems are often easy to articulate, but tough to articulate in a testable way. We have also discussed that a question or problem might be part of a well-understood ecosystem of data and ideas or a poorly understood ecosystem of data and ideas. And once a researcher has dug into these ideas, turning them into questions you can study is the next hurdle. Much can be lost in translation, even for a well-defined problem. Truly communicating what is meant can be very difficult. Sticking with the above example, if a researcher is exploring the turnover data angle to understand the employee value proposition, they might ask something seemingly simple, like "What was the turnover rate last year?" and expect the answer to be very easy. But in just a simple question like this, there are many aspects which the asker and the analyst need to agree on. For example, most people agree that the calculation for turnover looks like this:

$$\text{Turnover} = \left(\text{Separations} / \text{Headcount}\right) \times 100$$

But the details are more nuanced than this seemingly specific definition:

What counts as a separation? Common classifications include voluntary, involuntary, retirement, death, and severance. Which to include is not widely standard and often dependent on the context of the question being asked.

Next, how is "last year" defined? Fiscal year? Calendar year? Some other date marker which is important to the leader or the context? Both separations and headcount are influenced by the period defined by the researcher and may have different interpretations.

Once time is agreed upon, how does the researcher calculate headcount? In the example, some might want to use an end-of-year headcount. Others might want to use a start-of-year versus end-of-year average. Still, others might want to use an average comprised of quarterly or monthly headcounts. Furthermore, should the researcher include the people that separated in that headcount or not? Each of these approaches has pros and cons which will affect the results.

The point here is not to teach you how to calculate turnover, but to illustrate that a researcher must be aware of many considerations for their definitions, even for a seemingly simple point-in-time question.

In research methods, going through this process when considering what will be measured is called creating an operational definition. An operational definition is the precise specification of measurement for a phenomenon which is not directly measurable. Said more simply, an operational definition is *exactly* how to measure something. In the above example, the researcher must not simply ask for "last year's turnover," but instead provide exactly what is meant in terms of what should be included for separations and how headcount should be calculated. The practical reason for operational definitions is that (1) they demand that the researcher actively choose what is and is not included in their data and (2) they allow another person other than the researcher to replicate results. If a researcher must make all these little choices about how to define turnover, then they will make them while thinking about the problem and thus the choices will make the definition fit the problem better. Also, if this definition leads to an algorithm or report, but the researcher has not documented which separations were used or how they calculated headcount, then someone else may have a very difficult time understanding why the turnover numbers look the way they do.

Great HR analytics teams define and document everything. Furthermore, they are always seeking to create alignment with other teams about how phenomena are defined so comparisons across groups can be more consistent. This has great application to machine learning because the inputs used to create models, algorithms, and insights must be well understood and well documented if researchers hope for them to add the most value and to be explainable to stakeholders.

5.4 Tables: The Language of Data

In HR analytics, the phenomenon analysts want to measure are often concepts or ideas familiar to them, like turnover, compensation, potential, engagement, hire quality, leader quality, and performance. Some of these are reasonably straightforward

and only require alignment around some easy-to-discuss specifics (the earlier turnover example is a good illustration of such a phenomenon). Others are incredibly abstract, like engagement or leader quality, and often require bringing many data sources together and are still at best an approximation of what the researcher is trying to quantify. This becomes critical to machine learning because approximation and abstraction combined with relatively small data sets can create challenges for the mathematics underlying machine learning. In Chap. 8 we will discuss latent factors and in Chap. 10 we review a concept called "The Construct Chasm"—these each review some of the challenges regarding quantifying abstract phenomenon.

For now, once the analyst or researcher has solid definitions, they must produce data which is ready to work with. This is done with tables. We use our definitions to create computer logic, queries, and other means to build tables containing data. The phenomenon defined is often contained in fields, which is another name for the columns of the tables. The ideas these fields represent are often referred to as "variables." Variables are very important in analytics and machine learning because they represent, in data form, the ideas the analysts are attempting to quantify. Quite simply, a variable is anything that is likely to change. Fields/variables are an important way for the analyst to think about measurement because it can help them ask questions in an answerable way to IT or other data-owning partners—when an analyst considers research questions in terms of what variables they are asking about, how those variables may change from case to case, and what tables they live in, the questions may become easier to define. To dive a little deeper, let us talk a little more about what a "variable" is and how can you use them to ask better questions. Take this example:

In most research, data is collected and stored in tables, which are made up of rows and columns, like so:

ID	Name	Job code	Job family	Title	Gender	Ethnicity
568214	Applegate, Joanne	D1S23	Sales	Sales Director, B2B	F	White
853667	Blackburn, Letitia	D2F22	Finance	Senior Director, Finance	F	African American
333587	Franklin, Marcus	M2M51	Mktng	Sr. Manager, Marketing	M	White
159882	Meredith, Robert	M1I74	IT	IT Manager	M	African American
952847	Remington, Bethany	M2I74	IT	IT Senior Manager	F	White
753951	Vasquez, Martin	E5C85	Eng	Lead Engineer, Supply Chain	M	Latin American/ NonWhite Hispanic

From a data or statistics perspective, a variable is just anything that varies in a dataset. And almost always in a dataset, variables are stored in columns, whereas cases (also called records, rows, or other names) are stored in rows and are what the

variables are describing. Above, each row is a person and the variables are describing their job, their gender, and their ethnicity. As such, when an analyst looks at employee 159882 they know that Robert is a male, African American IT Manager. The variables (columns) provide data about the people (rows).

However, not every data table is one row per person. Sometimes rows are other things, like job applications, training activity, or internal movement. Here is another example:

Course ID	Type	Start	Comp	Participant ID	Participant name	Test score
D58444	Mandatory	010119	012519	952847	Remington, Bethany	85%
E20000	Voluntary	022519	022519	159882	Meredith, Robert	90%
R99952	Mandatory	022819	022819	952847	Remington, Bethany	100%
R99952	Mandatory	030319	032019	753951	Vasquez, Martin	100%
D58444	Mandatory	030819	030919	159882	Meredith, Robert	100%
E20000	Voluntary	041519	041519	568214	Applegate, Joanne	80%

In this case, the data is describing training course participation. Each row is a learning event, and the variables, in this case, are describing the course (ID and Type), the timeline of the event (Start and Complete), who took the course (Participant ID and Participant Name), and performance (Test Score). Even though the information is different—courses versus people—the information design is the same. That is, each row is a thing that needs describing (a learning event) and each column tells part of the descriptive story (course name, participant, assessment, etc.).

How does this help an analyst ask better questions? The answer to most HR analytics questions come from tables since they are the language of data. So, when an analyst or researcher asks a question, it makes sense to design it in harmony with the language in which the answer will come. Let us take one more example:

ID	Name	Action	Action description	Effective date	Old job code	New job code
568214	Applegate, Joanne	Promotion	In-Line Promo	022219	D1S23	D2S23
853667	Blackburn, Letitia	Lateral	Lateral Scope Increase	032519	D2F22	D2F22
333587	Franklin, Marcus	Termination	Involuntary	042819	M2M55	—
159882	Meredith, Robert	Termination	Voluntary	052119	M1I74	—
952847	Remington, Bethany	EE Update	Name Change	061519	M2I74	M2I74
753951	Vasquez, Martin	Promotion	Promotion	070219	E5C85	D1C65

Here we can see that each row is an action of some kind (terminations, promotions, name changes, etc.). Now, let us use our knowledge to examine the quality of a question. Say the Director of HR, Mary, wants to learn about the quality of the

talent management at her organization. As such, she wants to know if people are moving around and getting new experiences so they can grow, develop, and stay engaged. She asks, "how many people got promoted in 2019?"

What challenges do we see with her question? Go back to operational definitions for a second. Mary asked for "promotions," but her original thought was to examine the quality of talent management. In this case, do we think Letitia's Lateral Scope Increase should count as part of that story? Probably. After all, any good talent manager will tell you that internal, non-vertical movement is critical to the success of talent management strategy. But because Mary asked the question without understanding how the variables are laid out in the data, she is likely not going to get all the information she needs.

Asking good questions starts with the researcher understanding how to truly quantify what is being investigated. HR people speak in terms of business operations and people outcomes, but operating costs, revenue, turnover, promotions, and others are not so easily or literally defined in data systems. In order to ask questions in a way that the analyst can be confident in the data they get back they must (1) operationally define the variables in the question and (2) design the questions in a way that they can be answered in the data.

Section Breakout: Application to Machine Learning
Operational definitions and speaking the language of data tables are extraordinarily important concepts to machine learning. In almost all applications of machine learning, the data scientist is creating datasets just like the above (albeit much larger), which must be fed into a model, algorithm, or statistical process. These datasets must be very well understood and defined beforehand so that both the data scientist and end-users understand what the results mean. The old adage "garbage in, garbage out" can be slightly modified to, "understood going in, understandable coming out"—the researcher must know all the variables, how they are defined, where they come from, how they are maintained, etc. so that the ingredients to the machine learning project create understandable, usable, and repeatable results.

5.5 Manipulate Versus Measure

The beginning of this chapter started by talking about how science is a method for observation and experimentation. To do this, researchers and analysts must define what they want to observe very specifically in terms of which variables they want, and then use the language of data tables to ensure they can get the information they need to answer the questions at hand.

There is an important classification for labeling variables which will be a useful tool to have in the machine learning toolbox: **independent variables** and **dependent variables**.

On the surface, labeling variables as "independent" and "dependent" sounds very formal, but they are named this way for good reason. An **independent variable** is a controlled input. That is, it does not vary based on the values of other variables. Their values in the dataset are what the analyst chooses them to be when designing the analysis. On the other hand, **dependent variables** are variables whose value *depends* on the independent variables for what their value will be. To simplify, think of independent variables as what the analyst is *manipulating* while dependent variables are what they are *measuring*. Let us illustrate why thinking this way about HR data is useful.

In HR, there are usually two ways independent and dependent variables show up—an analyst is exploring differences between groups or when an analyst is implementing some sort of change or intervention.

The first example is extremely common in HR outside the world of machine learning but may also be very applicable when researching your research like we discussed in Sect. 5.2.

Assume you think the Southeast Region of your company is struggling to retain employees. How would you provide evidence for that? Simply compare the Southeast Region's turnover to other regions and to the company overall. It might look something like this:

Region	Headcount	Voluntary terminations	Turnover rate (%)
Northeast	2241	493	22
Northwest	1833	422	23
Southeast	2176	609	28
Southwest	1996	379	19
Total	8246	1903	23

You may have seen a table like this before, but what does it have to do with independent and dependent variables? This table's evidence is built using four different variables:

1. Region
2. Headcount
3. Voluntary Terminations
4. Turnover Rate

This group comparison actually contains a simple illustration of independent and dependent variables. In this case, what was the controlled input (independent) compared to what varied based on that input (dependent)?

From a data perspective, we controlled Region. We did not create the Regions, but as analysts we decided that we were going to calculate turnover-*by*-Region. What did we not know for sure? Turnover. That was the reason we pulled the data in the first place. As analysts we had an idea, but until we pulled in the Headcount and Voluntary Terminations by Region, we were not sure these data would be different when comparing Regions to each other. The Headcount,

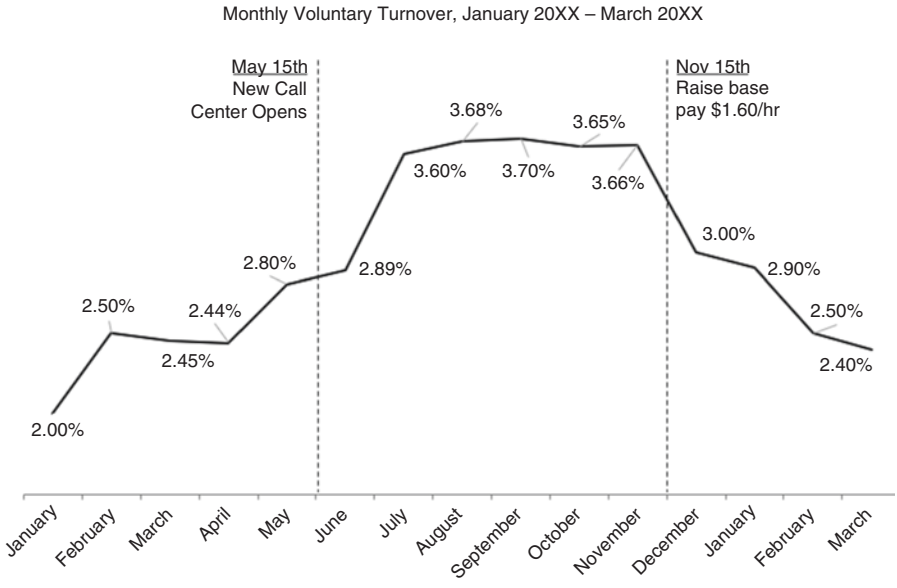
Voluntary Terminations, and therefore Turnover *varied based on the Region*, and that is what makes them the dependent variables in this case.

This is a very common descriptive use case in HR Analytics—leaders often ask to compare Group A to Group B on Idea C, so they can understand a variety of circumstances in business. Understanding which of these are independent (manipulate) versus dependent (measure) will help later when discussing how machine learning techniques leverage independent variables to *predict* dependent variables.

The other common place analysts and researchers will see independent versus dependent variables is when they are assessing the efficacy of an intervention of some kind. Let us say you are part of the HR team which manages 1000 call center employees. A competing call center opens on the other side of town, and over the next 3 months, your turnover spikes from 2.5% per month (30% per year) to 3.5% per month (42% per year). After some competitive analysis, you find out that the other call center is paying $1.50/h more for the same entry-level call-agent job and you think that is the main reason people are leaving.

After building your business case, you convince leadership and finance to agree to bump your entry-level pay rates up $1.60/h, so you can draw the level and hopefully stop losing talent. How might you show evidence that your prediction was right?

This is an oversimplified example, but your data visualization might look something like this:



Seems like an all-around success, but what were the independent and dependent variables in this case and how did you use them?

In this example, what did you manipulate? Base pay. And what did you measure? Monthly Voluntary Turnover. You made the prediction that the spike in attrition was being caused by base pay being too low in the local market. So, you manipulated it in attempt to get the outcome you wanted. This makes base pay the independent variable in your research design.

Conversely, voluntary turnover was what you measured. Did the change in pay impact the change in turnover? The answer in this example is clearly yes.

Admittedly, HR work is rarely this clean. In an example like this, turnover is likely seasonal which would add noise to the data. Manager quality, culture, and benefits all play into why people quit as well—pay is only part of the issue. And turnover data usually takes months or quarters to settle enough to observe a change like this. The point here is to help you think about what factors you are manipulating compared to which you are measuring. When an analyst or researcher can see the difference between these types of variables in descriptive and intervention-based examples like these, they will be better equipped when designing studies and when working with data scientists to build machine learning models.

5.6 Who Are We Studying?

As seen in the previous section, operational definitions for variables can extend beyond *what* is measured into *who* is measured. In the turnover-by-region and turnover-by-base-pay examples, the independent variables were groups. In the first example that was multiple groups: the Southeast Region versus other regions. In the second example, it was one group over three time periods: before the competing call center opened, after it opened, and after it opened with an adjustment to compensation strategy.

This all lends itself to helping define the phenomenon (turnover, engagement, etc.), but HR almost always measures that phenomenon within a group (e.g., a team, a business unit, an organization, an entire labor market, etc.). This is significant to dig a little more deeply into and answer: who are we trying to measure?

This is particularly critical in HR analytics because effect sizes (i.e., the meaningfulness of results) are usually derived from comparing groups to each other or a single group at two or three different points in time. Some popular categories to compare are:

- *Leader group*: People managed by different leaders in the organization.
- *Level*: People at different vertical levels in the organization (executives, frontline, etc.).
- *Demographics*: Different ways to describe your employees as people. This includes obvious ones like age, gender, and ethnicity, but may also extend to things like tenure, level of education, or others.
- *Location*: Where employees work.

- *Type of Job*: This can be functional like sales versus finance, or job responsibility-based like people managers versus individual contributors.
- *Benchmarking*: Looking at a group compared to higher level groups to create a comparison point. For example, comparing the Atlanta sales office to the whole region, or comparing turnover in your engineering function to US labor market trends of engineer turnover.

There are others, but this is a good working list. Why are these categories important? When looking for insights in data, researchers are almost always trying to find differences within or across groups because it is the differences in what is measured that indicate the researcher has found something insightful. This means understanding *what* we are studying has to be done well, but researchers must also ensure to define *who* they are studying well. To do this, we will turn our focus to two key techniques for defining groups: populations and samples.

A population is simply all the inhabitants of a group. Therefore, any group can be defined as a population. An analyst can say "all engineers" and be referring to everyone where Job Family = "Engineering." That group then becomes "the engineering population." Often times, the business likes to compare populations: Are sales employees stronger or weaker on engagement scores when compared to marketing or to the rest of the organization?

However, sometimes the researcher does not want to (or cannot) examine an entire population. In this case, what the researcher must do is look at a sample. A sample in research methods terminology is very similar to what it is in plain English: it is a subgroup, or part, of a population. Samples are hugely important in research because, in many situations, it is not feasible to capture every data point. For example, election polls use small samples to attempt to predict how the population will vote. Polling every potential voter would give the most accurate result, but it would be too expensive and time consuming to do. Instead, polling agencies sample a small subset of the population. The hope is that the sample is representative enough of the broader population to predict how the rest of the population will vote.

The difference between a population and a sample is intuitive and obvious: a population equals everyone of a given group while a sample equals only some of them. When it comes to research methods and statistics this difference becomes extremely important. In the voting example, using the evidence found when studying a sample allows the researcher to infer something about a population. "Inference" in this case is a way to use what is known to make a reasonable guess about something else. Researchers use what they know about a sample of a population's voters to *infer* that the same pattern exists in the population. If Candidate A is favored 60% to 40% over Candidate B in the sample, then they infer that they know what will happen in the actual election. In an HR example, they might sample 10% of the 20,000 employees and discover that the sales team in the sample has significantly lower job satisfaction. From that sample, they might infer that this difference exists in the whole population and thus say that the Sales population overall (not just in the sample) has lower job satisfaction. They have used the evidence from the sample to make an inference about the population.

The word "inference" keeps coming up intentionally because it is the namesake of an entire branch of statistics which we will discuss later. Inferential statistics is a field that has existed for over a 100 years and serves as the underpinning of much of modern statistical analysis, including the field of machine learning. Inferential statistics provides a framework for making informed judgments with the data that the analyst does have about data that the analyst does not have. This is helpful when the analyst can only get a sample of their population, but also when they want to make predictions about data that does not yet exist (i.e., the future). And just like earlier when discussing how important it is to define the question well, researchers and analysts must also ensure who and how they are measuring will answer the question in a valid way. This means a well-designed sample is as critical as a well-designed question.

The strategies used to design samples can get quite advanced, but as promised this text will not get too technical. Instead, we will introduce a few basic principles and techniques to get you started: random vs. systematic samples, as well as two main types of systematic techniques—pilot studies and stratified sampling. We will also briefly discuss sample size. Other techniques, like voluntary sampling, convenience sampling, cluster sampling, multistage sampling, and others are methods covered in more advanced research methodology or survey methods books. And as always, make sure to connect with analytics experts or consult more advanced texts when necessary.

The first major principle of samples is that they can either be systematic, where the researcher uses known factors to create the sample, or random where the researcher intentionally uses no guidance and chooses members for the sample from the population arbitrarily.

An example of a systematic sample would be when an analyst uses variables to divide up a population: How does the Atlanta sales office compare to other sales offices? In this case, the population is Sales, and the sample is Atlanta (note the similarity between systematic sampling and independent variable groups). An example of a random sample might be "100 sales representatives from across the company." In this case, the population is still Sales, but the sample is now "a random 100 employees."

Why might an analyst use one versus the other and how might you use them when building data for a machine learning project?

5.6.1 Random Samples: The Science Behind True Experimentation

Randomization is a critical aspect of an experiment because it is a primary ingredient to finding causality. The reason for this is that in any population, the quantity of variables a researcher *could* define is essentially infinite. Think about any group of employees at a company and how they differ. There are the obvious ones: the type of work they do, their demographics, their pay, etc. But if you dig deeper, the differences go on forever—educational background, where they grew up, their personali-

ties, their genetics, the mood they were in on the day of measurement, their cognitive ability, where they are physically located, their physical health, and on and on and on.

Obviously, depending on what is being measured some of these variables matter more than others, but the crux of the matter is that no analyst can control them all. Even if they wanted to measure every single aspect of every single person, they could not. Therefore, whenever a sample is created, the analyst or research team does not really know how these unmeasured attributes exist in the sample versus the population and how that might make the sample dissimilar to the population.

If a researcher cannot control everything (and we just established they cannot), then the best they can do is randomize the sample of people they are measuring, so they can assume that the differences which cannot be controlled are distributed similarly within the sample as they are in the population—this is why randomization is so important.

Let us illustrate with an example: A corporation has a population of 5000 sales representatives and they want to measure whether a new commission structure will increase performance. Marco on the compensation team mentions that the amount of satisfaction an individual has with the *current* commission structure will impact how a *new* structure would be perceived by them. Said differently, the impact the new system has on an individual's performance will be a function of how they felt about the old system compared to how they feel about the new one. His rough hypothesis looks something like this:

	They like the new structure	They do not like the new structure
They like the old structure	Moderate increase in performance	Decrease in performance
They do not like the old structure	Significant increase in performance	Minor decrease in performance

Marco has a point—simply saying that the new system is better overall does not mean it will be better for every individual. That said, it would not be reasonable to attempt to measure every individual's personal feelings of the quality of their company's current commission structure. The company could survey, but there is no way to get 100% response and even if they did and had somehow built a perfect survey, assuming 100% candor from employees on the topic of pay not reasonable. What do they do?

The company picks a random sample of 200 sales representatives to test the new commission structure. In a random sample, the analyst assumes the number of likers versus non-likers is distributed the same in the sample as it is in the population. It might look something like this:

	They like the new structure		They do not like the new structure	
	Population	Sample	Population	Sample
They like the old structure	1000 (20%)	40 (20%)	500 (10%)	20 (10%)
They do not like the old structure	2000 (40%)	80 (40%)	1500 (30%)	60 (30%)

Because they chose a truly random 200 people, the analyst can assume that the quantity of likers versus not-likers is approximately the same in the sample as it is in the population. And that means if Marco's hypothesis is true, then it will have the same effect on the sample group as it does on the population. And if it has the same impact, the analyst has neutralized how this uncontrolled variable might bias his results.

This brings up another important sample attribute: sample size. Sample size is the number of observations, cases, people, or replicates in a sample. In general, the larger the sample that is taken, the more accurate inferences about the population will be. That is not to say that one should always pursue the largest sample possible because often the cost/benefit trade-off does not justify the additional effort required to gather larger samples. Also, in very large populations, sometimes very large samples can accidentally magnify bias. For simplicity's sake, we will not venture deeply into sample size theory, but more information is available in textbooks dedicated to research methods and statistics.

That said, in HR applications it is most common to run into the risks associated with small sample size. In the example above if Marco only had a sample of 15 employees instead of 200, then the likelihood that he would get representation in all quadrants is much lower. In small samples, every individual matters more, which means their personal differences weigh too much on the overall results.

5.6.2 Systematic Samples: The Reality of Experimenting in Applied Settings

Unfortunately, the above example is not reality. For one, companies cannot apply commission structures randomly in a population. If Jolene and Casey are doing the same job in the same location with approximately the same skills and experience, they cannot be compensated differently—that is simply fair compensation practice. Additionally, the business processes and computer systems used to calculate and deliver pay would have a hard time with this experimental design, not to mention the social, interpersonal, and cultural impact of two people on the same team getting paid differently for the same work. Basically, organizations have a responsibility to treat everyone fairly, and true randomization of almost all HR practices would violate that principle.

For this reason, in almost all applied settings where employees are concerned, researchers must use systematic samples. A systematic sample is a sample where the analyst acknowledges that they know there are differences between the sample and the population, and control for them the best they can. Three main reasons to do this when designing research are to respect the realities of scale, to mitigate risk, and to ensure representation:

The Reality of Scale means many companies are too large to study the entire population at once. Even if they wanted to, the amount of stakeholders, moving

parts, and interdependencies are too great to realistically change something across a large population all at once. This means that organizations often choose one particular location, type of job, or other characteristics to start with.

Mitigating Risk means making changes inherently comes with risk. In the above example, what if the commission structure failed? Better to have it fail for just one region than for the whole company where the financial impact could be disastrous. Companies understand risk and scale, so often choose to hedge on the breadth of implementation for the sake of testing. In fact, one of the most common techniques which highlight these first two principles is present in almost all companies: The Pilot Study.

Section Breakout: Pilot Studies
A widely leveraged technique is the "pilot study." Pilots are used to create systematic samples, so an organization can assess impact before scaling an approach further. In practice, they are often geography-based (e.g., Pilot the new commission structure in the Southeast region); by leader (e.g., Try this new mid-year review process in Tom Anderson's group this year and see how it goes); or by type of job (e.g., Roll out the new order entry mobile app with the Senior Sales Reps first).

The advantage of this approach is that drawing the lines of implementation is logistically simple, legally defensible, and is often used to generate feedback from experts. The research design is intended to marry a reasonable amount of randomization while not ignoring the realities of implementation and the fair treatment of employees.

5.7 Ensuring Representation

The third reason systematic samples are so useful is because when designing a study there are often variables that must be represented in the sample. For example, an analyst may want to ensure a particular job code, gender, or employees with a certain amount of tenure are represented. The trouble with true mathematical randomization is that (1) there is a real possibility a random sample might not end up representative of the population, (2) the sample may be biased, and (3) the analyst must consider relevant Title VII implications. To illustrate, here is an example from a technology people likely use every day: music playlists.

5.7.1 The Shuffle Problem

An interesting piece of trivia is that the shuffle setting on music players and music software is not totally random. When the first shuffle settings came out on CD players, boom boxes, and mp3 players, the tech companies started fielding complaints they were not working properly. The companies checked and double-checked and could not find anything wrong. Turns out, that in a truly random selection algorithm

there is a very small chance that a listener might end up hearing the same song six times in a row, end up stuck on one album for an hour, or hear the same song four times in six songs. This makes sense when you think about it: "mathematically random" in plain English means, "anything is possible." And while the chance of these occurrences is small, if there is a consumer base using the software to listen to tens of millions of songs, it is bound to happen. Though it is hard to believe, companies began making their randomization algorithms *less* random, so that people would think they *were* random.

In HR samples, researchers and analysts must often do the same thing. If a team is going to test the effect of a new learning course, they may want to choose a random 100 people to test it on. But if it is truly random, they might accidentally recreate the Shuffle Problem. Truly random might mean the team selects all women, all junior employees, or all people from one office location. Analysts must ensure that all relevant user groups are accounted for, so they may start with a random sample, but then check how much representation they have from key groups, and if it is low in a particular category, randomly add participants from that specific group to ensure accurate representation of the population. In research methods terminology, this is called "stratified sampling," and is a hugely common technique to ensure samples are representative of the population.

This has a legitimate impact on machine learning design as well. In very large samples (which machine learning is best at), this is not usually a problem either because the researcher is using the whole population or because the sample is large enough that there is a very small chance of this happening (although they should always check). But a lot of machine learning in HR deals with groups smaller than the typical machine learning effort, so extra attention to the distribution of characteristics in the data can be quite impactful.

5.7.2 The Dewey/Truman Error

Another critical reason to understanding sampling is that poor sample design produces one of the biggest risks to the integrity of results: sample bias. Sample bias is when some members of the population have a lower probability of being selected for the sample than others, which adversely impacts the results. Machine learning is especially susceptible because it often uses extremely large samples and unintentional bias can impact results in ways that may not be obvious to the researcher.

A prominent example of this occurred in 1948 when the Chicago Tribune wanted to predict the outcome of the upcoming presidential election. The newspaper decided that for feasibility's sake, they were going to sample the population by randomly calling registered voters. However, they missed the critical bias that not all potential voters owned a phone. In fact, in 1948 owning a phone was highly correlated with substantial wealth. Furthermore, wealth correlates with conservativism and as a result, propensity to be republican. Unknowingly, their sample was overrepresented with conservatives and thus the results of their analysis showed a significant advantage to the governor of New York, Republican Thomas Dewey. Based

on their inferential statistical techniques, the Chicago Tribune published one of the most infamous pieces of journalism in history—a November 3rd headline reading: "Dewey Defeats Truman," when in fact Truman had defeated Dewey.

President elect Harry S. Truman, holding the infamous paper November 3, 1948

It is critical to understand the approach when sampling a population. If not, you may end up with the Dewey/Truman error. One seemingly small oversight—that phones correlate with wealth and wealth correlates with political views—can mean the difference between victory and defeat.

5.7.3 Other Forms of Bias

Sample bias is not the only form of bias that can appear while building a sample, especially for stratified sampling techniques and surveying in general, both of which predominate sampling activities in HR. Unlike a random sample of products on an assembly line that are chosen for inspection, surveys that require action by individuals can inherently be skewed based on who chooses to respond. When there are patterns in who does and does not respond, it is called nonresponse bias and it can quickly corrupt the results of a survey. This is one example, but others such as self-selection bias, survivorship, overmatching, prescreening, Berkson's fallacy, and exclusion are other types of bias. Many of these grew from the medical field. For example, Berkson's fallacy essentially posits that samples taken from a hospital are sampling from a population which is already less healthy than the general public. While this is not usually a consideration in HR, it is helpful to think about how the principle is analogous. Consider taking a sample for an engagement study from a group in a particularly toxic corporate subculture or a sample on turnover from a particularly competitive talent market. An HR practitioner may see similar challenges.

This is not intended to get deeply technical into the science of sample design, but rather to introduce the broad concept of bias and how analysts and researchers should watch for it when investigating answers to HR analytics questions. Please refer to a deeper research methods textbook to learn more about how bias can negatively impact your machine learning project design.

5.7.4 Title VII

Another critical aspect of ensuring representation from the HR perspective is consideration of Title VII of the Civil Rights Act of 1964. In HR circles, it is usually simply referred to as "Title VII" and encompasses all the principles brought into law to protect employees from discrimination based on race, color, religion, sex, or national origin. In practice, the federal group known as the EEOC (Equal Employment Opportunity Commission) as well as state or city-led FEPA's (Fair Employment Practice Agencies) enforce these regulations on behalf of the citizens of the United States[2]. All HR practitioners or data experts designing tools, processes, algorithms, or making decisions using data must comply with these laws when designing and conducting research, as well as when making decisions based on the results of research. Additionally, in many cases, they must be able to prove that a decision or process complies with these laws if the EEOC conducts an audit or investigation.

One of the most common places this appears is in the space of employee selection. While virtually all organizations have digitalized the majority of their selection process, many are now using algorithm-assisted tools, or even entirely automated machine learning based tools to prescreen applicants. This saves huge amounts of time, and therefore money, for organizations who see millions of applications a year. However, these tools must be explainable. Do they work? How well do they work? And are they fair in that they do not systematically exclude applicants based on any characteristics protected under Title VII?

Later when we review different types of machine learning, we will see that some techniques are very transparent. That is, you can explain how they work easily. Others, due to their complexity are often opaque, which means the researcher can see *that* they work, but not *how* they work. Techniques which work like this (a commonly recognized example being the neural network) pose challenges in places like selection because when the process is not clear, tuning your model (and possibly defending it) can be tough.

[2] This section is based on the governmental framework from a US perspective. Similar directives exist to combat employment discrimination in places like the European Union where the European Parliament Committee on Employment and Social Affairs oversees their versions of these laws through legislature like the Employment Equality Framework Directive and the Racial Equality Directive. Please note this text cannot serve as legal advice, but rather as an overview of the relevant agencies and frameworks to consider when making decisions concerning US employment law. Make sure to consult your legal counsel if you have questions about a specific situation or decision.

5.8 Data Privacy

A final consideration for the HR data practitioner is that of data privacy. There is no doubt that most organizations have a code of conduct which outlines how to treat customer data, but many government bodies are taking a stance on ALL data given, created, and stored by organizations. For example, in early 2018 the European Union implemented the General Data Protection Regulation in an effort to protect its citizens. Among other things, the GDPR requires that anyone who controls personal data must have appropriate technical and organizational measures to protect it. Much of this work was being done already because consumers demanded it (would you work with a bank who had a reputation for having their data breached?), but these regulations took the requirements much further and implemented significant penalties to those who do not comply. Later that same year, California signed into law the California Consumer Privacy Act with much the same purpose, and at the time of this writing, New York, Massachusetts, Hawaii, Maryland, and North Dakota all have similar laws in various stages of creation intended to provide the same protection to its citizens.

And while these laws have been primarily aimed at the consumer market to date, the data which organizations have about their employees is not excluded in many cases and is being included more and more each year. HR people handle a lot of sensitive information—home addresses, emergency contacts, personal email addresses, pay, demographics, and others. Individual employees are increasingly gaining the rights to (1) know what information organizations have about them, (2) how that information is stored, and (3) how the data is being used. Furthermore, governmental regulations are increasingly demanding that these data are collected, stored, handled, and used according to very specific guidelines which means an entirely new domain of compliance has been born in the space of employee data. And although HR practitioners are accustomed to protecting privacy and ensuring fairness, most are not well versed in the specific regulations of proper data handling. This has a significant (and still somewhat undetermined long-term) impact on the future of HR operations, analytics, and machine learning.

In this chapter, we have reviewed at a high level how to ask questions in a way that defines phenomenon exceptionally well. We have done this by researching your research and operationally defining things within the parameters of variables and data tables. We have learned about the difference between independent and dependent variables and how they influence the design of research. Finally, we spent time learning about who we study—samples and populations enable us to compare groups to each other and to themselves over time but come with many considerations when designing them.

The next chapter will shift from creating good questions, samples, populations, and variables, to understanding what to do with data once it has been sourced through an introduction to the math behind analytics and machine learning: statistics.

Discussion Questions

1. Consider a challenge you have tried to solve at a job. Break the problem down into its component parts: how many problems were there in total and how did they relate to each other? Frame these distinct pieces into individual questions or problems to solve.
2. What is the difference between exploratory, constructive, and empirical research? Give examples of when you might use each.
3. Create operational definitions of three things you would like to measure. Include all component parts necessary to define the phenomena.
4. Explain the difference between independent and dependent variables. Create two examples of something to study and define which variables are independent and which are dependent.
5. Why are systematic samples so important in HR Analytics? Name and explain four major considerations when building systematic samples.

Chapter 6
Statistics for Non-Statisticians

The other portion of the bottom section of HR Analytics Ikigai is the domain of statistics. And while statistics is not typically the preferred academic subject for HR practitioners, it is necessary to at least understand some basic concepts in order to think critically about, and have meaningful conversations with, the parts of the organization who will be handling analytics and/or machine learning projects. That said, this book is not a statistics or research methods textbook, and we will not treat this chapter as one. However, we will set some clear expectations before we dive in.

First, this chapter will help you understand the context of statistics—what makes it such a practical and applicable skillset for day-to-day life and why is it so important for good machine learning? Second, this chapter will introduce some critical concepts and explain them thoroughly enough that they can be used during exploratory activities and problem-solving exercises. And third, this chapter will frame those concepts in practically applicable examples so you can think about how they may augment exploration and problem-solving endeavors. This is not an exhaustive review of statistics, but rather a curated selection of statistical concepts which will be relevant for HR practitioners to understand.

Statistics has been one of the driving forces of scientific innovation this past century and for much of modern history. Indeed, the world of research methods helps people ask the questions, design the experiments, and collect the data, but none of that activity matters if the researcher cannot quantify and explain what the data means. Statistics, and specifically modern statistical methods, have enabled that and as such have aided the advancement of the modern scientific method which is overwhelmingly responsible for the technologically advanced world we live in. From weather forecasting to pharmaceutical testing to economics, statistics is the widely applied subset of mathematics which helps humans understand their world in a way that is more accessible and practical than almost any other branch of mathematics.

But what is it that makes statistics such an accessible part of mathematics? Unlike vector calculus or trigonometry which are quite complex (important, yes, but dense), statistics in its basic form is incredibly graspable for most people. At

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_6

its core, statistics is just collecting, organizing, analyzing, and understanding the messages found in data. In fact, data has been collected, tabulated, and used to solve problems for millennia. People use statistics every day. Sure, they do not run a t-test to decide what to eat for dinner, but let us take a practical example: you know your spouse hates spinach and is allergic to shellfish. Then, your friend tells you she is making Shrimp Florentine for your double-date at her house next Friday. Based on those three pieces of data, we bet you can make a prediction of how your evening will go given the current menu. This, in its most basic form, is statistics. How do you intake and analyze the data: what your spouse eats versus what is being served, and then infer the outcome: an embarrassed host, a hungry spouse, and a generally poor time had by all. Most importantly in this example, statistics allows us to then make inferences and intervene based on what the data shows us. By communicating that data to your friend, she can change what she cooks and avoid the situation entirely. Statistics enables intervention to create better outcomes.

Statistics gets far more complicated than navigating dinner dates. As recently as the late nineteenth and early twentieth century, what is known about modern statistical methods has arisen and increased science's ability to wield this branch of mathematics. By building on probability theory and concepts of experimental design, modern statistical methods provide a framework for analyzing and interpreting the world around us. Thanks to these approaches, analysts and researchers can take the research methods from earlier in this book and use math to provide insight as to whether or not our ideas are supported by the evidence collected. This concept has extended to nearly all fields of industry, addressing uncertainty in the world and encouraging the use of data and observations instead of anecdote and intuition to solve problems.

The power of modern statistical methods lies in their ability to help make mathematically grounded inferences about the world. Starting with a set of observations or samples, one can make generalizations (inferences) about the broader population, and sometimes even predict future outcomes. We reviewed the basic premises for how to do the work of creating these data in Chap. 5. Statistics' part of this story comes after the research methods drive the creation of data and the information needs to be analyzed. This is done using the two main kinds of statistics: descriptive statistics and inferential statistics. These relate nicely to the analytics types from Chap. 2: Descriptive, Predictive, and Prescriptive. Prior to harnessing data for prediction and prescription (which is the realm of inferential statistics), it makes sense to first understand the essential features, patterns, and limitations of the data, which is the realm of descriptive statistics, known most usually simply as "descriptives."

6.1 Introducing Descriptive Statistics

Datasets are like anything else—they have characteristics which make them unique. If someone were going to describe a person, they would talk about their hair color, their height and weight, and whether they had any noteworthy features (like big ears or an eyebrow piercing). If they were describing a car, they would talk about the color, body style, and the interior. Datasets are the same. There is an important set of characteristics and methods used to describe data. Analysts must understand them, at least at the basic level, because these characteristics not only help to do basic observations and pattern recognition in data, but they also inform what the data can and cannot be used for. For example, if someone described a low-riding, 12-cylinder, bright red sports car, you might be very excited to get behind the wheel. But if that person then said you were going to go off-roading with it, how would you react? The sports car sounds great and you could have a lot of fun with it, but probably not in the woods. In this way, descriptive statistics help describe two major things about a dataset: (1) the data's characteristics (what kind of car are you driving?), and therefore (2) what the data is suited for (where can you drive it?). Let us get into these concepts a little more deeply.

Descriptive statistics by their nature generalize and simplify, which means detail is lost in the process—just like a person is not merely the sum of their big ears and eyebrow piercings. That said, those descriptions are helpful if someone is trying to pick them out in a crowd or help someone get an idea of what they look like in general. Doing this requires a conscious sacrifice of some level of detail for the sake of brevity and clarity. For example, Bill could read through 1000 employee scores on an engagement survey, but he probably would not be able to remember everything, and he also would then not be able to explain it simply to everyone else. Furthermore, that level of detail does not help Bill see any emergent patterns. Basically, analysts need these simple descriptive characteristics so they can understand and describe datasets in a parsimonious way.

Descriptives are used to explore and understand the characteristics of a dataset so that the analyst can describe it and even make basic observations and inferences. Many common descriptive statistics and techniques which leverage them allow for things like trend identification, summarization of patterns, and differences between groups—all of which can lead to the extraction of basic insights.

Now, it is very tempting to take descriptive statistics to the bank as currency for making decisions. And while this is often done, we must realize that descriptive stats are purely descriptive. They do not "prove" anything. They simply describe historical data. They do not say *why* a pattern or trend exist and they do not show causality in any way.

Section Breakout: Confirmation Bias
Sometimes descriptive statistics are used to prop up preconceived notions. Chapter 5 referenced many types of bias, and there is an important one to talk about which is specific to data interpretation: Confirmation Bias. Confirmation Bias is a phenomenon where a preexisting opinion about something causes one to unintentionally look for evidence which supports their theory and ignore evidence which refutes it. This is one of the major watch outs of using descriptive statistics. While they are often well-suited to describe a situation, and triangulating lots of descriptives may even lead to a preponderance of evidence[1], researchers and analysts must be vigilant in not overweighting descriptive statistics in their own minds, especially when they "support" the point of view they started with.

Despite their inability to show causality, descriptive statistics are very useful. Say Lesley is trying to budget her family's income. She wants to know how much money she spends on groceries, gas, rent, and the like. She could look at last month, but will that tell the whole story? What if she and her family took a road trip to her mom's? She spent a lot on gas, but mom fed the family for 2 weeks, so they bought fewer groceries than usual. Maybe the month before that then? Well, that month Lesley hosted two parties at her house, so the groceries were out of control, but she did not go anywhere so gas was cheap. The idea is that one data point never tells the whole story. But what is the question Lesley is trying to answer? She does not care about any specific month, she wants to know how much is spent during a *typical* or *common* month. She may want the average or want to know what the most usual amount of money spent in a category is. She may want to know how much her spending varies each month, or what she is likely to spend this year overall.

These core concepts of descriptive statistics appear frequently in our daily lives. Many of the core ideas are immediately recognizable, easy to understand, and can serve as a common denominator when describing data and sharing insights in various settings or to broad audiences. These descriptive statistics and their methods can be grouped into three distinct categories: measures of central tendency, measures of variability, and measures of relative position.

[1] In legal terms, a preponderance of evidence is the greater weight of the evidence required for a judge or jury to decide in favor of one side or the other. Basically, this means there is enough evidence on one side to make a reasonable judgement. In business, analysts must often piece many data points together to essentially do the same thing when making choices in HR.

6.2 Measures of Central Tendency: Describing Data by Calculating a Center

Measures of central tendency are the most commonly used and widely understood methods of descriptive statistics. They summarize a dataset by calculating a center point and in this way, they seek to provide a number that represents the most typical or central characteristic of a dataset. The most familiar of the descriptive statistics is average (also known as mean) because it is by far the most common. However, there are several other statistics that can be used to calculate a central value and they all have different strengths and weaknesses. There is no one statistic, one measurement, or one technique which fits all scenarios. All statistics have their pros and cons. What makes a savvy builder and consumer of statistics is knowing which statistics to triangulate to understand data best. Knowing how to do this comes from the combination of (1) understanding which statistics do what and (2) how they apply to the specific question at hand.

6.2.1 Average (or Mean)

The average, also known as the mean, is probably the most used statistic in popular culture today. It is (over)used frequently in a variety of fields and contexts. Average is a powerful statistic in that it has the ability to summarize data simply and in a way that is very easy to understand. Average is calculated by dividing the sum of a set of values by the number of data points. For example, if the grades for five student tests were 97, 90, 85, 84, and 84, the class average would be 88:

$$(97 + 90 + 85 + 84 + 84)/5 = 88.00$$

Simple. Easy. Straight to the point.

Although the average is the most widely used measure of central tendency, the average does have very significant limitations. Unfortunately, in common usage the term "average" if often conflated with the word "typical," when they can actually be different things. This premise has even permeated everyday language:

- That guy's an average Joe.
- Is Tommy a good student? I would say he is a little above average.
- How was the movie? Average.

Average *can* mean typical. But the data needs to meet a very specific set of criteria for that to be true. An average can be heavily influenced by extreme values (also known as outliers). If outliers are not evenly represented on both sides of an average, the average will not be an accurate description of "typical." Furthermore, if there are too many outliers, even if they are symmetrically distributed, then average will give a false picture of "typical."

Revisiting the prior example, if a sixth student did not take the test or make it up and received a score of 0, the average would fall from 88 to 73.

$$(97 + 90 + 85 + 84 + 84)/5 = 88.00$$
$$(97 + 90 + 85 + 84 + 84 + 0)/6 = 73.33$$

That is a grade-and-a-half shift in average! The average can accidentally be very misleading and overly simplistic in describing a set of values. Later, when we discuss distributions, we will provide more examples of how data can vary in ways where an average does not do a good job of describing data.

6.2.2 Median

Another common measure of central tendency is the median. The median seeks to find the middle point[2] in a set of data values. To remember this, think of the median on a highway—the line or physical barrier which divides the road in half. Why would an analyst or researcher want to report a median? The median answers the question: what is the true middle of the data? At what point are half the scores higher and half the scores lower? This sounds quite a bit more like "typical."

Using the data from the prior example, see that the median of the grade data is 85. This is close to 88, so it shows in this case both average and median are decent statistics to represent "typical." However, when we add in the outlying 0, the median only shifts from 85 to 84.5! This is because one data point, or even a group of outlying data points, does not influence the median like it does an average.

97 90 85 84 84 Median = 85
97 90 85 84 84 0 Median = (85 + 84)/2 = 84.5

In a nutshell, this is the major advantage of median over mean: it is robust to outliers. As median finds the middle value based on sort and rank, a small percentage of high or low values will not distort it. This is useful when the dataset has significant outliers or skew (which we will discuss later in this chapter).

That said, the median does not consider the total of the dataset at all. When doing analytics involving salary, for example, leaders often need to know the total (e.g., dollars spent, allocated, or forecast). In an example like, "how much do we pay the typical employee," a median will help get at the middle of the dataset but does not have any relationship at all to how much is spent overall.

[2] If there are an even number of data points, there will be no single middle value. In this case, the average of the two numbers closest to the middle is the median.

6.2.3 Mode

Finally, the mode is the least frequently used and least understood measure of central tendency. The mode of a set of values is calculated by determining the value that occurs most often. In the grades example, 84 was the most common grade received by students, which makes it the mode. It is possible to have multiple modes if there are multiple sets of numbers that occur with the same frequency. If all values are unique, there is no mode.

97 90 85 84 84 Mode = 84
97 90 85 84 84 0 Mode = 84

A limitation of the mode is that it does not have to have a relationship to the center of the data. That is, the most common values may be far away from the center. The fact that multiple modes may exist for a set of values can also be confusing and difficult to describe cleanly, so modes are not often used for numeric data. However, the mode can be interesting if the potential values are discreet, meaning they are limited to a finite number of possibilities. For example, if an office manager were conducting a survey about what people would like to order for lunch (pizza, tacos, or shawarma) or voting on a peer to receive an award (Jola, Dave, or Allie), then the mode would be the statistic to look at. It can also be helpful when dealing with what are called "multi-modal" distributions, where values in a dataset are bunched together in different places (also discussed more in Sect. 6.4).

6.3 Measures of Variability and Measures of Relative Position: Describing Data by Summarizing How Data Points *Differ* from the Center

The second and third types of descriptive statistics are measures of variability and measures of relative position. They build on the measures of central tendency because they provide insight into how data is distributed. Whereas Central Tendencies are concerned with bringing all the data together and describing a center of some kind, Measures of Variability and Measures of Relative Position do the opposite—they are concerned with where data points are relative to each other or relative to the center. How data points are distributed across a dataset tells a lot and helps analysts understand what they can and cannot use the data for. Going back to the car example: if someone was trying to explain a new car to you that you had never heard of, how would they do it? They would explain how that car was similar *and different* to cars you are familiar with. In fact, this is how people describe new things in most scenarios. They frame descriptions by comparing similarities and contrasting differences to frameworks and examples people already understand. Measures of Variability and Measures of Relative position do the same thing. They use something static (like a center) or something

known (like other data points in the dataset) to create comparisons which summarize the data.

6.3.1 Range

The range provides insight into the overall spread of a set of values. It is calculated as the difference between the maximum and minimum values within a dataset. Using the earlier example, the range would be calculated by subtracting the maximum value (97) from the minimum value (84), resulting in a range of 13.

97 90 85 84 84 Range = 97 – 84 = 13

The range can be useful for understanding the breadth of values. It is great because it is simple to understand and gives a great picture of the outer limits of data. This makes range ideal as a summary statistic for things like objective performance metrics or salary. However, it is very sensitive to outliers. An extreme low or high value in the dataset can make a range very big and suggest that the data are spread less centrally than they are. Add that 0 back into the example:

97 90 85 84 84 0 Range = 97 – 0 = 97

Therefore, range is usually best when paired with other statistics (like median, mean, variance, or standard deviation) so that a very high range does not miscommunicate the centrality of the dataset.

Section Breakout: Quartiles and Percentiles

An important concept often used when describing data is the idea of a percentile or quartile. When data is rank ordered from highest values to lowest values (like you would do to calculate a range) that data can then be segmented into chunks. If there are four chunks (which is most common), they are called quartiles. Many are familiar with this concept via their experience taking standardized tests and receiving a result expressed as a "percentile." If someone is in the 75th percentile, that means they scored higher than 75% of test takers. To visualize this, imagine 1000 people who took the SAT all came together on a football field and got in a line from the highest scorers to the lowest scorers. If a test taker was in the 75th percentile or higher, they would be standing with the first 250 students in line. Their score puts them in the highest-performing 25%.

This idea is important because when we talk about distributions in an upcoming section, we will talk about the probability of scores landing at different percentiles. In a perfect scenario, the distribution of scores follows a very predictable pattern, but this does not always happen. When this is the case, descriptive measures like range help the analyst understand what a distribution looks like. This is critical because the size and shape of a distribution is ultimately what tells the analyst about the probability of having a particular score, which helps them know which sorts of statistics are appropriate for use in analyses.

6.3.2 Variance

Another useful measure of variability is variance. Variance is calculated by taking the average of the squared difference of each value from the mean. Said more simply, variance compares every data point in the dataset to the average and gives one number to explain how far apart the data points are. If variance is very high, that means the data points are very far away from the average. If variance is very low, then all the data points are very close together.

Note that there are two formulas[3] to calculate variance depending on whether the data represents a sample or the full population, though for the purposes of this book we do not need to get into the details of why. Also, almost all spreadsheet and statistical software will calculate variance for you with a simple formula, but almost always assumes the data comes from a sample.

How might an analyst want to use variance? Here is an example of a job satisfaction survey. Consider these two sets of 20 scores from Company A and Company B:

Scores from Company A		Scores from Company B	
90	80	98	86
89	78	97	85
89	78	96	83
89	76	96	82
89	76	95	62
88	75	92	56
88	75	92	55
84	70	92	52
81	65	90	51
80	60	90	50

Since this example data is relatively small and sorted by overall score, the differences in the data might be more readily apparent than in a real dataset with hundreds or thousands of data points. For example, we can see the difference between the highest and lowest score is 30 points for Company A, but 48 points for Company B. This range tells us that the difference between the most and least satisfied employee is much bigger in Company B. We might also see that in Company A, the least satisfied 25% of employees are scoring in the 60s and 70s, while in Company B, the least happy employees are all below 60 and score as low as a 50!

Most HR practitioners can likely see that there is a significantly sized disgruntled population in Company B. If 25% of people are this unsatisfied, there is much work

[3] In a population, the denominator of the variance equation is N (where N is the number of data points), but if calculating the variance of a sample, then the denominator of the equation is N-1.

to be done. But what is obvious to the naked eye in this dataset of 20 might get masked in a bigger dataset if we only used measures of central tendency. How? If you simply looked at a mathematical mean, these two companies have the exact same score—the average of both these datasets is 80! And in this case even the median does not help much—it shows that Company B actually has the higher median (median for Company A is 80, while for Company B it is 88).

This is where variance and other measures of variability help. The population variance for Company A is 69.2 while the population variance for Company B is a whopping 305.3! This shows that even though the average is 80, there are many employees whose score is nowhere close to 80. This would raise a huge red flag for a practitioner who could then say, "Even though on average the score is 80 (mean) and the midpoint is 88 (median), where is all this variance coming from?" They could then dig in and find that there is a significant population who needs attention. In this and many cases, variance provides a single number which indicates a much bigger problem that needs investigation.

6.3.3 Standard Deviation

The most useful measure of variability is standard deviation. Standard deviation is calculated as the square root of the variance. Why would an analyst do that? That sounds like variance with extra steps. Here is the main reason why standard deviation is so useful to the HR practitioner: it explains variance using numbers that are similar to the scores in the dataset. In the previous example, the variance for Company B was 305.3. And while a practitioner can tell that is much larger than 69.2, by itself it does not mean much when engagement scores range from 0–100. It is just not a very practical way to describe your data.

On the other hand, standard deviation transforms the variance into a number that is similar to the data. We like to call it the "plus or minus" metric. In the example, the standard deviation for Company A is the square root of 69.2, or 8.3. This means that the average score is 80 and deviation from that average is usually about 8.3 points. Do you see where the name comes from? The standard (or typical) deviation from average is 8.3. So, if the data is normally distributed (a topic we will get to soon) then an analyst can say, "On average the engagement scores are 80, plus or minus about 8 points."

Conversely, the statement for company B would be, "On average our engagement scores are 80, plus or minus 17 points." On a scale of 100, that means that even though the average is 80, it is very common to waver up to 17% in either direction! Standard deviation raises the same red flag that variance does, but does so in units that will make more sense to talk about.

6.4 Let's Get Visual: Distributions

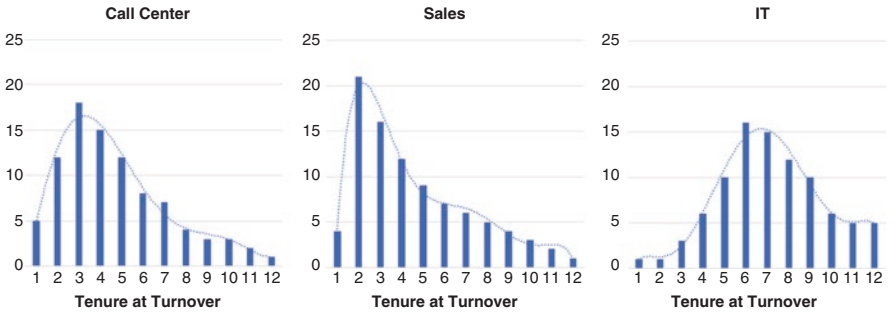
Another way to summarize data is by using what is called a frequency table. A frequency table summarizes data by showing how many times a particular score occurs. Instead of one row per data point, a frequency table shows one row per potential score. The rows can represent all possible values or can be grouped into buckets (also called binning) based on information the researcher already knows about the data's ranges or the distribution overall. A researcher may also choose different bins simply to explore if any patterns emerge. Here is an example of Tenure at Turnover:

Tenure at turnover (months)	Call center	Sales	IT
1	5	4	1
2	12	21	1
3	18	16	3
4	15	12	6
5	12	9	10
6	8	7	16
7	7	6	15
8	4	5	12
9	3	4	10
10	3	3	6
11	2	2	5
12	1	1	5
Total	90	90	90

This table shows that each job function group had 90 terminations in the first year. But the way that these turnover numbers are distributed is very different across that first year. In the call center and in sales teams, most of the turnover happens within the first 2–5 months, whereas in IT the turnover happens more often in months 5–9. This sort of table is easy to create in most data or statistical software programs and is usually called a pivot table or cross-tabulation describing data.

Frequency tables like this are often visualized in something called a frequency distribution, histogram, or bell curve (due to the shape the curve often takes).

Histograms estimate the probability of distribution for a continuous variable. Said more simply, a histogram uses the height of the bars to show how often something occurred. By comparing the bar heights to one another, the analyst can use the visual support to provide insight not only into the range of values but also to how values are clustered or distributed. Here is how the table above would look in histogram form:



For this visual, the metric Tenure at Turnover is on the horizontal axis, while on the vertical axis the frequency of turnover at different tenures is plotted. This means that all the way to the left of each graph is the date employees were hired, and as the graph moves right there is more tenure. The higher the bar is, the more employees quit when they had that tenure. In this way, the shape of the curve tells an important story about the relationship between tenure and turnover for these different teams.

This is a very important type of graph because analysts and business leaders can use it to see when something is typically occurring. In this case, if an analyst wanted to combat turnover in the call centers by having managers conduct "stay interviews," when should they do it? If they used the average, they would think that around 5 months makes sense to have the conversations because the average of the above dataset for the call center department is 4.7 months. But looking at this graph an analyst does not need to do any calculations to see that most people quit around 3 months of tenure. In fact, with a little calculating they would be able to see that by month 5 almost 70% of the people who are going to quit in the first year have already left! How might this graph help an analyst tell that story? And how would that story change if they were conducting an analysis for the IT department?

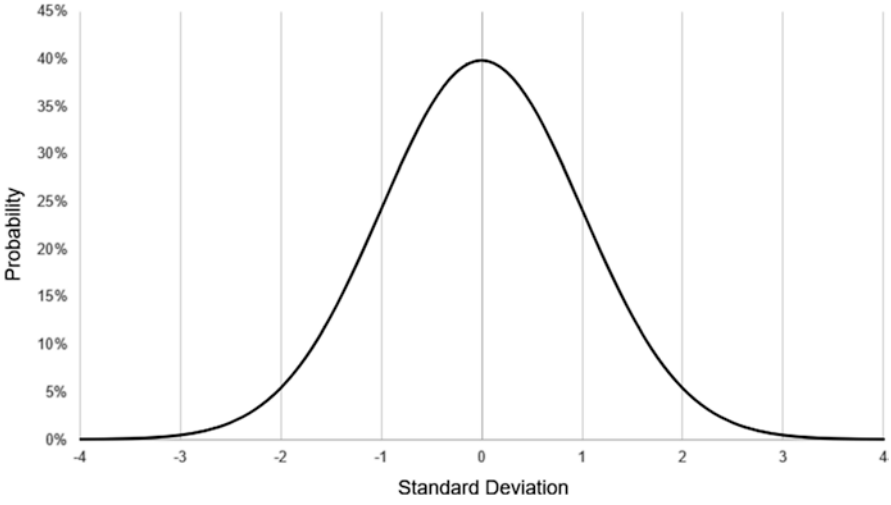
Histograms are one of the most useful charts in descriptive statistics because they show the distribution of data in a very simple way. Here are some other fundamental ways they can be applied in HR:

HR domain	Example 1	Example 2
Talent acquisition	Age of open requisitions	Requisitions closed by week or month
Employee relations	Resolution time of in-bound employee complaints	
HR operations	Service level times for employee transactions in an HRIS	
Learning and development	Training hours completed by employee (sorted high to low)	Training dollars spent by month (potentially filtered by platform)
Compensation	Distribution of salary, bonus, total compensation, Percent in Range, or other measures of dollars-paid	Distribution of labor dollars by month
Workforce planning	Headcount distribution by level	

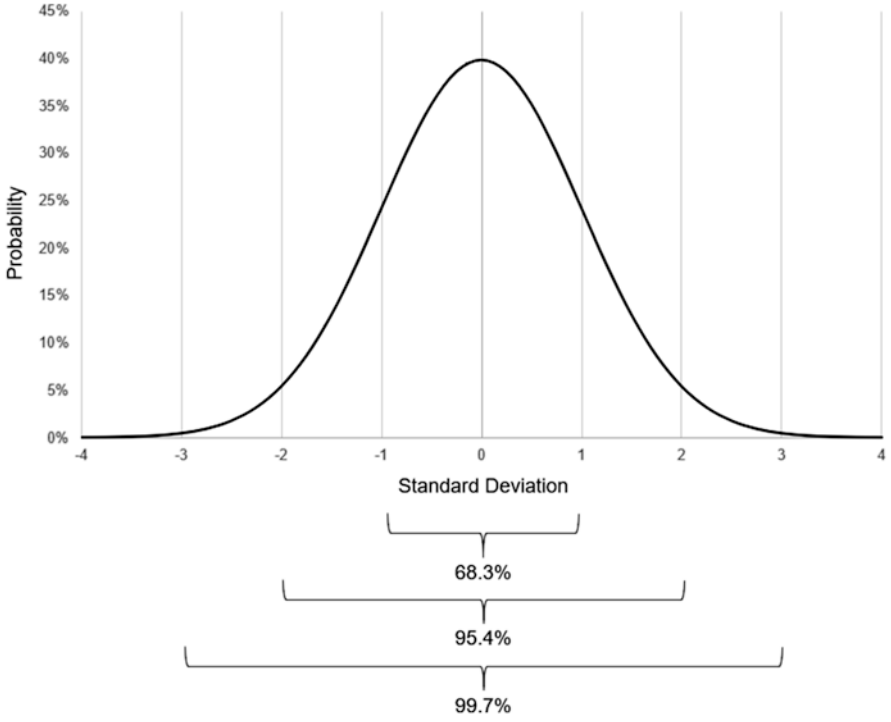
6.4.1 Why Bell Curves Matter

We have established how plotting a histogram can create a bell curve to show how data is distributed in a dataset, but to the practitioner the question always comes back to: "so what?" Why does the shape of a bell curve matter to practitioners? And why does it matter to machine learning?

A bell curve with a large enough sample on a well-defined metric is essentially a visual of probability. That is, the more space you can see under the curve, the more likely it is for a data point to have that value. The middle of a normal bell curve is where most scores fall, and as data gets farther away from that mean, median, and mode (they are all the same in a normally distributed curve), the results are less common. In a normal bell curve, there is a nice smooth departure from average:



The best part about a normal curve is that an analyst can tell what percentage of cases exist on each side of the average. This tells them about the "commonness" or "rareness" of a given score. The percentages look like this:



In a normal curve, 68.3% of all scores are within one standard deviation above or below the mean. Earlier when we discussed standard deviation as the "plus or minus" metric, this is what was meant: a bell curve shows visually that more than 2/3rds of scores are plus or minus one standard deviation from the mean.

Additionally, it allows the analyst to calculate any given score in the form of a percentile. If they sum up the percentages below the score, they can tell what percent of scores are above or below any given data point. Here is a well-known example:

Traditional IQ has an average of 100 and is normally distributed with a standard deviation of 15. Thus, we can say that 68.3% of people have an IQ between 85 and 115. This means, if we gave a truly random 1,000 people an IQ test, 683 of them would score between 85 and 115. Also, if we sum up the percentages from right to left we can say that if Amanda has an IQ of 116, she has an IQ higher than 84% of people:

00.1% with an IQ more than 3 standard deviations below the mean
02.1% with an IQ 2 to 3 standard deviations below the mean
13.6% with an IQ 1 to 2 standard deviations below the mean
34.1% with an IQ 0 to 1 standard deviations below the mean
34.1% with an IQ 0 to 1 standard deviations above the mean

84.0% of people have an IQ of 115 or below

As a practitioner, this is useful because if an analyst is investigating anything and produces a distribution, they can make reasonably sound inferences about how that metric exists in the population. Here is a good exercise for those who work in a medium or large company (and are authorized to see these data): create a histogram of the compensation metric for Percent in Range for employees, rounded to the nearest tenth of a percent. You will likely find a normal distribution!

This applies to machine learning because it applies to the statistical concept known as "assumptions." Whenever an analyst is going to employ a statistical method, machine learning or otherwise, there are assumptions about the data that must be met. Assumptions are characteristics of the data that must be true in order for a statistical process to work properly. Think of these as rules or prerequisites which must be met in order to use that statistic. All the descriptives discussed so far describe characteristics of data and certain conditions must be met if an analyst plans to use the data in certain ways.

This is not a statistics book, so the objective is not to get into all the details of which assumptions are meant for which sorts of statistical processes. At a high level, it will be helpful to remember that concepts like normality, linearity, and equality of variance are the most common assumptions and that if data is significantly departed from any of these concepts, it is cause for pause.

- *Normality*: Variables adhere to a normal distribution
- *Linearity*: Relationships between variables can be graphed on a straight line
- *Equal variance*: Different samples may have different means, but their overall variance should be equal (also known as homoscedasticity)

Remember, not all data are suited for all manners of math. Researchers and analysts must do their homework, talk with their experts, and make sure that their data can be used in the way they want to use it.

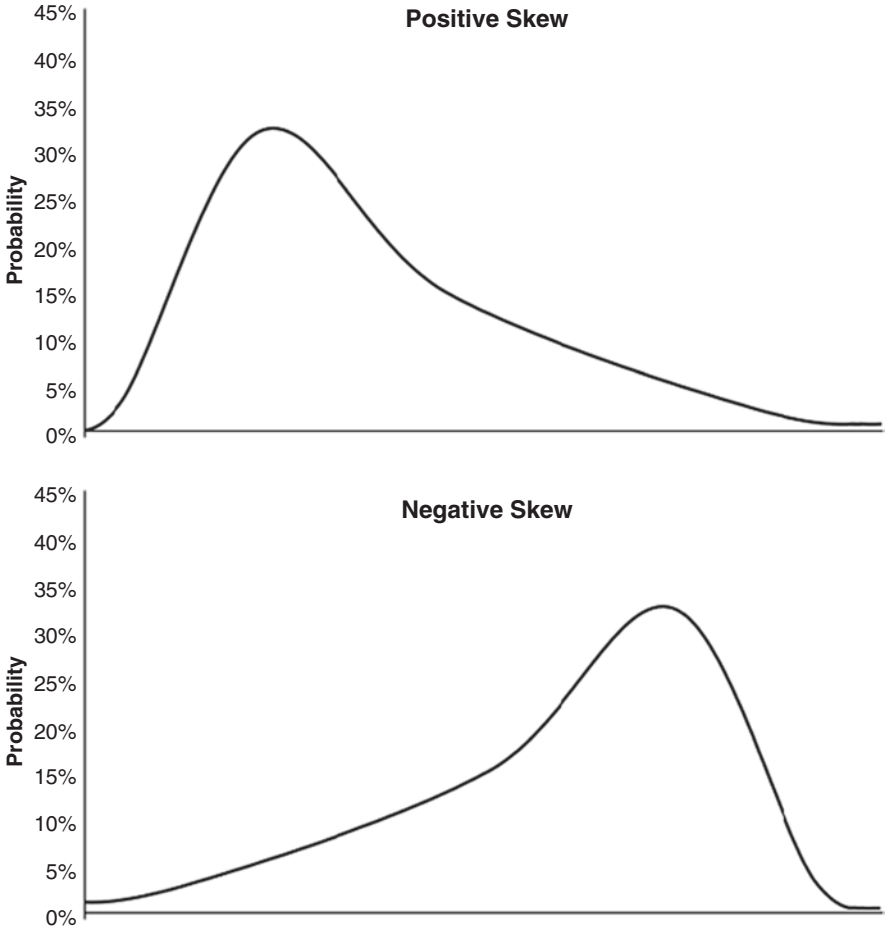
But what does not-normal look like and how will an analyst know if something is wrong? Let us introduce a couple of ways for an analyst to describe curves, so they can tell when they *do not* look normal. This will not go deep into the mathematics behind these shapes but understanding the basics of what exists in general will be helpful. If an analyst or researcher is getting deep into using bell curves, it is a good idea to consult with a more advanced text or an analytics department, so they can provide guidance.

6.4.2 Skew (Leaning Left or Leaning Right)

Skew occurs when something happens more often to one side of the peak of the graph than the other. Another way to say this is that the outliers in the data occur more often on one side of the peak of the curve. This is called skew and occurs in a great many types of HR data. Skew can make the graph "lean" left (positive skew) or "lean" right (negative skew).

In HR, skew often happens because of something called the Floor Effect or the Ceiling Effect. Floor Effect references the fact that something cannot happen before a certain spot in data. The above example of Tenure at Turnover is a great example of the Floor Effect. People cannot quit before they are hired, so the lowest value here is 0 months. However, they can quit *any time after that*, making the right of the graph essentially infinite. This means that outliers in these types of data will always be greater on the right side of the curve, which makes the curve look like it "leans left." More formally this is called "positive skew."

The opposite also occurs. The Ceiling Effect is when data has more values on the left side of the curve, which gives the appearance of "leaning right." Engagement survey data almost always has this type of skew. Like the earlier example used to explain variance, let us say the scale for the survey is 0–100 with 100 being perfect. A few people might give a 95+ score, but most will not. The majority of scores are then bunched somewhere between 80 and 95, and the majority of outliers existing somewhere between 25 and 80. This makes the "tail" longer on the left (also called "negative" skew).

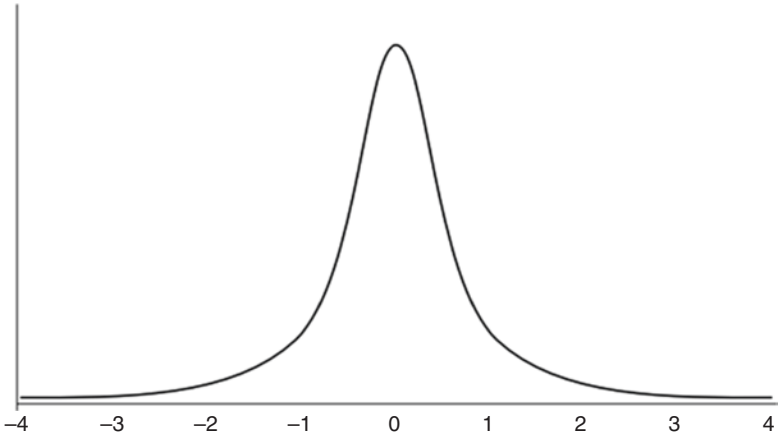


6.4.3 Kurtosis (Fat Curves and Skinny Curves)

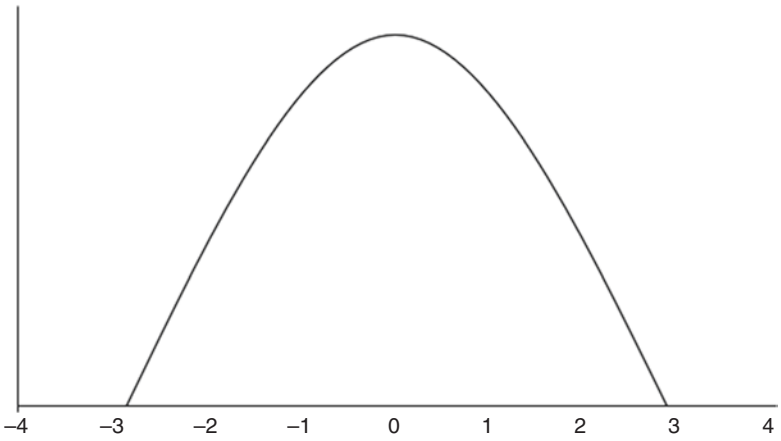
Kurtosis sounds fancy, but in fact just refers to the shape of the "bell" in the bell curve. Whereas skew looked at "leaning," kurtosis looks at where the data points fall as they move away from the average.

Extreme kurtosis means the curve does not follow a normal distribution pattern. There are two types of kurtosis which sound even fancier: leptokurtic (skinny curves) and platykurtic (fat curves).

A leptokurtic curve means that the data has extra outliers. It means many of the scores are bunched around the mean, but the outliers make it looks "pinched," like so:



When the bell gets "pinched" in, it gives the appearance of being skinny, and this happens because the tails on both sides are extra-long. That is, the extremely high or low scores at the tails approach a 0% probability of occurring much more slowly than in a normal distribution. The important takeaway here is that the probability of a score occurring which can usually be assumed from a normal curve does not apply.

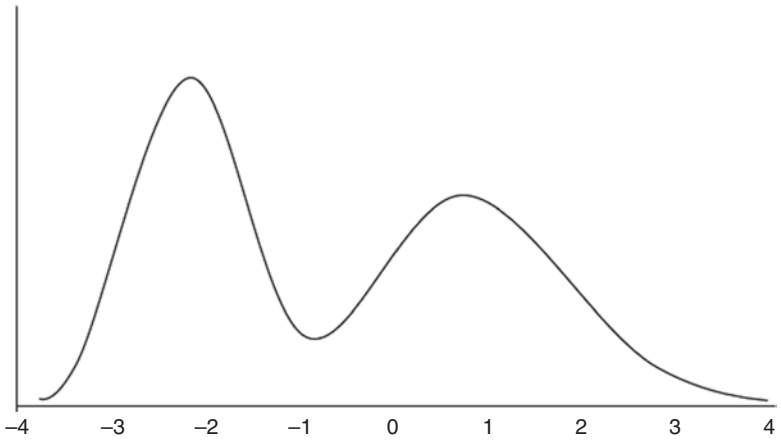


Platykurtic curves are the opposite. In this case, instead of a "pinch" in the shape, there is a fatness with no curve down into the tails. In this case, instead of having too many outliers, the data now has too few outliers. In this case, there is not enough variance away from the mean and the outliers stop abruptly. Though the problem is the opposite, the takeaway is the same: the probability of scores from a normal distribution cannot be assumed.

The first point to take home about this, which has already been discussed, is that if data has too much skew or kurtosis, then many descriptive statistics will not work properly. Specifically, in highly skewed data, average and range may be misleading and if the skew is extreme enough, even median can be affected. Significant skew or leptokurtic kurtosis will also inflate standard deviation. So as an analyst begins to use descriptives to describe "common," they must ensure their data does not have too much skew or kurtosis.

The second and most important point for a machine learning practitioner is that many types of statistics work only when all assumptions have been met. In some cases, data must be normally distributed for models to work and will not produce reliable results if their data has too much skew or kurtosis! For example, a common advanced statistic used in many types of machine learning is regression (which we will discuss later). However, regression does not work properly on skewed data. Analytics teams can help control for this with concepts like logarithmic transformation or by using other statistics which are not adversely affected by skew. However, when venturing into the realm of advanced stats, understanding the shape of the curves and other factors about the distribution of variance is critical to the validity of the models you produce. Always check and engage experts if something seems abnormal.

The final type of unusual curve to explore is what is called a multi-modal curve. Since mode is the number which occurs most often, then it makes sense that a multi-modal is a curve with many modes. Importantly, multi-modal refers to a curve with two or more distinct peaks. This makes the data look like a rollercoaster:



The challenge with multi-modal curves is that it usually means at least two different factors are impacting the probability of where the score lands. An example might be a distribution of salary of two groups with different market baselines or a time-to-fill curve for a group who fills both hard-to-find as well as easy-to-find talent. Just like with skew and kurtosis, a multi-modal distribution is a cause for pause, because many statistical methods are not prepared to deal with multi-modal distributions.

6.5 Getting Started with Inferential Stats

The statistics focus so far has been on descriptive statistics to summarize data and how to visualize them in a way that can give clues to their normality. However, earlier we referenced that we do not always have all the data we need. Sometimes datasets are incomplete, or they are only a sample of the total population. This is sometimes because all the data are not accessible, or because the analyst or researcher is trying to predict what will happen in the future, so the data literally do not exist yet. In these cases, is it safe to assume that the relationships observed in descriptive statistics are accurate? If not, how do analysts make the leap from observations about a sample of data to observations about the population as a whole or predictions about the future?

Other times, an analyst is trying to investigate whether there is a true relationship between two or more variables. Do high amounts of Factor X really influence whether someone is better suited for Job Y? And if so, is the relationship strong enough that the business should allow candidates' levels of Factor X influence whether they are selected for Job Y?

At the end of the day, these are all just ways to make a prediction, but this idea is a common topic that HR practitioners must navigate when working with data. When

considered from a results angle, stakeholders often pose this concern in the form of a question: "is that relationship statistically significant?"

To illuminate this idea, let us go back to populations and samples for a minute. When Marco and his colleagues were testing their new compensation structure, they wanted to see the effect it had on a *sample* of 200 employees. The *population* in that example was actually 5000. If Marco and his colleagues had given the new structure to all 5000 employees, they would know exactly how well it worked because they would have given it to everyone in sales. For reasons already reviewed, this is often not the chosen approach, so they used a sample. However, the trade-off of using a sample is that Marco does not know for sure what the other 4800 employees would do. He must use the data from the 200 to infer the effect it will have on the other 4800.

The threshold used to decide if an effect exists in a population based on how the effect appears in a sample is what is called "statistical significance."

If the results do not reach that threshold, then typically it is assumed that the differences seen in the data were due to random chance or noise in the data and probably do not apply to the population. This is a critical differentiation to make in HR data because HR studies often look at data about an entire population, not samples. And when looking at a population, "significance" takes on a different meaning. Here are three scenarios to always keep in mind:

Scenario	Example	Do you need statistical significance?
Studying a sample	Testing a skills training program on 100 of the 1000 high performing call center agents to see if it increases performance.	Yes—the results will need to extrapolate to the other 900 employees, so the data must infer significance using statistics.
Studying a population to understand something happening now	Looking at this year's compared to last year's engagement survey scores for the 250 Supply Chain employees at a company.	No—in this case, the difference is for the populations, so whatever difference exists is the true difference. No need to infer anything.
Studying a population to attempt to predict something about the future	Investigating turnover trends across different populations to forecast next year's attrition rate.	Yes—since the purpose is to predict, the researcher must use past data to infer something about the future, so the results need to be statistically significant.

Researchers must differentiate these scenarios into "statistically significant" versus "practically significant." Statistically significant simply lets us know if we can extrapolate our findings from a sample to a group or from the past to the future, but it does not necessarily translate to business value. Practical significance, on the other hand, requires translation of findings into something that actually matters. If in example #2 in the previous table the researcher finds that the supply chain employees have lower engagement scores by 10%, what does that mean? Does it mean that scores are 10% lower across all of supply chain or that a small population

went down much more than 10% and is pulling the overall average down? And either way, how does that impact the business in the form of employee outcomes like turnover, absenteeism, or productivity? Recall the sections on your Central Tendencies versus Measures of Variability—these will help tell the real story of what is actually happening in the data. Then, the analyst can begin investigating what outcomes it may be driving. Simply quantifying that something is "statistically significant" is only the first step (and sometimes not even necessary). However, translating quantified differences into something practically significant, like relationships to work outcomes such as turnover or productivity, is where "significance" really matters.

6.6 Translating Ideas into Testable Stats and Interpreting the Outcomes

6.6.1 p-*values*

Statistical significance in inferential statistics is often measured by using a statistic called the p-value. P-value (or probability value) is a number between 0 and 1 that expresses the probability of observing similar data in the population if H0 is true. Whether a result is deemed statistically significant or not depends on what the p-value is relative to a threshold called the alpha value. The alpha value is essentially the level the experimenter is willing to accept for the possibility that the results may be incorrect. For introductory purposes, think of it as the "close enough" value. The most commonly taught and standard alpha value in behavioral research is 0.05, which means there is a 95% chance that the results you are seeing in your sample will generalize to the population, and a 5% chance they will not.

In other industries, like the medical field, alphas of 0.01 are more common, due to the higher need to be more precise. If a researcher is doing a study that they will need significance for, they should decide what alpha values make sense before the study is conducted. The reason for this is that it can be tempting to set an alpha that allows p-value to sneak under the threshold and prove the hypothesis. For example, say a result comes back with a p-value of 0.055. At this point, the analyst's mind has been biased and anchored to that number, and it will be much harder to set an alpha value without that result influencing the decision. It is important to do this thinking upfront so that results do not influence perceived validity. There are many ways to misuse p-values, such as p-value fishing or p-hacking. The famous economist Ronald Coase once said, "If you torture the data long enough, it will confess to anything." This is a similar principle to confirmation bias and is important to keep in mind as a researcher and consumer of research—always ensure preexisting biases and desires do not cloud research methodology, analysis techniques, or the interpretation of results.

6.6.2 Hypotheses

We have now reviewed how to define questions well for the sake of research methods and data collection, but how must questions be framed so they can be tested statistically? This might seem like the same thing, but when defining the problem in a testable way, we found that asking a question must be framed in the language of the variables being used to define the problem. When creating a formal hypothesis for statistical analysis, something similar must be done.

When embarking on empirical research, inferential statistics can be used to answer several types of questions. A hypothesis is a statement that may or may not be true and that one wishes to test. In inferential statistics, this takes the form of two separate statements. The first is the alternate hypothesis (commonly written as H_a) which is the original statement and the observation that the analyst or researcher wants to prove right or wrong. In addition to the alternate hypothesis, there is a null hypothesis (written as H_0), which states that the observations are purely due to chance and are not statistically significant. For example, if an analyst is testing whether a peer mentor program increases rep performance by 5% or more, they would start by formulating the alternate hypothesis as follows:

H_a: The average sales dollars for poor performing reps who participated in the peer mentor program is greater than the average sales dollars of poor performing reps who did not participate in the program.

Next, the analyst would define the null hypothesis to state that any observed differences in the samples are due to chance:

H_0: The average sales dollars for poor performing reps who participated in the peer mentor program is not greater than the average sales dollars of poor performing reps who did not participate in the program.

6.6.3 Results

After testing, the result has a p-value of 0.02. Based on your alpha value of 0.05 (determined prior to the study), the result indicates statistical significance of the program. The analyst can now communicate with confidence that the program has a statistically significant positive impact on poor performing representatives. The formal statistical statement is obscure, but would read as follows:

Reject H_0 at $\alpha = 0.05$ ($p = 0.02$). There is statistically sufficient evidence to suggest that the population mean sales performance for poor performing representatives who participated in the peer mentoring program is greater than the population mean performance of poor performing representatives who did not participate in the program.

Fortunately, practitioners do not typically have to write results this way, but this style of communicating is specific and detailed for the purpose of clearly and consistently communicating what the research found. What a formal article in an academic journal lacks in warmth and engaging writing style, it makes up for in clarity, consistency, and specificity, which is an incredibly important part of scientific communication.

6.6.4 Standard Error, Margin of Error, and Confidence Intervals

Given that results and predictions are not failsafe even when using sound sampling techniques, the importance of properly setting expectations for the accuracy of inferences is essential. For example, for any given sample a researcher can calculate the standard error to help contextualize the confidence of how close the mean of the sample is to the population. Without getting too detailed, the idea is that based on the size of the sample and the sample's variance, the researcher can infer how confident they are that the mean in the sample reflects the mean in the population. From there they can add a constant based on how confident they want to be (usually 95%) to calculate a "margin of error." You may be familiar with this term as a saying in everyday language, but it is actually a statistical metric! Once the margin is determined, the researcher can apply that margin to the mean of the sample and say they are 95% confident that the mean falls between two values.

6.7 Introducing Bayesian Inference

So far, all the statistics reviewed fall into a philosophy of statistics called "frequentist inference." For those who are not statisticians or data scientists, it may be surprising that more than one type of statistics exists, and that is okay—most of the statistics in day-to-day life fall into the frequentist camp and it is what is most commonly taught in school. That said, there is a different kind of inferential model which is increasingly making contributions to advanced analytics and therefore machine learning: Bayesian inference.

Statistics using frequentist inference is all about calculating the probability of an event given a static set of circumstances. This means there is some amount of possibilities against which the statistician can calculate the likelihood of one or more of those possibilities occurring. That may sound confusing, but our brains understand the concept intuitively: if Mike flips a coin, there is a 50% chance of heads and a 50% chance of tails. 1 outcome (heads) divided by 2 potential outcomes (heads and tails) creates a probability of 0.5, or 50%. However, in any random trial, Mike may flip a coin 10 times and it is possible to get 8 heads. That is not as likely to happen

compared to an outcome that is closer to 5 heads and 5 tails, but it is possible. Frequentist theory will tell you Mike can get a more likely result with more flips: if he flips a coin 100 times the results will likely be closer to 50, if he flips a coin 1000 times, it will almost definitely be very close to 500, and if Mike flips a coin an infinite number of times, he will eventually get to that perfect 50/50 ratio. This is the fundamental basis for how most of the statistics you have come across work (and a good illustration of why sample size is important).

This whole philosophy—that probability is a ratio of outcome (heads) to all possible outcomes (heads and tails)—is based on the very important fact that every flip Mike makes truly adheres to that 50/50 probability. Frequentist statistics assumes the probability for the event follows the given probability model every time, in this case a 1 in 2 chance of getting heads.

But this book is about *data*. Data tells stories. Data helps researchers understand the past so that they can predict the future. What if Mike could use his data about past flips to influence that 50/50 probability model? What if, after many flips, Mike was seeing that he was getting more heads than he should? Maybe there is something about the coin, or the table, or the way he is flipping that is making it more likely to get heads. Can he use that information to make inferences about future flips that are better than just continuing to assume 50/50 every time?

This is where Bayesian statistics starts. Bayes' theorem basically says the probability of an event is not just the outcome divided by all potential outcomes, but rather must also include the data that has been observed about the event. In other words, use the real outcomes Mike has and use those data to influence the probability.

To put it in plainer terms, what Bayesian inference does to statistics is to provide a feedback loop for the data which can influence the probability model itself. When Mike starts a statistical analysis, he has a probability model in mind. For coin flipping it is 50/50. Then, after many flips, he has new data about flipping. If he ended up with 8 heads in 10 flips, Bayesian statistics would use that data to slightly change the probability in favor of the heads outcome and potentially improve future inferences. If Mike then collects more data and continues to get extra heads, the model will continue to shift the probability. If he regressed back and began seeing more tails, Bayesian stats would also take that into account and rebalance the probability.

We will not go any further into the technical details or applications of Bayesian inference, but it is important to at least know it is a distinct and influential field of statistics which is increasingly influencing how practitioners use inferential statistics to build machine learning models. As data has gotten easier and easier to collect, process, and use, Bayesian inference becomes more and more feasible to implement at scale. The increase in computing power has changed what sorts of statistics can be used, what kinds of machine learning models can be designed, and even how people live their daily lives, which brings us to the next part of Analytics Ikigai: Computer Science.

Discussion Questions

1. Why are descriptive statistics so important?
2. Explain confirmation bias and how descriptive statistics can contribute to it.
3. What is the difference between a measure of central tendency, a measure of variability, and a measure of relative position? Why might you use each?
4. Why is standard deviation so important? Give two examples of how it might be used in an applied setting.
5. Why are distributions an important part of descriptive statistics? Provide two examples of common HR data which do not demonstrate normal distributions.
6. What is the difference between statistically significant and practically significant? Give two examples of when you might need each.

Chapter 7
Why Now? Computers Enable a Future with Machine Learning

On February 10th, 1996 a young man sat down in Philadelphia to play a game of chess. He was known for his thorough pre-match preparation, early-game aggression, and ability to switch tactics mid-game. Just 32 at the time the young man, named Garry Kasparov, was thought by most to be the best chess player in the world. Garry was born in 1963 in the Soviet Union, and chess had been his life. Showing immediate aptitude for the game, by age 7 he was attending a special school to develop his skills and by age 10 he was studying under Vladimir Makogonov, one of the premier chess players of the 1940s and a renowned coach in the USSR.

By 1976, the 13-year-old Kasparov won the Soviet Junior Championship and then repeated in 1977. By 15 he had qualified for the grown-up version—the Soviet Chess Championship—and by 22-years old in 1985, became the youngest player in history to achieve the World Chess Federation's #1 ranking. Suffice it to say, Garry was pretty good at chess.

Across the table, Garry's opponent hailed from Pittsburgh, Pennsylvania. His parents were of Taiwanese, Indian, and Canadian descent. His calling card was his ability to process incredible amounts of in-game possibilities and by 1988, he had beaten his first grandmaster. However, the following year he was easily dispatched in two games by Kasparov himself, so it is no surprise his fans were eager for a rematch.

He was born with the name ChipTest, developed by Feng-hsiung Hsu, Thomas Anantharaman, and Murray Campbell at Carnegie Mellon University. First developed and sent into competition in 1985, by the end of 1987 ChipTest was the North American Computer Chess Champion. Seeing opportunity for improvement, ChipTest's designers developed an improved model which they called Deep Thought, and by the late 80s their machine moved on from beating computers to beating humans. The project was so successful it was picked up by IBM where the system underwent its final naming evolution, being redubbed Deep Blue.

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_7

Between that first match in 1989 and the rematch in 1996, a great deal had changed for Deep Blue. Names aside, IBM and its creators had invested more time, money, and technology. More processing speed, more algorithms, and more code warranted another opportunity to pit man against machine. Garry agreed to play again.

The rest is history. In that 1996 match, Garry Kasparov won the day defeating Deep Blue by a score of 4:2, but it was a monumental day for computer science nonetheless. The computer had drawn with Garry twice, but more importantly it won the first match outright, which was the first time a computer had defeated a reigning world champion.

The following year, things got worse for humans when an even more improved Deep Blue beat Kasparov twice and drew three games in a best-of-six, giving team-machine an overall match victory of 3 ½:2 ½.

So much for human intellect.

You may have heard this story before and may even remember the news coverage. But why does a chess-playing computer matter so much to computer science and why is its ability to beat a human important? The answer is that up until that day in 1996, chess was thought to be "computer-proof." This is because of the overwhelming amount of directions a game of chess can go. At the beginning of a chess game, the first player to move (white) has 20 possible options. This means that after one move, black will see one of 20 possible board configurations. After black's first turn, there are 400 possible board configurations for white's second move (20 possibilities from the first move times 20 possibilities from the second move). After each player goes one more time there are 197,742 possible board configurations and by the time each player has gone three times, there are 121,000,000 potentialities. When you consider that a typical game of chess runs about 40 moves, one can see how quickly the possibilities diversify. David Bronstein (a chess grandmaster) once said, "Chess is infinite, and one has to make only one ill-considered move, and one's opponent's wildest dreams will become reality."

Given near-infinite possibilities, chess must then require judgment, something computers before 1996 were not thought to be able to do (and certainly not better than humans). In simpler games, judgment is not technically important for high performance. For example, in tic-tac-toe, a computer can simply process all the different board configurations and win or draw with you every time[1]. But in a game with an infinite number of outcomes, a computer cannot simply out-calculate its human opponent. So how did Deep Blue "think" in a way that would allow it to process long-term strategy when it did not know all the possible outcomes? And how could it leverage tactics like pausing between moves to feign uncertainty (something Deep Blue actually did) during the match?

The technical answers to these questions are extraordinarily complex and warrant books of their own. Indeed, many have been written on the subject, including Behind Deep Blue (Hsu, 2002), Deep Blue: An Artificial Intelligence Milestone (Newborn, 2003), and Beyond Deep Blue: Chess into the Stratosphere (Newborn, 2011).

[1] There are 255,168 ways to play a game of tic-tac-toe. In 131,184 of them the first player to move wins, in 77,904 the second player to move wins, and in 46,080 there is a tie.

For the purposes of this book, we are less interested in the specific nature of the complexity and more interested in how technology evolved to allow for such a feat. And more specifically, how this technology is going to influence how HR is practiced.

Learning about computing will help us understand why computers have grown to touch virtually every facet of human life and briefly reviewing this history will reveal the massive leap forward that machine learning represents. In the 80s and 90s the desktop computer and internet revolutionized how we work by digitalizing our day-to-day lives, but what has been happening since Garry and Deep Blue? There is no debate everyone can feel the difference in how they experience technology today versus in 1996; that impact has been transformative. But what has been happening behind the scenes, and why is the HR industry just now poised for the impacts of advanced computing?

7.1 Computers Everywhere

Section Breakout: From the Author's Perspective
As I write this section, I have just arrived at my teenage son's swim meet. The team needs time to get changed, wait for the girls' events to finish (they race first), get their lane assignments, and warm-up. I typically take that hour to pull out my computer and write.

Before he got out of the car just now, he did something he very rarely does: he handed me his phone (they are not allowed in the locker room or on the pool deck). I put it away, find a parking spot, take my bag, and go search the unfamiliar high school halls for a place with a table, chairs, and—most importantly—an outlet. I settle in the school cafeteria and set up shop. My work phone emerges from my pocket. Then, my personal cell phone. Then, the cell phone my son entrusted me with, my laptop, and finally the tablet (which I need for the internet hotspot). It seems fitting this section is about the evolution of computing and how it has come to be so ubiquitous in modern life.

Whether you have personally experienced carrying five devices around, have spent time on an airport floor so you can charge whatever device is running low, or have become hopelessly lost while driving because your phone died, you understand that computers have been entirely integrated into modern life. If you had to wake up next Saturday morning and realize you cannot find one thing until dinner time, would you rather have it be your keys or your phone? When Kasparov lost to Deep Blue in 1996, would you have had the same answer?

You probably did not even purchase this book by walking into a bookstore and picking it up off the shelf. You likely ordered it from a company on the internet who delivered it to your home, office, or school. If you did get it from a store, did you

order it online or at the very least check the website to ensure the book was at the store before you arrived? You might not even be reading this in a book, but rather on a laptop or e-reader.

From microwaves to bill payment to smartphones, computers are in every part of life. Bill Gates once famously dreamed of a computer on every desk. In 2021, we've even got one in every pocket! Computers run cars, schedule sprinklers, forecast the weather, notify friends of their friends' moods, and source pictures of cheeseburger-eating cats. However, in 2017, there was an Apple commercial where a young girl is running around a city all day with her tablet. She uses it for everything from drawing to homework to pictures and eventually ends up in her urban backyard, still connected to her device. An adult enters the screen and asks her what she is doing on her computer. The pre-teen replies, "what is a computer?"

Setting aside that a 13-year-old in 2017 would obviously know what the word "computer" meant, the message is clear: computers have transcended the transactional machines of the twentieth century to become integrated into every aspect of life. With the whole of human knowledge accessible from our pocket, "computers" have become part of how we live our lives.

But amidst the social media, memes, automatic lawn watering, and over-the-top commercials have you stopped to think in the last decade about what a computer actually is? That is, if someone asked you, "what is a computer," what would your answer be? We know them when we see them, but what *are* they?

Computers are not "something that gives us access to information." Books do that, but they are not computers. It is not an "electronic device that performs tasks" either. Bedside clock radios from the 80s did that, but few would call those computers. And it is not simply "something that accesses the internet" because a computer is still a computer when it is offline.

When we do not know how to define something, our mind goes to examples. If we cannot articulate the common characteristics that define a thing then we give samples and the commonalities will make the definition obvious. What is a computer? Smartphones, laptops, tablets… you know, computers.

7.2 Input—Process—Output, Faster

Technically, and in the modern sense, a computer is just a machine that follows instructions via computer programs. But the concept of computers and computing goes back further than even electricity. The term computer did not originally have anything to do with digital machines. The word "compute" simply means to calculate or evaluate. Therefore, a compute-er was someone or something that computed things. Originally, "computer" was a job title!

In ancient times, a human plus an abacus was a computer. If many humans were together with many abacuses inside a building, it was called a counting house. Countinghouses were basically pre-twentieth century CPUs filled with humans cal-

culating sums, interests, and other accounting details to run early economies. "Computer" as a job title has been around far longer than its current meaning as a pronoun for these electronically driven intimate objects. Men and women for the majority of human history could and would be employed as computers, where it was their job to calculate things all day.

A great example of human computers is the scientific pursuits of Alexis Claude Clairaut in the 1700s. Clairaut and two colleagues wanted to determine the timing of the return of Halley's comet, but the calculations were too great for an individual to undertake on their own. By dividing the process between them they not only figured it out but became one of the first documented human computers in the process.

This is important because it illustrates that computers are not machines at their essence, but rather tools to process information. When you reframe your idea of a computer in this way, you realize that the successful computer is meant to do three things:

1. *Intake information*
2. *Apply rules*
3. *Output new information*

And, almost more importantly, what makes a computer better is its ability to do its job:

1. *Quickly*
2. *Without error*

Therefore, all advances in computing since the countinghouses of ancient Rome are aimed at improving intake > process > output abilities. Said differently, every computer ever employed, invented, or built was done in an effort to process information faster and/or with fewer errors.

When we boil computing down into the idea of intake > process > output and the improvement of computing into the science of doing that faster and more accurately, we can follow history along the timeline of improved speed. Computer terms define that using a simple metric called "instructions per second."

Instructions per second is how to measure the speed of a digital computer. The concept is this: how fast can a computer handle the input you give it? If I threw you a ball, you could catch it, and then put it in a bin. Then I could throw you another ball, and you could catch that one too. Then another, then another. Faster and faster. At some point, we would reach a threshold: you could only catch and place so many "balls per second" into the bin. In essence, this is what instructions per second measures in a computer, where the balls are inputs from interfaces like a keyboard, mouse, or touchscreen interfacing with a program and you (the person doing the catching) are the computer.

Modern computer systems are so sophisticated that multiple computers (called "processors") can take instructions at once within a single device. This would be like having a team of people catching balls all at once. If a laptop has a "quad-core" processor, that means it has something like "four brains"—it can take instructions four-times faster than a computer with one processor. This makes loading your favorite social media pages while listening to music and writing emails far easier than it was a decade ago.

For the majority of human history, computing could only run at the speed of a group of humans that was of moderate size. By the 1800s, humans were creating big machines that could mechanically automate counting and other basic forms of calculation. Then, the invention of the transistor enabled more sophisticated machines, like the one Alan Turing and team used in WWII to crack the Nazi enigma cipher. By 1958 the integrated circuit had been engineered, which allowed engineers to put more computing power in a small space and in 1965 Gordon Moore famously predicted that the number of transistors in an integrated circuit would double every year for the next decade. He was right, and in 1975 revised his forecast to say doubling every 2 years for the foreseeable future (this is known as Moore's Law).

Let us consider the magnitude of Moore's prediction for a second. Imagine you could catch a ball every 6 seconds, or 10 per minute. But you could train yourself to catch twice as many balls per minute as you could 2 years ago. And in 2 more years, you could catch 2× more than that, and so on. The math looks like this: 10, 20, 40, 80, 160, and 320. In just a decade you went from one ball every 6 seconds to more than five balls every second!

To give some real examples from the history of technology, consider this: the Apollo Guidance Computer which landed the first humans on the moon (built in 1966) had roughly the same computing power as the original Nintendo video game console, released in 1985. Less than 15 years later (about the same distance between Apollo and NES), Nintendo 64 is released in 1996 with more than 10x the computing power of NES. Four years later, the Playstation 2 delivers more than 10x the computing power of the N64. And by 2014 (just seven years after that), the iPhone 6 brought the same computational power as that PS2, but now it fits in a pocket! From a computational perspective, an iPhone 6 could manage the navigation of 500 Apollo spacecrafts… at once. And in 2021, how state-of-the-art does the consumer electronics market consider an iPhone 6?

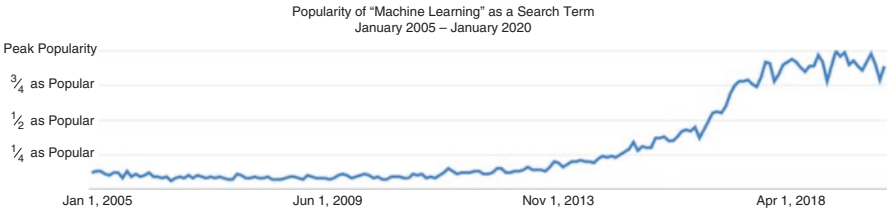
It is largely due to the increasing ability to put many tiny transistors on a small strip of silicon that humanity has enjoyed the results of this explosion of computational power. It is also what has enabled machine learning to become a game changer. It is in this window of time (the 1980s through the early 2000s) when humans realized that computers were so good at going fast, that they needed to figure out how to get out of the way. This was a major shift in thinking because throughout most of human history, computers just computed. Remember the original definition: intake > process > output. That system relies on humans setting up all the rules, then sticking the input in one end (intake), and reaping the benefits of the output. Computers only processed the instructions that the humans gave them. They do not "think," they do not "judge," they just process. And so it had always been all the way back to the person in the countinghouse whose job it was to count the gold. It was not his job to judge whether it was too much gold, to find patterns in the gold-spending, or to tell anyone that the gold was better spent elsewhere. It was his job to count it. Period.

Sometime between Apollo and Deep Blue people realized that computers could process so much so fast that they could change the computer's job. No longer did computers need to be just recipe followers. Engineers could design them to recog-

nize patterns, use those patterns to influence the recipes, and even have them create recipes of their own. Instead of simply increasing the speed of the process, computers would be able to increase the quality of the process. So that is exactly what the industry did.

7.3 Machine Learning Arrives: Basic Concepts

Machine learning as a term has been around for a long time, but recently reached a tipping point in the popular media. Between January 2014 and January 2020, the phrase "machine learning" has become roughly 4x more popular in Google searches. A YouTube query for "machine learning" yields about 548,000 videos. The same search for books on Amazon.com shows over 20,000 titles, and in just the last 3 years (January 1, 2017—December 31, 2019), the Wikipedia page for Machine Learning has enjoyed the attention and maintenance of over 800-page edits (that is more than an edit every day-and-a-half). As a society, it is safe to say we have become somewhat enamored with the concept.



Popularity of machine learning in Google searches (retrieved February 2020)

But what is it? The name originated in the late 1950s and originally used the two words literally. Machines in this case are computers—hardware and software working together to intake, process, and output (as defined before).

"Learning" is more complex. Learning is usually applied to organic things—people, animals, and even plants have behavior, and researchers can see behavior change based on the experiences organisms have. In behavioral terms, learning is any relatively permanent change in behavior based on experience. The key part is that past experience influences future behavior. As an example, think of a child who touches a stove and burns his hand. The next time he sees a stove, he does not touch it because he "learned." The boy's past experience with stoves guides his present behavior when he is faced with a similar situation. He comes upon a hot stove and his little brain makes a prediction: "if I touch that stove, my hand will hurt." Next, he makes a judgment and decides on an action: "I do not want to hurt, so I will not touch the stove." Finally, his behavior is to avoid the stove and he gets the outcome he wants (no pain). That is when we say, "he learned."

That is a sophisticated process. Can machines really do that? Not exactly, but the way we use computers in machine learning follows a similar process such that "learning" is a pretty good word for it. When a computer is using "machine learning," what we mean is that we feed the computer data that represents "past experience," like the boy touching the stove and hurting his hand. To the computer, these data (often called "training data") are what the computer "learns" from.

This "learning" is really the computer exploring the data for patterns so that it can tell the researcher something useful. Sometimes the researcher knows what they are looking for and they ask the computer to tell them about that (like predicting who is going to quit) and sometimes the researcher does not know what they are looking for (like how they should segment a population in a market analysis). Either way, when a computer can take data from the past and use it to find patterns that can help it make accurate predictions about the future, we say it can "learn." To illustrate further, let's talk about peanut butter and jelly.

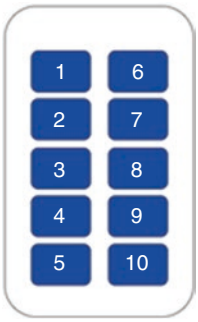
7.4 Peanut Butter and Jelly Programming

Computers are good at following instructions and so historically that is all computer scientists ever asked them to do. This is what a computer "program" or "application" is (or "app" for short). It is a set of instructions. Whether it is for word processing, accessing the internet, or throwing cartoon birds at pigs, that is all programs and applications are to a computer: sets of instructions.

So the programmer gives a computer a program to follow, and it follows it to the letter. In this way, think of the computer-to-program relationship as someone following a recipe in a kitchen.

Anyone who has been in a kitchen knows that recipes are not all created equally. If someone wanted to make a peanut butter and jelly sandwich, the recipe would be short and simple to get from ingredients in a cupboard to a sandwich on a plate. If instead someone wanted to make a salmon soufflé and blanched green beans with almonds, there would be quite a few more steps to get from ingredient to table. This logic is also true for computers. For example, the average internet browser has about 5,000,000 lines of code, while the average smartphone operating system has about 12,000,000. That is a long recipe.

The reason computer programs need so many instructions is the same reason chess was computer-proof for so long: a computer program has to account for every single possibility a user of the program might come across. To illustrate, imagine an extraordinarily simple phone app. Literally just a screen with 10 buttons on it:



When a button is pressed, the button turns yellow. Press it again, it turns green. Press it again, and it turns red. Press it one more time, and it turns "off." That is all the app does: Off, yellow, green, red, and off again for each button.

There are 1,048,576 potential screen outcomes for that app. The math is easy enough—four possibilities per button (off, yellow, green, and red) gets raised to the power of the number of buttons (10), so $4^{10} = 1{,}048{,}576$.

This app does not take over a million lines of code to produce—software engineers are smarter than that. The point here is to illustrate that as software gets more complicated, there are exponentially more potential outcomes, interactions, and user experiences that need to be accounted for. That is what makes writing software so complicated. Imagine taking this concept and putting it up against your favorite video game or phone app.

Where does machine learning come in? In the history of computing, computers did exactly what they were instructed to do. No more, and no less. They follow the recipe. People give them ingredients and a recipe, and they follow it to the letter.

Machine learning is the new ability of computers to "learn" from what they "see." And when we say "learn," we mean "observe patterns in past data and use that to either make predictions about, or influence the way they process future data."

Let us take the peanut butter and jelly sandwich recipe. A traditional computer program would say, "give me a PB&J recipe (i.e., program), and the necessary ingredients (input), and I'll give you a sandwich (output)." You give them this:

Ingredients:
- Creamy Peanut Butter: 1 Jar
- Bread: 1 Loaf
- Grape Jelly: 1 Jar

Materials:
- Plate
- Knife
- Napkin

Instructions:
- Take two slices of bread from the loaf, lay them next to each other on the plate
- Spread peanut butter on one side of one slice of bread

- Clean knife off with napkin
- Spread jelly on one side of other piece of bread
- Put pieces of bread together with the peanut butter and jelly oriented toward one another
- Use knife to cut sandwich in half

This set of instructions is simple enough for a person, but it is far too general for a computer to follow. For example, the instructions did not tell the computer to open either jar, how to open the jar, to actually insert the knife into either jar, or about 1,000 other things a computer would not know to do unless the program told it to. If, for example, there was less than a full loaf of bread, the program would crash. What if there was only raspberry and not grape jelly? Crash. Only crunchy peanut butter in the house? Crash. The list goes on.

This gets to the point of what limited computers for so long: their literal nature. They can only follow instructions. They can only function if they are explicitly instructed how to handle every potential outcome and every possible contingency. Any unexpected outcome creates a crash. Machine learning changes that paradigm.

At a high level, machine learning works like this: have the computer "watch" 100,000 people make a peanut butter and jelly sandwich and look for commonalities, so it can pick out its own ingredients, materials, and instructions. Instead of the human setting up all the rules (the recipe) and just expecting the computer to follow the rules to the letter, the human can say "here are many examples of inputs and outcomes. Make up your own recipe."

This gets around some of the literal nature of computers because the computer can then "figure out" that it is not the "grape" that is important when considering which jelly to use, it is the sweet, edible jelly part (e.g., not aspic or petroleum). Also, it will see you must open a jar before you can get at what is inside it. And some tops screw off counterclockwise, while others pop open and the jelly gets squeezed out. Instead of trying to force fit top-down logic on the computer (which makes it fragile and prone to breaking), machine learning allows the machine to observe a huge quantity of examples and infer what the recipe should be.

Does this make you think of something we discussed in Chap. 4? Remember our comparison of Deductive Reasoning (top-down logic) and Inductive Reasoning (bottom-up logic). Machine learning is fundamentally inductive because it uses patterns it can find to create "rules." This makes machine learning quite powerful–since it can observe enormous amounts of data (due to the significant increase in computing power we talked about in Sect. 7.2), it can find patterns a human or team of humans could never hope to get through. That on its own has revolutionized how people can get insight from data. In the HR space, practitioners can think about data sources like benefits, labor markets, surveys, or attendance information. With tens of thousands of events across hundreds of different types of employees and situations, an HR team could never hope to look at every possible combination or outcome, nor create a computer program to analyze every scenario. This is where machine learning adds value.

In fact, in some applications, like when (a) the data is big enough, (b) the past is very likely to look like the future, and (c) the input data accounts for almost all the potential variance, this form of evaluation can be extremely accurate. That said, because machine learning is not deductive, it cannot by itself tell us anything with the level of certainty empirical research using the scientific method can. Again, machine learning is a tool in the toolbox and is best fit for some kinds of problem-solving, and not for others.

Machine learning is not totally carte blanche for the computer—the data scientists cannot simply load every piece of data they can find into the computer and reap the reward (this is sometimes called the "brute force" method and in HR is almost always too problematic to use). Analysts still must guide the computer on which pieces are necessary to watch and which are irrelevant. For example, in the PB&J example, imagine all 100,000 cases of sandwich-making have the lights in the kitchen on. Without guidance from a person, the computer might think the lights are a critical prerequisite to making a PB&J. And while it is definitely easier to make sandwiches in a well-lit space, it is not entirely necessary.

The overall point is that while traditional computing was the art of creating extraordinarily thorough recipes which account for every possible ingredient, step, and contingency, machine learning is more the art of setting up models which can take inputs and use them to create their own rules and observations which lead to outputs humans could not have come up with on their own.

In the next two chapters, we go deeper into this idea of how to enable computers to explore ideas humans cannot. We will revisit the principles we have already learned about reasoning, research methods, and statistics to discuss (1) how humans make machine learning work, (2) some limitations and considerations for its use, and (3) some techniques you may come across when venturing into the machine learning space.

Discussion Questions

1. What are the three main functions of a traditional computer? What are the two metrics by which those functions are measured?
2. How does the literal nature of computers limit them? How have advances in computing enabled machine learning to remove these limitations?

Chapter 8
Introducing Machine Learning

Machine learning is better defined as a field of study than as a label which can be stuck on a particular type of analysis. A good working definition for the average practitioner is that machine learning is the study of computational algorithms based on mathematical models which improve automatically with experience. Machine learning identifies patterns in datasets and makes predictions or decisions based on those patterns without being explicitly programmed to do so.

In this way, machine learning is not a binary category at all. Try to avoid looking at different forms of computers making predictions and say, "this process has the 'correct attributes' to be machine learning, but this process does not." This is really just a semantic difference. Part of the reason for this is just that there are so many mathematical approaches to so many different problems that it would be impossible to classify them all. Furthermore, those techniques and processes often get ensembled, stacked, or otherwise connected such that the mere combination of different types of approaches is essentially infinite. Frankly, it is more attainable to take a simplistic and inclusive definition, then describe the characteristics and considerations a practitioner must use to make choices about which techniques are appropriate in which scenarios. As a practitioner, the goal is to solve problems, which means understanding the basics of how the techniques employed work. Being overly concerned with how those techniques are academically labeled is not as important. In this book, we group them under the heading "machine learning" for the sake of their general similarity, but defining *exactly* where the definition lines are is not a particularly useful exercise for the practitioner or practitioner-in-training.

We will help illuminate this class of techniques by explaining common characteristics which exist between methods, by comparing machine learning to other similar fields, and by explaining at a high level how the methods work. By doing this we can triangulate where machine learning lives among its brother and sister fields without getting overly specific. This chapter will also get into high-level explanations of how machine learning works from the statistical perspective, provide some broad-stroke definitions for types of machine learning to give the reader

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_8

familiarity with them, and review considerations which must be made when deciding whether to use specific methods. Then in Chap. 9, we will get into specific techniques and how these techniques work.

Before diving in, we would like to introduce three main purposes which can be of value to the practitioner when using machine learning. Machine learning can (1) Identify Drivers, (2) Create Groups, and (3) Predict Outcomes.

Identify Drivers: Machine learning models can be used to understand which variables influence an outcome and their relative amount of influence. For example, in Sect. 8.1 we will talk about Meredith and her need to solve a turnover problem. A predictive attrition model could be used solely to produce a list of how many people will leave or who is likely to leave. This could be useful for many reasons, but if you want to know *why* they are leaving, a "transparent" model (Sect. 8.4) can tell you *which factors* in your data are most strongly correlated to attrition. This does not provide causality (remember, correlation is not causation), but depending on the strength of the relationship a practitioner may be able to infer causality and those insights could drive organizational change to reduce undesirable turnover. For example, if a factor associated with turnover is lack of training, the organization could institute enterprise-wide training programs. This alone could provide substantial value without accurately predicting who is likely to turnover.

Create Groups: Models produced by machine learning can also provide insight into the structure and relationships between data in a dataset. For example, machine learning algorithms can be used to automatically group and identify similar types of employees based on a variety of different traits. A real example conducted by one of the authors was a research project to determine what a "new hire" was in a retail environment. The business knew that on day 1 someone was definitely a new hire and that by day 1,001 they definitely were not a new hire, but up until that study the lines the organization drew to distinguish one from the other were arbitrary and anecdotal.

Machine learning allowed the business to look at three main categories of work outcomes: performance, promotions, and turnover and let an algorithm guide what a "new hire" was. By examining the patterns across the variables, clear categories emerged, and the organization was able to make better distinctions for "new hire" groups. This created strategic advantage for many teams who work with new hires, including training, workforce planning, and talent acquisition. These data also allowed the team to do the next thing machine learning is good at: predict outcomes.

Predict Outcomes: The third use of models, and what comes to mind for many when they think of machine learning, is predicting a specific outcome. From a predictive perspective, the attrition model mentioned above would want to create a count of future turnover or a named list of individuals who are at risk of leaving. Predicting outcomes is probably the most desirable use of machine methods but is also the hardest to produce reliably.

8.1 Machine Learning and Inferential Statistics

Let us refine our understanding of machine learning by comparing the idea of machine learning to a topic we have already reviewed: inferential statistics. We started the conversation about what machine learning is by approaching it from the computing angle. But by the end of Chap. 7, we were again talking about models and predictions and inferences which sounds a lot more like statistics. We even came back to inductive versus deductive reasoning and the scientific method.

In fact, "building models to predict something about the future" sounds a lot like inferential statistics. Is machine learning just a fancier name for doing statistics with a computer? If you are a little math savvy you might even say, "all the statistical and other mathematic processes underlying machine learning *can* be done by hand. I learned how to do logistic regressions by hand in school, so that is not really machine learning, it is just doing statistics faster."

The answer to this is, "of course." And technically a statistician could analyze all the weight potentials in the forward-propagating section of a neural network too… it might just take more hours than they have left in their natural life to do it.

This returns to the premise that computers are so good at going fast that humans need to figure out how to get out of their way. The question is not whether humans *can* do the calculations manually or not, it is whether humans can set up a computer, so that it can do the calculating and evaluating without them having to explicitly tell it how to answer the question. And more importantly, when humans provide more optimized data, the computer gets better at performing whatever it is the humans have asked it to do.

Remember where data science and machine learning sit in Analytics Ikigai, *right at the intersection of computer science, statistics, and research methods*! So, the answer is that machine learning is built *upon* the foundational pillars of both statistics and computer science but is more than simply the application of statistics using a computer.

If you recall from Chap. 6, we reviewed how inferential statistics is the branch of statistics which uses math to infer relationships between variables. Humans can then use those inferences to make predictions about what will happen in the future. So it follows that the main purpose of inferential statistics is to *uncover whether or not relationships exist between variables*. When we do this well, the results of inferential statistical analyses will allow us to predict the future, but importantly, that is a secondary goal. The purpose of inferential stats is to provide evidence for the relationships that exist between variables.

In this way, think of inferential statistics as support for good research methods. Remember that the scientific method is all about creating testable hypotheses, collecting data, and *then* using statistics to see if the hypotheses are supported or refuted. The statistics come in at the end to help understand if there is reasonable and quantifiable support for the idea. Inferential statistics when used for science is concerned with *why* relationships exist.

Machine learning, in contrast, is concerned with *patterns*. The goal of a machine learning model is to use data to train and validate a mathematical model which can spot patterns for the sake of identifying drivers, creating groups, and predicting what data in the future will look like. In this way, machine learning is less concerned with *why* the patterns in data exist (like inferential statistics) and more concerned with *how* the patterns in data exist. Let's illustrate the difference with an example:

8.1.1 Understanding Turnover Using Four Approaches

Turnover is a challenge for the sales reps in Meredith's retail organization. As the head of HR for multiple geographic regions in a big company, Meredith sits at the table with all the sales executives who struggle year-in and year-out with the well-understood challenges high turnover creates. Lost people need to be backfilled and then new people need to be onboarded. People leaving is expensive for the organization and stressful for the employees they leave behind. It disrupts operations, reduces revenue for the stores, and has a negative impact on the customer experience.

Turnover also creates significant uncertainty. Not knowing how many people will leave makes it difficult to forecast volume for talent acquisition (who have to find new employees), learning and development (who must train new employees), and incumbent workers (who must bridge the gap with overtime and/or hire contractors to help out). This makes for a rollercoaster when it comes to figuring out how to allocate resources to stores, regions, and the organization overall.

As the HR leader for this part of the organization, the Chief Human Resources Officer asks Meredith to help reduce turnover to a more manageable level. There are several ways she can approach this request.

Approach 1: Qualitative Investigation

The traditional HR method would be to use human insight and anecdote. To do this, Meredith would marshal her resources to gather insight about employees who left. She would have teams interview HR personnel, peers, and managers who supported turned over employees. Meredith would conduct focus groups in places where turnover is highest and ask people what the problems are. Then, she and her team would triangulate all those data to create a list of the drivers of turnover. Finally, she and her team would build an action plan from those insights and hope actions influence turnover in the future.

Approach 2: Descriptive Statistics

The next rung on the empirical ladder would be to gather data on both employees who left and employees who have stayed and look for trends that might indicate factors causing attrition (this approach is often done in conjunction with Approach 1). Meredith could, for example, graph turnover by key employee categories such as performance ratings, education, tenure, compensation, and absenteeism (recall the different ways to analyze data from Chap. 6). She might find that turnover is particularly high for employees of a certain category. Those insights would lead to actions of their own. For example, high attrition in a certain location which also has poor engagement scores might lead the team to focus on organizational culture in that location, whereas high turnover in employees with more than 3 years since their last promotion might lead to conversations about more effective talent management.

Approach 3: Inferential Statistics

The final level is an inferential approach. While approaches 1 and 2 will both probably be leveraged in some way, neither of them really tells Meredith *why* people are leaving. Remember from Chap. 6, as intoxicating as strong patterns in descriptive statistics are, they are only correlations. To know, in scientific terms, she must take further steps.

If Meredith's descriptive analyses led her to believe that poor culture was causing turnover, she could create a hypothesis to explore: "Newly hired employees hired into locations with engagement scores below 75 are 15% more likely to turn over in their first year than employees in locations where scores are above 75." She might even define her idea slightly differently: "Every point of engagement in our annual survey for an employee's location drives a 0.01% increase in their likelihood to stay with us for at least one year." These are testable ideas that inferential statistics can help answer.

Importantly, by the end of any of these three approaches, Meredith will not have a *prediction* of who or how many people will leave. She will have inferences into the relationships that exist between many work-environment factors and turnover which will help her make educated guesses about how to improve the situation. *This is the distinction between statistics and machine learning*. Inferential statistics uses math to tell us whether the relationships between variables are real and dependable. Machine learning uses statistics to find patterns and/or predict how the data will look in the future, which in this case would tell Meredith who or how many people will leave and when.

Approach 4: Machine Learning

At some point between Approach 2 and Approach 3, Meredith realizes something important: "Even if I diagnose why people are leaving, it will still take a long time to reduce turnover. Sure, we will see positive change in our workplace in the long term, but that does not help forecasting efforts for TA, L&D, and other folks negatively impacted by the rollercoaster that is our staffing model today. But if I could predict *who*, or at least how many, were going to leave then we could get out in front of that turnover and minimize its impact on our operations."

Meredith has reframed the problem in an important way. *Why* people are leaving is an important thing to figure out for the sake of making her company a better place to work. However, simply predicting *who or how many* would provide an extraordinary operational advantage in the meantime, as well as potentially provide clues as to why people are leaving.

Taking this approach, Meredith could partner with a data science team to create a machine learning model which takes many of the inputs used in Approaches 2 and 3 and use them to look for patterns in the data. This is a complex process and we will not get into it in detail here (see Chap. 9), but the idea is Meredith has shifted her mindset from "figure out why" to "figure out who and/or how many." Machine learning is well suited to, for example, look at all the data Meredith has about a person (e.g., tenure, time since last promotion, education level, performance metrics, engagement scores, absenteeism) and figure out which attributes are more likely to be linked to turnover. This does not tell her cause, but it does tell her that, "If employee X has a given profile of attributes, they look like someone who might quit."

Machine learning automates much of the data analysis and interpretation that would be performed manually in the previous approaches. In its purest form, machine learning takes two elements, the data the researcher has and the outcome that they want to predict, and then calculates the rest. This is an oversimplification, and machine learning capabilities are broader than just this example, but the idea is that machine learning can look farther, wider, and at more combinations than any team of researchers could manually. And as an added benefit, machine learning produces results that are more reliable, accurate, measurable, and statistically valid than models constructed by hand. The machine learning results are far less subject to human error and bias[1] than approaches 1, 2, and 3.

Suffice it to say, machine learning models use statistics but are not the same as inferential statistics. Descriptive and inferential stats are a critical part of the scientific method leveraged to investigate the relationships and causality which exist between phenomena we have defined through variables. Machine learning, on the other hand, uses statistics to find patterns in data and uses those patterns to identify drivers, create groups, and make predictions about the future.

[1] Machine Learning models are not immune to bias; we will talk about this more in Chap. 10. The point here is that when done well, a machine learning model's objective nature tend to make it less susceptible to the various forms of bias.

Now that we have differentiated machine learning from inferential statistics, we would like to review a few other machine learning-like terms you may have come across recently, so that we may define and differentiate them from the activities of machine learning.

8.2 Fields Related to Machine Learning

Artificial Intelligence and Deep Learning: Probably the most overused term in the popular media when talking about the advent of advanced computing, artificial intelligence is even tougher to define than machine learning. At its core, AI occurs any time a computer is mimicking the cognitive functionality someone would observe in a human. The interesting part of AI is that "mimicking cognitive function" is subjective and has made the classification of AI somewhat of a moving target. Previously it was thought things like optical character recognition (being able to "see" and interpret a visual field) or understanding human speech were sophisticated enough to be called AI, but since they have been so well understood and built into computer processes these days, few still consider them to be true artificial intelligence. When you call the customer service line of your bank and it says, "please say your account number" and it can understand what you said, would you consider that AI? Not likely.

This begs the question because every time computer science figures out how to create an automated way to handle tasks previously only done by humans, the industry says, "well that is not *real* intelligence." This essentially means humans have implicitly defined AI as "anything humans can do that we have not figured out how to do with computers yet." This is called the "AI Effect" and really demonstrates the struggle humanity has with classifying what behaviors and cognitive functions qualify as uniquely human.

So where does that leave AI compared with machine learning? In today's state of computing (and based on the content above), the term AI is usually reserved for systems which model complex behavior which require extremely powerful computation and produce sophisticated behavior. Some examples given today's technology are:

- Self-driving cars
- Competing in Go competitions (a strategy-based board game)
- Advanced robotics able to interact in real space
- Mimicking emotional or social intelligence to allow programs to interact with humans to solve problems or route calls (e.g., high-quality chatbots)

These and other examples are where the AI industry currently lives and is specifically different than machine learning because most AI today requires observable behavior from a system which is considered to be "artificially intelligent." Driving a car, playing a game, and talking to a person online are processes which require real-time input processing and incredibly powerful computation. Juxtapose

this with machine learning because machine learning does not necessarily require real-time behavior or human interaction. Machine learning is more concerned with automating the pattern recognition and prediction development process. A powerful algorithm that can predict who will succeed in a management development program based on the data it is fed is an example of machine learning, but it will simply ingest data, process it, and output information. It will not "behave" in the way most things considered AI today would. All that said, most current definitions of AI still include machine learning as either under the AI umbrella, or at least as an important technique used to enable AI.

Another term you may have heard in the popular media related to both AI and machine learning is "deep learning." Deep learning sounds mysterious and makes a great headline, but it is more of a marketing term than a realm of mathematics or computer science. In Chap. 9, we will talk about many different machine learning models, one of which is the neural network—a complex and sophisticated model for prediction. Deep learning is essentially using complex, neural networks with multiple hidden layers (discussed later) to produce sophisticated prediction and even real-time behavior (which is why deep learning and AI often go together). Deep learning is far too complex for the purposes of this book, but if you are getting into machine learning, it is only a matter of time before you come across the term.

Natural Language Processing (NLP): NLP is a subset of artificial intelligence and leverages machine learning for its advancement. It also has significant relevance in the HR space because it is specifically concerned with using computers to get accurate meaning from language. NLP's focus is to get meaning from written data. Surveys, frontline performance monitoring, benefits assistance, and even coaching are spaces where NLP stands to make a huge impact on the HR industry. It is also a challenging space because even though language is reasonably simple for humans, it is complex for computers. Concepts like sarcasm, figures of speech, and local colloquialisms (like a company with many acronyms that are used in regular speech) are more challenging for computers to handle than many other kinds of predictive signals and in addition must also often be paired with speech-to-text AI in order to work properly. And although there have been great advancements in the last five years, these challenges continue to make NLP difficult to commercialize in any meaningful way, especially in HR. As an example, most modern engagement survey companies claim they use text analysis to "automatically find themes in survey comments." However, if you have ever dug into these algorithms, the systems they use to categorize are usually extremely general and have a lot of trouble picking out even the simplest patterns which are obvious to human readers.

Organizational Network Analysis: ONA is another form of advanced analysis gaining popularity today and is sometimes conflated with machine learning. ONA is the construction and investigation of social networks to map and better understand how individuals communicate. Think about your organization's "org chart." It starts with the CEO, who has direct reports, who have direct reports, who have direct reports, and so on. It looks like a tree with your highest executives at the top. Now imagine, instead of those "lines" being who *manages* who, draw the lines based on

who *communicates* with who and how often. Did this totally redesign where the people in your organization are? If so, this is because the "organizational networks" in a company are based on the work they do and *who they rely on to get things done* which is related to, but not the same as, who they report to.

ONA is a rising form of analytics in human resources because it is such an interesting and potentially impactful way to understand how work gets done among and across groups of people. That said, ONA is not technically machine learning, though it can be used to make inferences and predictions.

A practitioner encountering ONA is likely to be in one of two situations: first, a team conducting ONA might implement a detailed survey asking individuals to identify who in the organization they communicate with and rely on to get things done. The resulting data is then used to construct a network. This approach is thorough and produces the best data but is time intensive and disruptive.

The second approach is one which may already be occurring in your organization. Given how digital the life of employees has become, some organizations partner with their technology providers (the vendors who provide physical computers and software) to analyze the data from email, calendar, internal chat, and internal social media programs to quantify the interactions between individuals. The data gathered is then used to build a network map and insights. This only captures electronic communication, but the tradeoff is that there is virtually no disruption to individual workers.

To support hypothesis testing and statistical inference using ONA, techniques such as Exponential Random Graph Models (ERGMs) were developed. They provide a framework for analyzing networks using a statistical model to support findings. That said, ERGMs far more resemble the world of inferential statistics than they do the world of machine learning.

8.3 Considerations for Machine Learning

We have now defined machine learning by talking about what it is *not*, but before we dive into a more detailed review of its characteristics and techniques, we would like to spend a little time reviewing some considerations to keep in mind when weighing whether machine learning is the right tool for the job.

Beware of the allure of correlations: We have already established, "correlation does not imply causation," and indeed we have mentioned this concept a few times already. But we want to be very clear on this point, machine learning models can identify factors that are predictive of a given outcome, sometimes extremely well. However, always remember that even though models may be very effective while appearing quite smooth, they are still built on correlation, not causation. In correlation, factors might appear to be causally associated with the outcome. This association *does not prove that the factors directly cause or influence the outcome*. Though there may be a causal link underlying the relationship between the variables, the model itself does not prove that; it can only suggest possible links.

To prove causation requires research and analysis beyond the scope of a machine learning model and is more the realm of empirical research supported by inferential statistics.

Models tell you "what," but not "so what": A related concept to correlation not implying causation is that machine learning does not suggest actions. Machine learning techniques can certainly improve decision-making by helping a business understand what is happening, but *the results do not directly recommend action*. The insights on data relationships and predictions about the future made by machine learning models are valuable inputs to decision-making, but they do not tell a leader what they should do.

In the example about Meredith, let's say the machine learning model finds out that education level and time-in-role interact to drive the turnover of their highest performing employees. That is a great insight, but it will not tell Meredith how to fix that. This is where HR experience becomes critical. Meredith and her team must now create an intervention to attack the problem the model has uncovered. This may ultimately require more research to solve. She might use empirical research to help uncover causality, and/or constructive research to help test potential solutions (see Chap. 5).

The past must look like the future: Machine learning models' ability to make predictions is based on their ability to generalize information from the past to predict the future. In much the same way that humans take in data and create generalized mental models of how the world works, machine learning models provide idealized, abstract representations of the world *based on what they have seen in the past*. Recall when we first defined what "learning" was in Chap. 7 by explaining how a young boy learns not to touch stoves. If he comes to a stove that is not hot, he may still avoid touching it because he assumes that the future (the current stove) is like all other stoves he has encountered (the hot stove). This would cause him to make an erroneous judgment.

For our purposes, when considering whether to leverage machine learning to investigate or solve a problem, the practitioner must ask themselves if the past will look like the future. Organizational, economic, and strategic shifts often impact the sorts of data HR examines and the extent of these impacts will influence how effective a machine learning model can be.

This concept also holds true when considering the randomness of outcomes. Machine learning is not good at predicting random events like who will win the lottery. Since machine learning relies on the assumption that the past predicts the future, randomness and machine learning do not go well together.

If the model is good enough to create change, the change will probably break the model: Sometimes a good model will help monitor and predict outcomes, but not necessarily change how business gets done. Other times, a good model will help the business decide to change how they do business in an effort to make things better. This brings the last two considerations together: First, the business has understood the "what" and created an effective "so what." Second, because it is effective the past will no longer look like the future!

If the world changes, models can lose their effectiveness. This means that model maintenance (see Part III) is a key component of any data ecosystem which contains machine learning. The better a model is, the more likely this will occur, and the more important it is that analysts continually pay attention to how well a model predicts, so that they can ensure the models always deliver the best insights possible.

Models can only collect so much data: The outcomes machine learning models are trying to predict can be heavily influenced by variables that cannot be measured, captured, or predicted effectively. With attrition, for example, it is not possible to have data on all the factors that could lead one to leave a company. For example, a spouse getting a new job could drive an employee to move to a different region and thus leave a job. It is not reasonable to expect to have access to all possible information on employees. And without everything, a model may miss significant elements which could impact its accuracy. This is a consideration to keep in mind when deciding whether machine learning is the right tool for the job—does the data cover the important aspects of what will drive the outcome? This gets back to *Researching your Research* from Chap. 5 because human behavior is complex and difficult to measure which makes many HR problems particularly challenging for machine learning models. Understanding those limitations allows the practitioner to approach and frame projects appropriately.

Size matters: Machine learning models work best when the outcome being predicted occurs consistently and frequently. Rare events, or events that only occur in small numbers, pose significant challenges for machine learning models. Machine learning models need data to understand events and how they occur. Though there are techniques to deal with rare events, they stretch the limits of statistics and machine learning and require advanced knowledge. Some techniques require more data than others, and you should partner with your analytics or data science team to understand if you have enough data to train a model to predict what you want to predict.

As complexity of relationships increases, so does the complexity of the model: Traditional machine learning algorithms work best when the problem is not too complex. That is, the problem must be able to be solved through a relatively straightforward set of interdependent correlations. If a problem has a large number of intertwined factors influencing the result, the model may have trouble finding the relevant patterns.

A more practical way to frame this is that sometimes an analyst is trying to predict things that are the result of multiple, layered decisions or a complex chain of events. In these scenarios, traditional machine learning techniques may struggle to predict. These types of problems require even more advanced methods (and typically huge data volume) like deep learning and reinforcement learning, which will be discussed in Chap. 9.

Always start with good data: Finally, the number one key assumption in every model is that the underlying data is accurate and of good quality. Going all the way back to the first mention of data quality in Chap. 2, practitioners must always keep "garbage in, garbage out" in mind when creating models. HR data is particularly notorious for having quality issues, so specific attention must be paid to data integrity. This typically requires a significant partnership between analytics teams (who understand data and data systems) and HR practitioners (who understand what the data is trying to quantify).

Now that we have differentiated machine learning from a few other types of analyses and reviewed considerations for its use, let us talk about some of the characteristics that you can use to define types of machine learning, as well as introduce some common techniques which traditionally qualify as machine learning.

Machine learning is whenever a computer-based model can use data from the past to find patterns which help us predict the future. When working with analytics and data science teams, there are a few key components of the methods which are worth a conversation. These are not meant to be an exhaustive list of every characteristic and consideration for choosing a model, but it is a good list to get the conversation started. We will review a few of these in more detail in subsequent sections.

Characteristic	Scale	Definition
Transparency	• Transparent • Opaque	Is it feasible to look "under the hood" of the model and understand how the model created its results?
Overfitting risk	• High • Medium • Low	Does the model tend to over-estimate its effectiveness or have data preparation aspects which may result in accidental collinearity? Essentially, is there a risk that an effect is seen when there is not one?
Supervision	• Supervised • Unsupervised	Do we feed the model examples of the outcome we want to predict, and ask it to learn how to achieve that outcome? Or, are we asking the model to search for patterns without a specific outcome in mind?
General complexity	• High • Medium • Low	How much technical acumen is needed to build, execute, understand output, and explain the results of this model?
Data required	• Very big • Big • Moderate	How much data is needed to design and validate this model? Includes cases (rows) and variables (columns) as well as training data versus testing data. Some models need dozens or hundreds of data points to get started, while others need 50,000+.
Computing time	• Hours+ • Minutes • Near instant	Once data is prepped and ready to be run, how much time is needed to execute the model?
Ease of tuning	• High • Medium • Low	How easy is it to use your results to interpret effectiveness and edit the input or model settings to iterate on your model and make it better?

8.4 Transparency, Opacity, and Overfitting

Key Definitions

- Transparent: Easy to perceive or understand
- Opaque: Difficult or impossible to understand
- Overfit: When a model predicts existing data well but predicts future data poorly because it relies too heavily on the idiosyncrasies and noise in the training data

In Chap. 5, we briefly mentioned that some types of machine learning are "transparent" while others are "opaque." In the following sections, we will be using these terms often because it is one of the defining characteristics of different machine learning techniques.

Essentially, transparency is another term for "explain-ability." The more transparent a model it is, the easier it is to understand how it works and why it is making the predictions it is making. Remember, some machine learning techniques examine millions or billions of possibilities while they are searching for patterns. In general, the larger the data and more sophisticated the patterns a model finds, the harder it is to explain.

Lack of transparency is not always an issue in machine learning. Often, the analyst cares only that a model predicts well, but not how or why that model predicts well. However, in HR "explain-ability" is often a critical, if not legally required, attribute of a model. This is especially true if the model is going to be used to influence decision-making around activities like hiring, promotion, or admission to developmental opportunities like high potential programs. Additionally, from an ethical and organizational culture perspective, these sorts of decisions should be explainable to leadership and potentially even to candidates and employees who may be rejected due to the prediction of a model. It is frustrating enough to not be selected for something, but even more frustrating to have the only developmental feedback be "the algorithm said so."

The last few years have enjoyed exceptional focus on driving transparency in models which have historically been difficult to interpret. As an example, tools like SHAP allow us to peel back the layers of traditionally opaque methods like neural networks and raise to the surface the drivers of these very complex models. This is critical, because legally the logic for these decisions must be understandable in case they need to be justified during an audit by a government agency like the Equal Employment Opportunity Commission or defended during legal proceedings. Opacity of a model is not a legitimate legal defense. More on this in Chap. 10.

Overfitting is a different phenomenon to understand. To explain, let us first revisit a concept discussed in Chap. 5: randomization.

The reason for randomization of a sample is that every dataset is a little different. The attributes which could be measured about individual cases in any dataset are near-infinite. Therefore, when designing samples choosing truly random participants will give the best shot of creating a group who represents the population as a whole. The same principle is true when using data to train machine learning models. Every dataset will have these idiosyncrasies, often referred to as "noise" in the data.

Overfitting is when a model picks up on this "noise" and treats it as important for prediction rather than as random. This means that it might make the model look better for this particular sample or training dataset, but then does not do as well when being used on data it has not seen before.

Some methods are more susceptible to overfitting than others, and combating overfitting is definitely a job for advanced analytics professionals and data scientists. That said, it is an important concept to understand when partnering with these teams or venturing into basic modeling for yourself.

Now let us talk about the two major ways machine learning models are classified: Supervised and Unsupervised.

8.5 Supervised Learning: Taking a Road Trip

One of the most important adjectives attached to machine learning methods is "supervised" versus "unsupervised" learning. These are two very important distinctions because they differentiate whether the methods being employed are (a) aimed at mapping a solution from examples of inputs and outputs or (b) trying to pick out the important features in a dataset.

Supervised learning is the type of machine learning from the PB&J example. In each of the 100,000 cases, the computer saw examples where the end result was a successful or unsuccessful PB&J sandwich. This makes sense when thinking about the general concept of "teaching" (which is the process to facilitate learning). If a parent is trying to teach a child to make a sandwich, taking them outside to have them watch someone wash the car will not help much. They need to observe the actions linked to the outcome being taught. As they watch, they can pick out the patterns of what leads to a successful sandwich. It is not unlike following a map. If Erin wants to get from New York City to Jacksonville, Florida she can look at a map and see that many roads go south and that Jacksonville is about 950 miles away. In fact, cars do it every day. Stated in a different way: "Erin can predict that if she drives south from NYC for 950 miles, she will be in Jacksonville, Florida."

Supervised learning is the branch of machine learning dedicated to following trails that have already been blazed. New York City exists, Jacksonville exists, and many roads between them exist. If Erin watches enough examples of the successful journey from one to the other, she will be able to do it too, just like learning to make a sandwich by watching someone do it many times.

Supervised learning does get more complicated. There is semi-supervised learning, where the computer is only given partially labeled data; reinforcement learning, where the new information is only given as feedback to the computer's actions; and active learning, where the program can proactively seek information when it needs to. We will introduce some of these later, but the computer science behind these is advanced and not really fit for this book. Suffice it to say that supervised learning has many different permeations which fit many different scenarios data scientists help solve for.

Because supervised learning has such applicability to practical concerns, like whether someone will buy a product, survive a medical procedure, or default on a loan, supervised learning is an extremely popular form of machine learning. Supervised models are given a defined goal that they are expected to work toward. And like the learning concepts from earlier, the model is trained using data that includes data points which relate to the desired outcome. This approach, using a specific objective and representative sample data, is why it is considered supervised: the learning takes a predefined path from the data it has to the outcomes it is trying to understand.

In HR, a predictive attrition model is a good example of where a practitioner might apply supervised learning—the algorithm is instructed to predict whether a given employee will leave the organization. There are two possible predictions: the individual will depart or they will stay. The algorithm is provided with a training dataset that includes potential predictors (tenure, absenteeism, performance, etc.) along with samples of both possible outcomes. Recall Chap. 4's discussion of variables (columns) versus cases (rows) when defining our question in a way that it is answerable with data. To a supervised machine learning algorithm, rows represent occurrences, while columns represent the predictors and the outcome which can be different from occurrence to occurrence (row to row).

Another potential application in HR for supervised learning are applicant performance prediction models. Choosing the best candidates out of a pool of hundreds or thousands can be a challenging task fraught with risk and bias. Many companies have turned to machine learning models to improve their candidate selection process, reduce bias, and optimize quality of hire. Machine learning works best for this type of problem when a company hires many people in the same type of role, like call center agents or warehouse workers. In this case, companies would likely have many examples in their data systems of hires who turned into successful, high performers as well as those who did not. They can then train the model to find what makes high performers unique and hire for those particular attributes.

To bring it back to a simpler example, let us imagine each row of data is one time a PB&J sandwich was attempted. Each column would be something about the event (e.g., type of jelly, type of knife, size of a piece of bread, etc.). Importantly, one column must be: "was the attempt to make the sandwich successful or not?" Some of the rows will be successful attempts, while some rows will be unsuccessful attempts. Then, the algorithm will be able to look at what the successful attempts have in common, what the unsuccessful attempts have in common, and be able to tell the difference. Then, in the future where there is no value given in the "successful or unsuccessful" column, the algorithm will be able to predict what value should go there.

8.6 Classification versus Regression

There are many common types of supervised machine learning techniques. We will not get overly technical, but it is important to have an overview of these approaches and generally how they work. If you are an HR professional, this will help you partner with data scientists and analytics professionals and if you are already a data scientist, we hope some of these explanations will help you do the same when explaining differing approaches to your business partners in HR.

There are two main categories of supervised learning: regression and classification. In regression, the goal is to predict a numeric result based on a set of input variables. In classification, the goal of the algorithm is to assign data to specific categories or classes. Both forms of supervised learning are common, though classification is used more frequently in human resources and in many other industries.

Type	Name
Classification	Logistic regression
	K Nearest neighbor
	Support vector machine (SVM)
	Decision Tree
	Random forest
	Neural network
Regression	Linear regression
	Polynomial regression
	K Nearest neighbors
	Support vector regression (SVR)
	Neural network
	Regression trees
	Random forest

As mentioned earlier, and just as with statistics, when working with supervised learning there are two categories of variables that are used—independent (manipulate) and dependent (measure). In machine learning models, the independent variables are often called "predictors," "features," or "parameters," while the dependent variable is often called "the outcome." A major difference between classification and regression is that in classification models, the outcome variable is a category or class (yes/no, stay/quit, blue/yellow/green). In regression models, the outcome variable is a number.

To make a prediction, the algorithm will be fed one or more variables. These are the independent variables or predictors. The independent variables are what we manipulate, or in this case choose to feed the model, so that the model can understand what relates to the outcome. Ensuring that the appropriate independent variables are chosen to predict a dependent variable is essential to the success of supervised learning models.

As mentioned, classification is the most common form of supervised learning used in human resources and in most industries. The goal of classification is to predict which group (or "class") each case belongs to. Classes are predefined from a fixed list of labels. In the example of the PB&J sandwich, this was whether making the sandwich was a success—yes or no? Like this example, most implementations of classification models are binary, which means that there are only two possible classes or labels. The earlier example of a predictive attrition model may class employees as "stay" or "depart," which is a great use case for a binomial classification model. Multinomial classification, or classification that involves assigning an observation to a class that contains three or more possible labels, may also be used. However, not all classification algorithms support multinomial classification in their implementations. An example of multi-label classification would be a model which can identify multiple features in a picture to classify a complex object. A model could be honed to recognize doors and windows. This is not commonly used in human resources currently and is used primarily for video and images.

As an additional part of generating a classification, many classification algorithms also produce a probability value along with their prediction (between 0% and 100%). The intent is to provide the researcher with the likelihood that the classification is right.

That is to say, the algorithm might predict, "Yes! Successful sandwich completed, but only 25% certain." This means the algorithm thinks it found a good sandwich but is not sure it got it right. Having access to these probabilities allows data scientists to "tune" the model, which means achieve a balance between error rates like false negatives (we said no sandwich when it was in fact a successful sandwich) and false positives (we said good sandwich when it was actually an unsuccessful sandwich).

Regression models, on the other hand, are forms of supervised learning that produce quantitative results. Specifically, they generate continuous, numeric values based on a set of input variables. The output is a real number as opposed to a label or category like would be produced for classification. An example of a regression problem is a model that predicts home prices. The algorithm would be provided factors such as square footage, number of bedrooms, and age of the home. It would then predict home price based on those variables. The key is that the algorithm produces a number (e.g., $150,000) and not a category or class.

An example in HR might be to attempt to predict the headcount requirements for a given frontline location like a plant, warehouse, call center, or retail outlet. Using data on workload, performance metrics, expected volume, etc. of the location (and similar locations), a regression algorithm could help identify how many person-hours would be required to produce acceptable performance levels. Again, the important difference is the algorithm is producing a number, not a label.

8.7 Unsupervised Methods: Blazing a Trail

In 1802, the United States consisted of most North American land east of the Mississippi River. The Spanish had claimed modern-day Texas along with most land flowing west and north through modern-day states like New Mexico, Arizona, and southern California. The French had laid claim to a huge swath in the middle, starting in modern-day Louisiana, running north along the Mississippi River, and extending west where it met its land border with the Spanish lands. The French had only explored and settled a tiny piece of this land: near where the Mississippi empties into the Gulf of Mexico.

In 1803, the United States purchased all France's land for $15 million (over $250 million in today's dollars), nearly doubling the size of the United States. There was just one problem: no European had ever set foot on the vast majority of its acreage. It was entirely uncharted territory[2].

[2] For the purposes of this book we are not discussing the fact that these lands were *already* settled, but the authors would like to acknowledge that they were.

Enter Captain Meriwether Lewis and Second Lieutenant William Clark. Shortly after the purchase, President Thomas Jefferson commissioned a party of Army volunteers, led by the now legendary Lewis and Clark, to explore the territory. The United States saw potential in what the land had to offer, and they wanted to reap the benefits of the territory. However, they did not know where to start, so they set off to explore.

If supervised machine learning is taking a road trip, then unsupervised machine learning is exploring uncharted territory. If supervised learning allows us to follow the United States' highway system from New York to Florida by learning from what we know works and modeling against it, then unsupervised learning is American settlers figuring out how to get from St. Louis to Oregon at the turn of the nineteenth century.

Unsupervised machine learning is like Lewis and Clark being sent out into the North American wilderness because Lewis and Clark did not have roads, speed limits, or gas stations. There were no documented trips for them to model their trip against. Nobody could say, "A successful trip from Missouri to the Pacific Ocean looks like this."

This is how unsupervised learning differs so greatly from supervised learning. In supervised learning, the data scientist knows what data are the predictors and what data are the outcome. They know where they are trying to go (outcome variable) and what variables they think are going to get them there (predictors). In unsupervised learning, the data scientist knows in general where they are *trying* to go, but their data is either not labeled or not structured in a way that they know what they want the outcome to be.

Then what is the point? If the data scientist is not trying to predict something or show some relationship, then why would they ever use unsupervised learning? For the same reason Jefferson commissioned Lewis and Clark: they think and hope there's something to find, even if they do not know exactly what it is yet.

Unsupervised machine learning methods can be categorized into three major types. These key concepts will provide an overview of the different purposes of unsupervised learning, and we will get into a few specific methods in Chap. 9 which you are likely to come across in HR. If you are very interested in a deep dive on unsupervised machine learning techniques, we recommend a more technical text on machine learning or a class in advanced data science.

Clustering: Probably the most common form of unsupervised learning is clustering. Clustering is intuitively named, since the goal is to create "clusters" of data which resemble each other and then label them as such. In clustering, the data is provided to the algorithm and minimal direction is given on how to analyze it. The algorithm looks at all input variables and data to determine logical groupings of instances. Specifically, the algorithms look for points that are similar to each other, but dissimilar from other data points. Two popular types of clustering used in machine learning are k-means clustering and hierarchical clustering. There are others, such as OPTICS and DBSCAN, but we will not review them in this text.

Additionally, it is important to note that clustering is not just a machine learning category, but also a class of optimization problem-solving techniques. The idea of looking at data and then segmenting it appropriately to understand similarities and differences has applications across virtually all fields of science and engineering. Machine learning's claim to cluster analysis is but one segment of its applicability to science and industry.

In HR, cluster analysis can be used to do many interesting things. In the introduction to this chapter we discussed "Create Groups" as a main value-add for machine learning, and indeed cluster analysis is one of the primary ways to do this. The example of using machine learning to help define what constituted a "new hire" was a great example of clustering's applicability in HR. In fact, any time a practitioner wants to explore potential segmentation of a population of employees, clustering can be an interesting method to pursue. Patterns in benefits utilization, training activity, performance metrics, or others are just a few examples of avenues to explore which can create valuable insight.

Anomaly Detection: Another intuitively named category, the purpose of anomaly detection is to find data that does not fit and call it out. In practical application, anomaly detection is most often used in spaces like fraud detection, cyber-security, and data quality auditing where outliers can have a major negative impact.

HR has not yet found widespread application for anomaly detection. There are cases in HR where anomalies can be good (e.g., sustained exceptional performance) or bad (e.g., absenteeism), but data size and other limiting factors have not yet been overcome to show large value for this type of unsupervised learning[3].

Dimensionality Reduction: Finally, dimensionality reduction is one of the most important, though least glamorous, uses of unsupervised methods. Data scientists want the input data used for their models (i.e., the predictors) to have the strongest relationship possible to the outcome variable, while using the simplest model achievable. But data is often big and complex. Dimensionality reduction is the art and science of reducing what is input into the model to its simplest form and sometimes discovering new variables along the way.

This idea in practice can get quite complex, but to simplify let us start with three challenges that exist in many datasets and explain how they get in the way of simple-yet-effective models:

1. *Covariance*: this is when two or more predictors vary together. If two variables covary it means they are distinct variables, but they may be telling the same story. For example, consider a person's height and a person's weight. Does weight cause height? No. Does height cause weight? Sort of. There are tall light people and short heavy people, so the relationship is not perfect but overall, these two attributes will vary together, and depending on the research question may not provide unique value to a model because they are so closely related.

[3] Anomaly detection is not restricted to unsupervised learning. There are supervised and semi-supervised versions as well. We included it here because unsupervised is one of the most popular and widely applied methods of anomaly detection.

2. *Noise*: This is data in your dataset which is irrelevant. Said differently, data that does not help predict the outcome you are looking to predict.
3. *Latent factors*: Latent factors are variables which influence data, but are not, or cannot, be directly measured. Latent factors are a huge part of working with HR data because so much of what HR attempts to predict is not directly measurable. "Engagement," "potential," "manager quality," and so many others are simply not measurable directly in the same way someone's height, work location, or job code are. A great example of a famous latent factor is introversion/extraversion. Nobody has ever been able to directly measure how introverted someone is in the same way they can measure their age or weight. But by asking several related questions (which in a data table each look like their own variable), a researcher can *infer* the presence of this *one* latent factor. We will discuss more about latent factors in Chap. 10 when we review a concept called "The Construct Chasm."

A few unsupervised methods which help reduce dimensions in our data by combatting covariance and noise and discovering latent factors are (1) principal component analysis, (2) independent component analysis, (3) non-negative matrix factorization, (4) discriminant analysis (of which there are many types), and (5) factor analysis. Admittedly, the line between advanced inferential statistics and machine learning gets blurred here, but we would like to err on the side of inclusion for the sake of introduction to dimensionality reduction.

Section Breakout: The Cocktail Party Effect

A simple way to remember dimensionality reduction is by thinking about what is called the Cocktail Party Effect. When you are at a crowded, noisy event like a dinner in a restaurant, out at a bar, or attending a cocktail party, you can still have a conversation with a friend near you. Despite all the noise going on due to waiters, other conversations, and maybe even a live band, you can somehow focus on just the sound that matters to you—the voice of your friend.

The Cocktail Party Effect is usually used to demonstrate the neurology of auditory attention and signal separation (deciphering mixed signals in audio data), but it draws a nice analogy. Your dataset is just a different version of all that noise at the cocktail party, while what your friend is saying is what you are trying to get out of all that data. Machine learning for dimensionality reduction can help you hear what is important to you.

This chapter touched many topics and potentially brought many new terms to the reader. Supervised, unsupervised, inferential, artificial intelligence, deep learning, natural language processing, classification, regression, opacity, transparency, overfitting, clustering, latent factors… if you are new to statistics and machine learning, this may have been overwhelming. But do not worry, we are not preparing for a

deep dive into highly technical content. This chapter was meant to set the stage by introducing some important concepts to walk away with:

- Machine learning helps (1) identify drivers, (2) create groups, and/or (3) predict outcomes.
- Machine learning is related to, but not the same as, many fields. Some of the most prominent are inferential statistics, artificial intelligence, deep learning, natural language processing, and organizational network analysis.
- There are several key considerations when using machine learning to problem-solve, like continuity between past and future, and how much transparency is needed based on what the model will be used for.
- The two main types of machine learning are supervised and unsupervised, each having their own methods and best applications. Supervised methods are for when the researcher knows what success looks like and wants to train a model to find the patterns that predict that success. Unsupervised methods are for when the researcher is not sure what the data needs to predict, or the data is not well-structured or labeled. Use unsupervised methods to explore data to find patterns when unsure of what the outcome will be.

In the next chapter, we will keep the concepts from this chapter in mind as we dive into many of the machine learning methods which may have application for the HR practitioner of tomorrow.

Discussion Questions

1. What is the difference between machine learning and inferential statistics? Why does it matter?
2. Name four major considerations for machine learning and explain why they are important.
3. Explain what transparency is and why it is usually so important in HR Analytics.
4. What is overfitting? Why is it so important to watch out for?
5. What is the difference between supervised and unsupervised learning? Which is more commonly used in HR?

Chapter 9
Common Machine Learning Techniques

This book is not designed to be significantly technical, and so far has navigated research methods, statistics, and an introduction to machine learning without going too deep down any technical paths. To keep to that mission, the machine learning methods discussed in this chapter will be high level and introductory. As with statistics, this is *not* intended to be an exhaustive view of all types of machine learning techniques nor a detailed how-to manual. This chapter will (1) introduce many of the machine learning techniques HR professionals may come across as a practitioner getting involved in analytics, (2) will explain in general how the techniques work, and (3) in what applications the analyst or practitioner may find them useful in HR. Keep in mind while reading:

1. If you are an HR practitioner, we would like you to become familiar with the techniques and, in general, what they do and how they can be used.
2. If you are a data scientist, we would like you to become familiar with some potential applications for these techniques in the HR space.
3. For both, these overviews can help create common language to facilitate conversations between HR and analytics/data science teams.

So if you are not a deep statistics or computer science professional, do not worry—the following sections will review many techniques which can be used to generate machine learning models, but not in a way that is too technical to follow. And if you are from a data science background, skim this chapter for helpful tidbits and HR examples as well as learn some ways to help discuss these topics with HR and business partners. If you are interested in a deeper understanding and would like to learn the skills required to execute the methods we are about to discuss, we strongly encourage you to explore more specialized texts in the field of data science and machine learning.

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_9

9.1 Linear and Non-Linear Regression

The best place to start the exploration of methods is with a method at the intersection of statistics and machine learning: linear regression.

Linear regression is one of the most common advanced statistical processes in use today, and often makes its first appearance to students during basic statistics courses. Due to its simplicity, it can look inferential in many ways, but it is included here for one major reason: it produces an equation which can be used to make predictions.

Numerous algorithms are available for use in regression problems. They vary in complexity, power, and flexibility. Picking the regression algorithm best suited for a specific problem requires an understanding of the nature of the data and business goals. And even though it dates from the early 1800s, linear regression still possesses many advantages that make it a practical option and a good place to start when working on a regression problem.

There are two types of linear regression and one type of nonlinear regression to introduce here—they differ based on (1) the number of predictor variables in the model and (2) the expected relationship between the variables.

Before getting into that, why use the term "linear" as a label for this type of regression? Linearity is a concept we do not discuss deeply in this book but is a basic premise to understand. When variables relate to each other linearly, it means their relationship can be represented by a straight line (as X goes up Y goes up, or as X goes up, Y goes down). You may remember this as $y = mx + b$ from algebra class. You can refer to more comprehensive statistics textbooks for detailed explanations of linear and nonlinear relationships.

In linear regression, when the researcher is using only one predictor variable, it is called a simple linear regression. When using two or more predictor variables, it is called multiple linear regression.

The model for simple linear regression can be plotted and visualized in two-dimensional space. Let us use a business problem to illustrate.

Imagine you are the lead analyst for a team trying to help optimize staffing in your organization's call centers. You have 1500 call-taking agents spread across five call centers. An important performance metric you track is something called "shrink." Shrink, sometimes referred to as "shrinkage," is the difference between the time your agents are scheduled to be on shift compared to how much time they spend actually taking calls. Part of shrink is predictable: training, team meetings, lunch breaks, etc.—these all contribute to valuable time for agents but are not hours spent taking care of customers.

However, a large part of shrink is time which is unplanned and variable. Sometimes it is unexcused, like in the case of lateness, absenteeism, or larger-than-normal gap time between calls. Other times it is planned but still variable, like in the case of scheduled vacations, jury duty, or leaves of absence.

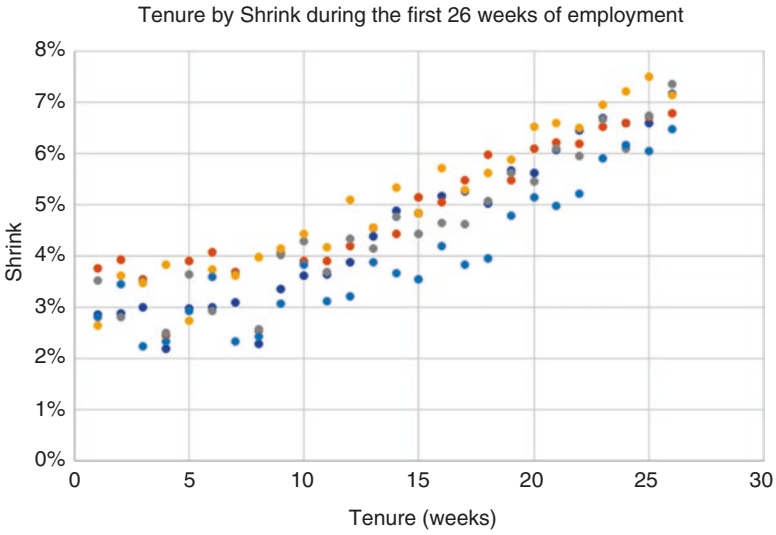
As an additional factor, your call centers have a high, but not particularly abnormal, turnover rate of 35%. This means that during any given year you are hiring

about 525 new agents, all of whom have different behavioral patterns than the 525 who left. This also means that at any given time, about 35% of your workforce is brand new to the organization.

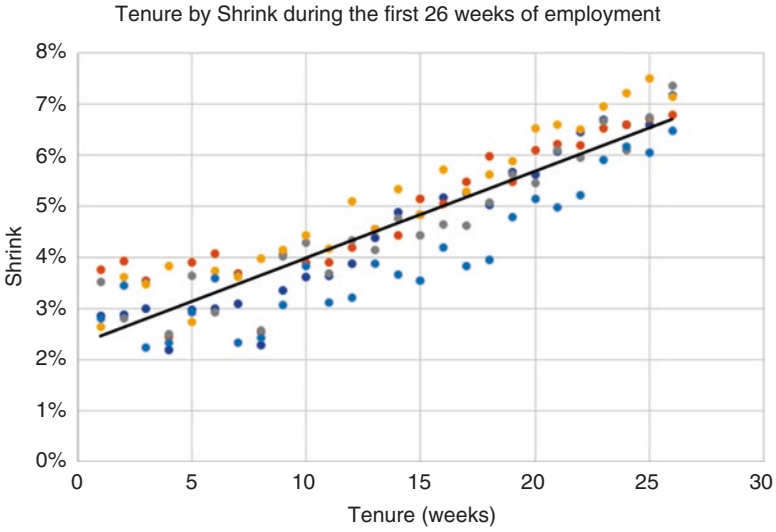
The final important piece of context is that you noticed employees do not act exactly the same when they are first hired as they do when they get more experience. When new hires are in training and in "incubation" (the first 4 weeks they spend on the phones after training), their unplanned and variable contribution to shrink is very small: absenteeism, tardiness, and the other factors are quite low. However, as time goes on you notice that these factors creep up until their rate is equivalent to the rest of the agent population.

Your leaders want you to help them predict shrink. If you can do this, they will better understand how many agents they need to have working at any given time to handle projected call volumes. How could linear regression help?

Simple linear regression uses one predictor variable to predict one outcome variable. Since you recognize this relationship between tenure and shrink, you think to yourself "I can predict how much shrink someone will have based on how long they have been with the company." When you graph one against the other for five random new hire employees, it looks like this:



Linear regression computes the relationship between the predictor variables and the outcome variable. The resulting formula can then be used to make predictions about the outcome based on the values of the predictor variables. The goal is to make an equation for a line that fits well on the data, called the "regression line."



Once you have the equation for the line, for any given value of tenure (X), we can now compute an expected value for shrink (Y).

This visualization is easy to follow when using one predictor to predict one outcome. Plotting multiple linear regression is much more difficult to do because you quickly run out of visual dimensions to plot the graph. For example, let us say you realize that there is more that can help predict shrink than just tenure, like what shift a new hire gets assigned to, their performance metrics, what queue they are in (i.e., what types of calls they take), or others. Multiple linear regression can do essentially the same thing as simple linear regression, except using multiple predictors. That is, instead of the value of a single X predicting the value of Y, there are multiple X's (all the predictors) which each contribute a portion of what the value of Y is expected to be. We will not get into the details of how this works, but the principle is the same: X (or many X's) predict Y.

A key advantage to linear regression is that it is transparent in both the underlying math and in its output. Linear regression produces a formula that is easy to interpret and understand. Each predictor is assigned a weight (how much it contributes) to the overall value of the outcome variable. By having a transparent formula, the factors that are important to the model can be clearly communicated to leaders and stakeholders. The actual drivers of specific decisions are also easier to see in linear regression than in other models. Because of this clarity, linear regression is one of the most commonly employed regression algorithms in human resources. Many newer algorithms are far more opaque, and thus require significant effort to understand and interpret.

From a computing standpoint, since the formula produced by linear regression is (algorithmically) simple, it can be also be executed efficiently on new data to make predictions. For situations where minimizing lag and processing time are important, linear regression can be a good solution. Also, given the simplicity of the resulting formula, models can be ported easily to other languages and platforms. For example, a model that was developed in Python can be rewritten in C++ or Java relatively easily.

For all its advantages, linear regression is not without constraints and limitations. First and foremost, the algorithm assumes linear relationships between the variables and the outcome. If the true relationship between a key variable and the outcome is curvilinear, exponential, etc., the model will struggle to predict. This could lead to inaccurate or unreliable predictions. When this is the case, sometimes the data can be transformed[1] to make the relationship linear.

Linear regression will also struggle if multiple variables in the model covary with one another. As we mentioned earlier, covariance can confuse a machine learning model because the algorithm cannot tell that two things are telling the same story. This also can lead the algorithm to distort the value and weight of the individual predictors. The developer must be diligent to identify multicollinearity and remove it where possible. Dimensionality reduction as reviewed in Chap. 8 is a great way to do this. Variable pruning, which is essentially a systematic approach to removing predictors from your model and seeing how it changes predictive power, is another method to reduce multicollinearity.

A final common risk of linear regression brings back a watch-out discussed in Chap. 8: overfitting. Overfitting is when the model performs well on the training dataset but poorly on the test dataset or new data because the model pays too much attention to the unique attributes of the training data which do not generalize to non-training data. Said more simply, the factors that the model thinks are important are not important when looking at data the model has never seen. There are multiple techniques that can be employed to reduce overfitting such as regularization, which penalizes large coefficients to avoid overfitting but is not a technique we will get into here.

9.1.1 Multiple Regression and Polynomial Regression

It's important to note that data transformation techniques like log transformation are a very common way to deal with suspected non-linear relationships. If a data scientist can use math like this to make the relationships linear, this is usually the preferred route. However, when they cannot, a great way to address the limitation of linear regression's dependence on linearity is by using polynomial regression. But what is "polynomial" and how is it different?

In linear predictions, we explained that the relationship can be represented with a straight line. In linear lines, $y = mx + b$ where y is what you are predicting, x is the predictor, m is the slope of the line (how "steep" the line is), and b is where the line crosses the Y axis (i.e., where $x = 0$).

In simple linear regression, the equation looks a little different, but essentially gets at the same concept:

$$y = \beta_0 + \beta_1 x + \varepsilon_i$$

"x" and "y" are still the variables whose relationship you are exploring. The difference in linear regression is that β_0 and β_1 have taken the place of the y-intercept

[1] We review the basics of data transformation in Chap. 13.

and slope respectively. There is also a new value, "ε_i" which represents the error in predicting the value for *Y*, given a value for *X*.

In multiple regression, we just add more *x*'s with their own coefficients:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon_i$$

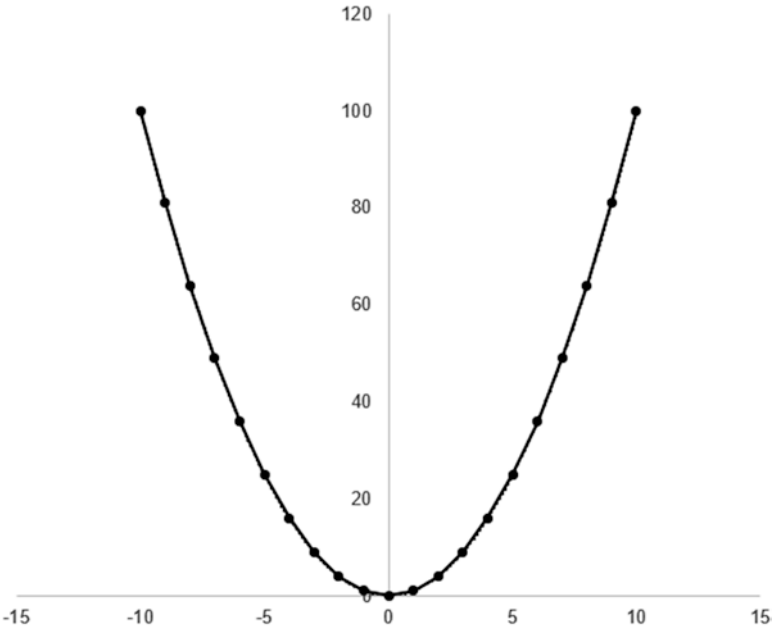
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon_i$$

…and so on for as many predictors as needed.

Polynomial regression introduces exponents to this equation. When exponents are added to a variable, the lines are no longer straight. Here is an example of a common exponent: a parabola defined as $y = x^2$.

When a number gets squared, it gets multiplied by itself which means (1) negative numbers become positive and (2) as the number gets bigger, the product gets *much* bigger:

When $x = -2$, then $y = -2^2 = 4$.
When $x = -4$, then $y = -4^2 = 16$.
When $x = -8$, then $y = -8^2 = 64$.



The possibilities for polynomial regression do not end at x^2. Any exponent can be part of a polynomial regression: x^2, x^3, x^4, and on and on (all of which create different shapes). Polynomial simply means exponents have been introduced to the formula for the regression line.

9.2 Logistic Regression

Another common machine learning method is logistic regression. However, there is one confusing part about the naming of logistic regression: it is not for regression problems!

Logistic regression is actually a method for classification. Whereas linear, multiple, and polynomial regressions are for predicting continuous outcomes, logistic regression is for using predictor variables to predict and place a class label on the outcome variable.

Then why is it called regression? Logistic regression gets its name because it uses the same fundamental mathematic principles we talked about in the other regressions: use predictor variables to build a regression equation (i.e., line) that can predict the value of the outcome variable.

The difference with logistic regression is that instead of using weights of x to compute a value for y, logistic regression uses a logit or logistic unit. A logit is the logarithm of the odds that an event will occur. This sounds very complicated, but the principle is simple:

Back to the example of you as an analyst working on shrink in your organization's call centers. Now that you have helped predict shrink using multiple regression, your organization would like you to help combat it. To do this, you would like to predict which newly hired agents will drive the most shrink in the future and provide them remedial help to increase their productivity before shrink becomes severe. You hypothesize that by looking at things like attendance, shift tardiness, break tardiness, formal disciplinary action, and shift schedule during the first 3 months of employment, you can predict if an agent will end up in the worst 20% of shrink offenders by the end of their second year of employment.

Logistic regression can help here because the problem has been defined as new hires falling into one of two classes: "people who need shrink help" and "people who do not need shrink help."

A logistic regression model will look at the predictor variables and outcome variable you train it on, but instead of predicting a value for amount of shrink, it will predict what side of the "needs help" line that a new hire is likely to fall on. Said differently, a logistic regression equation will build a model that calculates the

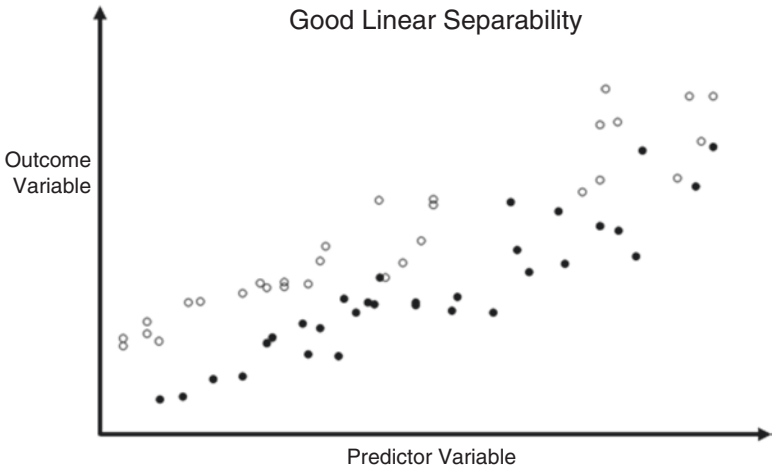
probability that an employee will end up in class 1 (needs shrink help) or in class 0 (does not need shrink help).

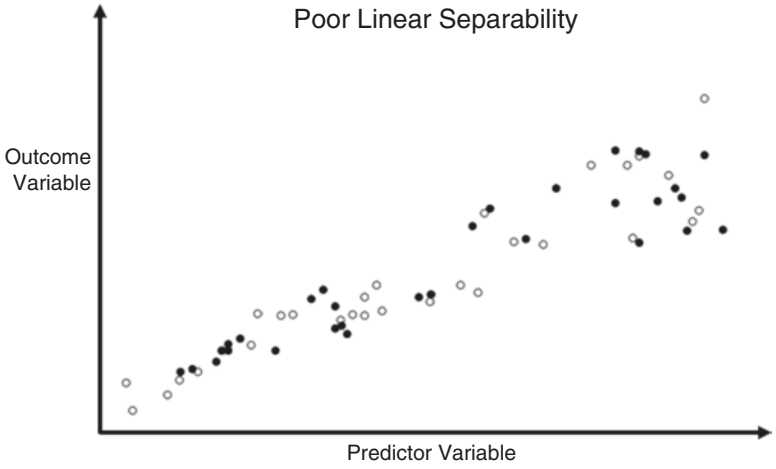
Two of the advantages of logistic regression are that it provides both visibility into the factors driving the model overall (in general what predicts bad shrink) and the factors driving an individual prediction (confidence about a particular case). This transparency is the main reason logistic regression is one of the primary classification algorithms used for human resources projects. As we will talk about more in Chap. 10, with most HR efforts it is important to understand the factors driving a model, especially if that model is being used to drive a decision (like whether or not to provide preemptive remedial training). When transparency into the factors is needed, logistic regression is a good choice. It is easy to understand and interpret, and few other classification algorithms provide the level of transparency that logistic regression offers.

Though logistic regression is frequently used, it does have considerations which essentially mirror its regression cousins. First, there is a significant risk of overfitting, which regularization can help mitigate. Second, multicollinearity is important to watch for. But as with linear regression, dimensionality reduction and variable pruning can help here.

Third, logistic regression assumes and relies on linear relationships. So just like simple linear and multiple linear regression, there must be linearity in the relationship between the predictors and the outcome variable. That said, data transformations can be used to attempt to establish a linear relationship, though this is advanced and can require significant effort. Often, if the relationships between variables are likely nonlinear, other algorithms would be a better choice.

Finally, logistic regression works best when the classes are linearly separable. This means that the regression line can be drawn in a way that puts most of class 1 together on one side of the line and most of class 0 on the other side of the line.





If there is too much overlap between the points, the classes are not linearly separable, which means logistic regression will have a tough time performing optimally.

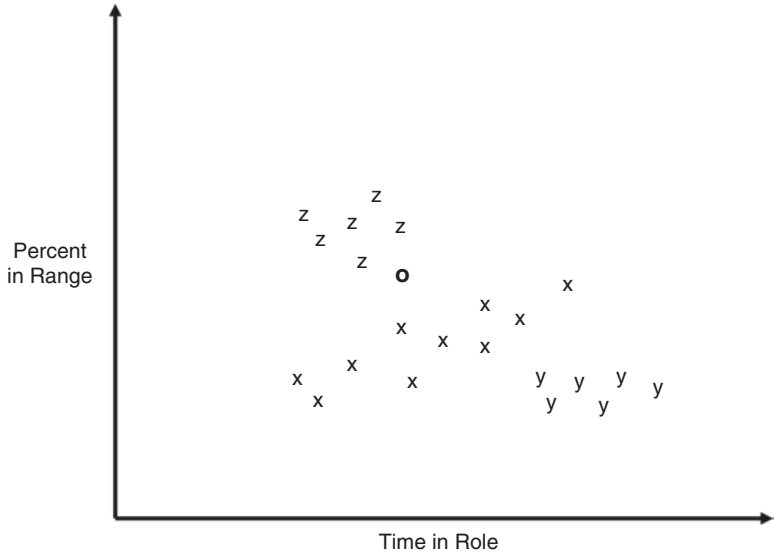
A final note to keep in mind when creating logistic regression equations is that they are not particularly flexible when attempting to predict complex relationships. Logistic regression is relatively simplistic in form with a single equation that drives all predictions. For complex relationships, it may overgeneralize and fail to produce reliable results. If you are modeling a complex relationship, there are other options. First, consider algorithms that can deal with complex relationships better, such as random forests. Second, you can create separate models designed for specific segments of your data. This breaks the complexity down by tuning for specific groups or parts of the relationship you are attempting to predict.

For example, during the research and tuning of your shrink model imagine you find that "shift schedule" significantly impacts which predictor variables matter: shrink for new hires on the morning shift is predicted by start-of-shift tardiness and absenteeism, whereas evening and overnight employee shrink is predicted by between-call gaps and return-from-break tardiness. In this case, you could try separating employees by shift and creating unique logistic regression models for each.

9.3 K-Nearest Neighbors (KNN)

Another popular form of classification is K-Nearest Neighbors (KNN). One of the simplest forms of classification, KNN is mostly used in academic settings. With KNN, an analyst predicts a class for a new data point based on its similarity to existing data points for which the algorithm already knows the class (i.e., based on what it learned from the training data). Calculating which class a data point belongs to is done by calculating the distance from the data point to its nearest "neighbors" (hence the name). When evaluating a new piece of data, KNN simply compares the data point to other data points with similar values on the predictor variables and then examines which classes (outcome variable) those data points belong to. Then, the algorithm simply uses majority voting to determine the classification of the new point. The "K" in K-Nearest Neighbors is the predefined number of points that the new point will be compared to.

To illustrate, let us say an analyst used KNN to create an algorithm to cluster people by Time in Role and Percent in Range. The training data helps determine that 3 clusters exist, *X*, *Y*, and *Z*:



Cluster *X* are average employees; their pay relative to their level (that is what percent in range typically measures) goes up as time goes on until they get promoted to the next pay band.

Cluster *Y* might be called "comp stuck." These are folks that have been in their pay band for a long time, but for whatever reason their pay is not increasing at the rate of their peers. This is often the result of the combination of being hired at low pay and not getting a band-changing promotion for a long time.

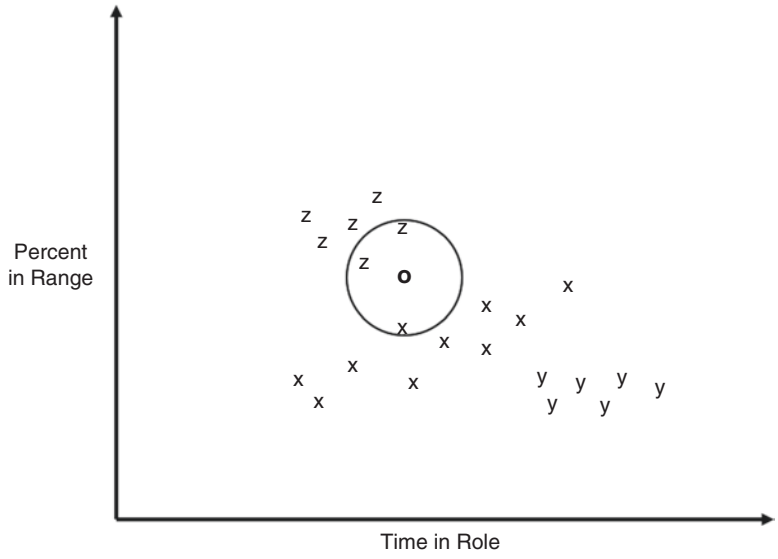
Cluster *Z* might be called "competitive talent." One might think this category would be called "high performers," but the truth is high performing employees typically have *low* PIR because they are getting promoted and changing pay bands so often. Outliers as seen in cluster *Z* are usually the product of hiring highly desirable people from outside the organization, which means the organization chose to pay a premium for them.

This type of analysis might be useful (1) if the analyst wanted to focus extra retention efforts on cluster *Z* (since the company has invested more in them as a group), (2) if the analyst wanted to make a case to adjust the compensation of cluster *Y*, (3) facilitate internal movement for cluster *Y*, or (4) investigate dissatisfaction with compensation in general. It might also be a useful exploratory research method for a workforce segmentation project.

Once trained on a percentage of the organization, this model could extrapolate classifications for the rest of the company, and then provide classifications for new employees and existing employees over time (like employees who drift from cluster *x* to cluster *y* or from cluster *z* to cluster *x*).

Whatever the case, let us illustrate how KNN helps an analyst "understand" new points in the dataset. For example, how would someone classify point "o" in the graph?

If the analyst sets K to three, the three points nearest to the new data point are used to determine the class of the new instance.



In this case, two of the points are "*z*" and one is "*x*," which means the new point would be classified as "*z*."

With KNN, the key parameter that must be defined is the number of data points that the new instance will be compared against to determine its classification. If you set $K = 1$, then outliers may have a larger impact on your model and the risk of misclassification goes up. When K is larger than one (like the example above), then the analyst is using multiple examples to classify new data.

With a two-class problem, an odd number is typically chosen for K to prevent ties in the voting. When there are three or more classes, there is a possibility of ties regardless of the size of K. Selecting an optimal K requires testing and analyzing the error rate of various Ks and choosing the one that performs best. The larger the K the longer the model will take to calculate and make predictions. As a general rule, remember that too small of a K will incorporate noise and let outliers or uncommon values weigh more heavily. Too large of a K will be less precise because it may consider too much information and have trouble making an accurate decision.

KNN is typically used for its simplicity and accuracy. It is also easy to implement and requires minimal programming. For most implementations, there is only one parameter to tweak, the size of K. KNN is one of the simplest machine learning models available for classification. Another benefit is that it is easy to explain to stakeholders how it works.

Also, KNN does not rely on or assume specific probability distributions in the data. When we talked about distributions in Chap. 6, we mentioned that some methods require data to be distributed in a certain way for them to work properly. KNN is not one of those techniques and does well with decision boundaries that are nonlinear.

KNN stretches the definition of machine learning a little since it is more memorization (or remembering) than learning. Most machine learning methods look for patterns and relationships in the data and produce a model that generalizes what it finds. This means that algorithms are trained to look for the relationships that exist between the predictors and the outcome, and then apply that logic to make a prediction. Said differently, many algorithms "learn" what predictors are important and then apply that "knowledge" to data it has not seen before to make predictions. This means the algorithm it generates can predict future data points *without looking back* at the data it was trained on.

KNN and other instance-based learning are more like an exercise in matching. When KNN gets a new piece of data it has not seen before, it looks back at all the data it has, and simply matches the new data into the patterns it sees in the old data. This is a key difference and can influence things ranging from higher maintenance needs to how much time and memory it takes to run new data through the model.

While KNN is a simple approach to classification problems, there are some considerations for its use:

1. *It can accidentally become too memory intensive.* In order to make predictions, KNN requires that the model remembers and *recalls* the training data. This means that KNN requires ongoing access to past data to make predictions. In a

model using large amounts of data, this can be memory intensive and a significant performance limitation. Algorithms like KNN that require access to training data to make predictions are called "lazy learning" because they wait until prediction is requested to do all the computational work! Most other classification algorithms employ what is called "eager learning," which means they find all the relationships and generalize during the training phase. Then the learning is applied to new data with minimal effort.

2. *Beware high dimensionality.* Another reason it can be memory intensive is high dimensionality. This is just another way of saying: "use too many inputs to try and predict an output." Since KNN uses distance, the more dimensions that are in the model, the less likely that a given point will be close to another, reducing model effectiveness. There are methods for dealing with this "curse of dimensionality," but are too technical for this text. Suffice it to say dimensionality reduction and feature selection are key considerations for improving a KNN model.
3. *Very reliant on input accuracy.* Because KNN does not generalize it is dependent on the accuracy of the local data. If the data is wrong or noisy, it will produce incorrect results. Models that generalize can smooth out many of these kinds of bumps to avoid these types of issues.
4. *Explainability is tough.* While it is easy to explain to a stakeholder how KNN works in general, inferring which predictors are the most important in a KNN model can be a challenge. Also, KNN works best when only those variables that are most influential on the outcome are included, so using large amounts of predictors to "explore" predictive power does not work well. Pruning the model for performance can provide indication into general variable importance, but ultimately decisions on why an individual instance was assigned to a particular class by the model cannot be clearly attributed to specific predictors. Since KNN is based on the concept of nearness, there are no specific inputs that can be used to explain any given decision other than that the data point "looks like" these other data points.

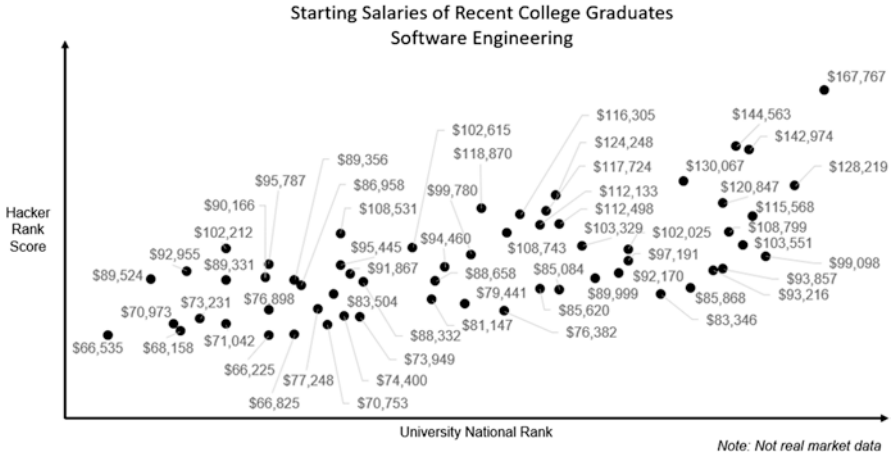
9.3.1 Two More Ways to Use KNN

KNN is a simple, popular, and flexible machine learning method. And as such, there are two additional ways it can be used: "weighted" KNN's and for solving regression problems.

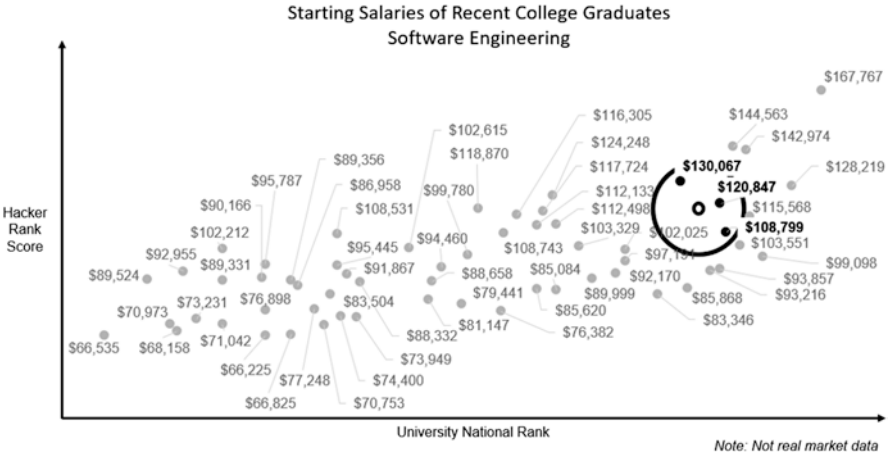
Weighted KNN means that instead of using a majority vote based on the value of K, more weight is given to data points which are closer. In some cases, this can significantly improve predictive power because it can do a better job of controlling for outliers.

Second, though KNN is primarily used for classification, it can also be used for regression. In classification, the classes of the nearest neighbors are counted and sometimes weighted, to determine the class of a new data point. With KNN for regression, the values of the outcome variable for each of the neighbors is averaged

to calculate the value of the new data point. For example, let us say a team wanted to predict the likely starting salary that a software engineer coming out of college would accept. They might want to do this to improve the offer-acceptance rate for their university relations team. The two main data points they might have to use as predictors are (a) how prestigious the university program is and (b) how highly the candidate is ranked by the Hacker Rank metric. The data might look something like this:



Using KNN for regression would take a new data point (where the team does not have a starting salary), and graph it based on the predictors. With K set to three, it might look something like this:



KNN for regression would then take an average of $130,067, $120,847, and $108,799 and predict that a candidate from this caliber school and this rating on Hacker Rank would accept a starting salary of $119,904.

KNN for regression maintains many of the advantages that KNN presents with classification. It is a simple algorithm that is easy to explain and understand. It does not generalize or look for an underlying relationship in the data, but instead uses memorization and comparison to known data points to make predictions. It does not require linearity in relationships and has no assumed distributions of the data.

And as with other methods of KNN, the key element of tuning KNN for regression is determining the proper K. Specifically for regression this can be tuned by using Mean Squared Error (MSE) instead of classification error metrics like accuracy or F1 score[2].

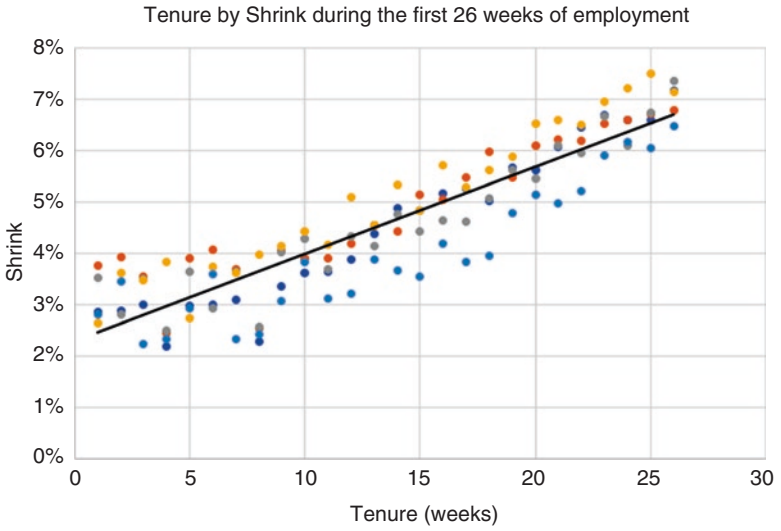
The disadvantages of KNN for regression are similar to classification. The model is memory intensive and slow as it requires access to prior data points to make new predictions. And like KNN for classification, KNN for regression also requires judicious feature selection and is at risk of the curse of high dimensionality. If too many variables are included, it can struggle to make accurate predictions. And like KNN for classification, KNN for regression can be weighted.

9.4 Support Vector Machines (SVMs)

Support vector machines (SVMs) is an algorithm that has been a popular alternative to traditional classification methods such as logistic regression and KNN. Offering higher accuracy and the ability to effectively model nonlinear relationships, SVMs experienced rapid growth after their utility got better in the 1990s thanks in large part to using kernel functions. More recently, more flexible and high performing algorithms such as tree-based models have taken much of the popularity SVMs once had, but they are still an important method to review.

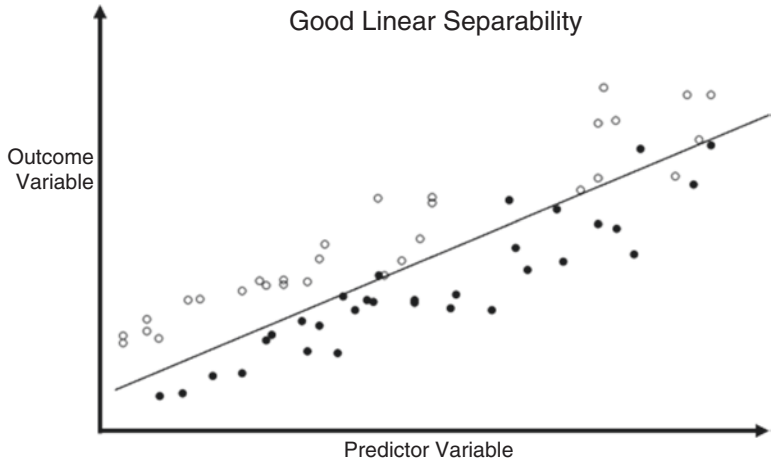
To explain support vector machines, let us go back to the idea of a regression line. Recall from the previous few sections that the predictor variables are used to draw a line on a graph:

[2] We will not be reviewing the skills and methods for tuning models in this book. We call it out here just to illustrate that tuning methodology may differ depending whether KNN is being used for classification or regression.



In the case of linear regression, the goal is to draw the line to be "best fit" to the data points. Which line can we draw which reduces the distance between its points and the training data points? Visually, it looks like it goes right through the middle of all the data.

Then, in logistic regression, instead of drawing a best fit line, the goal was to draw a line that "cut the data" so that it could be classified into groups, like so:

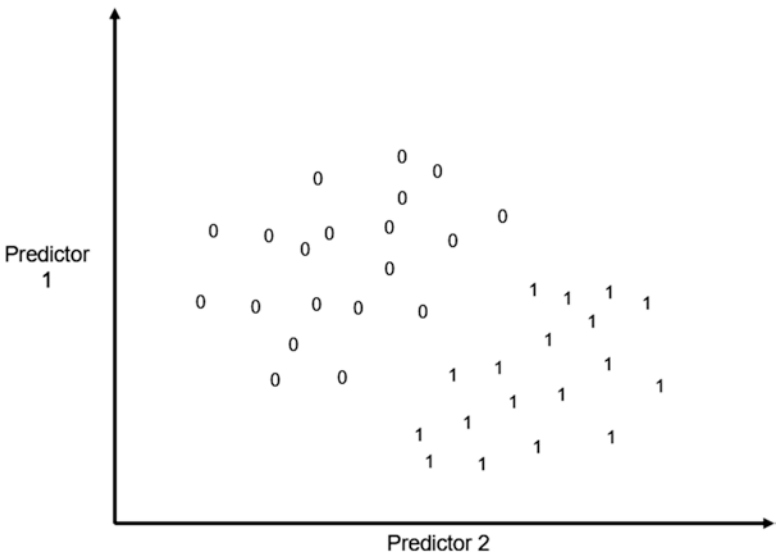


This is why linear separability helps logistic regression so much—the more linearly separable the data is, the easier it is for logistic regression to find the best equation.

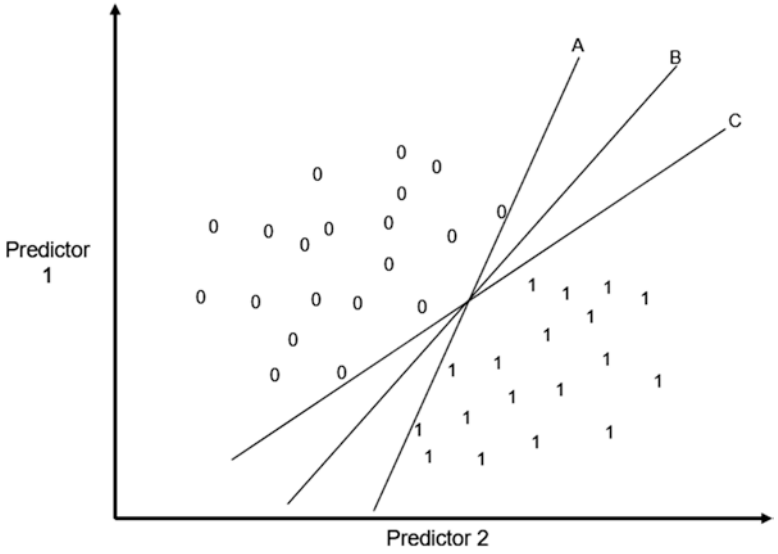
Support vector machines want to draw a line with the same purpose as logistic regression: separate the data into classes. However, SVMs focus on very different

characteristics in the data to accomplish the goal. They attempt to find what is called "decision boundaries." Decision boundaries are exactly what they sound like—the area where data points are close to the boundary between class 1 and class 0.

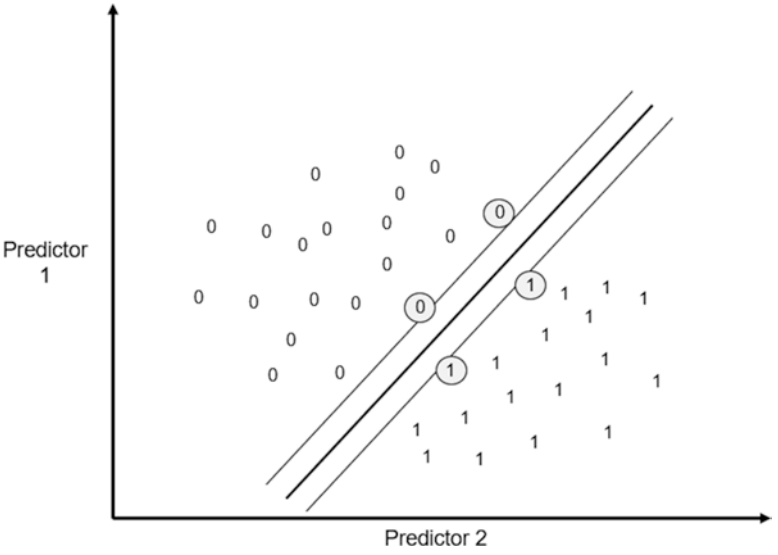
Another way to describe these data points would be: "Which class 1s look the most like class 0s and which class 0s look the most like class 1s?" SVMs focus on these data points which are very close to the boundary and use them to determine the optimal line to divide the classes. We will use an even more linearly separable dataset to illustrate:



If an analyst were to draw that line that you can picture in your mind, where would it fall? Where would the line which does the *best* job of separating the classes be? This is really important if the goal is to minimize misclassification of new data. Here are three lines which all make a good case to be the best line:



Lines A, B, and C all predict the training data perfectly (not to mention the infinite number of lines which exist between A and C), so how would an algorithm choose the best? This is where SVMs help—they examine the data points closest to the boundary between classes to determine the best line:



By looking at the 0s that are closest to class 1 and the 1s that are closest to class 0, an SVM tells the data scientist where to draw the best line.

The outside lines here, or the ones close to the outlying 0s and 1s, are called support vectors—these are the lines that produce the margin that exists between the classes. The goal of an SVM is to make that margin as big as possible. The bigger the margin, the more obvious it is that the classes are distinct and not overlapping.

But why call them "support" vectors? What do they "support" exactly? Support vectors support the middle line, which is called the "hyperplane." The hyperplane is the line in the middle of the margin and is ultimately the boundary between what gets called a 1 and what gets called a 0.

"Hyperplane" sounds like a science fiction term, but there is a good reason for the name. In geometry, a plane is a two-dimensional space that extends in all directions. To visualize the idea of a plane, think about how you would cut an apple in half; the knife travels down a plane that divides the apple.

Classification is just a mathematical way to do this slicing, and hyperplanes are how it is done. What makes a hyperplane different than a regular plane is that its dimensions correspond to the type of data it is slicing. On the previous page, we were able to graph the 0s and 1s in two dimensions (up/down and left/right). This means that the hyperplane only needed one dimension to do the slicing (lines are one dimension; they only have length, not width).

All linear relationships (and even some nonlinear relationships) follow this pattern—a straight line can cut them well. However, sometimes the math required to model data relationships cannot be done using two-dimensional equations. When this happens, the hyperplane required to slice the data must also increase its dimensions to keep up. Even though support vector machines were originally designed to model linear relationships, they have this ability to model these much more complicated relationships as well.

The way hyperplanes do this is through kernels, sometimes called a "kernel trick." We will not get into the details of how they work, but essentially kernels "translate" the hyperplane equation to higher dimensional space. We introduce them because kernels are a key parameter to set when using support vector machines. The standard kernel is the linear kernel, but as data relationships get more complex, there is a radial kernel, a polynomial kernel, and others. Choosing the correct kernel is a significantly important part of SVMs and is beyond the goals of this book. If you are interested in kernels, talk with your analytics team, or consult a more detailed text on SVMs.

A final big plus for SVMs is that in addition to being able to classify very complex relationships, SVMs are also not a big risk for overfitting, even when the model has a significant number of predictor variables.

Though SVMs have much utility, they do have some key drawbacks. First and second, they are memory intensive and can also be difficult to tune. There are many parameters that can have significant impact on model efficacy, but tuning a model by

testing various hyperparameters[3] like kernels can run for hours or days. It is processor intensive to search for the optimal decision boundary, especially with many features.

Third, SVMs struggle on large datasets due to the time to train, though this may be seen as a positive in HR since datasets are often smaller than average from a data science perspective.

Fourth, though overall variable importance can be obtained, the specifics of why a particular prediction was made is not directly available from the SVM model output. Additional tools and techniques such as SHAP can be used to determine the reasons why a particular decision was made by an SVM model; however, this adds an additional level of complexity.

Finally, SVMs predict classes but do not inherently provide probabilities for class assignment. There are some implementations where probability can be provided; however, it is not a true probability which means a confidence metric to help contextualize results is not easily available.

A far less common application of support vectors is Support Vector Regression (SVR). SVR is a form of regression based on the concepts of Support Vector Machines. As with SVMs, the algorithm starts by identifying support vectors that lie along the margin of the relationship between the independent and dependent variables. That margin is maximized to optimize the definition of the relationship. In classification, the margin is used to make predictions on class depending on which side of the hyperplane the new data point falls on. With regression, the resulting line is instead used to make numerical predictions.

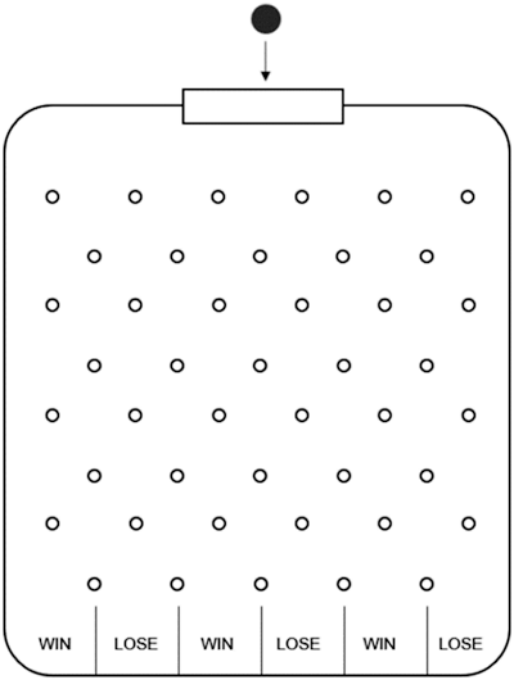
As with SVMs, SVR is kernel-based and supports multiple types of kernels (standard, radial, polynomial) that can be used to model nonlinear relationships.

9.5 Decision Trees

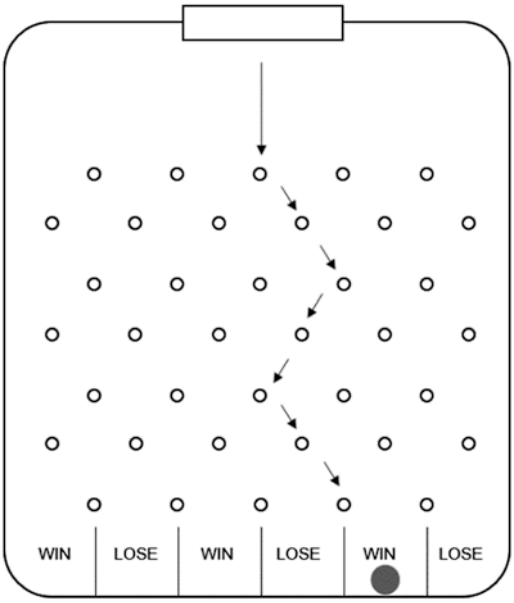
The next form of classification is decision trees. Decision trees use a branching method to create predictions. The result is a diagram with nodes and branches that can be followed to determine the model's prediction. Trees are made of nodes and leaves, with the root node at the top that is the start of the decision process. In this way, it might make more sense to call them "Decision *Upside-Down* Trees" because visually a decision tree gets more branches and leaves as it goes down, which is the opposite of an actual tree.

A great analogy for how decision trees work is a common gameshow and carnival game called "Plinko." In Plinko, a contestant drops a ball or a disc into a transparent covered box, and it bounces off pins until it gets to the bottom where it lands in a bucket or compartment that determines the contestant's prize. If you are not familiar with the game, it looks something like this:

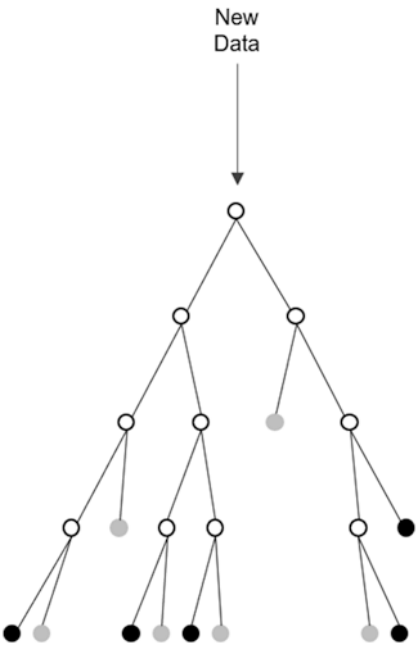
[3] "Parameters" are another term for the variables used to make predictions, whereas "hyperparameters" are the parts of the algorithm the researcher manually tweaks to help a model learn. We have discussed things like K in KNN and kernels here in SVM—these are examples of hyperparameters because we manually set them as part of the model design process, but they are not themselves predictors or outcome variables.



At each pin, the ball makes a choice: left or right. And the culmination of these choices lands the ball in a certain compartment. One path might look something like this:



In Plinko, the path of the ball is random and the contestant is hoping for a good outcome. In decision trees, each pin is an evaluation of data and based on the result, the data goes in a new direction until it reaches a compartment where a prediction is made. It might look something like this:



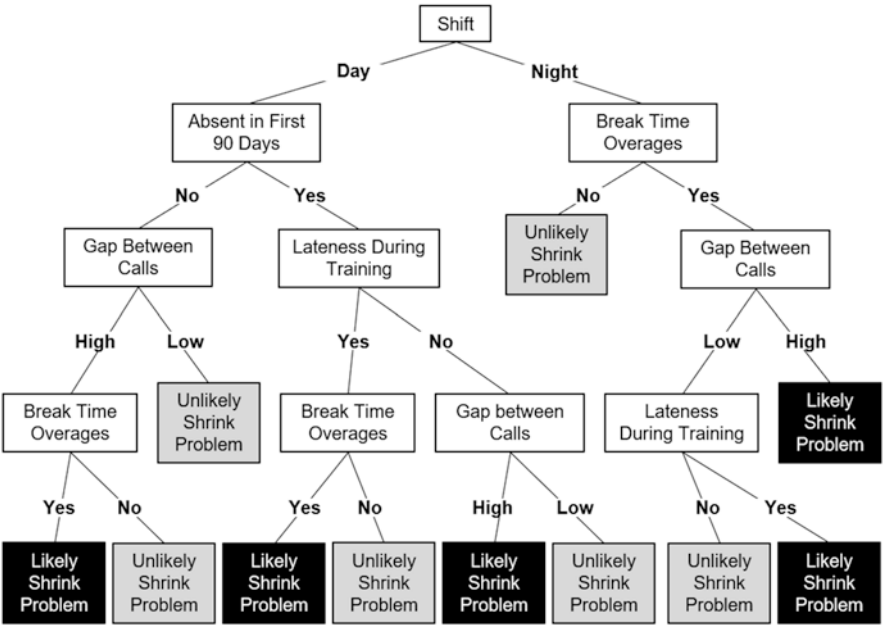
In this visual, a white node is where the data is being evaluated (go left or go right). The grey and black nodes are called terminal nodes, and once there the model has arrived at a prediction.

The nodes and thresholds for the nodes (what sends data left or right) are built with the training and validation data. Then, new data gets sent into the Plinko machine to see how it will be classified.

As mentioned, at each internal node on a decision tree there is a condition that is evaluated. A decision is made based on the value of a variable in the dataset. For example, if you were trying to classify who would need help with shrink, a decision tree model might discover a few important variables:

- Which shift
- Lateness during training days
- Absenteeism during first 3 months on the job
- Average gap time between calls
- Average break time (20 minutes allotted)

The model might decide this is the tree that makes the best predictions:



This tree follows the same pattern as the model on the previous page, it just uses the shrink example. If a decision tree was trained and found this pattern to be true, then you could take employees in the future who have completed their first 3 months of work and run their stats through this model. The results would then classify these new employees into grey or black buckets, predicting who will likely struggle with shrink in the future.

Given the visual nature of decision trees, they are incredibly easy to interpret. That is, practitioners can visually follow a decision tree to understand why it made a particular classification. In fact, the model itself can be communicated to individuals without having to explain and understand complex math.

The clear nature of the decision process in a decision tree extends to the evaluation of variable importance. It is possible to evaluate importance of variables and the elements that were involved in deriving a specific decision. In this way, they are on par with logistic regression in terms of model transparency. Further, with classification trees, you can also assess confidence levels or probability for the decisions.

The generation of a decision tree can be performed using a variety of different algorithms. The most common methods are ID3, C4.5, C5.0, and CART (Classification and Regression Trees). All methods ultimately generate a tree by splitting the data at each condition to maximize model performance. There are also different measures to evaluate the success of a split such as Gini index and cross entropy. However, a common criticism of decision trees is that they are myopic in generating splits since they do not take the whole problem into consideration. Instead, they maximize success for the current split which does not always lead to an optimal model. We suggest if you are interested in different decision tree

algorithms and optimization, you connect with your analytics team or a more advanced text.

Decision trees are often chosen due to their flexible data requirements. In particular, they work well with variables that have nonlinear relationships to the outcome and they do not assume a particular distribution for the data. Moreover, they are robust to outliers and also scale well. As the decision mechanism is simple, they perform well and can be used to quickly make predictions. There is no need to do feature selection with decision trees as it is done automatically. Variables that do not help predict the outcome are simply ignored.

An important drawback for decision trees is they do not do well with continuous variables. You may have thought in the example, "what do we mean by 'high' gap between calls? That seems like it would be a type of data well fit for a median or average." After all, in Chap. 5 we spent a lot of time talking about how to define things specifically, and these definitions seem ambiguous!

Decision trees branch on all variables based on thresholds, which means it is hard for them to see into the nuance of continuous data. They work better when the data has clear delineations as we see in data with a discreet set of options.

That said, part of the data exploration and wrangling process is the art of investigating the data for opportunities to do this. In Chap. 12, we will talk about binning, which is the technique of transforming continuous variables into ordinal variables with discrete categories. Doing this well can be a very powerful addition to the design and creation of decision trees.

Another disadvantage of decision trees is that they are at risk of overfitting. As a result of this, trees left to their own devices can grow very complex in their effort to minimize error. They can have duplicate leaves with different probabilities and other issues. And in addition to the risk of overfitting, the more complex the tree gets, the more difficult it is to read and interpret. Penalization is a good technique to prune trees and keep them manageable.

Finally, decision trees are very sensitive to changes in data. Retraining a model on slightly different data can result in a fundamentally different tree. This can be unnerving to stakeholders and those not familiar with the complexities of models like decision trees.

9.6 Random Forests

Many of the shortcomings in decision trees can be addressed through a set of related tree-based algorithms, the most famous of which is random forests. Random forests are intuitively named because essentially, they are just a group of trees! Random forests are a form of ensemble method, which means they are multiple machine learning models working together.

Originally developed in the mid-1990s, random forests are now one of the most common algorithms for classification. Based on a technique called bagging, results from multiple decision trees are combined to produce a more optimal result. More specifically, the algorithm selects a random amount of predictor variables and/or cases to create many unique trees (often hundreds of trees!). Each tree makes its

own prediction based on the subset of the variables and data it is given. The random forest algorithm then combines those results to produce the prediction.

In the shrink example, this would be like making 101 different Plinko boards and running the new data through all 101 of them. Whichever classification came out 51 times or more would be what the random forest recommends. This helps reduce variance and avoid overfitting because the noise in the overall dataset (see Chapter 8) is somewhat neutralized by triangulating predictions across many different subsets of cases and variables.

Random forests are also one of the easiest algorithms to use and configure. One element that makes random forests simple to use is that there are few hyperparameters to tune. The primary hyperparameter is the number of decision trees to generate. This can vary in size though, as noted above, hundreds of trees are not unusual.

Using random forests also eliminates the need to do extensive feature selection, as variables that are not relevant to the output will be ignored. In fact, random forest models are often created as an exploratory step to identify which variables are important to predicting an outcome. This is often done even if the final model will not be built using a random forest.

The random forests algorithm has many advantages in how it works with data. For example, variables do not need to be scaled to be useable by the model. It also works well with variables that are correlated with each other. Another interesting capability in some implementations (and other tree-based models) is the ability to handle categorical variables directly without one-hot encoding[4] the data first, which is more efficient and can offer a performance boost.

A key drawback to random forests relative to decision trees is that you lose transparency, which is often critical in HR. Though random forests are based on decision trees which are all individually transparent, they do not produce a final, composite decision tree that can be examined visually. There is no resulting diagram to inspect or analyze so the reasons for individual decisions are not easily available. That said, variable importance in a random forest model can be determined through measures like the Gini index and full decision explainability through the use of tools like SHAP. In general, the main takeaway for choosing a random forest versus a decision tree is that you lose the transparency but typically gain significant predictive power. You must weigh if that is acceptable within the confines of the business case.

9.7 Regression Trees and Forests

Both decision trees and random forests can be repurposed to solve regression problems. Regression trees are just decision trees which produce a numeric output and attempt to predict a dependent variable that is continuous.

As with decision trees for classification, they are transparent in how they arrived at the predicted value which makes them easy to visualize and explain. However,

[4] One-hot encoding is a method of transforming categorical data into a more binary format that is easier for many machine learning algorithms to understand. We will not get into the details of how it works here–consult a machine learning textbook for more details.

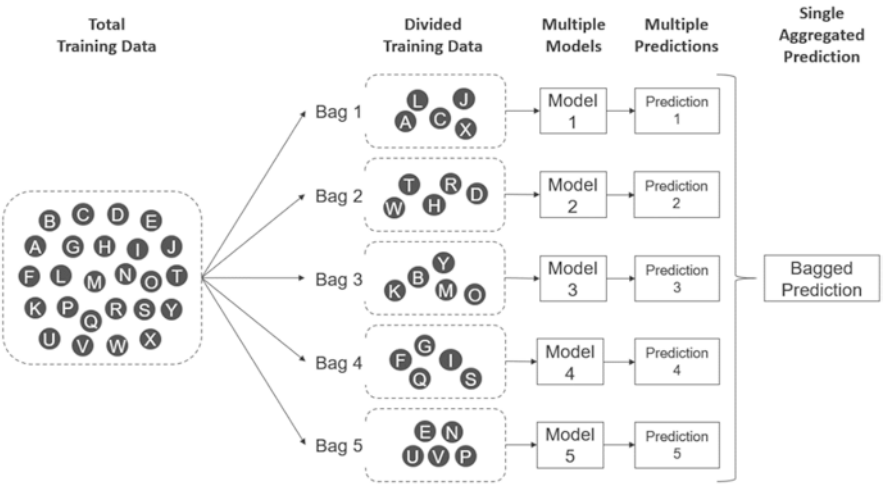
regression trees cannot produce continuous values like true regression methods such as linear regression. Each terminal node produces a static value, which lacks some of the nuances a true regression algorithm brings to the table.

Random forests for regression share the same benefits as for classification: limited feature selection is required, and predictive power is high. They also share the same major drawback: lack of transparency.

The final important callout with random forests for regression is that since they are an ensemble of decision trees which each have a numeric result, instead of using majority voting like in classification, random forests for regression take the average of the results of each decision tree in the ensemble to create the prediction.

9.8 Other Tree-Based Algorithms; Bagging and Boosting

Many modern tree-based algorithms use techniques called bagging and boosting to combine the results of multiple decision trees to produce more optimal results. Above we discussed random forests which use a variation on bagging–which is short for bootstrap aggregation. Bagging was developed in the mid 1990s and works by creating groups of samples from the training dataset (called "bags") and generating multiple models based on the bags. Much like the advantage of a random forest over a single decision tree, the benefit is that instead of one model, we can use the results of multiple models to generate a final prediction. This can help reduce overfitting and sometimes greatly increase predictive power:



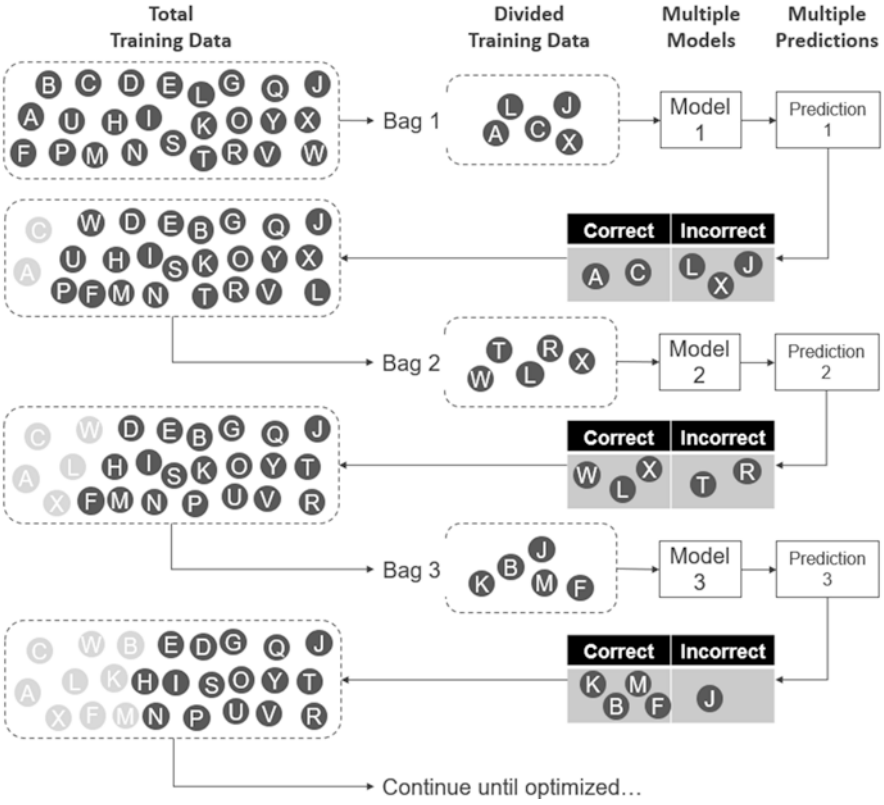
In this example, we "chopped up" the training dataset into five different training datasets[5] and built unique models according to each. The final prediction the model

[5] Note: The bags do not necessarily have to include mutually exclusive data points—they can have-overlapping points such that one data point can be evaluated by multiple models during training. We show it this way for illustrative purposes.

makes is based on the results of those five individual predictions. For classification this is determined through majority vote whereas regression takes the average of the predicted values.

Bagging in its natural form is powerful and effective; however, it has been largely superseded by variations on bagging such as random forests or related alternatives like boosting.

Boosting is a slightly advanced form of bagging that has proven highly effective. When we use bagging, we randomly create samples that drive the creation of multiple models. In boosting, the results of the first model are used to determine which data is sampled for the next model. Data that the model struggled to predict are more likely to be included in the next sample. The goal is to increase the performance of the model by focusing on getting better at predicting data it failed to predict correctly in the past. Think about it like this: Joe is a baseball player trying to become a better hitter. David pitched him 50 fastballs, 50 sliders, and 50 curveballs. He hit 49 fastballs, 45 sliders, and 20 curveballs. The next time around, what should David pitch to Joe if the goal is to make Joe a better hitter? Obviously, he has trouble hitting curveballs, so that makes sense to focus on. Boosting does this with model development—it progressively makes the sample more filled with data that the model had trouble with, so it can improve performance:



As you follow the arrows around the example above, you can see that each time we build a model with boosting, we use the results to help us choose the next bag. If the model guessed incorrectly, it is more likely that data point will be chosen again. This means later iterations of the model are progressively more focused on data points the model is having trouble with.

There are a wide range of algorithms that are based on boosting techniques including some of the leading algorithms used by professionals today. AdaBoost was one of the first and paved the way for boosting. XGBoost was a breakthrough implementation of boosting, is used extensively, and has helped realize the potential of boosting as a practical technique. More recent implementations of boosting that expand and increase its applicability include LightGBM and CatBoost.

With the rise of boosting, tree-based models have overtaken traditional algorithms such as SVMs, and logistic regression and are even preferred in many situations to more complex algorithms like neural networks.

9.9 Ensemble Methods

Bagging and boosting combine the results of multiple models to create a more optimal prediction. The idea that many models working together can predict better than one model alone actually falls under a general umbrella of techniques called ensemble methods. The approach that bagging and boosting algorithms take is to incorporate ensembling as part of the algorithm itself. The algorithm controls the models that can be produced and how to best combine the results.

There is another form of ensembling called stacking that allows you to combine results from multiple different models. This gives you the ability to merge the results from fundamentally different algorithms (like, for example, logistic regression and random forests) to produce a single prediction. This can produce significantly better performance than using the results of an individual model. Though using stacking can improve prediction performance, it does require running multiple machine learning models in parallel and then reconciling the results, which takes time and computational power.

Ensembling requires careful planning to determine which models to use and how to best combine the results. Often, a simple approach where each chosen model is weighted evenly is sufficient, however a nuanced approach where models are weighted differently can often be beneficial. For example, you could produce a model that gives priority to your random forest model over a corresponding logistic regression model by configuring the ensemble to give 75% of the decision-making power to the forest, and only 25% of the power to the logistic regression.

Like random forests, any approach like this combines results from multiple models, making it potentially difficult to understand why a particular prediction was made. Each model has its own decision-making process and degree of transparency which makes it challenging to provide clear explanations. There have been strides made recently to apply explainability to ensemble models; however, the results can be limited.

Given the performance benefits and flexibility that stacking provides, the technique has gained popularity and is now commonly employed by data scientists and built into many machine learning tools and platforms. As with earlier algorithms, prediction power should be weighed against other factors like explainability and computational requirements when choosing whether to use an ensemble method.

9.10 Supervised Neural Networks

The final type of supervised method to review is the neural network. Neural networks have been mentioned a few times in previous chapters and are a commonly used machine learning algorithm across the industry of applied machine learning (though they are not often used in HR). These types of algorithms are great for use on audio, text, and image data, and have made their way into even more advanced applications recently. Neural networks also famously form the basis for deep learning which we discussed briefly in Chap. 8.

The concept of neural networks has been around for years, but there are several things that have led to a recent surge in interest and application. First, the growth in available data. Neural networks require large amounts of data and do better with more data. Second, neural networks require significant processing power. When we talked about computing in Chap. 7, we referenced how our ability to help computers "go fast" has increased exponentially since the 60s and 70s, and especially in the last 10–15 years—this has very much enabled the growth of all computationally intensive innovations. Third, more data and computing power has led to better tools for data scientists to use to enhance neural network capability. Finally, all this advancement and success has led to many previously unsolvable problems to be solved, which has made the methodology somewhat recognizable in the popular media, like Watson on Jeopardy.

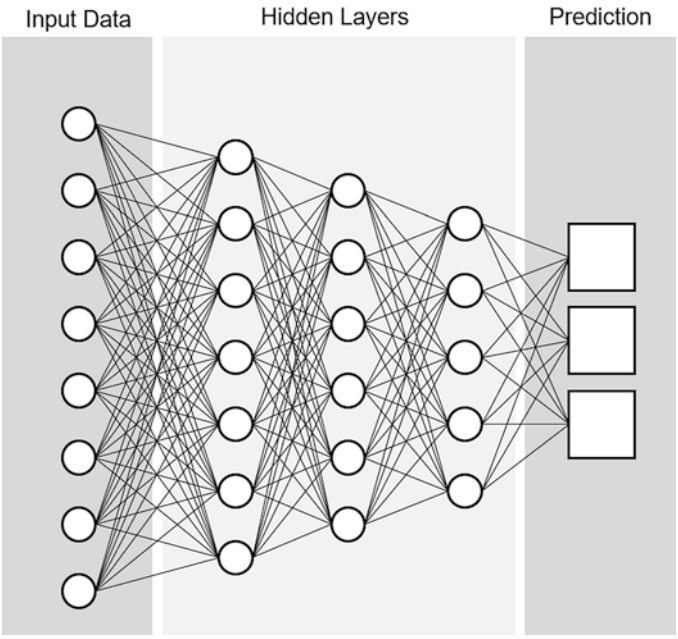
But what are neural networks? The name sounds very sophisticated, and in many ways they are. Neural networks are based on the basic ideas of how brains work, which is one of the most complicated systems known to mankind.

Brains are made of neurons, about 86 billion of them for any given human. These cells are all connected to their neighbors, creating a vast network with literally trillions of connections which regulate everything from breathing to emotions to you being able to understand this sentence. To vastly oversimplify, your brain cells work by activating their neighbors and convincing them to fire. Have you ever been in a house full of dogs when one starts to bark? All the others want to get in on the action, even if they do not exactly know what they are barking about! With enough noise, they may even get the neighbor's dogs barking and cause a larger chain reaction. Essentially, groups of neurons firing together and in specific patterns according to the stimuli they receive is the basis for how brains work. Neural networks in machine learning take that same principle:

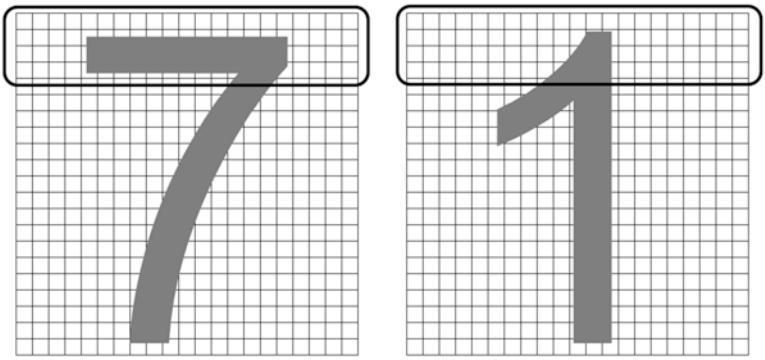
Enter Stimuli > Activate Nodes > Produce Output

…and use it to make predictions. There are many variations on the types of neural networks and how the layers connect, like convolutional neural nets which are designed for character recognition, but the general concept is the same.

A basic map of a neural network might look something like this:



The input data for a neural network are the predictor variables which the data scientist feeds it. In fields like image recognition, these could be data about the color of pixels in a grid. The patterns of pixels then begin the chain reaction of the hidden layers, which are the most important part of a neural network. If the data scientist is trying to differentiate a number "7" from a number "1," the algorithm will notice the pixels along the top of the image more for a '7' than for a '1'.



If you think of each pixel as its own predictor variable (with a value ranging from totally empty at 0% to totally full at 100%), then you can see how the pattern across the top four rows would be much more associated with numbers like "7" and "5" and far less with numbers like "1" and "4." This means the nodes which correspond with those pixels will "fire" much more strongly and lead us to a "7" rather than a "1."

During training, these hidden layers assign weights to each of those connection points and depending on how activated they are, they fire strongly, weakly, or not at all, causing the next node in line to fire strongly, weakly, or not at all and so on. In some ways, this is like the Plinko game we played with decision trees, except now we are not looking for a left-or-right threshold—there can be much more nuanced relationships. This is metaphorically like throwing multiple balls into the Plinko machine at once because the nodes light up in many places simultaneously to arrive at a judgment.

When the input has fired all the way to the prediction, that is called forward propagation. However, the algorithm does not always guess right the first time, which means it might have to go backward, so it can adjust the weights for what causes the nodes to fire and try again. This is called backpropagation and is how the network refines its predictions to reduce errors (i.e., "learn").

This is the basic idea of how to train a neural net. Create layers, feed in the input, and let the machine run through huge amounts of possible connections and weights to create the optimal network for prediction.

Tuning a neural network is mostly about choosing the type of network how many hidden layers are defined and the number of nodes in the hidden layers. The input layer node count is based on the number of predictor variables while the output node count is based on the number of classes to predict.

Given enough data, neural networks are high performing and often outperform all other machine learning algorithms. Neural networks do not require linear relationships, nor do they assume any particular distribution of data so in this regard they are very flexible. With classification, neural nets provide probabilities of membership in each class. Additionally, neural networks can support multinomial classification (more than two possible class labels) with relative ease whereas other algorithms often do not. They can even be used for regression problems!

So why haven't neural networks solved all the machine learning problems? While neural networks can create powerful classification and regression algorithms, there are a few significant limitations which are specifically important in Human Resources. First, they lack *significantly* in transparency. Out of the box, many neural net libraries do not even provide variable importance. Because of the lack of transparency, it is often challenging to understand what is driving a model. This might not matter when trying to automatically tell the difference between a 7 and a 1, but it does matter if an organization is making an employment decision and must ensure the decision is ethically and legally defensible. Neural nets are often referred to as "black-box" algorithms for this reason and is an element of neural nets that can keep them from being a useful methodology in a lot of predictive spaces. There is work in this area to develop neural network algorithms and inspection techniques

that will shed light on the factors driving a model, but in today's world neural nets sacrifice transparency for performance.

The second major reason neural networks are not used much in HR is that they require a large amount of data to train relative to what HR usually has available. Datasets with hundreds of thousands or millions of rows are typically ideal. Even the largest companies in the world simply do not have the data size on their employees needed to do neural network training. A company in the global Fortune 100 with 15,000,000 customers may very effectively be able to use neural networks to help their consumer insights or marketing teams, but that same company will likely only employ 100,000 to 200,000 employees. Some of the largest companies in the world, like Walmart or McDonalds, make it into the low millions due to enormous frontline populations, but even their turnover and job diversity reduce analyzable populations into numbers which are too small for traditional neural networks. This is why machine learning techniques which can predict with smaller datasets are typically preferred in HR[6].

Two other reasons (not as related to HR) which can make neural nets challenging are tuning and memory usage. There are no hard and fast rules on how many hidden layers and nodes should be used and there are many different types of structures of neural networks designed for different problems. For example, as we mentioned earlier convolutional neural networks (CNNs) are designed for character recognition, while long short-term memory (LSTMs) are designed for time series problems. Neural networks also have a large number of hyperparameters that require tuning. Basically, although simplicity and usability are improving rapidly in this space, for someone new to machine learning it is still best to consider the setup of a neural network to require significant expertise.

Finally, neural networks are processor and memory intensive and can take a significant amount of time to train (sometimes hours, days, or even months!). The very structure that makes neural networks so powerful—namely lots of nodes with lots of connections examining very large amounts of training data—is also a substantial limiting factor in its usability. However, though the feasibility can be challenging, depending on the problem it is often worth the wait.

So even though they have been gaining notoriety as the end-all-be-all of machine learning, neural networks are not currently a very useful tool for most problems in HR. They are a very powerful, but very specialized, method which is best suited for problems with extensive data, tolerance for opacity, lots of development time to train properly, and an expert data scientist at the helm.

[6] One exception to this rule in HR is in the space of data quality auditing. Large companies often process hundreds of thousands or millions of transactions a year and neural nets can be helpful to aid in the process of ensuring data integrity.

9.11 Unsupervised Learning and Reinforcement Learning

All the methods we have talked about so far have fallen into the category of supervised learning. Recall from Chap. 8 that supervised learning is where the analyst knows what success looks like. They have input data and are unsure how it predicts the outcome, but they do know what that outcome is and what it looks like. The models in supervised learning essentially figure out how the inputs predict the output.

However, there are two other major categories of machine learning which do not follow this philosophy: Reinforcement Learning and Unsupervised Learning. Reinforcement learning seeks to make a series of decisions to maximize a cumulative reward. Instead of predicting a single future action like supervised learning, reinforcement learning is based on the concept of exploring and learning an environment, and then adjusting future behaviors based on the results. This is where much of the machine-learning-bordering-on-AI comes from and begins to get close to the concepts used for applications like modeling general intelligence. To illustrate, think about games. If you were learning to play a game, you would want to know two main things: (1) what are the rules and (2) how do I win? Then, you would behave within the boundaries of the rules and adjust your strategy based on feedback to perform better. For example, if you were playing tennis and realized your opponent has a very strong backhand, you might avoid hitting it to that side. But how did you learn that? You learned because every time you hit it to that side, you lost the point. That caused you to change your behavior.

Reinforcement learning uses these ideas of exploration and exploitation in a machine learning environment. The model proposes an action, and then is either reinforced or punished with respect to their ultimate goal. This causes them to update the model and try again. This loop between "action," "reward," "update" is the essence of reinforcement learning.

Unsupervised learning is a form of exploratory data analysis in which the researcher does not necessarily know what success looks like as they do in supervised and reinforcement learning. This seems like a stretch because if a researcher is trying to *predict* something, they need to know what they are trying to predict. Without a specific label or category to anchor the exploration to, how can an algorithm analyze the data and make useful observations?

Unsupervised learning does just this and has some potential applications in HR. We would like to review two types of unsupervised learning: clustering and latent variable models.

9.12 K-Means and Hierarchical Clustering

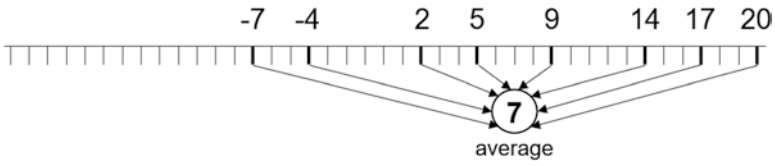
In Chap. 8, we talked about clustering as one of the key types of unsupervised learning because it helps us create groups. Discovering groups which are similar to each other, but distinct from other groups is of great interest to HR because as more data

can be collected, HR professionals are more able to use that data to curate and customize the employee experience.

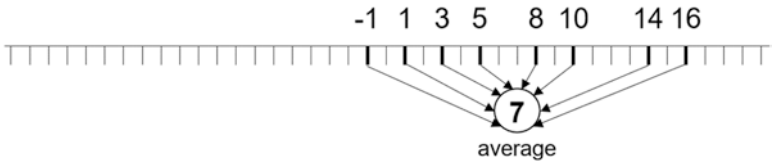
Understanding the similarities and differences in workforces and work environments allows practitioners to "group them," at which point they can better understand employees and organizational cultures. As an example, a research and development department filled with mid-20s, single software developers might want very different things from their employee experience than the finance department of the same company which might be comprised of mostly 35+, home-owning employees with spouses and children. When HR strategists can accurately find these general commonalities, they can improve the employee value proposition. And in today's talent market this equates to competitive advantage.

Two main types of cluster analyses common in machine learning are K-means cluster analysis and hierarchical cluster analysis. K-Means is the most common form of clustering and uses the distance between data points to determine similarity and create clusters. To explain this a bit further, we will return to Chap. 5's discussion of variance and standard deviation.

Remember that variance in a dataset is measured by looking at how data points are situated relative to the average:



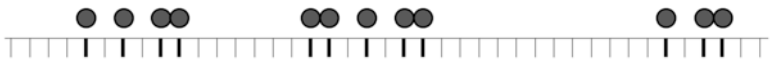
This dataset of -7, -4, 2, 5, 9, 14, 17, and 20 has a variance of 95.4 and a standard deviation of 9.8.



This dataset of -1, 1, 3, 5, 8, 10, 14, and 16 has a variance of 37.1 and a standard deviation of 6.1, even though the average is still 7.

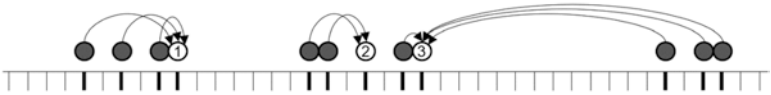
A good way to visualize variance is the cumulative length of the arrows. In the first example, if you laid all the arrows end-to-end, the overall length would be much longer than if you did the same exercise with the second example because of how far the numbers are away from the average.

K-means clustering takes this concept and uses it to find clusters of data points. Let us take another example, this time without numbers:



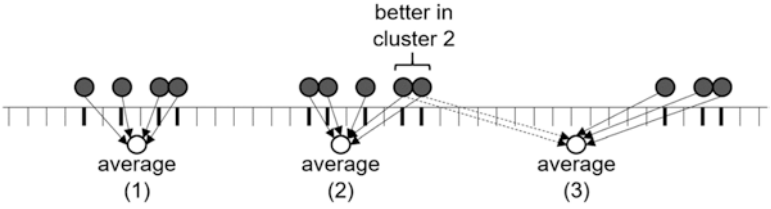
It is easy for the human eye to pick up that there should be three clusters here. But a computer cannot "see" that, and so k-means clustering does it by picking random data points to be the "center" of a cluster. But how many data points should it pick? Just like in K-Nearest Neighbors, "K" represents the number that the researcher chooses. But in clustering, the number K now denotes how many clusters the data scientist wants the algorithm to create. We will choose 3. The algorithm first randomly picks 3 data points:

The next step is for the algorithm to assign all other data points to one of the three clusters based on how close they are to one of the three center points. This is similar to our variance examples:



Once all the data points are assigned to a cluster, the algorithm uses the average of each cluster to check and make sure no data points need to be reassigned.

If any data points need to be reassigned, then the average is recalculated and the cycle repeats until no more data points need reassigning.



The final step is to repeat this whole process many many times using different random starting points. This allows the algorithm to look at all the cluster possibilities and choose the model which creates the smallest overall variance. Ultimately this is just a mathematical way to translate the idea that each data point is similar to the data points in its cluster, but different from the data points in the other clusters.

This example using one variable is easy to visualize on a number line, but as the data has more and more predictors, simply think about the "distance" being calculated with more and more dimensions. Two predictors would mean the data could be represented on an *x*- and *y*-axis graph. Three predictors would use a three-dimensional graph (*x*, *y*, and *z*). Anything above that is not visualizable but can still be represented with equations.

It is important to remember that the results of K-Means clustering are not predictive. Whereas regressions and other forms of supervised learning build an algorithm into which the analyst can insert new data and predict an unobserved value, K-means

cluster outputs are descriptive because they are describing data as a function of how they are similar and different from each other. This can be slightly stretched into the world of prediction because once new data becomes available, the model's thresholds can be used to classify new data, although this work then begins to look more and more like K-Nearest Neighbors (KNN) or SVMs.

Despite its less-than-predictive nature, the output of K-Means can lead to a better understanding of inherent relationships or patterns in data. This can be used by the business to institute change or it can serve as input into subsequent supervised learning models. For example, K-Means clustering could be used to identify key employee segments by grouping employees with similar attributes. The HR team could then analyze those clusters to create employee profiles and then design differentiated and targeted developmental or engagement strategies for each group.

Also, in machine learning projects clustering is often an initial step of exploration, followed by more targeted analysis. The first step is to determine if the groups make sense and represent some underlying relationship, which involves exploration and partnership with subject matter experts and stakeholders.

As mentioned earlier, the number of clusters has to be manually specified which means the ideal number of clusters is not necessarily known. Further, there is no way to validate the "correct number" of clusters. Instead, the model has to be tuned to determine the best number of clusters. This typically involves looking at cluster stability (i.e., do cluster assignments change radically when new data is introduced). Another method is to have the business or other subject matter experts validate the cluster assignments.

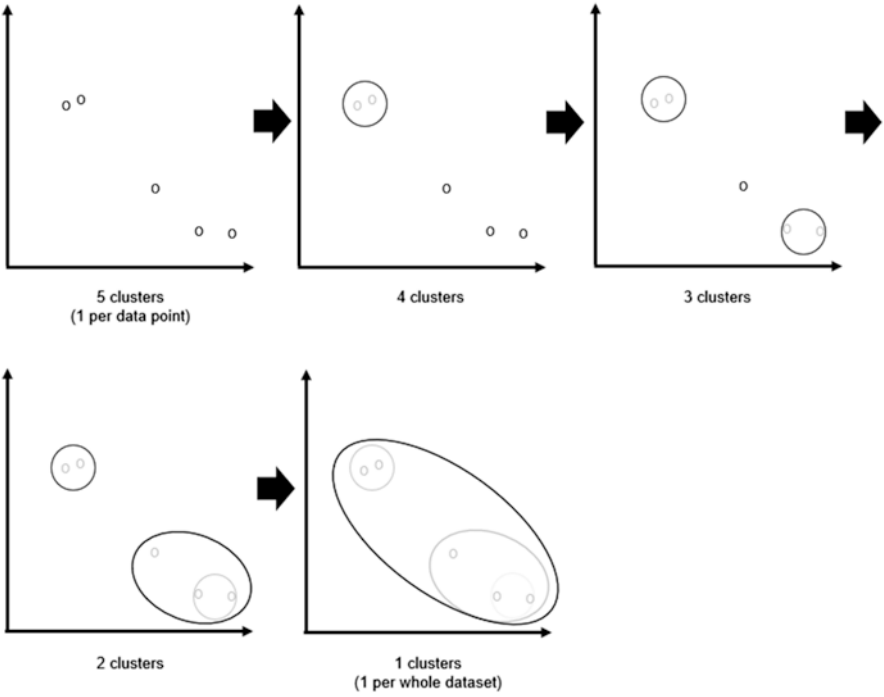
A challenge with K-Means is that because the data scientist is randomly selecting center points to generate clusters, there is a random element to how the algorithm works which can result in different clusters each time it is run. This can be stabilized using some more advanced methods; however, new data will often disrupt the model.

Another limitation of K-Means is that it works well only if the clusters are best represented using many different center points. That is, K-Means somewhat assumes that all clusters should be of about the same size and shape. However, if the clusters are best represented in different sizes or asymmetrical shapes, K-Means may fail to properly identify those underlying relationships. This can limit the effectiveness of K-Means. There are clustering algorithms that are designed to deal with odd-shaped, varying sized clusters.

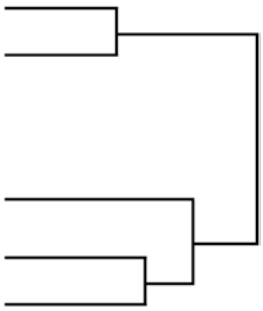
Finally, given that K-Means is based on the distances between points, like with KNN, feature selection is important for the effectiveness of the model. If too many features are included, it can muddy the clusters.

The second type of clustering is hierarchical clustering. Also, known as Hierarchical Cluster Analysis (HCA), this type of clustering uses the strategy of progressively reducing the number of clusters by looking for commonalities across observations. Instead of randomly choosing a starting point like in K-means, HCA starts with the most similar data and then works its way toward dissimilarity. It begins by looking at just the two observations which are the most similar across all predictor variables and groups those observations together in one cluster. Then,

HCA starts the process over, treating the new cluster of two as one observation. The process repeats until all the data points have been grouped into 1 cluster. Here is an example of 5 observations using 2 predictor variables (graphed on *x* and *y*):



Once this process is completed, hierarchical clustering creates a visual called a dendrogram which shows the progression from one-cluster-per-data point (left) to one-cluster-per-dataset (right):



Each branch and combination show where the clusters were formed and the length of the branches shows how much similarity exists within the cluster. It is up

to the researcher to decide where to set the threshold for how many clusters are appropriate based on the similarity they can see in the results.

Also note, the above is an example of "bottom-up" hierarchical clustering. HCA can also be done "top-down," which is essentially the opposite—the data starts with one big cluster and progressively "chops data off" until it has become one cluster per data point.

There are a few important considerations for HCA. First, we will not get into the details of "linkage," but it is a critical aspect of HCA. Linkage is the guidance the data scientist sets on how to define the comparison points for a cluster. Another way to say this is "how do you define similarity between clusters?" The researcher may want to compare other data points to a cluster's center, to its nearest neighbor, its farthest neighbor, or another approach. Linkage methodology is an important choice to make when setting up an HCA.

Second, HCA does not handle very big data as well as K-means does. Due to the nature of its math, bigger datasets can be a challenge.

Third, HCA results are far more reproducible than K-means. We mentioned earlier that retraining or slightly changing the data in K-means can result in a fundamentally different set of clusters, which can be challenging and confusing for stakeholders. HCA's methodology is quite a bit more stable, so this variability is not as much of an issue.

Fourth, HCA does not require you to know how many clusters should be in the data at the beginning of your exploration. Remember with K-means the data scientist needs to set a value for K, and the algorithm will then create that many clusters. This is an advantage if they are less familiar with the data.

Finally, HCA is better suited for differently shaped clusters. Since K-means uses the averages of clusters based on a random *center*, it is better suited for clusters which have density and symmetry; terms like "hyper-spherical" and "globular" are adjectives used when investigating cluster shape. Since HCA deals with similarity between individual points first rather than centers and averages, it works better for asymmetrical clusters. These sorts of details and ideas are best discussed with data science or analytics teams when choosing a method.

9.13 Latent Variable Models

The final category of unsupervised learning we would like to briefly touch on is latent variable models. In the next chapter, we will discuss what we call the "Construct Chasm" which demonstrates the real-world problem which latent variable models attempt to solve, but here would like to briefly introduce the techniques used to combat it with statistics and machine learning.

In the most general sense, a latent variable model is any statistical model which attempts to create and define an *unobservable* variable by triangulating many *observable* variables.

Behavior, potential, emotions, and many other concepts HR professionals want to quantify fall into the category of "latent" or "not-directly-observable" phenomena. When this is the case, the researcher must use things they *can* observe and hope they can model what they cannot observe. For example, nobody can directly measure "employee engagement," but if a survey asks someone to tell it how they feel about their manager, the direction of the company, how much time they spend putting in extra effort, and similar items, a researcher may be able to use those measurable opinions to uncover one latent variable called "employee engagement."

Latent variable models are the statistical processes used to mathematically analyze this. We will not get into any specific models, but as we mentioned earlier, here are some common latent variable models practitioners may come across:

- Factor Analysis
- Item Response Theory (IRT)
- Latent Profile Analysis
- Latent Class Analysis
- Principle Component Analysis
- Method of Moments
- Expectation-Maximization Algorithm

These methods are on the complex side and are best approached with a statistics expert close by. Also, they are less in the realm of machine learning and more in the realm of traditional statistics because they are fundamentally aimed at quantifying relationships between variables. That said, latent variables are a critical phenomenon in the world of HR data and will aid an analytics team in their design and development of data which is useful for advanced analytics and machine learning.

So far we have reviewed the basic underlying research methods and statistics principles needed to interact with advanced analytics, discussed how the evolution of computing has enabled the rapid advancement of analytics, and introduced machine learning as a natural outcome of these various historical achievements. We then reviewed some of the terms and techniques in use today which can help practitioners and organizations get value from this new force of industry.

Essentially, this book so far has talked about *how* machine learning developed and *what* machine learning is. In Part III the goal is to *help you get started in HR*. We will review how HR and people science is different from other fields where machine learning has had an impact, as well as use some examples from history to demonstrate how we can ensure we do it well. Then, we will talk about executing machine learning projects within an HR function, with HR stakeholders, and within the parameters and realities of corporate human capital decision-making.

Discussion Questions

1. Choose three supervised methods.
    - Explain in general how they work.
    - Create a detailed example of how you might use each in an applied setting.
2. Choose two unsupervised methods.
    - Explain in general how they work.
    - Create a detailed example of how you might use each in an applied setting.

Part III
Getting Started with Machine Learning

Chapter 10
What History Can Teach us About Using Machine Learning Well

The advent of machine learning will enable us to take care of employees better, which is not only key to the sustainability of healthy corporate culture but has been shown to create stronger, more profitable enterprises. To begin this journey, we must set some context around where and why HR practitioners seek to apply machine learning. This is because simply knowing how to use a tool is not enough. To this end, history is a wonderful teacher because there are many examples in history where humans have allowed the potential benefits of an innovation or technology to overshadow the risks. As HR practitioners, we are entrusted with the well-being of employees, and therefore it is not enough to simply know how to use machine learning but to seek to use machine learning *well*.

Indeed, companies that invest in their people reap huge benefits in the long run and are often held in high regard as the best places to work. Having a reputation as a desirable organization to work for is one of the most valuable assets a company can possess, especially in a day and age where talent is hard to find and even harder to keep. As organizations start to take this concept seriously, they realize this is far more than competitive pay, free lunch, and a great holiday party. Especially as companies grow (or are already large), the needs of people vary widely. Different cultures, life stages, physical locations, job types, and many other factors make "investing in your people" a complex and nuanced concept. Mary, who is the mother of two young children and works at a retail location has a very different opinion about what she needs from her organization than Adriana, the empty-nester and director of finance at headquarters. And while there are a great many ways to attack all the intricacies of these issues, it is far from objective science.

Machine learning, and advanced analytics in general, poses an attractive and even intoxicating solution: "If we can just get enough data" or "the right data," then we can solve all the ills of the employee experience. This leads to rapid solutioning. When business leaders and data scientists talk about machine learning with HR data, the conversation typically turns tactical quickly. What processes can be automated? What outcomes can be predicted? How can return on investment be

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_10

measured most effectively? And while these are absolutely the right end states to get to, the unglamorous, time consuming, and often overlooked parts of quantifying and predicting behavior must be considered first.

In order to design for that kind of value, the practitioner must first understand behavioral data. And to understand behavioral data, the practitioner must understand behavior and how it translates into data. Essentially, the tricky part about behavioral data is that it is almost always an approximation of what is being measured. There are some easy ones, like turnover: either they work for the company or they do not. But *predicting* turnover is a different story. For that, the practitioner has to measure things like disengagement, dissatisfaction, work-life balance, and others, all of which are much more complex to quantify.

Fortunately, science has been thinking about and studying human behavior for over 250 years and more recently, they have been specifically studying people at work. This is where psychology offers a lot of insight into the new world of machine learning with employee data. When we consider what we have been able to learn through psychological science and theories, we gain a lot of important perspectives about the nature of measuring behavior and using it to make predictions and decisions.

It might seem strange to delve into the history of psychology and industry in a book about machine learning. However, there are important thematic nuggets throughout the history of the development of psychology, and a few from other industries which have helped science learn about how people work, about what sorts of applications of behavioral science are effective, and how their pursuits need to be considered in order to remain both effective and ethical when beginning your journey down the path of machine learning. These tenets are critical because they will accelerate your ability to effectively wield advanced analytics in your organization as well as ensure that the analytics you use are providing you with insights which are valid and useful. After all, the ultimate purpose of using machine learning with employee data is to successfully predict behavioral outcomes: Who will succeed if we promote them? Who is going to quit? How much return will we get in performance if we implement this new training? Competitive advantage is the name of the game. Introducing quantified behavior through the lens of its academic and commercially applied development will illuminate its nature which is what ultimately underpins the accuracy of your data and insights. These ideas and lessons are the foundation on which you want to build competitive advantage.

10.1 Lessons from the Early Science

Like all sciences, psychology has its deepest roots in philosophy. Dating as far back as Aristotle, philosophers used logic, reason, and reflection to make inferences about "human nature" or "the nature of the soul." And since "nature," at its root is describing character or essence, we know that this is what those ancient philosophers were trying to understand: what makes humans act the way they do? Why do

some choose apples and others, oranges? Why are some philanthropic and others greedy? Why do some prefer risk for big reward and others less reward with more guaranteed security?

This philosophy was only philosophy for thousands of years until the science of psychology was undertaken in Germany in the 1830s. Originally framed as the exploration of how humans perceive physical stimuli, a line of German scientists launched the field of experimental psychology, followed shortly by experimental psychiatry. The most famous scientists of the day, men like Gustav Fechner and Hermann von Helmholtz dedicated their labs to psychophysics or the study of sensory perception. Psychology as a predictor of behavior was figuratively born from the scientific pursuit to understand how humans take in stimuli from their five senses and translate it into perception.

This quickly grew into more advanced experimental psychology and psychiatry, when scientists like Wilhelm Wundt built on the scientific advances of harder sciences like chemistry which were focused at the time on the structure of material. Following the paradigm of the day that everything can be broken down into its smallest components, these scientists focused on reducing human mental processes down into their simplest and most common attributes.

By the late 1800s, psychology had spread beyond Germany to much of Europe, as well as across the Atlantic to the Americas. Universities like Cornell in the United States had begun their own psychology programs which gave rise to the expansion of theories like Structuralism and Functionalism, which sought to explain more advanced mental processes than the biological endeavors of psychophysics. In parallel, institutions like the University of Buenos Aires in Argentina kept this more biological and experimental approach to psychological research.

Russia followed a similar path of biological exploration. We can see the marks of this era today in behaviorist and cognitive theories, each of which brought unique and important insight into our modern understanding of behavior. Some examples you might be familiar with are:

- Ivan Pavlov and his salivating dogs, known now as “conditioned reflexes” (circa 1900)
- B. F. Skinner and his pigeons, whose research taught us why gambling is addictive and how to treat other maladaptive learned behaviors (circa 1955)
- Jean Piaget and his theory of childhood development which gave us concepts like object permanence, which is why peek-a-boo is fun for babies, but not 5-year olds (circa 1960)

The effects of the work of these early scientists are far-reaching, but specific to the mission of this book, this era marks the first time humans *began attempting to quantify behavior and reverse engineer it into an objective model of the human mind and its decision-making processes*. This is also where history sees the first very important difference between behavioral data and other kinds of data. Most outcomes people want to understand are the products of systems similar to the computers we talked about earlier: input, processing, and output. To understand the “processing” part, scientists typically can open up the system and observe it.

To illustrate, think about the human gastrointestinal system. To understand how mammals digest food, early biologists literally cut open animals and cadavers and took a look. There is a mouth connected to a tube, connected to a stomach connected to another set of (much longer) tubes, and an exit at the rear. Enter food, process food, exit waste—and all (mostly) observable. Neurologists have been observing brains where they can; by the 1500s scientists like Andreas Vesalius were publishing about the anatomy of the brain, but until very recently technology has not had any ability to help observe the actual workings of neurological systems. In fact, the now accepted number of approximately 86 billion neurons which a human brain has was not even discovered until 2005 when Dr. Suzana Herculano-Houzel pioneered a way to count them. Science literally did not know how many neurons were in a brain until about fifteen years ago! Science had made some educated guesses: the best guess until Dr. Suzana's research was 100 billion, which is off by about 14 billion, or an entire baboon's-brain worth of cells.

This matters quite a bit because it illustrates just how opaque the brain is as a system to study. Psychophysics started at the logical beginning—measure the inputs and measure the outputs. That began the quantification of perception. Then Structuralism, Functionalism, and the more advanced schools of thought they birthed took the next step: quantify the inputs and outputs in a more complex way. Instead of light, sound, taste, touch, and odor as inputs, they began thinking holistically about experiences which are processed, and then result in a new system which may or may not react the same way the next time. If a stomach gets an ulcer, science can see it. The system does not react the same because it has changed in an observable way. In the brain, these differences are often virtually undetectable. Science can observe the input and the output, but the processing is still largely a mystery.

10.2 The Construct Chasm

This applies to working with people data because it illustrates a concept we call the "Construct Chasm." In behavioral data, there is a big inferential gap scientists must leap between what they are *trying* to measure, and what they can *actually* measure. Earlier, we mentioned turnover as an easy outcome to quantify, and it is. Yesterday they worked for the company, today they do not, and that means they left. Divide the number of people who did that by the number of people at the company over a period of time, and a rate is produced (e.g., 10 people per month in a group of 1000 is 1% turnover per month). This is a relatively easy way to describe what happened in the past. Therefore, measuring observable behavior from the past is relatively easy. Where this gets complicated (and where machine learning hopes to help) is explaining two things: (1) why did it happen and (2) what is going to happen next? This is where the construct chasm enters the equation. *Risk* of turnover is the likelihood that an employee will voluntarily terminate their employment with an organization or engage in behavior which results in their involuntary removal from an organization. The factors leading to these two outcomes are where science and

statistics must start making inferential leaps. We will illustrate with a decidedly opposite example:

"Iron"-Clad Benefits: When we use statistical analysis to quantify the cost of the production of a car running through a sophisticated supply chain, we are measuring concrete things with concrete outcomes[1]. If steel costs $650 per metric ton, then the cost rises $25 per metric ton, and you use 1000 metric tons each month, then it will have $25 × 1000 × 12 impact on your annual operating costs. The fluctuations and variances in the price of steel have an objective impact on your process and your finances. If that $300k is too much, you may slow production, find an alternative metal, or solve the problem another way. However, the variance requires no interpretation, and any fluctuation can be explained by reasonably clear factors.

Similarly, the solutions have reasonably clear pros and cons. Changing to a different metal will have pros and cons that structural engineers and supply chain professionals can calculate—fail rates, sourcing costs, equipment changes, etc. Slowing production is a conversation for sales, finance, and legal (where contracts with customers are involved). These are not simple problems to solve, but they are reasonably objective.

However, if you spend $100 per month per employee for your 1000 employees to subsidize their health benefits and find out that the rates will be going to $125 to maintain the same out-of-pocket for your employees, that $300k lands differently. Similarly to the steel, you can absorb the $300k into your operating costs, in which case there is no impact except to your bottom line. But the solutioning phase is quite different. Maybe you reduce your coverage to maintain the same out-of-pocket for the employees. Cutting some benefits is a way to save money on the premiums. Most people do not use all the services you offer anyway; they will hardly notice. Or maybe you raise the out-of-pocket you demand from your employees. After all, that $300k must come from somewhere and it is either this or impacts to compensation next year. The difference here is that there is no structural engineer to calculate the fail rates of these decisions. No scientist can perfectly quantify how 1000 people will react to the decision you make and there is no way to measure it truly objectively.

For option one, you might analyze benefits usage. Which benefits are heavily relied on and utilized, and which are largely ignored? You could use the insights there to infer what people will not miss. For option two, you may examine salaries and other compensation factors to infer how much $25/month really means to your employees. Or you might be very inclusive and send out a survey to ask them what they think. The data you get back will allow you to infer what the opinion of the group is.

You are trying to quantify how individuals are going to process this information and how it is going to affect their life and their opinion of the organization they work for. The input is clear—changes to the out-of-pocket insurance costs. The outcome is opaque because you cannot directly observe how that input is going to be

[1] We do not intend to diminish the complexity of the art and science of supply chain or the automotive industry—the example is just simplified for the sake of illustration.

processed by the system. This is what makes this challenge so different from the steel example and is what illustrates the Construct Chasm. The problem of the steel is the same: $300k impact on operating costs. However, the big difference is the car production system is far more observable. Your engineers, supply chain experts, finance folks, salespeople, and the like can observe the impacts on the system and weigh the options. Again, it is not an easy problem, but it is a problem which you can objectively observe.

In our benefits example, the $300k is there, but there is no direct way to observe how your employees are going to process the decision you make. The best you can do is be proximally familiar with the potential impacts. If you collect survey data, you can infer how each decision will land with your group. There will be different opinions related to all the differences in your employee base, and you can infer how those employees will impact the organization if their opinion is on the other side of the chosen solution. It could be direct like reduced performance or turnover, or chronic like sustained financial hardship which adds stress and leads to burnout. Either way you cannot define those impacts like you can with the fail rate of steel which is based on physics. You can only infer, which is what makes behavioral data often a chasm to jump.

10.3 War on Intelligence

By the late 1800s to early 1900s, psychology developed enough traction that groups like La Société de Psychologie Physiologique were founded in France (1885–1893) and the American Psychological Association (1892–Present) was born in the United States. And although psychology was growing rapidly, it was still a largely academic and medical pursuit.

Then during World War I, psychology had its first breakout in the applied world when Robert Yerkes developed Army Alpha and Army Beta tests to classify over 1.5 million soldiers. The US government wanted a scalable way to assess fit for the Army, so they commissioned Yerkes to build the tests to evaluate the cognitive function of soldiers. They were built to measure verbal ability, numerical ability, ability to follow directions, and knowledge of information. The government used this measurement of recruits' intellectual and emotional abilities to make decisions like their job classification and potential for leadership positions. Alpha was for literate, English-speaking recruits, while Beta was for non-English speakers and the illiterate.

The test gained so much popularity that the Rockefeller foundation bankrolled funding for more advanced behavioral research. This gave rise to, among other things, the movement of eugenics in the United States. The test was thought to measure native intelligence but was actually found (many years later) to be better at predicting acculturalization, or how well one had assimilated to life in the United States. This was a big miss at the time because it means that if a person was born in the US, they had a big advantage on the test and in the 1920s it also meant they were

probably (a) white and (b) descended from ancestors in northern/western Europe. We know this today because follow-up research has shown the test correlates almost perfectly with years spent on US soil.

Nonetheless, at the time it *seemed* to work and after the war Yerkes was appointed as an Expert Eugenic Agent and put on The House Committee on Immigration and Naturalization. During his tenure there he helped draft and implement the National Origins Formula, which was essentially used to restrict immigration of certain nationalities because, in Yerkes' opinion, American citizens could not "afford to ignore the menace of race deterioration."

These early applications of the beginning of modern psychology show us the first time people realized that they could use psychological testing to judge people's abilities. Up until this point, psychology was used to *assess* differences, not *judge* them. Army Alpha, eugenics, and assessments like the Stanford-Binet IQ test were the first industrial-level application of institutions and governments using assessments to judge whether someone was fit for a job, an officer's rank, or worse, equitable treatment as a person.

This critical shift in the application of theory raises two major considerations for the modern practitioner seeking to predict behavior today:

1. *Does the test, algorithm, approach, or process explain accurately and reliably enough to be a legitimate tool for decision-making? And what is the risk if you are wrong?* Often times, and as seen through the atrocities of World War II, science might think it knows that something predicts, but finds it is mistaken. That is part of science. However, the critical error not considered was the *risk* of acting on incorrect information—millions of human lives were devalued as a result of not understanding the true nature of the effects which were observed and has left an ugly, yet indelible mark on history.
2. *Is what you are doing legal and ethical?* Sometimes what seems like a small decision or change can have big downstream impact. One should always carefully examine the implications of research *before* it is conducted, as well as consider the implications of the decisions it drives.

These two points bring us to an important book by Cathy O'Neil.

10.4 Objectives Matter

In her book *Weapons of Math Destruction*, Cathy O'Neil outlines a great many places across industries where advanced mathematics aimed at business optimization have gotten out of control. In her view, when mathematical tools like machine learning (1) do harm to people's lives, (2) are unexplainable to the populations they are used on, and (3) are used at large scale they qualify as a "weapon of math destruction." The pun makes sense—in a world where we increasingly rely on software and statistical models to evaluate truth, unregulated tools and systems are poised to be as harmful as some of our most powerful bombs. However, instead of

leveling cities, these weapons attack subversively, gnawing away at infrastructure and social systems until they collapse. Just ask any average person who lived through the economic crisis of 2008 and subsequent recession.

While that market collapse was a glaring example caused by a few bad apples wielding "scores" generated by "math," O'Neil has a great many other cases which outline everything from crime fighting software to college admission to prehire assessments. She goes into detail about how these tools can hurt great numbers of individuals by creating harmful confirmation bias, self-fulfilling prophecies, and just plain bad insights resulting from shoddy math and questionable research methods. A great example is the short-term loan industry. When these institutions recruit new clients, they want to find people who have money issues already, and who feel that a short-term loan would help get them out of trouble. Practically speaking, this makes sense—market your product to the people who need it.

In theory, yes. But the way these companies generate their leads is questionable at best. They spend exorbitant amounts of money to canvass enormous amounts of people. They use click habits, online activity, and demographics to triangulate what an optimal customer would look like. Then, they use those data to generate interest rates. Unintentionally (at best), they end up charging higher rates not just to people who are more likely to default, but also to the people who mathematically resemble them, but have never necessarily behaved in a way which would indicate they should pay higher interest.

On the surface, this seems like a smart business. If an algorithm can find someone who is more likely to default on a loan, they should be charged a higher interest rate. That is how any company who loans money manages their risk. And no model is going to be perfect, there are outliers in every bell curve.

Three important points on this: first, this entire premise only holds up if the assessment of their likelihood to default is accurate and fair. If they are not accurate, they do not work. If they are not fair, they are not ethical. Accuracy is easy to show, so let us talk about fair. In today's world, businesses cannot use race to determine an interest rate—that is illegal. But many of these models use zip code, and zip code can be highly correlated with race. In practice, skin color then becomes a de facto variable in these models which, though unintentional, breaks a great many civil liberties Americans fought hard to create. Second, scale must be considered. You might be thinking, "predatory loans are not a new thing." And that is correct, pawn shops have been preying on cash-strapped people for decades and we are not going to defend their business models. But pawn shops do not have (1) everyone on the internet as a potential customer nor can (2) a pawn shop infiltrate your day-to-day life through web-based marketing. In the world of targeted marketing, these predatory loan companies can find the millions of people who need their services and subtly stitch their messaging into potential clients' online experiences.

This leads to the third and scariest point: negative feedback loops. When they prey on someone and the cash-strapped person takes their loan, they then are likely going to fail to pay it back (partially because of the enormous interest charged). Does that person now need the same services more, or less than before they engaged with the establishment? Obviously, the answer is more.

So, if a company only markets to people who need cash, then basically guarantees most of them will fail to pay it back, they are zeroing in on a societal problem and making it worse. And they are doing it on the grandest of scales—the internet. They are literally making poor people poorer and rich people richer.

What does this have to do with machine learning with employee data? One of Cathy O'Neil's chief theses in her book is that objectives matter. Weapons of Math Destruction are only defined as such if the tools are directly or indirectly harming the people they are being used on. And that is where to take a lesson from—we must use machine learning techniques, and any other form of analytics, to help companies *and* employees. If you are not doing the former, it is not efficient business. If you are not doing the latter, it is not ethical business. If you are not doing both, it is not good business.

10.5 Some Fat is Good

In the first Matrix movie, a sci-fi trilogy about AI taking over the world, Agent Smith (a piece of software with human-like intelligence) spends some time talking to one of the humans resisting their new world order:

> *"Every mammal on this planet instinctively creates a natural equilibrium with its surrounding environment, but you humans do not. You move to an area and you multiply and multiply until every natural resource is consumed. The only way you can survive is to spread to another area. There is another organism on this planet that follows the same pattern. Do you know what it is? A virus."*

Another word for this equilibrium is homeostasis. Systems of various sizes create a homeostatic balance between their component parts so that all the parts may perpetuate. As an example at the individual level, mammals have a system of complex feedback loops to regulate body temperature so that the other various internal systems can operate and give the proper nutrients and signals to the temperature regulation system. As an ecological example, algae in the ocean regulates water temperature by growing with the sun. When they grow too much, they block the sun, the water cools, and they die off until they do not cover the water as much and the temperature comes back up. The growth works with the heat to balance the algae population. These sorts of examples abound in nature from the micro to the macro.

In business, we do not always see this. Humans seek to win, not to balance. Agent Smith had a point—we often sacrifice long-term equilibrium in a market for the sake of winning in the short term. Capitalism is a zero-sum game and we want to "get ours." Our shareholders are happier when we make more money, so do what it takes to get there. However, when we are getting that win, is it going to work in the long term with respect to an industry, the whole market, or the overall success of society? Those predatory loans might be driving revenue right now, but they are also contributing to the polarization of wealth and collapse of the lower-middle class. The employees of Lehman Brothers thought everything was great in 2004, but how did they feel in 2009?

When performing machine learning, sustainability is a key hallmark characteristic of a good model. As we see in Cathy O'Neil's work, you can use advanced analytics to squeeze money out of a market. Likewise, in HR Analytics you can use advanced analytics to squeeze money out of your business processes and/or employees. But unchecked, you run the risk of taking it from the wrong place which may cause you to run afoul of sound ethics and/or ultimately hurt your business in the long term.

To be clear, we are not advocating inefficiency. We are not anti-profit. Companies need and do work to make their operations and employee bases better. We are pro-equilibrium. Too often, we mistake economic efficiency as the sole predictor of a business win. We want to make it clear that there is no free lunch. Every dollar you take out of a system has an effect. If we eliminate a job, that might be good for the company's top line, but it is not good for the person who lost the job or their peers and coworkers, and often creates disruption to operations as well as reduced morale and productivity.

Again, and we want to be abundantly clear on this point, we are not anti-efficiency. There are times when reorganizations, layoffs, and other organizational design or operations decisions may not seem rosy for the employees on the surface. We are not so naïve to suggest that. But we also want to ensure that we are not blind to the homeostatic effects of those decisions. Driving efficiency to the extreme microlevel with operations and employee analytics has to be done with prudence. Taken to an extreme, imagine if a cheeseburger restaurant is the only business in a small town and is so efficient that it only requires one employee. The business might see that as operational excellence. For the first couple of years or so the restaurant experiences record profits because the work is so streamlined. But over time, not sharing the revenue with the rest of the local economy will cause them to lose. That is, eventually, one employee will be the only person who can afford to eat the cheeseburgers. The population will bleed out their dollars and have nothing left to give.

Business Optimization is not the Same as Profit Maximization.

In their book Gardens of Democracy, venture capitalist Nick Hanauer and civics professor Eric Liu analogize money in an economy to blood in a body. They talk about how we are on a trajectory to metaphorically concentrate all our "blood" in a few upper-echelon parts of the body and starve the rest. Loosely speaking, if people cannot afford to buy the products and services of the companies they work for, then the economic system is not sustainable. If our brain and lungs do not share their oxygen-rich resources with the arms and the legs, we cannot walk to where the food is and put it in our mouth to ensure the whole system stays alive. This is a recipe for disaster.

Another good anatomical analogy is fat. In America, most people have too much of it. We have villainized fat. You cannot check out of a grocery store without seeing 15 celebrities tell you how to get rid of your fat, what to eat to reduce your fat, or just showing people who are not fat as "good" and people who are fat as "bad." No doubt that too much fat is bad—there is well-documented science of how too much fat stresses your internal systems, reduces the quality of life, and ultimately shortens

life span. But the opposite is also a problem. Anorexia, bulimia, and other eating disorders are real issues on this other side of the spectrum.

Fat is important. It plays an incredibly pivotal role in the effective operation of a human body. In addition to insulation, fats store energy and nutrients for later use. Fat acts as a messenger, helping proteins do their jobs and start chemical reactions to help control growth, immune function, reproduction, and other aspects of basic metabolism. But on the surface (literally) we tend to be more preoccupied with our waistlines.

Unfortunately, we have taken business operations and employees to the same extreme without thinking about "good fat versus bad fat." Any inefficiency is too much inefficiency. We must have the lowest possible operating costs and the smallest possible staff to get the work done. This maximizes margins and that is good for business.

The problem with this of course is the same as with predatory loans. In the loan industry, they rejoice in their huge profits, not concerned that they are sucking the life out of the poor people they prey on. In organizations, we run the risk of sucking the life out of our employees and cultures by making things, dare we say it, too efficient.

Predatory loan companies do not care much—and will not care much—until they run out of people to take from or the market collapses. Companies have a more urgent, but still veiled, issue to contend with. Driving efficiency too hard might remove important fat that maintains implicit systems (like culture) as well as important employee affect (like organizational commitment and job satisfaction). The trouble is these long term, chronic pains are masked by short term wins. For example, finding an algorithm that screens résumés 10% faster might seem great, but if the recruiters are already maxed out by the social and emotional demands of their job, they might not be able to take 10% more requisitions on. In the short term, it looks like the team is filling more jobs, but eventually it will drive burnout which at best leads to give-up-and-quit turnover and at worst leads to give-up-but-stay employees.

In the world of venture capitalism, they have a similar concept called "dumb money." Dumb money is money a start-up gets from an investor, but it comes with strings they do not want attached. The money may come, but it comes at a cost that is not worth it in the long run. Getting too efficient is a different version of dumb money—the benefit can be seen in the short term but is blind to the long-term chronic repercussions of the decision. Remember that any money saved in the name of efficiency has to come from somewhere.

This consideration is paramount when beginning to embrace machine learning in HR Analytics. What kind of fat are you trimming with your models? Are you helping to bring an obese process down to a fit weight so that it can live a long, healthy life? Or are you starving an already undernourished department in order to meet some ulterior objective?

10.6 Bias, Authority, and Effectiveness

Over-indexing on efficiency is not the only thing which can prevent an organization from using machine learning well. Three other intoxicating advantages can distract teams when building models and are well demonstrated in the history of science and technology: bias, authority, and effectiveness.

Machines are not biased. One of the authors' favorite myths to dispel during analytics discussions is when they hear, "It is not perfect, but using (insert automated technology) is better than people. At least machines are not biased."

The problem with that statement is well illuminated by the old saying, "Guns do not kill people, people kill people." Regardless of your feelings on firearms legislation, in this capacity we encourage the reader to see the analogy for what it is: Machines do what humans program them to do. And whether that is a fully-coded piece of software or a flexibly designed machine learning algorithm, there are humans behind them. And since a human is writing the code or setting the guidelines, all machines come with the biases (intentional or not) of their human designers. Part of human nature is to be bias, and we all have unconscious biases that we are not even aware of. Therefore, it is important to design machine learning models with bias in mind—triangulating opinions, checking with a diverse set of stakeholders, and examining the effects of outcomes on all possible groups will help you ensure that bias does not negatively impact your employees.

"I just did what they told me to." In 1961, Stanley Milgram did a set of experiments which showed the extreme lengths to which people would go when they were following orders. Milgram wanted to know to what extent people's beliefs would bend when they perceived they were following orders from a credible authority.

He advertised in a local New Haven, CT paper that he was conducting experiments on learning and memory. When the participants arrived, they were "randomly" assigned to be a student or a teacher—except every participant became the teacher. The other "participant" was actually an actor who was in on the experiment. Milgram then explained that the student would be memorizing words and depending on their performance, the teacher would be administering progressively more severe electrical shocks. The shocks were labeled across several categories, ranging from mild to moderate and severe and ending with a red label marked "XXX."

The experiment was actually about following authority. After the "student" was connected to the shock-giving system (no shocks were actually administered), the experiment would begin with the authoritative scientist giving orders to the participant based on performance of the actor, who would progressively get more and more questions wrong. The experiment would continue until either the participant refused to administer more shocks, or they had administered the most severe shock possible.

Over the experiment there were many variations, like whether the participant could see the person being shocked, the words the scientist would say, or the amount the subjects were paid for their participation. But the net result was that in most cases, over 60% of the population tested were willing to go to the highest level of

shock when being directed by the scientific authority. That is, they were essentially willing to kill somebody in a lab setting if their actions were perceived as coming from a higher authority. Though the experiment is now regarded as unethical at best and psychologically damaging at worst, Milgram's studies have been used as the basis for understanding how people relate to sources they see as authoritative.

In some parts of the world, human rights violations are still a very real threat, but there is a new force which threatens to keep people from thinking too critically about their decisions: machine learning. In Germany during WWII, it was the SS and the government telling people how to act, regardless of what they believed, or thought was right. But then at the Nuremberg trials, these accused parties used the "science" of eugenics to defend their actions.

In Milgram's lab, it was the authoritative scientist who played this role. Today, many use data, machines, and the "answers" they provide to justify behavior. They want a number or an algorithm to do the thinking for them. When models *replace* decision-making rather than *augment* decision-making, trouble is not far behind.

Allegiance to a number without critical thought takes away the human part of human resources. If practitioners conflate the machine's abilities with authority, then they are at risk of making decisions which do not align with what is right. This is a difficult line to balance on because empiricism and logic encourage the creation of sound methods and data to guide decisions, while theory and ethics demand a more subjective approach. Furthermore, bias creeps in from all angles to threaten the integrity of both approaches! This balance is not easy to do well, but simply passing the accountability off to a higher (or automated) authority will not lead to long-term success.

The "It Just Works" fallacy. Just after WWII, a West German soap making company called Chemie Grünenthal was doing research to address the need for more antibiotics. Research and development in this space has a great many work streams, one of which is preparation of reagents (chemicals used to start chemical reactions). While in this process, one of the pharmacologists discovered a drug which behaved much like a well-known and widely used sedative at the time, Doriden. Further research began to refine the drug into something marketable.

As the researchers spent more time with the new compound, they discovered that in addition to sedation, the drug was also a highly effective antiemetic—it was very good at reducing the symptoms of nausea and vomiting. Recognizing a great opportunity, the drug was quickly moved to market and by 1957, the brand names Contergan (in West Germany), as well as Distival (in the UK, Australia, and New Zealand), were being heavily marketed toward pregnant women to alleviate morning sickness.

Then, something bad happened. Thousands of infants across the drug's marketed footprint began to be born with teratogenic defects. Specifically, this means missing, deformed, and under-formed limbs began to show up in the babies of pregnant women who took the over-the-counter drug before their third trimester. By 1959, the drug also began to show links to peripheral neuritis, or damage to nerves in the peripheral nervous system, and the drug was made prescription-only (not taken off the market, just more tightly controlled).

By 1961 the effects were so obvious that outrage in the public and press got the drug pulled off the market. Today, Thalidomide (as the drug is more famously known) is a case study in imprudence; a rush to market without the consideration of potentially life-threatening side effects. And while this seems on the surface like a flagrant failure of corporate responsibility and the regulatory bodies in place to monitor companies, consider for a second the precedents for pharmaceutical exploration in the 1950s: drugs taken by pregnant mothers were not strictly controlled because scientists did not believe medications taken by pregnant women could cross the placental barrier. This belief, and the aligned regulatory laws, were the result and Thalidomide its consequence.

In Germany, a large criminal investigation was launched in 1968, but from a legal perspective Chemie Grünenthal had a reasonable case. And by 1970, settlements with victims had been reached and the cases were closed with no findings of negligent homicide or injury by the German government. The UK and Australia tell similar stories: companies paid significant damages because they recognized their actions directly hurt others, but criminality was dismissed.

From the ethical perspective, consider two other important facts. First, the "nothing-crosses-the-placental-barrier" scientific belief did not exist without challenge. Studies had already shown (by 1957 at the latest), the effects of alcohol on the fetus during gestation, so there was legitimate evidence that compounds could cross the placental barrier. This fact makes the second more important: Thalidomide was going to be specifically marketed to pregnant women. So, while it is understandable that in most drug exploration a company does not need to investigate the effects on the fetus (because that is not the scientific paradigm of the day), if the primary market is pregnant women, it is important to weight any evidence, regardless of how fledgling, more heavily.

The case of Thalidomide demonstrates a dangerous fallacy masquerading as evidence: it works! The encouraging evidence from abbreviated research blinded the organization to the fact that they quite poorly represented the drug's target demographic, yet quickly scaled to market anyway.

Said differently, they used the ends to ignore the means (and the scaling of those means). The fact that a model or algorithm "works" in that it shows some positive outcome, such as relieving short-term symptoms like financial constraints, does not mean that the organization is supposed to just accept it as "good." We must not mistake "it works" for "it is good," because that would mean no further critical thought is required.

We must demand that "it just works" is not good enough, nor is it a substitute for critical thought. To be a good model, it must stand up to scrutiny from triangulated sources. Business experts, ranging from finance to human resources to operations must be given the opportunity to understand how the model works and weigh in on potentially long-term consequences. Ignoring or not investigating the long-term implications of our models for the sake of short-term relief means we open our businesses up to a digital dose of Thalidomide. Nobody wants morning sickness for 9 months, but nobody wants teratogenic birth defects either. And given the choice, we think the mothers would have chosen nausea.

10.7 Meat Packers and Machine Learning

Fortunately for today's modern employee, the world of employment law is not as unregulated as the world of predatory loan marketing. To understand why (and how machine learning fits in), we will use an example back from just after the industrial revolution of the early 1800s.

Imagine you are a young man living in Chicago. You are the son of a Polish immigrant and your family arrived in the Windy City just before your birth in 1854. It is now 1879 and you have been employed at a local meat packing company since the age of 10. Let's review a bit about your probable working conditions:

- 14–16 h per day, 6 days a week
- $0.10 an hour (~$3/h in 2020 dollars); your mom makes ½ that because she is a woman, and it is more than double what you made when you were a kid
- No breaks, except to eat twice a day
- No safety precautions when working with machinery
- No protective equipment for the cold environment
- No light or air filtration in the freezers or on the warehouse floor

But all in all, you do not complain (because that is how you get fired). Then one day, a hook gives way and an entire pig lands on you, breaking your leg. Your employer simply fires you and hires someone from the queue of prospective employees who line up each day hoping for a job. While you are out of work, your family cannot afford to eat, so you send your kids to stand in the line, hoping to get employed while your leg heals. Unfortunately, since you cannot afford medical treatment, your leg gets worse and they ultimately amputate, keeping you from doing essentially any manual labor job you are qualified for.

This example is grim, but it was literal reality for working-class families at the turn of the twentieth century. Millions of men, women, and children were abused by the companies which wielded the power to hire and fire at will, as well as treat employees as expendable commodities.

Then, in the 1920s and 1930s, the United States began to take significant action to correct the injustices that grew from unregulated working conditions. With the introduction of the legislature like the Railway Labor Act (1926), the Norris-La Guardia Act (1932), and the various aspects of the New Deal (1933–1939), the American worker began to see more equitable treatment.

Since then, employment law has grown into a rich tapestry of regulations and laws designed to protect the worker. As an example, some of the most significant subsets of the US government (at the federal level) dedicated to workers include the Department of Labor, the Occupational Safety and Health Administration, the Employee Benefits Security Administration, and the Equal Opportunity Employment Commission.

These do not include tangentially involved government agencies that have stake in worker treatment, such as the Commission on Civil Rights or the Department of Justice Civil Rights Division. And *those* do not include any of the nongovernment organizations dedicated to fair treatment, like the Fair Labor Association, the American Civil Liberties Union, and Workplace Fairness.

But what was the underlying condition which leads to the problem in the first place? There were many, and it is not the place of this book to review the sociopolitical undertones of postindustrial revolution America. But one of the glaring issues which is relevant to this book is: *the ability to work in new ways went faster than the ability to regulate it*. Think about it like this: humans never needed to regulate airspace before they could fly. They never needed to regulate airwaves before the radio. Basically, society does not need rules about things until after they get created. And by the time they are created, industry does not wait around for people to figure out how to do them right because they are in a hurry to reap the benefits.

The digital era brings us its own version of this problem. Bias, adverse impact, negative feedback loops, violation of privacy, and others are all the modern-day equivalents of the risk workers must navigate today. And while it is certainly not as grotesque as child labor and the abject poverty of people working 90 hours per week, its effects can be just as damaging to the economy and financial health of the average person.

"But I did not know." Nobody sets out to build a biased model (extraordinarily few at least). Most people do not consciously say, "I know zip code correlates with race and I cannot use race in my model. I do not care if I am discriminating as long as I am making a profit." The trouble in this instance is it truly is honest-to-goodness ignorance.

Unfortunately, ignorance is no excuse for the law. Ignorance of a legal violation does not excuse the perpetrator from the legal consequence nor does ignorance of an ethical violation do much good for the people being violated. The difference is there is no legal consequence for an ethical violation until it is backed up by a law. In the meantime, as practitioners of machine learning and analytics, it is our duty to examine and understand the models we are creating, how they work, and what adverse impacts they may have.

This chapter reviewed many things to consider when working with machine learning and other forms of advanced analytics. We have compiled a Top-8 summary sheet with the main points for this chapter. We encourage you to copy it or take it right from this book (it is on its own page) and keep it somewhere it will be of use to you. Given organizations' focus on business results, which must always be a primary consideration for profit-driven companies, these tenets may not always naturally float to the top of project plans or model design. That said, we hope they will remain as part of your skillset and help you be a well-rounded practitioner.

Example	Observation	Takeaway
"Iron"-Clad Benefits and the Construct Chasm	Behavior is very hard to observe and define objectively with data.	Always consider what you are trying to measure because if it is intended to predict or describe behavior you might be making a big inferential leap from your data to your theory.
War on Intelligence	Despite this challenge, we used psychological assessments to make judgments about people in WWI. This went badly.	Ensure your judgments and decisions are truly valid, and understand the ethical, legal, and business risks of being wrong.
Objectives Matter	Modern data science, which machine learning is a part of, is enabling us to make judgments faster and across larger groups of people.	Is your work going to ultimately help the business without harming employees (directly or indirectly)? Sometimes these lines are obvious and clear, but sometimes they are not.
Some Fat is Good	We must drive efficiency, but not at the expense of sustainability.	All efficiency comes from somewhere. Are you helping take an obese problem down to a healthy weight, or unintentionally starving an already undernourished system?
Machines are Biased Too	All machine learning is built by humans and therefore is at risk for bias.	Triangulate opinions, source diverse perspectives, and recognize that everyone is biased. This will help you find potential bias in your models and outcomes.
Stanley Milgram's Shock Study	Machines can seem authoritative and let us think we do not have to think.	Statistical and machine learning models must *augment* decision-making, not *replace* it.
Birth Defects from Thalidomide	Good results can distract us from thorough investigation of impact.	Do not allow positive results to distract you from exploring all potential impacts of your models.
Working conditions in the late 1800s	Legislation always lags innovation, and data science is still ahead of our ability to regulate it.	Always examine the effects of your models beyond what is legally required, since legislation is still new or nonexistent in many HR analytics spaces.

Discussion Questions

1. Define the construct chasm. Provide three examples of HR metrics which have large construct chasms and discuss how you might bridge the gap to measure them well.
2. Think of a time when an assessment or model did not measure what it was supposed to or did not measure it well. What happened? Why did the measurement miss the mark? What were some of the repercussions of the mismeasurement?
3. Think of a time when a change at a job did more harm than good. Explain what happened, and why the negative impact was missed.
4. Discuss a time when bias went unnoticed in the creation of a new process, assessment, or data model. Why was it missed? How was the bias discovered and remedied?
5. Provide two examples (not necessarily at work or in HR) where algorithms have replaced human decision-making. Choose one you think has benefitted society and one which has caused challenges. Discuss why for each.
6. Discuss the current state of data legislation in your country and the world. Which countries have most recently created legislature to protect employees specifically? How do you expect that to impact HR Analytics functions of the future?

Chapter 11
Machine Learning Project Management

The final four chapters in this book provide readers with frameworks and tips for executing machine learning projects. Learning some key phases of machine learning efforts in Human Resources will help you begin to think about how to make this work happen in your organization. Whether you are a student, an analyst learning how to get in the advanced analytics game, a qualified data scientist getting started in HR, or a leader creating and delivering a vision, these chapters should help you get started.

11.1 Seven Reasons to Do Project Management Well

In its most basic form, project management is a set of standards, processes, and skills used to achieve specific goals and objectives. Virtually every industry and every organization use project management of some kind, though different groups take the formality of project management to various levels. However it is executed, quality project management techniques increase transparency, align expectations with stakeholders, improve consistency, and reduce risk to delivery. In this sense, machine learning projects are no exception—project management can be particularly beneficial in improving the likelihood of success and minimizing the chance for delays and rework.

Project management with machine learning is important because, like any piece of a larger system, a machine learning model developed in a vacuum will have significant trouble fitting in with the processes and teams which surround it, regardless of how well it predicts the desired outcome. An engineer would not design a piston without understanding how the rest of the engine works. Not considering important context leads to friction—a force which prevents the intended outcome. And just like the piston will likely not fit or move in harmony with the rest of the engine, practitioners must think about the system in which their model is going to exist.

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_11

A successful machine learning effort considers the context in which it will operate and therefore requires active partnership with stakeholders and subject matter experts. This extends from the initial definition of the business problem through to acceptance of the results and maintenance of the model. Furthermore, a good model that makes accurate predictions will have an impact by creating change in the business, usually by impacting decisions, policies, or processes. Good project management defines clear plans for these considerations and for how model-generated insights will create impact.

To drive this point home, here are *Seven Reasons to Do Project Management for Machine Learning Well*:

1. *Good project management helps you understand the business problem thoroughly*: Active stakeholder involvement combined with a consultative data team will ensure the model is designed so that it accurately addresses what it is intended to. As discussed in Chap. 4, understanding and feeling the effects of a problem is not the same as operationally defining it in a way that can be solved with machine learning. Business problems are usually complex and layered, requiring deep understanding of not just the pain points, but also (1) where the pain comes from and (2) the potential impacts of potential interventions. A good doctor treats the disease, not the symptoms, and does so in a way that does not compromise the health of other parts of the body. A complete understanding of the business problem and context will ensure you make all the right choices, so data models do the same.
2. *You will understand your data (and data owners) better*: Another deceptively simple reason for good project management is taking the time to understand your potential data sources and their business context (which may be different from the general business context). This requires active partnership with the subject matter experts who are data owners. This is necessary because you want to ensure that you consider all potentially valuable data sources and that you understand their data elements and the ecosystems they exist in.

   The stakeholders who are data owners are unlike the business or end-user stakeholders because they (1) often do not feel any of the pain you are trying to fix and (2) may have different considerations and constraints than your other partners which impact their ability to partner with you. For example, an internal IT team managing an applicant tracking system likely does not feel the pain of inefficient hiring processes. However, they may have a full plate doing data management—loading, refreshing, reconciling, and other tasks required to ensure the data is good. If your model needs a constant or regular feed of data, these two considerations are important because (1) what is IT's motivation to help? They already have a full plate and they need to understand *why* they need to help. (2) Your needs may conflict with their already maxed-out data management schedule.

   Data comes from disparate places, and stakeholders come in all shapes, sizes, and needs. Building the business cases, getting buy-in, and getting thorough engagement from these types of stakeholders is absolutely necessary for success.

3. *Good project management ensures clarity of model requirements and criteria for success*: Data models can solve all kinds of problems. Along with this diversity-of-problem comes the diversity-of-considerations with regard to model accuracy. In an ideal world, all models would be 100% accurate, but that is not the case. The next logical assumption might be, "I want the model to be as accurate as possible." This may seem to make intuitive sense but may actually significantly hinder model production. It is essential when designing machine learning projects that the business is actively involved in discussions on model accuracy: how accurate is a viable product? What are the risks of false positives and false negatives? For example, a turnover prediction model for a company with 10,000 employees and an annual turnover rate of 30% loses about 250 people each month. If an analyst built them a model that predicts 90% accurately it will deliver an insight like, "250 people will quit next month plus or minus 25." This may give the Workforce Planning and Finance teams accurate enough data to significantly improve forecasting. However, if one of the company's factories has 160 people, the predictive validity may go down substantially, since at 30% turnover the math would predict only ~4 people leaving per month. If the model works at all (which it probably will not), it would be looking at such a small sample that insights would be tough to generate. That does not mean this model is bad or wrong, it just means that it was built to provide insight on a specific problem to a specific group. No single model is going to solve a very wide breadth of issues. The point is to ensure the team is clear about which problem they are solving, and that they spend the time to accurately understand what success looks like.

   Additionally, an "as accurate as possible" mentality may cause models to take an extremely long time to create. Balancing delivery speed with accuracy is one of the key equilibria data teams must strike when building and tuning models. The upfront research to understand and set these thresholds is key to ensuring a model which creates a timely impact.
4. *Good project management allows you to plan for deployment across all impacted groups*: The groups who will be using the model or who will be impacted by the insights it generates must be engaged at some level in the model development effort so that upon completion there is acceptance and support for use of the model. This must go beyond the three or four leaders who are backing and driving the project—it must include representation of the group of employees who will be impacted by the changes a model will bring about. The whole point of a machine learning project is to influence decisions or processes, and so it follows that the groups who stand to have their decision-making and/or processes affected should have a chance to participate.

   In the world of biological transplants, sometimes a body's immune system will see a donated organ as a foreign body and reject its presence. Imagine how frustrating that must be: waiting months or years on dialysis to receive a new kidney, then finally receiving one, only to have the body simply says "Nope. I do not like it." This chapter began by talking about how parts of a system must work in harmony. The organ rejection analogy is a reality for some, but it is a figura-

tive risk for machine learning projects. Regardless of the quality of a model or its efficacy, it will not survive if the organizational body rejects it. Project management will help get in front of this very real threat. Good project management always includes good change management, and good change management will ensure something like organ rejection does not render an otherwise effective model unusable.

5. *You will reach clarity and alignment on resource needs*: Most anything can be accomplished in business if time or money is unlimited. Conversely, as either gets smaller the other must get bigger to compensate. No team has unlimited amounts of both, and most struggle with not having enough of either. Machine learning efforts often require resources from multiple teams with multiple agendas and priorities, so the earlier in the project these resource needs can be vetted and weighed against other initiatives and projects, the more likely you will avoid frustration from being under-resourced.

   Specifically to machine learning efforts (and advanced analytics in general), review of resource needs and setting expectations are critical steps because executive sponsors and/or senior leaders often simply lack the knowledge to know the resources required to do this type of work. As their expert, the analytics team must be able to size the resources required to deliver and make those expectations very clear to ensure the work meets timeline requirements.
6. *Drive cultural change*: Project management is particularly important for ensuring the success of machine learning efforts in Human Resources because it provides legitimate structure and outcomes for HR work in a way that enhances its credibility as a function. This drives differentiated value for the business.

   We have mentioned that although data about employees and the workforce is increasingly being gathered and analyzed, HR has historically not made extensive use of its data beyond operational needs. This is due in part to the complexity of HR data and, in particular, its behavioral nature. This state is magnified by the (in general) natural interests of individuals who choose to work in Human Resources—outside of the compensation function, most organizations have few individuals in HR with deep background in math and statistics. Solid management of machine learning projects helps bridge the gap between stakeholders who may not understand how to use data in advanced ways and those who are building the models.

   In this way, project management can serve as a catalyst for broader dialog with stakeholders on how to use and interpret data to improve decision-making in HR. This has potential long-term benefits as stakeholders realize the value of data. With each machine learning project, HR leaders will better understand where data can be applied and the questions that can be answered by analytics. Project management through its active stakeholder involvement can be a key enabler of a data-fluent culture in HR.

   The advancement of a data-driven culture via well-organized efforts also drives a key facilitator in any team or industry: trust. Good project management means active involvement with key individuals throughout the life cycle of the effort which goes a long way to establishing trust in both the team doing the

work and the results it creates. Trust is essential to the success of the effort, regardless of how technically impactful the model is. This matters significantly for the same reason many new ideas and processes fail: unfamiliarity breeds uncertainty, uncertainty breeds anxiety, and anxiety breeds retreat back to familiar strategy. Good project management ensures clarity and opens the channels of communication to help unfamiliar stakeholders get comfortable with the methods and progress, which ultimately creates trust and helps shift culture.

7. *Good project management confirms alignment to legal, ethical, and cultural assumptions*: In Chap. 10 we talked at length about examples where moving too fast or not planning well can leave blind spots which create negative outcomes. Good project management will ensure that these blind spots are reviewed with care and the efforts of the project will not violate relevant law and will be aligned with your company's ethos. With HR analytics, models often directly impact the lives of employees at work, and potentially even at home. A company's values should be consulted as part of the process to ensure that the outcome of the project is an authentic representation of the organization's operating principles.

11.2 What Makes Machine Learning Projects Unique in HR

The above seven reasons are a short list to keep in mind when ensuring you and your leaders make adequate time to plan and manage a machine learning project. Additionally, there are some attributes of machine learning which make managing their projects unique.

We do not know what we do not know. By its nature, machine learning deals with uncertainty. That is, most often teams initiate machine learning projects to solve a problem they do not know the answer to yet. These projects are often investigative in nature: Why are people quitting? What is an optimal starting salary? How is leadership potential best defined? Where are our biggest opportunities for improving engagement? And although the askers of these questions often have an idea of where to start, they must not let preexisting biases cloud the investigation.

Also, machine learning projects are often embarked upon at a point in time when the analytics team does not even know how they are going to answer the question yet. Often unknown at the outset and initial scoping of a project are things as simple as "which data is needed," "who owns those data," "which statistical techniques are appropriate," and others.

Clarity is brought progressively to all these facets as a model is developed. That said, this makes machine learning projects very different from projects HR professionals are more used to. For example, traditional IT projects (like implementing a new piece of HR software) can have nearly every aspect defined upfront and are simply a matter of executing. This does not make these types of projects easy—there are still many issues to discover and work through—but knowing what the end looks like at the start makes the project less ambiguous.

Another good example to compare machine learning to is process reengineering. If an HR team wants to change the process they use to approve promotions, they can create a project to evaluate and design a new process, and then transition from the old process to the new process. Again, we do not imply this is easy—there are many complexities in both IT projects and process engineering. The key difference is that in these types of projects, the stakeholders mostly know what the end looks like at the beginning. Machine learning projects generally operate under much higher-level objectives that cannot be transformed into specific actions until late in model development. This demands that scope and requirements should focus on the business problem and not make assumptions about the solution and the changes it will bring.

As an example of this, imagine an organization seeks to understand the key drivers of employee turnover. The high-level objectives (understand and reduce undesirable turnover) can be established early on. But until a mature model is developed, how the team plans to fix the problem will not be known. The model might identify compensation, training, career opportunities, or other factors as the key drivers of turnover. Each result would have fundamentally different recommendations and actions associated with solutioning. Traditional HR projects start with both the pain, diagnosis, and treatment identified. Machine learning projects typically only start with one:

Type of project	Pain	Diagnosis	Treatment
HR software	Inefficient TA	Outdated software	Better software
Business Process Reengineering	Unhealthy inflation in promotions per year	Poorly regulated promotion approval process	Reengineer the promotion process
Machine learning	Too much Turnover	???	???

Failure is an option. Second, an inherent risk in machine learning efforts is that the model may not be able to successfully predict or provide valuable insights on an outcome. Failure, in this sense, is a real possibility. This can be a bit confusing or scary to HR leaders since risk of failure is not as common in HR as it is in departments like Product Development or Marketing.

The history of HR has been largely transactional and incremental. Closely aligned with groups like finance and legal, HR departments do not usually tolerate high levels of risk and prefer to operate from safe positions, innovating incrementally lest they open themselves up to untenable disruption in core operations or great cost to the business.

This is not a bad thing—some groups within organizations must be more conservative than others. That said, it is limiting because small risk means small reward. The era of big data in HR means opportunity in the industry to find great competitive advantage by driving great improvement to the employee value proposition. This means taking chances on investigations and research which may not yield results every time.

This will be a culture shift in project chartering and risk tolerance for many HR teams. Asking for budget and time to work on a project that might not yield any value is not something HR executives are accustomed to approving. That said, ensuring these sorts of stakeholders are aware of this risk and are willing to accept failure is essential to undertaking a machine learning effort.[1]

It is also important to communicate that there can be value even when projects don't succeed. Even when an effort does not produce a useable model, they can still add value to the business and enable future efforts by identifying new data that should be captured or data quality issues that should be resolved. Findings like theses can lead to changes that enable future success. Along with this, it is important that stakeholders know they should not force implementation. The concept of "something is better than nothing" may not apply in a machine learning project and although this concept can be difficult for leadership to understand and accept, it is imperative to communicate effectively.

MVP: Minimal Viable Product. The concept of minimum viable product is not unique to machine learning, though it has a unique slant in the realm of predictive analytics—in this application MVP is the need to define early what the acceptable level of accuracy is. This does not need to be an exact measure, but serious thought should be given to required model accuracy as it will guide decisions made during model development. Required model accuracy is also a significant factor in determining whether a model and the project is ultimately successful. For example, the business may be willing to accept a model that is 40% accurate in predicting whether an employee will be promoted within 6 months whereas 20% may be too low and would not be usable.

11.3 When to Use Project Management

Though elements of project management would benefit most efforts, formal project management is not always required for machine learning development. When deciding whether an investigation should follow a more formalized approach, we would like to give you three key situational considerations to help you think through whether formal project management makes sense.

1. How cross-functional is the effort?
   For models that require partnership across multiple teams or departments, project management can help by facilitating the communication, expectations, and needs of each area. Periodic meetings, mutually developed scope, and regular stakeholder involvement can ensure that efforts that require partnership are successful. In machine learning projects, this goes for both the requirements for the

[1] This is not to say every machine learning project may fail. Often there are very high probabilities of success in a project. We just want to be clear that the variance in success probability is much higher in ML than in traditional HR projects.

data as well the subject domain of the project. For example, if a single group or very well-connected small group owns all the necessary data to complete the investigation and project, then less formalization is required because data can move very easily to where it needs to be. Likewise, if the subject of investigation is controlled by a single or small group and/or the potential impacts of the study will only impact a single or small group, less formalization is required. Conversely, the broader the group required for the data, investigation, and potential impact, the more important it is to formalize the management of the effort.

2. How big is the potential impact on business operations?
   If a model will require substantial process or organizational change to implement or support, then formalized project management is highly recommended. Coordinating process change typically requires stakeholders working together to define, validate, and test the new processes. Project management can facilitate these efforts and ensure all relevant stakeholder perspectives are considered.

   For the implementation of a machine learning model specifically, it is important to think about how this new way to inform or automate decision-making will impact how the business runs. It may have logistic impact, meaning the flow of business process may change and therefore needs minor to major process reengineering to accommodate the new model. It also may be a change management effort, meaning those who are part of the impacted business processes need to be bought into the model being trustworthy, effective, and an overall win for the team. This is especially important if those impacted by the change were not a significant part of the design of the model.

3. Will we create an asset which requires maintenance?
   Machine learning efforts which produce a final product such as an ongoing data model or a product for use (like a dashboard) typically benefit from a project management approach. Planning for maintenance should begin early in the effort because support needs must be designed, and proper resourcing allocated. Early deployment is usually highly monitored by the researchers and designers, but once moved to "business-as-usual" production, there must be a plan for its care. Formalized project management can improve the chance for long-term success and reduce the risks associated with this transition.

   Not all machine learning models have this sort of outcome. A one-time model built to diagnose a problem and influence a business process might be designed simply for the in-the-moment diagnostic work and then retired. However, sometimes a model will be built and then used consistently month-over-month, quarter-over-quarter, or year-over-year in effort to provide insights over time. An example of this might be a high potential program selection model. In this case, project management will answer questions like:

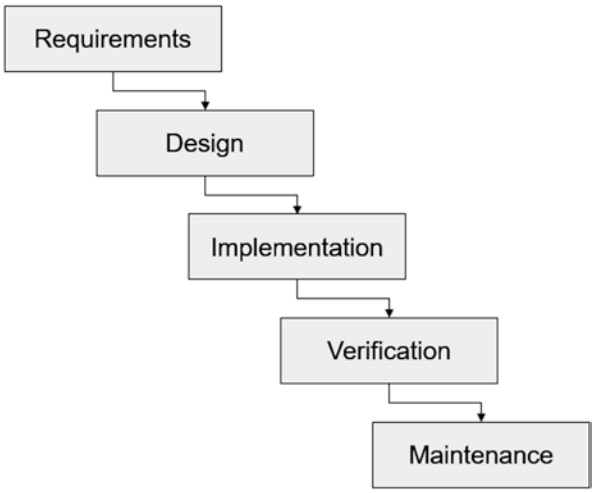
   – Who is sending us the data? When? And through what channel(s)?
   – How are we auditing the data feed to ensure quality?
   – Who is loading the data to the model and testing it?

- Who is quality-assuring the results?
- Who is maintaining the production of the data or insights for end-users to consume?

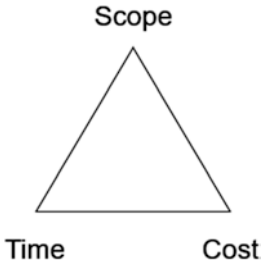
These are important questions to formally answer with project management because it is very often many different groups working together. The data delivery comes from whoever owns the data and can range from end-users of that data running reports to IT professionals setting up and maintaining automated feeds. The loading and running of the data once it has arrived and shipping the data once it has been run through the model is often a separate IT group who supports the analytics team or their end-user population. Finally, the data auditing, model quality assurance, and end-user aspects such as visualization and insight creation are typically analytics teams or end-users themselves. Each project, team, and organization are slightly different, but the idea is the same: the more you are building something which requires many people to be involved to deliver regular production and ensure long-term success, then the more formalized the development and resources requirements should be during design.

Once a team has decided they will use a formalized project management approach, which should they use? Here we will review a couple of common philosophies, and one which is less common, but more specifically designed for machine learning.

The first is called Waterfall, and is a traditional approach which is still around, but not as popular as it used to be with the advent of Agile (discussed next). Waterfall project management is highly structured and serial in nature, which means it moves step by step using very specific criteria. Projects move through gated phases that flow forward, hence the name waterfall. A typical waterfall model looks like this:



A major focus of waterfall projects is to balance what is called the "triple constraint" of project management: scope, schedule, and cost:



We will not go deep into the challenges and techniques to combat the triple constraint, but the general idea is this: as any of these three attributes of a project approaches an extreme, the others must balance to compensate. As time gets shorter, scope must decrease and/or cost must inflate. As scope gets bigger, time and/or cost must also get bigger to compensate, and as available money reduces, either time or scope (or both) must adjust to meet the financial restrictions.

There are several waterfall frameworks with two of the most popular being PMI (US) and PRINCE2 (Europe). Both frameworks are widely used and are supported through certification programs and memberships that cascade knowledge and ensure project management practitioners are well versed in the methodology.

Waterfall projects start with an idea that is developed in partnership with stakeholders. Next, requirements and scope are refined and documented, and a budget and timeline are established. Once all parties agree on what will be delivered, project implementation begins. Stakeholders are kept updated throughout the process via regular communication. After implementation, the deliverables are reviewed by stakeholders to ensure they meet requirements. Upon signoff, the project moves to closure. Each phase has requirements to begin and requirements to end and move to the next phase. It does a great job of ensuring all critical aspects of a phase are handled before moving on.

Though the waterfall model is well established and has been in use for decades, it has been subject to criticism recently. The primary issue cited regarding the waterfall model is that it is not flexible enough to accommodate the dynamic nature of most projects and suffers from lack of feedback loops. And due to the fact that stakeholders don't always ask for exactly what they need up front and that needs can evolve relatively quickly, deliverables in a waterfall model often stray or miss the mark without having the opportunity to be recalibrated. This makes projects notoriously difficult to control and means waterfall projects are rarely on-time and within budget. Further, "gold plating" is a common challenge where additional work continues to be added to a project without resulting in significant additional value. Finally, the ability to properly forecast and plan the effort required for the underlying activities is a well-known challenge that can quickly add to projects' tendencies to exceed budget and timeline.

A popular alternative to the waterfall model is Agile, which was created initially for software development and has been expanded to other types of projects. Agile in its purest form is iterative and incremental. Projects are broken into short sprints (typically 2 weeks) and delivery occurs in chunks. At the end of each sprint, a

revised product is delivered to a representative from the business (the product owner) who has a chance to review it and provide feedback. This ensures that the results align with expectations and allow the team to quickly adjust scope or direction. Instead of finding out at the end of a project that what is being delivered does not meet expectations (as was common with waterfall efforts), the business can be directly involved in defining and realizing scope.

Managing efforts in this way also allow for better and more predictable budget and schedule. Individual changes or work items can be prioritized and negotiated to ensure timeline targets are met. Budget can either be flexible, allowing a project to continue iterating as long as it is producing value that outweighs the investment, or rigid, shutting down the effort when budget or timeline targets are reached. The result in the latter option being a working product based on the scope of what was able to be completed within the allotted time or budget.

Agile is based on a set of software development values and 12 principles which form the foundation of the methodology. Though focused on software development, the principles of Agile can also be thought of in light of machine learning efforts:

Agile software development values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile's 12 principles:

1. Customer satisfaction by early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (colocation)
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective and adjusts accordingly

In theory, these values apply rather well to machine learning. For example, working models are key, responding to change is critical, and relying on motivated and self-organizing teams is often very advantageous. However, the nature of Agile can leave some gaps which cause its framework to struggle with machine learning efforts. Agile grew out of software development and is based on the idea of providing a base deliverable and then iteratively developing that deliverable through active partnership with the business and regular releases. In machine learning, a huge part

of the work is done during the data wrangling and model development phases, at which time there is little to show the business that would help redirect the dialog and partnership, rendering a big part of Agile's framework effectively useless.

This is not to say that feedback loops and iteration are not useful for machine learning projects, nor that none of Agile's values or principles apply to machine learning efforts. On the contrary, when done well, feedback loops, iteration, and other parts of Agile are well-met when managing machine learning. They just need to be organized slightly differently.

Now that we have reviewed (1) the basics of project management, (2) how machine learning projects are different in HR, and (3) some basic models, we will dive slightly deeper. The goals of a machine learning project are a unique confluence of linear steps, which are aligned with many opportunities to loop back and iterate. And while machine learning projects are also heavily frontloaded with significant investigation and setup, the investment in time can create extremely impactful results. The final two chapters will review how to do this well, as well as how to ensure that stakeholders understand these unique attributes so they can be effective partners and sponsors.

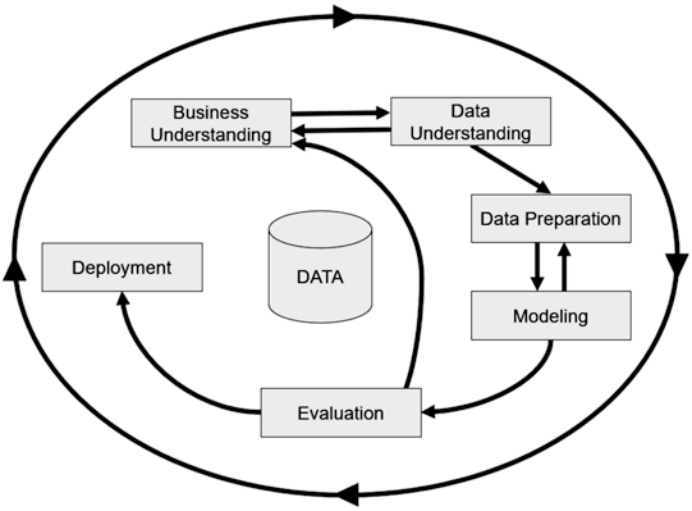
Discussion Questions

1. Name three of the seven reasons to do project management well you think are most important and explain why you think they are more important than the other four.
2. Which of the attributes which make machine learning project unique is most impactful in HR? Why?
3. Create two examples of machine learning projects: (1) where project management is necessary and (2) where project management is not necessary. Explain what the differences in the projects are and why they should use different approaches.
4. Explain the triple constraint. Why is it so important?
5. Name one benefit and one weakness of both Waterfall and Agile approaches to project management and how they apply to machine learning projects.

Chapter 12
The 3 A's of a Machine Learning Project

Though both the waterfall and agile frameworks can be used to manage machine learning efforts, their limitations create challenges when working to create a machine learning model. In response to these challenges, project frameworks have been developed specifically for managing machine learning efforts. These frameworks help overcome many of the issues with generic project methodologies when managing machine learning projects.

The most commonly used machine learning project framework is CRISP-DM (Cross Industry Standard Process for Data Mining). Conceived in 1996, developed in partnership with five companies (SPSS, Teradata, Daimler AG, NCR Corporation, and OHRA), and updated most recently in 2015, CRISP-DM has become the dominant machine learning project management framework. It includes several defined steps and a process flow that considers the typical path to care for all the needs of a machine learning effort:



C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_12

Instead of using project phases linked to discovery and requirements gathering, the first steps in CRISP-DM are focused on the iterative relationship between the business problem and the data landscape. Then, the method moves into data preparation and modeling where the team iterates between getting the data ready for a model and model creation. Finally, the model is evaluated and either sent back to be reiterated or is deployed.

Using a methodology like CRISP-DM for managing machine learning efforts provides several benefits. First, it allows for more efficient management of machine learning projects as it is attuned to the typical flow and structure of machine learning work. Second, it allows for clearer communication to stakeholders on the progress of the effort. The steps align more closely to how machine learning efforts typically evolve, which makes it easier to show status and next steps. Third, the methodology considers the iterative nature of the subcomponents of a project. The iterative cycle when understanding the business problem and data landscape as well as the iterative cycle of data prep and model design are the two most critical feedback loops the method has, neither of which exist in waterfall or agile. In addition, clearly showing that the "evaluation" stage can lead back to the initial phase of "business understanding," is beneficial for stakeholders to see and understand. It is not uncommon for machine learning models that are built to return to the drawing board if they do not meet performance expectations.

Methods like CRISP-DM that embrace the irregularities of machine learning when compared with more traditional projects are a great start, but often teams need to employ additional project management frameworks for overall management of the effort. Picking best practices from traditional methodologies, or even fully using them for given workstreams can often be beneficial for managing components not specific to model development. For example, we have mentioned that sometimes a model impacts business process and could result in business process reengineering. In this case, a team may want to use CRISP-DM to build the model, but agile or waterfall to develop the new business process or implementation strategy.

12.1 The Six Phases of a Machine Learning Model Lifecycle

Regardless of the specific project management method used, a general cycle for machine learning model development follows a predictable pattern which starts with ideation and ends with model deployment and upkeep. CRISP-DM is a great model for visualizing this flow, but for our purposes we are going to keep the components of the lifecycle general and methodologically agnostic. This will allow the reader to take the material and apply it to whichever project management approach they choose.

And though these stages will be shown in sequence, keep in mind that they are rarely executed in a strictly serial manner. Just like we have seen in both Agile and CRISP-DM, iteration and feedback loops are critical between stages as the project evolves.

The first step in a machine learning effort is to obtain an **(I) Understanding of the Problem**. This involves working with the requestor and stakeholders to define the problem in clear, tangible terms. During this stage, the team will also begin to explore data sources that could be leveraged to build a model. Upon completion of the stage, the team should have a solid understanding of what they are trying to solve and confidence that it can be addressed through machine learning techniques.

In the next stage, the team will **(II) Frame the Project**. The goal here is to produce a business case or other similar document to summarize the scope and potential value of the project. This includes the problem definition which was gathered during the first stage, along with potential solutions and estimates for scope, schedule, budget, and return on investment (short, medium, and/or long-term). The business case will be used by leadership to prioritize and authorize projects. Once approved, budget and resources can be assigned.

(III) Data wrangling is the next phase in the cycle and is where data scientists spend most of their time. We have mentioned previously that data wrangling is the part of the project where the team actively partners with the business, IT, and other data owners to identify potential data sources and variables that could be leveraged. The data can then be collected, cleaned, and integrated to make it usable by the model.

It is commonly estimated that 80% of machine learning is data wrangling. The discovery, transformation, and prepping required can often be very intensive. Working in close conjunction with the data wrangling stage is **(IV) Model Building**. In this phase, models are prototyped using the data gathered earlier. The data is shaped and optimized for use in machine learning in a process called feature engineering. The phase ends when a model is produced that meets the business' requirements or it is determined that a viable model cannot be created in the schedule or budget constraints that have been defined.

After the model is built, **(V) Results** are then communicated to stakeholders for review and acceptance. This is an essential step to ensure stakeholder support of the model and to obtain their commitment to action on (or not action on) the project's recommendations. The results are typically delivered in presentation form, summarizing findings and potential next steps. This review differs from the more detailed, technical review of the model results which is performed at the end of the model development stage.

The final step in the cycle is **(VI) Deployment and Upkeep**. This stage requires close partnership with the resources tasked with maintaining the model (often IT and/or HR Operations) to plan for deployment and support of the model in production. Once launched, the model should be monitored and tuned in accordance with a cadence deemed appropriate based on the nature of the data. A good rule of thumb here is that the faster the data changes and provides insights (weekly, monthly, quarterly, etc.), the more often it will need to be examined, audited, and tuned. This includes the time where the business determines that the model is no longer providing value, so it should be either retired and decommissioned or given a full overhaul in the form of a new project.

A pneumonic to use to help remember these steps is "The 3 A's of Model Development: Appreciate, Assemble, and Adopt."

- **Appreciate**: Learn the challenge you are undertaking, both from the data and the business perspective
  - Understand the Problem
  - Frame the Project
- **Assemble**: Apply that understanding to source, transform, and analyze the data in effort to design a model
  - Data Wrangling
  - Model Building
- **Adopt**: Leverage the learnings and outcomes from your exploration and engineering to communicate and implement a solution
  - Results
  - Deployment and Upkeep

The remainder of this chapter will be dedicated to Appreciate: how we Understand the Problem and Frame the Project, while the remaining two chapters will help us learn to Assemble and Adopt.

12.2 Understand the Problem

When initiating a machine learning effort, having a clear understanding of the problem is essential. It can be tempting to dive into an effort without fully understanding the nature of the underlying issue or question. After all, it is easiest (and fastest) to simply begin pulling data based on an idea and see what sticks. However, if the proper time is not taken to probe deeper and understand the underlying issues from both the business and data perspectives, (1) significant effort may be expended on a fruitless path and/or (2) major gaps may be left which leads to (2a) risk that the model will not address the core business need or (2b) may lead to significant rework.

The first step in understanding the problem is to work with the originator of the request to discuss and document the business goals and objectives. This is an opportunity to gain a deeper understanding of what is driving the request. Working through the details of the problem can also provide valuable insights that allow for accurate framing of the question. It is common for analytics teams to receive requests that are so specific that the overarching objective is overlooked. This can lead to efforts that address only part of the problem or one view of the problem. Stepping back and discussing the broader need can lead to a fundamentally different approach to the effort and a more effective result. The skills we learned in Chaps. 4 and 5 are great strategies to leverage here, as well as general probing and consulting methodologies.

Some additional steps that the team must ensure to take when understanding the problem fall into two categories: understanding the business and understanding the data.

12.3 Understand the Business Perspective

We have said before that any data effort, machine learning or otherwise, has to fit into the system, culture, and processes of the business it will affect or live within. In order to get a good understanding of the landscape and help clear the necessary paths for success, one of the first things to do early in the effort is find an effective and engaged project sponsor. The sponsor is the individual who has ultimate accountability for the effort. They are the project champion and will help align leaders and the organization to the goals of the project. Further, they will confirm the business need is valid and properly prioritized, provide approval to initiate the project, provide business support and funding (if needed), and will approve and accept the results. Though the original requestor may also be the person who becomes the project sponsor, this is not always the case. Having the sponsor clearly identified early on will reduce the likelihood of the effort being scrapped due to lack of support from the business.

It is very important to note that the sponsor and the project manager *are not the same person*. In a machine learning effort, a data scientist or analytics team member may be the project manager. In other efforts where the data modeling or machine learning is only part of the effort, a person outside of the analytics team may be the project manager. Either way, the project manager is *not the sponsor*. The sponsor is a more senior-level person who is responsible for using (1) their understanding of the business problem, (2) their passion for solving it, and (3) their influence, to clear obstacles, secure resources, and *enable the project team to get work done*.

The opposite (but also important) common misconception about sponsorship is that the sponsor is a ceremonial figurehead. That is, the project is on their radar, but they do not know enough about the project nor are engaged enough in its progress to effectively fulfill their duties of enabling the project team.

In machine learning for HR, sponsorship is critically important because the field is so new and not yet well-understood. This means that this type of work is especially vulnerable to de-prioritization. The reason for this is that much of machine learning work is unglamorous and takes significant time to yield observable results. This means that in the world of shifting priorities and changes in business environments which drive the ever-existing battle for resources among teams and leaders, machine learning work needs a sponsor to defend their usefulness and protect the resources (time, money, and people) to do the work. Without exceptional sponsorship, it is exceptionally difficult for new ways of working to get a foothold in business operations.

The next critical category to understand the problem from the business perspective is to identify and reach out to key stakeholders. A stakeholder is an individual

who can either influence the effort or be influenced by the results. Stakeholders can shed additional light on the nuances of the problem and help to better frame the core issue.

When we talked about Meredith and her call center shrink, a big part of the effort was connecting with the leaders and employees in the call centers. Why? Because regardless of the data she could have pulled from the system, there is no substitute for understanding the context which is driving the creation of the data. HR Analytics Ikigai demands the analyst understands both Business Acumen and Behavior. Stakeholders will almost always serve as subject matter experts in these two domains and will shed much needed perspective on the data which will ultimately be collected.

Engaging stakeholders can also help secure support of the results when they are delivered. For example, if the call center teams were not engaged until after all Meredith's research was performed, they may have problems accepting the validity of the results and may even steer the team away from the proposed solution. People want to be part of work that impacts them—it is a lot easier to champion something when they helped influence its development and had an opportunity to get their perspectives considered. Good stakeholder engagement and management is a win-win for both the project team and the impacted population.

12.4 Understand the Data

The other side of the Understand the Problem coin is getting a firm grasp on the data landscape. The general industry term for this process is called exploratory data analysis (EDA). EDA should be performed to better understand the data aspects of the problem. Most of the work during this phase is consultation and research rather than detailed analysis of the data. The analysis that is performed is generally descriptive analytics supported by descriptive statistics and visualization for the sake of communicating early findings. Machine learning is not yet employed, and even basic statistical analyses should be limited. The focus is on assessing the data landscape relative to the question being asked.

A key goal at this point is to get a sense for which data sources are available and the potential variables or features that could be employed. This should not be a comprehensive inventory; that will come later in the data wrangling phase. Rather, this is an initial evaluation of whether sufficient information exists to solve the problem. Stakeholders and other subject matter experts can provide guidance both on potential data sources and factors that are thought to be associated with the outcome.

In addition to learning the "what" about the data, part of the assessment at this point is also the "where" and "how much" aspects of the landscape. The data sourcing and quantity are as important to the diagnostic work as which data the team has access to. As reviewed in Chaps. 8 and 9, different machine learning models have different data size requirements. What the analyst learns in EDA will help them understand which models may be best for their project. And as always, make

sure to consult with an analytics expert to understand which methods are best fit for the task at hand.

During this stage, the team should also assess whether there are relevant factors that are not currently available or that are not being captured. For example, if education data is not captured in the HRIS system or is out of date it may need to be gathered manually for the purposes of an effort. In cases where potential high-value data is not available and which might be leveraged by future analytics efforts, it is important as an analytics team to point out these opportunities. This is part of the skillset we talked about in Chap. 5 when we learned to research our research. Exploratory research is all about understanding what data is available and how that data does, or does not, help quantify an answer to the question.

Often, the current project may not be large enough to push the organization to start capturing a particular type of data, but it is important to raise the potential of future use as part of the business case. This applies both to internal and external sources of data. Though most HR analytics efforts solely leverage internal data, there are a growing number of external data sources and vendors who curate data for HR analytics which can be used to improve machine learning results.

Finally, though it may be tempting to begin to look for correlation (and answers) in the data, always remember where exploratory research and correlations sit in the process: it is best to use these insights to *inform* model building which will be a much more impactful solution. Analysis of the data without research rigor can lead to invalid conclusions. Simply cross-tabulating and graphing performance against a variety of factors can cause what is called "data snooping." Basically, if an analyst looks at enough things, they will eventually find something that looks interesting, whether it is meaningful or not. In addition, the team may waste effort looking for a solution if the problem has not been fully fleshed out.

12.5 Applicability and Feasibility

The reason to understand the problem from both the business and the data perspectives is because ultimately it leads to the main question: "knowing what we know about the business problem, can it be improved or solved with the data we have (or the data we can create)?" In short, can data be applied to the problem at hand? To analogize, the "business understanding" is like a doctor examining the patient and running tests to ultimately create a diagnosis based on the symptoms she can observe. Conversely, the "data understanding" is like a review of the available treatments—medicines, therapies, changes in patient behavior—which could help remediate the symptoms. Sometimes the answer on how to apply one to the other is straight forward, and sometimes it is not.

By the time you have a thorough understanding of both sides, an outline of the problem and data landscape will emerge. The team (business and data experts) can then discuss potential approaches including whether machine learning (or other data

science tools) are the most appropriate to address the problem. Here are a few points which should be brought up about applying data to a business problem:

- *Data quality*—Is the available or creatable data of high enough reliability and validity to be usable?
- *Data size*—Is there a large enough amount of data, data being created quickly enough, and/or enough variance in the data such that machine learning (or another analytical approach) can be used?
- *Definable factors*—Is what you are trying to measure measurable? This might be a challenge with the Construct Chasm (Chap. 10), or simply a problem with what data you have access to.

If the data can be applied to the problem, the next hurdle is feasibility. That is, even if the doctor knows what the diagnosis is and knows which treatment will work, that does not necessarily mean the treatment can be executed. For example, the solution to a weak heart might be a transplant, but the patient may be too old or ill to handle major surgery, or maybe a donor heart is not currently available. Basically, knowing *that* a solution exists does not necessarily mean that the solution is possible given the circumstances and resources.

The feasibility of machine learning's ability to solve a business problem falls into a few main categories you should assess. First: generalizability. In some ways, this is much like the concept of "does the past look like the future" and it will literally affect the validity of your model. That is, can you rely on past data to inform a likely future? In other ways, generalizability is about taking the model and applying it to other groups. For example, if the problem is specific to manufacturing plants or in the DC Metro market, does the problem also exist for sales or in the Pacific Northwest? If it does not, does that make the cost (time and money) of developing a solution for a smaller population prohibitive?

Second, will the insights you gain be actionable? Just because something is accurate, does not mean it can be feasibly, legally, or ethically actioned against. For example, if you predict that the likely reasons for the business problem are based on economic, competitive landscape, or labor market factors, what can feasibly be done? Sometimes these are surmountable challenges (like discovering a problem with compensation relative to the market), but other times, it becomes apparent that the challenges are more a function of natural circumstance than remediable business circumstances (like knowing that the company has entered a new, highly competitive market and they cannot afford to pay more than they already do).

Another key consideration of feasibility falls in the category of legal and privacy compliance. Early in the project, the team should assess potential privacy and legal concerns that could arise when trying to answer the problem posed by the business. HR personnel are no stranger to data privacy and legal compliance. That said, if you are new to HR you must become familiar with the guidelines for handling and using information about employees for the sake of making business decisions. In the United States, rules in domains like Title VII, HIPAA, and the ADA (and enforced by groups like the EEOC) play a significant role in what sorts of data you can and cannot handle, and what decisions can be made with those data.

For the last few decades, these fair treatment and personnel's right to privacy rules are where efforts stopped for HR professionals. However, HR is now tasked with an additional realm of responsibility: *data* privacy. This might sound the same, but it is actually an entirely different set of rules and laws. Whereas the traditional slant of HR data protection was aimed at *protecting employee privacy and reducing adverse impact*, new legislation is concerned with the protection of data so that it cannot be used illegally or stolen. As discussed in Chap. 5, this is where the rise of GDPR and similar legislature have begun to have legitimate impact on HR operations. Data processes like ingestion, storage, reproducibility, usage, and many other factors that were not typically worried about in traditional HR data privacy and fair practices work are now increasingly important to understand for all handlers of HR data.

When understanding business problems, these tenets must be a consideration. If the problem is solvable, but not without violating the rules, it is not really solvable. Business leaders and even data system experts will almost certainly not be versed in these laws, especially the new data-privacy regulations.

The final piece of this puzzle is the organizational culture, trust, and authenticity angle when it comes to privacy. Beyond the letter of the law present in Title VII, HIPAA, ADA, GDPR, CCPA, and others, you must respect *what works for your organization*. Different companies have different cultures that are the reflection of their leadership, histories, and industries and there is no single answer or approach that is right for everyone. What might seem like a participative pulse survey in one company might seem like an invasion of privacy in another. Respecting these norms is a critical part of understanding the business context you are operating in. That said, when done well, you can help evolve the organization toward a more people-data-oriented culture.

12.6 Frame the Project

Once you understand the problem and its solvability from the proper angles and perspectives, the next step is to build a business case. The specific definition of, and process you go through to create, a business case can take many forms. You have probably come across the term "business case" and likely have an idea of what one is. We will not dive deeply into the guts of the anatomy of business cases or try to prove or refute the various models of business case design. However, all business cases meet a very specific need: they effectively demonstrate the value of the effort you would like to undertake. Value is essentially the ratio of perceived benefits to perceived cost. Therefore, to oversimplify, a business case is your pitch of why the cost of the work (time and money) is worth it when compared to the benefits.

The business case also sometimes serves as documentation for the parameters of the project. It might entail financials, charter documentation, resource agreements, or other important details. Throughout the lifecycle of the project, the business case will serve as a reference for these details when working with stakeholders and spon-

sors. The business case is also typically used as a vehicle for project prioritization and approval. In this way, it is an essential deliverable as it helps to transition the effort from an idea to a real, working project. The format and content of the business case will vary based on the standards of the organization and the size and scope of the effort. The preferred format could be as simple as a slide deck presentation or as complex as a multipage, formal deliverable to a committee or project management office (PMO). If a template is available, it is best to use one that aligns with your organization's standards. For organizations with a mature project delivery function or PMO, they may even provide a resource to help create the business case and manage its review and approval.

Much like project management in general, not all machine learning efforts will require a business case. Small efforts that can be completed with minimal resources may be able to forgo creation of a formal business case. However, even if a formal document is not required, the core elements of the business case should be documented for any project that requires more than a few days of work. Collecting this information will ensure that members of the team understand the overarching goals and expected benefits that the effort will bring. At the end of the project, success can be measured in part based on how the results meet those objectives.

Even if your organization or leadership does not specifically require a business case, there are several situations where it should be strongly considered. These align perfectly to Sect. 11.3 when we discussed when to use project management:

1. Significant cross-functional partnership
2. Impact to business operations
3. Will require ongoing maintenance

Once you have determined you need a formal business case, documenting the value of the project is next, which we have explained can be thought of using the following equation:

$$\text{Benefits} / \text{Cost} = \text{Value}$$

Most usually it is easy for a project team to articulate the benefits of a project because that is where the passion comes from. That said, it is equally important to articulate the costs, which should be considered across these broad domains:

1. Monetary funding
2. Person-hour resources
3. Operational impacts (which are a form of cost from the change management perspective).

It is the duty of the project team and leadership to evaluate the benefits and the costs of those benefits, so the effort can be properly prioritized. The bigger the benefit, the higher the cost you will tolerate. And conversely, the lower the cost is, the easier it is to justify the project. But always, in order to show value, you must do a good job of articulating both parts of the ratio.

Though the sections and format of a business case will vary based on an organization's requirements and an individual project's needs, the common sections addressed in a business case are:

- Executive summary
- Business drivers
- Scope
- Assumptions
- Timeline
- Costs
- Benefits
- Risks

Business Drivers, Scope, and Assumptions: These sections should be built from the information gathered during the Understand the Problem stage. The focus is on the context that surrounds the benefits and costs of the project, as well as what you can (and cannot) handle within the bounds of the project. In a way, these sections are the story that supports the need for the project. In a machine learning project, these sections may be very technical because they may describe specific data systems, licenses, definitions of populations or metrics (in an effort to clarify scope), and other critical attributes. It is paramount to define things clearly, but also to summarize effectively. Often, the stakeholder-appropriate level of detail will be much more general than the in-depth information you have collected. In scenarios like this, ensure your business case has a robust appendix with well-documented details, but the actual presentation is kept at an audience-appropriate level.

Timeline and Costs: These explain to your business partners the realities and feasibility of your project. More detailed project management texts will explain the specific approaches to covering all bases here, but for our purposes think about these sections as an opportunity to explain what the project needs from a resource perspective. Time, money, people, vendors, hardware, software, ongoing maintenance costs, etc. are all relevant at this point.

These are very important to articulate effectively because when explaining machine learning projects to HR or business leaders who are unfamiliar with the nature of machine learning, you may meet some challenges. For example, machine learning projects typically need more time upfront for EDA and other exploratory work than other kinds of projects. In more general project frameworks, this is sometimes called the "discovery" phase. In our model, this covers all of the Appreciate (Understand and Frame) and half of the Assemble (Wrangling) sections. Machine learning projects are very front-loaded in this way which to an uninformed stakeholder may end up looking like a project whose wheels are spinning but is not getting anywhere. It is important to appreciate how this may appear to nontechnical leaders and handle their concerns appropriately.

Conversely, machine learning projects are usually less capital intensive than other projects. That is, other than maybe some software licensing and server space, the biggest cost to machine learning projects is usually people-hours. During development, this comes from analysts and data scientists, during implementation the

burden shifts to project and change management personnel, and post-implementation the final shift is more to IT resources. However, all in all, the financial costs when compared to things like professional development, team offsites, recruiting programs, and other typical HR efforts are lower-than-average.

The overall takeaway here is that the costs to execute a machine learning project are harder in some ways and easier in other ways than typical HR projects and so you must take care to explain these differences effectively.

Benefits: Calculating benefits can be tricky. Benefits of machine learning projects can be nebulous or require significant time to realize. For HR projects, in particular, the Construct Chasm (Chap. 10) can make quantifying benefits in terms of directly measurable return on investment very tough.

It is important when thinking about benefits to consider the three ways that analytics typically provides value, either because they are a project unto themselves, or as an add-on to another project:

- Tell a Story
- Solve a Puzzle
- Quantify an Impact

When a machine learning project tells a story, the model will be part of the diagnostic details required to make a larger business case. In this way, the work is exploratory in nature, but its insights will solidify the direction a team needs to go with other work. For example, a model which shows a significant difference in performance between two groups.

When a machine learning project solves a puzzle, the outcome of the model *is* the solution. This is more of an empirical research approach (Chap. 5) and is often the last piece of a larger puzzle. The benefit of this type of work is that now you know what the problem is and can act against it. For example, a model which shows the drivers of attrition.

When a machine learning project quantifies an impact, it measures the return on investment of work which has already been completed. This usually falls more in the world of descriptive statistics, but there are some methods we have reviewed in this book which could be used in such a capacity (e.g., linear and logistic regression). An example here might be running pre-post-test design like Marco suggested in Chap. 5—if he thinks a new compensation strategy will make an impact, he can examine the pilot population before and after implementation and then compare those shifts to groups outside the pilot group over the same period of time.

Nobody can tell you how to articulate the benefits of a specific project without knowing all the relevant details, but these above categories will help you begin thinking about where a proposed project falls in the landscape of value for analytics projects.

Risk: Risk is another category which more detailed business case material will handle in great detail, and we will focus on a high-level overview and how it commonly reflects in machine learning for HR. Risk typically falls into three main overarching questions:

– *Do this or Do that*: What are the risks associated with spending the resources required for this project?

  Any time or money a business spends on a project is time or money they *do not* spend on another project. This is the art of prioritization, and often contains comparative risk assessment of the other work your project is up against. "Who needs the resources more" is a function of where the work falls on the list of things which are vying for a share of limited resources.

– *Swing and a Miss*: What are the risks of doing the project and getting no answer or the wrong answer?

  A real risk of machine learning work is that you may not come up with an answer (remember, failure is an option). Therefore, you need to size that likelihood based on your understanding of the problem, as well as the consequences if you do not get an answer or you get an answer that turns out to be incorrect.

– *Not to Decide is to Decide*: What are the risks of not doing this project?

  Sometimes not creating action creates a bigger risk than trying something new. If a project is aimed at handling an incoming industrial or organizational headwind, or a preliminary analysis shows trending headed to a bad place, then the risk may come from *not* acting.

Executive Summary: Even though it typically goes first in the document, we include it last because the best summaries take into account all the details they support—therefore, they most usually benefit from being created near the end of the process. The contents of the executive summary are meant to be the proverbial "pitch" for the work. Which details you choose to include will be a function of the specific attributes of your project which you think are most important to highlight. This may even include the current landscape of the business, other competing priorities, and/or industrial context. A good mental exercise is this: Imagine you got on an elevator on the way to presenting this business case and your company's CEO got on behind you. They look over your shoulder and say, "what are you working on there?" What would you say? In two minutes or less, can you sum up the critical attributes (context, benefits, costs, and risks) simply enough for someone to understand? Summaries will always lack nuance and detail, but if your time to get the most important points out has been shrunk to an elevator ride—what would come to the surface? This is where the expression "elevator pitch" comes from but is an excellent way to test yourself on how well you understand the business context, challenges, and proposed solutions.

Once you have understood the problem and framed it appropriately, the work will receive a go/no-go ruling. The process by which this happens varies a great deal depending on organizational structure, department size, if you are internal support function or an external consultant, and many other factors. That said, at this point if you are authorized to proceed, you will begin to initiate your project via securing resources, engaging stakeholders, and kicking off the project management process and begin to refine your design for the project and its players. From a machine learning model perspective, this is where we transition from Appreciate to Assemble.

Discussion Questions

1. Name and define the three A's of a machine learning projects and the two categories within each.
2. Discuss what you think are the two most critical aspects of understanding the business and the two most critical aspects of understanding the data. How are they similar and how are they different? How are they interdependent?
3. What are the eight attributes necessary to frame the project? Why is each important?
4. What are the three major ways an analytics project provides value to a stakeholder? Create an example for each.

Chapter 13
Data Wrangling

Data wrangling is the first part of assembling a model and is the term applied to the collection, cleaning, and organizing of data. The term "data wrangling" began appearing in literature in the late 1990s and has stuck because it is a fitting description for the steps data teams have to go through to make data usable in the context of model development. In North America, the term wrangle is synonymous with the rounding up, herding, and the general organization of livestock—images of cows and sheep wandering in fields to be collected and brought back into their pens comes to mind. When we think back to the diversity of data sources from Chap. 2, and the complexities of operational definitions from Chap. 5, we can begin to imagine all that needs to be herded together to do data engineering and science. "Wrangle" is a solid analogy.

Data wrangling is an essential step in all machine learning projects (and most analytics projects in general). And like wrangling livestock, the process is time-consuming and often tedious—a commonly quoted statistic on data science estimates that 80% of any given machine learning effort is spent on data collection and preparation. To illustrate, let's think about the data lifecycle for a machine learning project in the same way we talked about our Analytics Ikigai Data Chef in Chap. 3. If we break the four main parts of working with data down into steps of the cooking process, it would look like this:

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_13

Essentially, we need to find our ingredients, prepare them for cooking, actually cook them, and then serve them to be consumed by a user. Wrangling is a big part of steps 1 and 2. Sourcing is where we go and how we get our ingredients. Are we going to a farmers' market or a grocery store? What are we buying? Are we examining the product to make sure it is fresh, has no bruises, and is ripe enough? Sourcing is all the things we do to get data into our kitchen. We may be getting a manual spreadsheet of data from a Subject Matter Expert (SME) via an e-mail. Or maybe it will be an automated report, a feed of some kind, or results from a survey tool hosted on a vendor's website. We must also consider things like cadence (how often the data can/should be refreshed), quality (is the data accurate and reliable), and the sustainability of that quality (how often might the quality change).

Once the data is in the kitchen, we then have to prepare it for analysis. If we were cooking, we couldn't simply put a whole bell pepper into a pot—it must be washed, seeded, and cut appropriately for what we plan on doing with it. Transforming is all the "prep work" that must be done to ingredients before analysis. From the data perspective, here are a few common things you may need to do in this stage:

Technique	Definition	Example(s)
Formatting and Encoding	Converting the format of data to the appropriate type	(1) Changing a date that comes to you as a number back to date format; (2) Parsing data that comes from a csv into unique columns; (3) Changing text-based categories to numeric categories so the software can understand it (e.g., Python)
Converting	Transforming data into common units	Transforming global salaries into one common currency so they can be compared
Mathematical Transformation	Using a mathematical process to change data into more useful values	(1) Z-scores (to normalize scale); (2) logarithmic transformations (to make nonlinear data linear)
Binning	Turning a continuous variable into a categorical one	Turning an engagement survey score (0–100) into a category like "high," "medium," and "low"

Technique	Definition	Example(s)
Imputing	Filling in gaps in your data with educated guesses based on data you do have	Assuming a score on an omitted survey item based on scores provided from similar questions
Free-form cleanup	Transforming free-form input data into a standard value.	Changing the values "RU," "Rutgers," "Rutgers U," and "Rutgers New Brunswick" to "Rutgers University" so they can all be tabulated together
Append, Merge, Filter, or Join	Bringing data in from different tables into one table.	(1) Adding columns to your data, like adding performance data to employee record data; (2) Adding rows to your data, like combining reps from the South Region to a data table containing reps from the East Region; (3) Using criteria to decide which records or fields are included, like creating a table of only employees who are present in both 2019 and 2020 performance reports

This is not an exhaustive list of all possible transformation techniques but rather is intended to give you an idea of many of the data conversion efforts teams must go through to ensure that data can be analyzed properly. We will discuss some of these in more detail later.

After the prep work is done, we can actually cook the food and ultimately serve it to family or guests. The majority of this book has been dedicated to learning how to do this part: solid scientific and research methods, statistics, machine learning techniques, privacy and ethical considerations, and most of the other topics have helped you think through *what* you need to cook and then *how* to go about cooking it. Data wrangling is the somewhat unpleasant, but critically important, step between knowing what needs to be cooked, and having ingredients prepared to begin the process.

And even though the "what" and the "how" are very important, from an "hours of work" perspective, the picture of wrangling compared to other steps is quite large. Let us reconsider our data lifecycle visual from this perspective:

If you are not a data scientist, this might seem strange. But to illustrate, let's continue to consider the restaurant industry. The cooking, presentation, and taste are the stars of the show at any given restaurant, which is why they get the most recognition. People outside the industry rarely talk about the teams (yes, teams) of prep chefs who show up at 4 am to unload the trucks, ensure the ingredients are of the right quality and quantity, and then spend literally all day washing and preparing the ingredients for the actual chefs who work in fancy restaurants. But make no mistake, no chef could function without them, most chefs started as one of them, and many chefs still work with these teams closely to ensure the product meets expectations.

In the same way, analysis, data visualization, and insights are the stars of the show in machine learning and analytics. Leaders and stakeholders rarely care "how the sausage is made"—they just want their data to "taste good." However, just like a restaurant cannot function without its prep chefs, algorithms and dashboards cannot work without quality data wrangling.

From a skills perspective, data wrangling is a suite of techniques and considerations (many of which we mentioned above) that are best learned through lots of practice. There is no "one way" to wrangle every dataset, just like there is no "one knife" that will work on every ingredient. In this chapter, we will introduce you to three main attributes for wrangling to help get you started. First, what are some of the most common types of HR systems you will encounter and what are some characteristics that make them special? Second, what are a couple of useful techniques when starting to wrangle data yourself? And third, what are some common tools you may want to consider becoming familiar with?

13.1 Wrangling Attributes: Quantity, Time, and Quality

Since HR data can be stored in various forms and be spread across systems and groups, curiosity and dogged research skills are essential. Much of what we learned in this book has been directed at teaching you about HR data ecosystems and how to think about and ask the right questions. That said, once you have asked the questions and gotten the answers, you will have to roll up your sleeves and dive in. Successful data wrangling is the active partnership with various teams who know the business (subject matter experts), data owners (when applicable), the groups who manage HR systems, and you—the person tying it all together. Wrangling is part data sleuth, part data cleaner, part architect, part engineer, and part construction worker. It is a simultaneously frustrating and satisfying process to work through to operationalize a research question in terms of actual tables of information which can then be analyzed.

But what characteristics of HR data make the wrangling process challenging? Every industry and type of data have common attributes which you must consider when diving into analysis, and we would like to enumerate some of HR's:

13.1.1 Quantity

By most data science norms, Human Resources has very little data. In the world of machine learning, the best models predict based on millions or billions of observations with trillions of potential interactions between predictor variables. There are two factors which keep HR data out of this class of data size:

First, population sizes. When studying behavior in data science, most are looking at consumer markets or geographic populations which puts the sample size in the millions. In HR, the populations are usually limited to the size of a given company or department which, at best, gets the population into the tens of thousands or hundreds of thousands. As such, people data is typically at least one or two orders of magnitude smaller than common sample sizes for data science.

Second is the rate of change. Other behavior-centered data science may be studying events which occur daily or even hourly (e.g., purchasing data in consumer science). Human Resources data moves comparatively slower. Machine learning is often concerned with predicting things like turnover, engagement, or promotion potential. These are events that happen once (e.g., turnover) or rarely (e.g., promotions), which makes their quantity much smaller and therefore more difficult to predict.

Combating data-size challenges is not easy and will often restrict which machine learning methods are appropriate. It is important when wrangling to develop an accurate understanding of how much data you can get, and how often it is refreshed or updated. One of the ways to increase your data size is to pull data over a greater span of time, but this comes with its own set of considerations.

13.1.2 Time

Time for HR data is incredibly important. As we said above, time can be one of the best ways to increase data size, which can be critical for wrangling enough data to build a useful model. Time can also be critical due to the fact that many machine learning models in HR are meant to predict things like a future behavior (e.g., performance or turnover) or potential (e.g., likelihood of success in a given scenario). These are fundamentally linked to time in that you must use past information about behavior and attributes to predict something which will happen in the future. This is distinctly different than other machine learning models. For example, in consumer science, a data scientist may use preexisting factors (e.g., age, gender, location, etc.)

to predict engagement with a product or service. In this case, time is not really as important as the description of the individual.

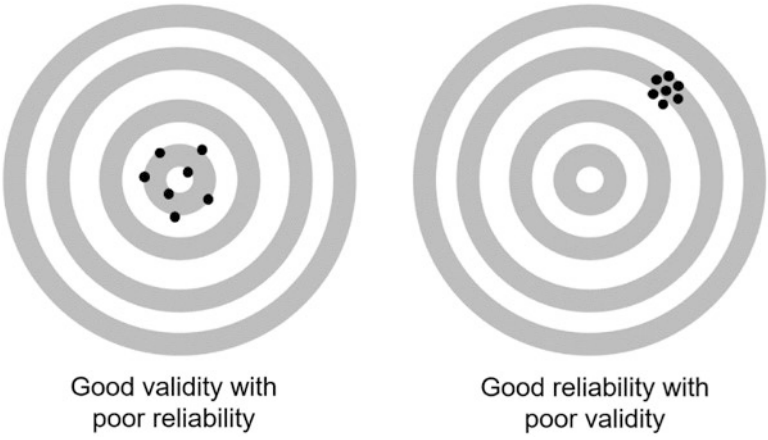
On the other hand, time can also be the enemy of HR data. Organizations and industries move fast. Competitive landscapes change—sometimes literally—overnight. Economic shifts change performance patterns, labor strategies, and promotion criteria. Though the term is often "five-year strategy" or "long-range plan," it often seems to be updated annually.

This means which *too much time* in data can create noise (see Chap. 8). Examining patterns over a period which is too long can mean that organizational or expectation changes can create inconsistency in the patterns a model is trying to detect. For example, in a predictive attrition model one of the authors built for a frontline organization, 2 years' worth of data did well, 3 years' worth of data did better, but 4 years' of data performed worse. When wrangling data across time, ensure to strike the proper balance between size and validity.

13.1.3 Quality: Validity, Reliability, and Variance

The balance between quantity and quality will always be a delicate one. When wrangling HR data, there are three main areas you must ensure to audit when collecting data.

Validity asks, "is my data correct" whereas reliability asks, "is my data repeatable?" Visualized, it looks something like this:



Both are equally important to understand. If you can get high validity and high reliability together that is obviously best, but they are often conflated or not considered separately. And in human resources data, they are both often suspect.

Validity in HR data will often be poor in older systems or in datasets which have little accountability or governance attached to them. Highly visible or impactful data, like salary or title, are usually valid because of the pain it causes when they are wrong. Others, like job code, are often not valid because getting them "wrong" carries little consequence. Here are two examples:

1. You need to pay $110,000 in salary to hire Juan as a software engineer, but the "proper" job code's hiring range tops at $98,000 because it has not been updated in a long time. Therefore, you use a job code with a hiring range which ends at $120,000. It is an incorrect job code, but getting salary right is more important.
2. In the West Region, "Widget Reps" sell Blue Widgets, Red Widgets, and Green Widgets. In the East Region, they have Purple Reps who sell Blue and Red Widgets and then Green Reps who sell Green Widgets only. However, they are all on the same compensation plan and so all three reps are labeled under the same job code.

Reliability, on the other hand, is a matter of consistency. Reliability issues are most often a challenge in large organizations where many teams are handling the same types of data. When reliability is low it means that the data being captured may be correct logically, but not be calculated or reported the same way every time or from every source. Here are two examples:

1. In the West Region, when someone gets a raise in salary it is always processed as a "Promotion" in the system. If someone changes teams without an increase in salary, the move is coded as a "Lateral Move." In the East Region, changes in pay are only captured as salary change, but not labeled as an internal move. "Promotion" is only used as a label when a change in manager and pay increase happen together. *In this case, both philosophies have merit, and neither is "wrong" per se. However, what counts as a "promotion" is not reliable across the company.*
2. Five years ago, your organization noticed that the technology used in your manufacturing plants was getting more technologically advanced, and workers with new skills were needed to do this type of work. They created a new job family with new job codes to do this type of work and began applying it to newly hired employees. However, some plants allowed their workers to continue in their old job codes despite having these new skills and responsibilities.

The third important idea to consider when wrangling HR data is variance. We talked a lot about variance in Chap. 6—it is essentially how close data points are to one another in a dataset. When a machine learning model is looking for patterns, it is essentially looking for patterns in that variation. If a dataset has no variance or too little variance, machine learning models cannot "see" anything.

A very common example of this in HR data is performance ratings. Performance ratings are the "one number" each employee is assigned every year to summarize

how well they did and are usually measured on a three or five-point scale. Unfortunately, they traditionally suffer from a significant lack of variance. To state it plainly, in three-point systems most employees rank as a two, with a small percentage of threes and typically very limited ones.

Data with very little variance is not an ideal candidate for machine learning. Part of wrangling is looking at the descriptive statistics of potential predictor variables and ensuring there is enough variance in order for the data to be useful.

13.2 Data Data Everywhere

The next major bucket to consider when beginning to wrangle workforce data is all the different places you can go to get it and the considerations which go along with diversity of source.

As part of this section, we encourage you to go back and revisit Sect. 2.3: *The Employee Lifecycle and Where Its Data Lives*, as the content is complimentary. In Chap. 2, we talked about the journey of every employee following a "lifecycle" from hire to retire, with 6 main parts:

- Attract and Select
- Onboard and Assimilate
- Engage and Reward
- Develop
- Advance
- Separate

We then talked about how data systems at organizations have largely grown to reflect those parts of the lifecycle, how data moves from raw data to the presentation layer, and how the information which comes from these systems serves as the foundation for all analytics.

In data wrangling, we now have to dive into this information. However, each organization has their own employee data ecosystem which is a product of their history, their governance strategies, their data vendor choices, and many other factors. Therefore, to make this content generic enough to apply to many organizations we would like to talk about (1) the *types* of data you will likely need access to and how they are generally connected, (2) some characteristics of these data, and (3) some types of systems you will encounter. This should help you get started, regardless of what specific setup your organization has.

13.3 Types of Data Systems

Over time, the data and data systems concerning employees have evolved to match with the general employee lifecycle. Rather than call out specific vendors or designs, we would like to orient you to the type of data via the topics these data and data systems usually store information about.

13.3.1 Finding, Selecting, and Onboarding New Employees

The first thing every new employee goes through is the selection and onboarding process. For most medium and large companies, this entire part of the lifecycle is managed by two or three systems:

Candidate Relationship Management (CRM) System—this is the software to help manage relationships with potential employees in the talent market before they become official applicants. Many recruiters spend a lot of time at college campuses, job fairs, conferences, and other events building up a healthy pool of applicants. Candidate relationship management tools help manage all these data.

Applicant Tracking System—ATS's are the most common type of prehire software and manage the lifecycles for filling jobs. You can think about the purpose of this software from two angles. First, the angle of the applicant. Once someone applies to a job, they are often stored in the system semi-permanently using an ID like their e-mail address or a random ID assigned to them. One person may apply to many jobs at once, or at many different points in time. An ATS helps manage an applicant's interface with the available jobs at a company.

The second is from the perspective of the job. Recruiters partner with business leaders and HR to fill a job. This requires review of several applicants who move through "the funnel:" 100 online assessments may become 50 applicants may become 10 prescreens may become 5 phone interviews may become 2 live interviews to become 1 hire. ATS software helps all the employees who touch this process manage their portion of the funnel.

Onboarding Software—once a candidate is selected, many things need to occur. The new employee usually has to sign an offer letter, clear a drug screen and background check, provide documentation of eligibility to work, receive a company e-mail address, get loaded into the HR and payroll systems, receive instructions about their first day, have a badge created, order hardware (like a laptop), and other things. Onboarding software helps manage these many moving parts to get new employees all set to start on their first day.

These three types of systems are best thought of from the perspective of how they organize data and who they organize data for:

By person or by job? When you think about prehire data you are usually either thinking about people or about the seats they sit in. These sorts of systems usually have data structured this way. They will have a unique ID for each person, as well

as a unique ID for each job. Furthermore, the job is usually IDed at least twice: once uniquely by HR for the purposes of management structure and once uniquely by finance for the purposes of budget and payroll. These types of software perform complex processes and their data can get quite complex as a result. And though this means they can do pretty amazing things, at their core they basically boil down to those main identifiers.

Applicant or Recruiter? These software systems only have several different views/presentation layers to make themselves beneficial to relevant user groups. They must be usable by applicants or at least efficiently connect to other software that applicants can use. The design and quality of these interfaces often impact the data that analysts get from the back end or from reporting. For example, an interface which asks applicants to provide information about their education history may provide a drop-down selection of universities or may choose to leave those fields as totally free form. The former will give relatively standard university data requiring minimal cleanup, whereas the latter will likely provide data which will require extensive cleaning to be useful.

Conversely, internal employees like recruiters, hiring managers, and HR personnel have unique user interfaces so they can click the right buttons and move applications and job openings along the path from posting to hire. The design and quality of these interfaces, as well as the governance and auditing of how they are used have a significant impact on what kind of data is available and its quality.

13.3.2 Managing Employee Records

The central system used for managing employee records is referred to generically as an HR Information System (HRIS) or a Human Capital Management System (HCM). The idea is essentially that once you convert someone from not-employee to employee, you have a responsibility to keep track of them. HR typically wants to know who they are, where they live, how much to pay them, what bank account to deposit their paycheck into, how to tax them properly, what job they do, who their manager is, when they hired them, who reports to them, and many other facts. All this information must be available at any given time, but also must be tracked *over* time (e.g., what it is now compared to what it was 5 years ago?). This information is used for very practical purposes, like payroll, benefits, processing a promotion, or change-in-name after a marriage. However, it is also used for strategic purposes, like workforce planning (e.g., how many salespeople do we have or need in Atlanta) or talent management (e.g., how quickly do people get promoted?).

This system is also usually connected to many other systems. It connects to prehire, finance, compensation, learning, operations, and many other systems either because it has to give information to those systems or receive information from those systems.

For data wrangling, this system will likely be the place to start for most projects as virtually every other HR data system starts or ends with a connection to this system.

13.3.3 Managing Performance and Schedules

Systems outside of HR are often utilized to track performance or productivity. These can be very industry-specific—a call center, a retail store, and a manufacturing plant are all going to quantify the performance of their employees very differently but will all likely have software of some kind to do it. Similarly, and especially in traditional frontline jobs, there will also be time and attendance software which manages work schedules, vacation, leaves of absence, and other such data.

From an HR analytics data wrangling perspective, you can think about these data two ways. First: as a person or group of people. An analyst may want to look at performance or attendance for a given retail store or an individual over some period of time. Second: time—an analyst may want to examine a period of time for a given metric or metrics. As such, data is often designed to be tracked this way.

One consideration for these types of systems is that they are often not cleanly connected back to the HR system. For example, an HR Information System might use a Personnel ID (e.g., 87247661) whereas a performance management system may use a Sales Rep ID (e.g., E7640-A532). From a data perspective, they are both random strings of characters, and if an analyst needs those data connected, they may have to find or build a way to link them.

13.3.4 Measuring Sentiment

Sentiment data is almost always managed by a third party to protect confidentiality for employees. The vendors who do this often offer a combination of three different services. First, they will build and help deliver surveys to employees. Second, they will curate benchmark data from similar companies so that you can compare yourself externally. And third, they will provide an interface for accessing your data.

The challenge with wrangling this type of data is that usually the interfaces and contracts put in place to protect confidentiality prevent data scientists from getting data at the individual level. This means that in many cases the data team can only get aggregated results, such as the average of a team or the count for a whole region. This prevents almost all types of machine learning which are usually built off of individual-level data. If you need sentiment data as an input to advanced analytics or machine learning, you will likely have to create a partnership with the vendor to deidentify or cipher the data so that it can be analyzed without violating the privacy of employees or simply use aggregated data within the privacy guidelines of your organization.

13.3.5 Managing Pay and Benefits

Labor is one of the costliest aspects on a business' profit and loss sheet. We do not like to think of people as overhead per se, but in the world of dollars and cents how much a company pays its people accounts for a very large percent of the dollars managed. As such, HR, Compensation, and Finance departments work together to ensure that all the people information maps back to all the finance information and ensures that the right people are receiving the right dollars from (and to) the right bank accounts.

These systems are as big and complex as HRIS or HCM systems and are equally as important. That said, they are usually managed from a very different angle. HR and the business tend to think about buckets of work: "Ryan has a team of 50 and they all work on the Blue Widget Account." Who gets what job, promotion, and recognition is all managed by Ryan and the people who report to him.

However, in the Pay, Compensation, and Finance systems "Ryan" does not exist from a data perspective the way he does in an HRIS. Instead, there is a financial entity called a "cost center" that Ryan is responsible for. In this case, it might be labeled "Blue Widget Cost Center." This cost center is essentially a bucket of money that Ryan uses to run the operations of his team.

Why is this distinction so important? Often times, labor dollars do not sit neatly within these centers. Ryan might need help from the Green Widget Team, and so agrees to pay for a person on that team. Now, the data has a "Green Widget Team member" who is "paid for" by the "Blue Widget Cost Center." This will create discrepancies in areas like headcount reporting where the financial view of the organization differs from the managed-by view.

This concept gets complex, especially in a very large or very matrixed organization (i.e., where there are a lot of cross-functional dependencies to get work done). It can also be complex because finance and HR are often looking at different outcomes, so their key performance indicators will differ. For example, a finance team may count interns as "half-of-a-head" because they only work part time, whereas HR will usually count them as a full head. Both these rationales make sense: finance may be looking at "dollars out the door" whereas HR may be considering "who is sitting in our building" or "who is a potential internal candidate for a job." The differences between HR and finance views of employee data abound, so when wrangling these types of data, it is important to create strategies to reconcile differences so that data models are valid, reliable, and make sense to stakeholders.

13.3.6 "Other" Systems

There are many other systems you may ultimately want or need to source and transform data from, which will all have their own special considerations. A few you may come across are:

- Learning Management System: Tracks access to, and activity within, learning and development content
- Travel and Expense: Tracks and analyzes employee expenditures
- Facilities: Tracks access to specific buildings, floors, and/or rooms and typically works with data contained within physical tokens (like a badge).
- Systems Security: Tracks access to software systems, LAN or VPN access, and other IT-related security
- Software Utilization: Many types of software use licenses to track who has access to specific software, who is using it, when, and how much.

An additional characteristic of these types of data systems is that regardless of what type of data they contain across the employee lifecycle, the system can have unique characteristics as well. These are not formal distinctions, but may be helpful for you to think through as you learn about the data systems in your organization:

13.3.7 Aggregated Systems Versus "Systems of Record"

As data science and IT get more advanced, it becomes easier and easier to duplicate, move, and store data in many different places. The bigger the organization, the more often this occurs, and when wrangling data you must ask yourself: "how did these data get here?"

In HR, all data starts with an input of some kind. Sometimes data is system generated, but it always ties back ultimately to the action of a user somewhere. The closer the data is to that input, the more likely your data comes from a system of record. This term can be used as the "truest," "rawest" form of data and is often inaccessible to most users, but it is also the least edited which often makes it most desirable to the data scientist.

Conversely, much data can be moved around, combined, and stored in aggregated systems to make access easier to a broader audience—these are sometimes called "data lakes" or "data warehouses." Say, for example, Nathan wanted to track headcount fluctuations and he knows that this is the product of three things: hiring, current headcounts, and turnover. So he builds an "aggregated" system which ingests data from multiple systems and then builds reporting from that data for his own usage. Then Ryan discovers he needs the same data. The advantage of this design is that the data is now all in one place and easy to get. The disadvantage is that in order to deliver what he needed, Nathan had to make assumptions to source and transform these data for his original purpose. If Ryan wants to use Nathan's data for a different purpose, he must first ensure that all the wrangling Nathan did aligns with what he wants to use it for. This is a critically important consideration when wrangling data which has "already been wrangled" before you get to it.

13.3.8 Vendor Systems

When we introduced sentiment data we discussed how vendors often have ultimate control over some of our data. This is important to understand during wrangling so you understand the process and privacy considerations of sourcing information.

Another main consideration is the concept of benchmarking. Vendors want to help their clients get a pulse on the market and want to do it in the most valid and reliable way possible. As such they curate huge amounts of information and offer it as a service to data teams around the world in an effort to help them "compare" themselves to their industry, geography, or other benchmarks.

However, when ingesting benchmark data for the sake of descriptive or advanced analytics in HR, there are two extremely important questions to ask yourself:

1. How are they defining the variable? Example: "Engineers"
   How you define "Engineering" at your company is likely not exactly how the vendor defines it. This is because the vendor must collect this information from dozens or hundreds of other companies and create one version of engineering. This requires generalization that you as the data team must be comfortable with. This will never be perfect, but make sure you ask the questions around what is, and is not, included in your benchmarking samples to ensure you are comfortable creating a fair comparison between your data and the benchmark.
2. How complete is their sample?
   Vendors only have access to data from their clients and data they purchase or share. What they may call an "industry" or "regional" sample may or may not represent what you think would be a complete sample. Ensure you ask who and how big these samples are so you are comfortable that the sample is broad enough to provide a valid benchmark.

13.3.9 De-centralized Systems

Sometimes even the same systems are not managed together. You may have a vendor who manages a type of data but may store it in completely different data systems based on region, country, or even business unit. These divisions are often made for financial, legal, or privacy reasons, but nonetheless can impact how much data you can get from a given system and how aligned data could be from system to system. For example, it would be easy to assume that if you use the same survey vendor across your company that all the data will link together easily. However, you may find that the company is managing entirely different servers in New York compared to Seattle compared to London. Ensure you ask these questions early, so you are not surprised later in the model building process.

13.4 Data Formats

As we talked about in Chap. 5, the language of these data systems is ultimately tables. And while columns, rows, values, and primary keys are the functional reality of these systems, the way that these systems are set up to ingest information and then ultimately provide you access to them can vary greatly. When wrangling data, it is very helpful to understand some of these attributes as well as their pros and cons.

13.4.1 The Spreadsheet

Though the accessing of HR data for most situations is done through HRIS applications, it is not uncommon for local teams to store key HR data in spreadsheets.

- (Pro) Exceptional flexibility and accessibility: If your data is small enough to fit in a spreadsheet it is very easy to work with, even if your computer science skills are limited. It can also be stored and accessed very easily.
- (Pro) Ease of sharing: Spreadsheets are small in size and can be shared between teams and parties very easily. This makes them a very desirable format to pass back and forth between teammates and stakeholders
- (Con) Security risk: Because they are so easy to pass, they are tough to secure. Passwords can be helpful, but ultimately the advanced security technology you can find in an HRIS or even data visualization software simply does not exist in most spreadsheet programs. This includes things like approved users, row-level security, and field-level security which allow only certain users to see certain data based on predetermined authorization levels.
- (Con) Data quality: Because it is flexible and easy to pass, it is difficult to maintain data quality. Backend systems usually automatically track changes and allow for documentation of changes, but spreadsheets are not automatically set up this way. You most usually need to manually build these features into a spreadsheet and take great care to ensure version control and ongoing data refreshes and quality in order for spreadsheet data to stay current and correct.

13.4.2 Text Files and .CSV

Another common source of HR data outside of HRIS systems is text files. If you have ever used a text file, you might think there is no way they can be useful for data—they are even more unstructured than spreadsheets. This is true, they can be difficult to wrangle and derive meaning from. However, they can be useful because they are small from a file size perspective, and do not have the row limitations that spreadsheets have. This makes it an attractive way to pass data, even though once you receive it you must do extra work to make it usable.

One way you will often come across text-formatted data is in the form of .csv files. CSV stands for "Comma-Separated Values" and is a very efficient way to export and ingest data. If you think about a typical spreadsheet, it has rows and columns. Well in a .csv file, each row is on its own line, but the values for each column are separated by a "," by default. You can set that value to nearly anything you want. Other common selections are "|", ";", or tab.

Once you know that every value is separated by the same character, the parse function in a spreadsheet program or other software will "chop up" the data into unique columns.

13.4.3 Survey Data

An increasingly common type of data in the workforce outside of HRIS systems is survey data. Surveys are routinely leveraged in many parts of the business to gather sentiment and feedback from employees and customers. The questions are typically behavioral in nature and attempt to quantify motivation, opinion, loyalty, engagement, and other affect-based concepts.

Surveys are important because they are the only practical way to get large scale information about the constructs that exist inside the minds of employees. There is no HRIS table that can label someone as "engaged." Tenure, job code, performance, and others are most often a reasonably objective concept that can be labeled independently of the employees' "opinions" on the matter. But many others, which have much wider construct chasms (see Chap. 10) do not—engagement, organizational commitment, faith in leadership, manager quality, etc. Surveys help us understand these types of data but come with their own considerations. In fact, survey methodology is quite an advanced science. You can take an entire graduate-level course on the topic or even get a Ph.D. in psychometrics. In lieu of that, we want to review some basics for you to keep in mind.

– Survey data is "Self-Report"
  There are certain considerations with self-report data which you must keep in mind if you want to ensure your usage of data is appropriate. Other texts get deeper into the challenges here, but essentially self-report data is a scaled-up version of opinion, which makes it susceptible to bias and inconsistency. This does not make the data bad or unusable but should be used with those assumptions in mind. For example, some areas of self-report measurement simply have consistently higher or lower baselines. In compensation, it is possible to objectively have the most aggressive practices in the industry, but this area is rarely rated highly at the company level. Social desirability bias is another example—depending on corporate and geographic culture, the sentiment results received on engagement surveys can be skewed due to the desire to be seen positively. Self-report data are important in the world of HR, but they are different from observable evidence and so datasets built from them must keep those differences in mind. This is also a cause to take extra care to ensure appropriate sample size when using survey data.

– Take time to think about design
  The design of surveys is important. There is significant science on all kinds of survey design attributes and considerations, from faking to anchoring to survey fatigue to double barreling and many others. The aforementioned field of psychometrics is dedicated to psychological measurement, and the creation and validation of assessments is a big part of it. The point here is not to provide significant guidance in this space, but simply to understand that when wrangling survey data you should be thinking not just about the availability of the data, but also how well the survey was designed in the first place. It could have a significant impact on the usefulness of the data.
– There is a difference between "confidential" and "anonymous"
  Often used synonymously, confidential and anonymous are actually different terms with distinct (and sometimes legal) implications. An anonymous survey means that there is no way to know who took the survey, whereas a confidential survey means that the identity of the survey takers is significantly guarded but is still technically discoverable. Almost all employee surveys are confidential, not anonymous. This is because (1) if an employee says something legally or ethically extreme (e.g., identifying real danger, threat to physical well-being, or other illegal activity), then the company and survey vendor may have a legal and ethical obligation to act and (2) because it is simply far more feasible to manage hundreds or thousands of surveys if you have an unambiguous identifier.

That said, these are rare cases and so for all intents and purposes, the identity of survey takers should be highly protected. For a data wrangler, this can be problematic. When high levels of confidentiality are employed, the value of survey data can be very limited and as we said before can often only be used for descriptive analytics. Feature engineering usually requires individual-level detail if it hopes to be used as a predictor variable. However, as organizations begin to realize the shortcomings of this approach and the potential value in the data, many are loosening the grip on confidentiality without compromising security. For example, data can be anonymized before it is handed to the analytics team so that it can be used without divulging individual responses.

13.4.4 Big (Passive) Data: Work Application, Social, and Unstructured Data

As e-mail, instant messaging, and collaboration tools become the dominant mechanisms for employee communication, organizations are generating huge amounts of data on how employees interact and work together via the applications and internal social networks they use. Though the datasets are large and enter the big data realm in size and complexity, they have high potential for generating meaningful insights.

As discussed in Chap. 8, organizations are beginning to tap into the potential of using this data through statistical approaches like Organizational Network Analysis (ONA).

Along with work application data, this idea of large amounts of data collected passively as the byproduct of other work has enabled practitioners to investigate unstructured data.

Unstructured data is data which does not arrive in rows and columns like the data we have talked about in the majority of this book. Text is the main type of unstructured data that HR researchers work with. Other forms of unstructured data include video and audio recordings. Below are examples of how this type of data is being used in HR:

- Sentiment Analysis: Natural language processing (NLP) is a technique that is used to analyze and derive meaning out of unstructured text data. One NLP technique commonly used in HR is sentiment analysis which is frequently employed in HR to assess whether the tone of text is negative or positive. This can beneficial when analyzing text feedback provided on an employee survey or interpreting social media comments.
- Video Recorded Interviews: In the same way that computer vision machine learning is learning to have cars drive themselves and robots navigate in real space, we are now starting to be able to pull useful insights from video recording candidates. Body language, tone of voice, eye contact, and other things that have long been indicators during face-to-face interaction are beginning to be quantified and used to create a more objective view of candidate quality.
- Retail or Floor Movement Analysis: Efficiency in work environments matters. Where employees and customers spend their time, how much movement it takes to complete certain tasks, and other efficiencies can be greatly improved with machine learning as a guide. Ingesting and analyzing video feeds of work environments can begin to help drive these work outcomes to improve both worker health (e.g., same task, less physical work) and businesses (e.g., improved productivity).
- Speech-to-Text: In order for NLP to do analysis to get meaning from unstructured text (bag of words, sentiment, word counts, tfidf, encoders, etc.), sometimes audio data must be transcribed to turn it into structured features for ML models. Often this is for sentiment analysis (see above), but can also be used for general digitalization efforts (e.g., call routing). As this technology improves, it also stands to impact areas like coaching and performance management (e.g., coaching bots).

13.5 Moderately Advanced Techniques: Binning, Lagging, and Z-scores

Now that we have talked about where you might want to wrangle data from and a bit about different data formats, we would like to return to a few practical techniques which might be helpful when assembling your data. As we said earlier, there are dozens and dozens of techniques to learn if you wish to become a master data wrangler, and we will not go too deep here. We have selected three which we think (1) have broad application to many potential scenarios and (2) are reasonably easy to understand and execute.

Please note, if deep details of wrangling techniques are not for you, skip this section. However, if learning how to get deeper into the descriptives of your data is interesting, here are the three techniques which you may find useful while exploring.

Binning is the process of taking a continuous variable and turning it into discreet categories. If you have a performance metric for sales representatives called "sales dollars," it is probably equivalent to a dollar value for how much a rep sold in a given period of time (like each month). You can explore that variable as continuous, or you could create "bins" which group the values into discrete ranges. You might want "high," "medium," and "low" or "Top 5%," "bottom 15%," and "middle 80%." The point is, it is often valuable to take a continuous variable and chunk it up so groups of similar employees can be analyzed together. And in today's world of advanced computing, there are programs and code which will even help you bin your data automatically.

Lagging is the art of pairing data at different points in time. We talked about how rows of data represent a "case" or observation of an event. But what if the effect of data in one row influences the value of data in another row? How can we see the effect? Let's stick with our sales example:

Imagine retail operations has a program to pair high performing representatives (e.g., top 5% in sales dollars per month) with low performing employees (e.g., bottom 15% in sales dollars per month) - eligible employees can volunteer to mentor and struggling reps spend a whole month shadowing the more successful representative. But you hypothesize that the positive influence of this training will not take effect in the first month of training, but instead take a few months to have an impact, since new skills take practice and reinforcement. So that means our data may not "line up" correctly:

Rep ID	**Month**	**Peer Training**	**Sales Dollars**
85212325	August	**Y**	$4573
85212325	September	N	$4621
85212325	October	N	**$5432**

In this sample data, the effect the training is having (represented by "Y") does not have an influence on performance until 2 months later (represented by the higher sales dollars). Lagging is a data transformation technique to move the data around so you can see this effect within the same record:

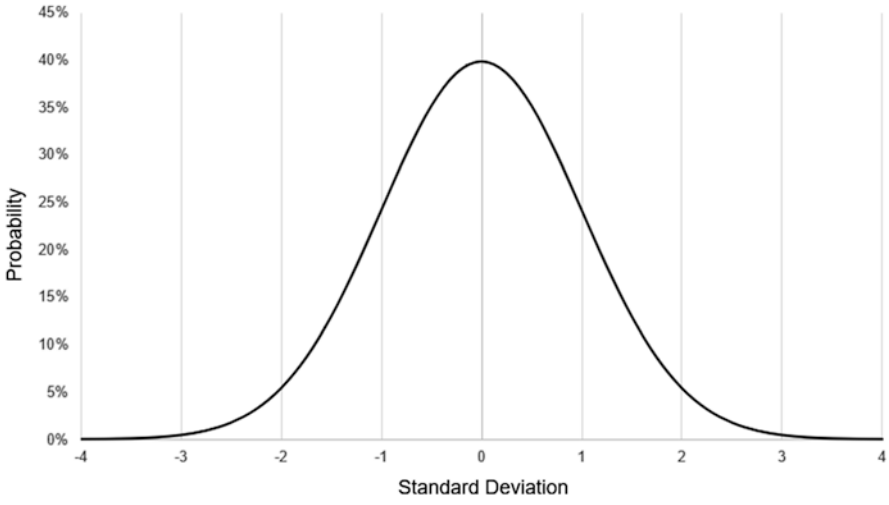
Rep ID	Month	Peer Training ***-2 Months***	Sales Dollars
85212325	August	N	$4573
85212325	September	N	$4621
85212325	October	**Y**	**$5432**
85212325	November	N	$5551
85212325	December	N	$6192

Lagging allows the researcher to see relationships they might otherwise miss.

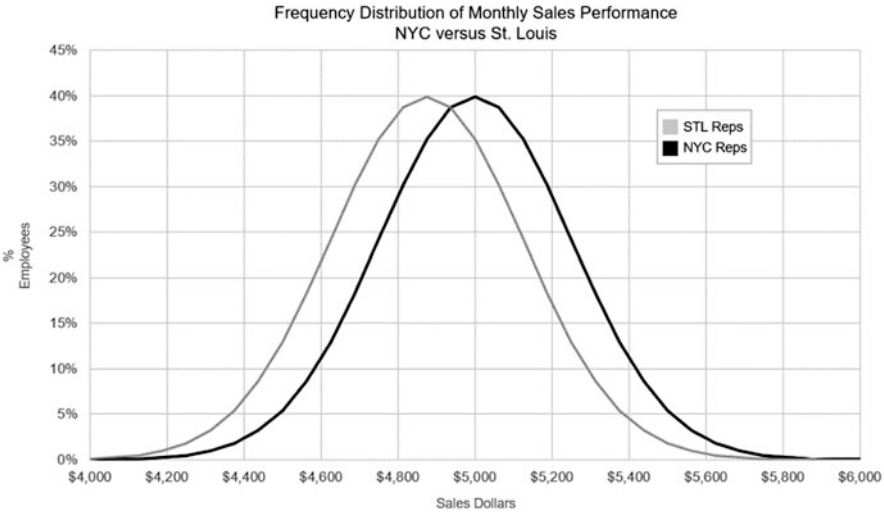
Finally, a *z-score* is a way to transform a metric so that it can more accurately be compared to other metrics which might not be on the same scale. For example, sales reps in New York City and Los Angeles probably create more revenue than sales reps in Des Moines or St. Louis. Higher foot traffic and higher product pricing will influence how much revenue a rep can produce.

Could we then say NYC sales reps are "better" than St. Louis sales reps? From an absolute revenue perspective, sure, but that does not really get at the human performance angle of what you are trying to quantify. If you took the location out of the equation, how would they look?

Earlier when we talked about normal curves, we showed a visual that described the curve as a function of its mean and standard deviations, like this:



What a z-score does is make the average score in your dataset a 0 and the standard deviation 1. This way, you can compare metrics that might not have the same baseline. Let us looks at our retail example:



You can see that on average NYC reps perform better, but we assume this is largely because of the market they are in. How could I compare a St. Louis rep to a NYC rep? Z-scores help by using the following formula to transform all the data points:

$$\text{Z} - \text{score} = \left(\text{score} - \text{Mean}\right) / \text{Standard Deviation}$$

Breaking this down, by subtracting the mean from the score we end up with how much bigger or smaller the score is when compared to the average. Then, dividing that number by the standard deviation tells us how big that difference is compared to the general variance of the dataset. Let's get a little more specific.

The NYC bell curve has an average of $5000 and a standard deviation of $250. The St. Louis bell curve has an average of $4875 and a standard deviation of $250. Jason works in St. Louis while Jennifer works in NYC. Last month Jason sold $5500, while Jennifer sold $5400. Who did better?

$$\text{Z} - \text{score}_{\text{Jennifer}} = \left(\$5{,}500 - \$5{,}000\right) / 250 = 2.00$$

$$\text{Z} - \text{score}_{\text{Jason}} = \left(\$5{,}400 - \$4{,}875\right) / 250 = 2.10$$

The z-score tells us is that Jason performed 2.00 standard deviations above the mean of his city, while Jennifer performed at 2.10 standard deviations. This indicates that, when controlling for the baseline differences in their respective locations, Jason actually did "better."

The great thing about z-scores is that they can be used on all kinds of populations. Anything you can create a legitimate average and standard deviation for can be z-scored. Level, location, tenure, job type, and others are great ways to be able to

"compare" populations that might have different baselines which makes comparison on the absolute values of metrics misleading.

Z-scores can also be used to combine *metrics* with different baselines to create indices. An index is when you summarize multiple metrics to create one metric. An example could be "sales rep performance," which might be more than just sales dollars. Let's say we want to combine sales dollars, closed sales, and Net Promoter Score into one number for "rep performance."

The problem is that sales dollars range from ~4000 to ~7000, while closed orders range from ~100 to ~300 and NPS range from −100 to 100. We can't just add them up because sales dollars is so much bigger than the other two and NPS exists in the negative range.

Since z-scores turn everything into an average of 0 with a standard deviation of 1, we could transform these data into z-scores and then add them up to create the beginnings of an index.[1]

One note about z-scores is that they do not change the distribution of your data. If your data has skew or kurtosis, the transformed dataset will also have these characteristics. It also does not change the variance. So, if you are going to combine metrics you have to make sure that they have similar distributions, as well as doing other descriptive analyses to ensure you are properly choosing and weighting your variables. Consult your analytics or data science partners if you want to use z-scores to transform data for the purposes of an index.

13.6 Tools

Data wrangling can be performed using a variety of tools, though some tools are more efficient than others. Below are a handful of common tools you may come across.

13.6.1 Microsoft Excel

Excel has long been a tool of choice for data cleanup and analysis for those without programming skills. Excel is a very flexible program and can be used to quickly analyze a dataset and create visuals to better understand patterns and trends. It is limited however in bridging multiple datasets efficiently and in a repeatable way.

An additional advantage of Excel is that it is typically the least common denominator for end-user groups. Most employees who work with computers have at least a rudimentary familiarity with the Microsoft suite, and so it can be a very effective platform to share information and data through.

[1] Note: Creating an index is a complex process which requires analysis of collinearity, latent variables, and other considerations. If you want to create an index, consult with your analytics team.

For advanced users, over the years Excel has also evolved to include add-ons which make it more powerful. Examples include Power Query, Power Pivot, and general integration of Visual Basic for features like custom macros.

In many ways for the average HR Analyst, Excel is like a favorite spatula or screwdriver—It's comfortable, easy, and works in 80% of situations. And while it has its limitations and those with programming skills often do not understand its appeal, it is a powerful and flexible GUI tool for data transformation.

13.6.2 GUI ETL Tools (Examples: Alteryx; Trifacta; Tableau Prep)

There has been a recent boom of GUI software packages designed to assist with data manipulation and curation. Many still require limited programming skills; however, there is an increasing number of low-code / no-code GUI ETL tools that, as the name implies, require minimal programming skills. Anyone who can point, click, and use basic formula logic can use them. Extract, Transform, and Load (ETL) implies the data wrangling part—these tools are built to manipulate data tables in advanced ways so they can be prepared for reporting or further analysis. Even if you are a programmer, if you are getting into the advanced reporting, advanced statistics, or machine learning game you should absolutely get familiar with these programs for two reasons:

First, they broaden your collaboration base. If you are one of—or the only—person on your team who can truly work in a SQL or other programming environment, you will eventually bottleneck advanced analytics work. These programs allow less technical folks to contribute and still integrate with all your favorite advanced environments.

Second, the outputs are self-documenting. Since they leverage a visualized canvas approach to drive their flow, it is very easy to share, communicate, and debug across groups. What they give up in the efficiency of pure code, they more than give back in transparency.

13.6.3 Data Visualization Tools (Examples: Power BI; Tableau; Qlik)

You might ask yourself, "what does data visualization have to do with data wrangling?" In principle, it should not be here, but virtually all data visualization and data exploration tools have data manipulation fields built into them. Most, if not all, transformation and prep should happen before data gets to this stage, but you may find yourself in a situation where these tools have features you need. Just ensure you do not rely on their transformation abilities too much, since they are typically too far downstream to be sustainable.

13.6.4 R and Python

Of the free tools available for machine learning, two have emerged as the forerunners. The first is R, which is software developed and tailored specifically for statistics and visualization. Though it can be used for other purposes, at its core it is a language designed for mathematical models such as machine learning. Long the preferred choice for academics, R has many features that distinguish it and make it a good choice for machine learning. A major advantage of R is that there are numerous free libraries (CRAN) available for it that are designed for machine learning, visualization, and statistics. Much academic work uses R and many academics have built packages in R.

R is an interpreted language that is typically used via a tool called R Studio. As an interpreted language, one is able to create data objects and models and interact with them without having to restart the program with each command. This allows for easy exploratory data analysis. R also supports a feature called vectorization where commands can be executed easily on a multielement data object without having to use programming techniques such as loops that you would have had to use in traditional languages. This makes it easier to work with data for cleaning and model preparation.

As we mentioned, R is also very good for visualization. With libraries like ggplot2, visualization in R is integrated with its other components like data frames. The results are clean, powerful, and aesthetic. Many academic publications use visualizations created in R.

The other major free language used for machine learning is Python. Embraced by many for its simplicity, robustness, and ease of use, Python has overtaken R as the primary language for pure data science. Python is more versatile than R and used for many other purposes including web application development. Python, like R, is also an interpreted language. There are many different integrated development environments (IDEs) for Python including Jupyter Notebooks, Spyder, and PyCharm.

Python also has an extensive set of machine learning libraries though the primary libraries are part of the Scikit-Learn package. It has similar structures to R's data frames in the popular Pandas library. The data frame in Pandas is excellent for data wrangling and EDA. For visualization, the popular base package for creating charts is matplotlib. There are also many other additional libraries available such as Seaborn for advanced graphics.

Python is embraced for its clarity of language. It is clean and esthetic and eschews many of the traditional elements of prior languages such as the use of brackets to define structure. In its simplicity, it is easy to use and learn. Python also has a deep set of free libraries that rivals R's libraries for data science.

A key advantage to Python over R is that it can be used to create other important data science functions such as REST APIs. It can also be used to code machine learning on big data platforms such as Hadoop and Spark (Spark-ML).

Discussion Questions

1. Why is data quality a challenge with HR data? What are some ways to combat this challenge and what are the associated risks?
2. What is the difference between validity and reliability? Create two examples of how each might manifest as a challenge in HR data.
3. Name four different types of HR data systems. Explain how data is typically structured for that type of system.
4. Choose two different data types and enumerate the advantages and challenges of each.
5. What are some advantages of binning, lagging, and z-scores? Provide an example of when you might use each.

Chapter 14
Bringing Your Model to Life

This chapter will complete the three-part cycle review of Appreciate-Assemble-Adopt by providing an overview of the main steps and considerations when building a machine learning model and implementing it at your organization. This final chapter will provide a summary of the key steps and objectives of the model building phase and then focus on the considerations that will be important when transitioning from design-and-build to communicate-implement-maintain.

Prior to the model building phase, expectations for how the model will be used should be discussed and agreed upon with stakeholders. That is, the plans for how the output of the model will be translated into actions should be clearly defined. To frame this idea, go back to the different categories of research from Chap. 5: Exploratory, Constructive, and Empirical. The foundation of model building lies in part in understanding the nature of the problem you are trying to solve.

This is an essential step early in the project management process and is more accurately a part of the Appreciate phase. Then why bring it up here? Assuming the decisions have been made about the purpose of the model early on, it will now be time to begin actioning against those decisions. For example, when a model is designed to explore, specific predictions are not necessarily needed. Rather the model will be used to better understand the overall factors that are associated with a particular outcome. The actions that the business would typically take when using a model in this manner would be to try to identify and then influence the top factors instead of responding to specific predictions. Many of the techniques we reviewed in Chap. 9 have the capability to provide details on the primary factors that drive decisions. Choosing which technique or techniques best fit the combination of the data available and problem-to-solve should be done with this exploratory goal in mind.

Another use of a model is to make predictions that will drive specific actions. The intent here is usually for the model to receive new data on a regular cadence that it will use to make predictions. These can be individual predictions (e.g., "Giana has a 40% chance of being promoted in the next 6 months, but a 75% chance to term in

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1_14

that same window") or aggregated predictions (e.g., "the Pacific Northwest Region will experience 174 terminations in Q3, plus or minus 5%"). In these cases, the business will want to use the ongoing insights of the model to create actions (maybe Giana needs a "stay interview") or inform decisions (how does the PNW region need to flex their staffing model to handle the impending turnover). Though it may be insightful to understand the factors that drive the model and the reasons for specific predictions, the business may be more interested in actionable information as opposed to a causally informative model.

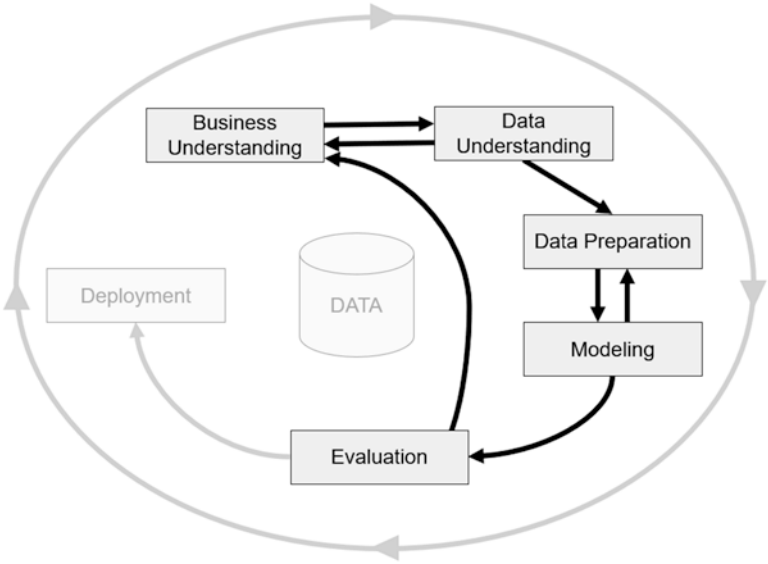
Recall from Chap. 9 that machine learning methods vary in the amount of transparency they provide into how they work. Some algorithms, like linear regression, show the calculations and are therefore readily available for post hoc[1] analysis which can provide details about how it makes its decisions. Other algorithms, like neural networks, are often "black box" which means they provide little or no information on how they arrive at the predictions that they generate. In some situations, predictions without explanation may be sufficient for the needs of the business. However, if the details are required on why a particular prediction was made, only a subset of machine learning algorithms will provide that degree of transparency without the use of a secondary tool like SHAP. This part of model assembly is where you must make these choices as it will greatly influence which techniques you have at your disposal.

14.1 The Model Development Lifecycle

Appreciate-Assemble-Adopt is not a waterfall-type progression. Model development is an iterative process that occurs in parallel to, and in partnership with, the data wrangling stage. It is informed by decisions and research performed much earlier in the discovery phases. Rarely is there a serial transition from data wrangling to model building because as models are developed, new questions and ideas arrive based on early findings or unanticipated roadblocks so additional or different data sources must be sought out to improve performance. Further, though the selection of the predictors is technically part of the model development stage, many of these features are derived from data sources that are typically investigated and created during the data wrangling stage.

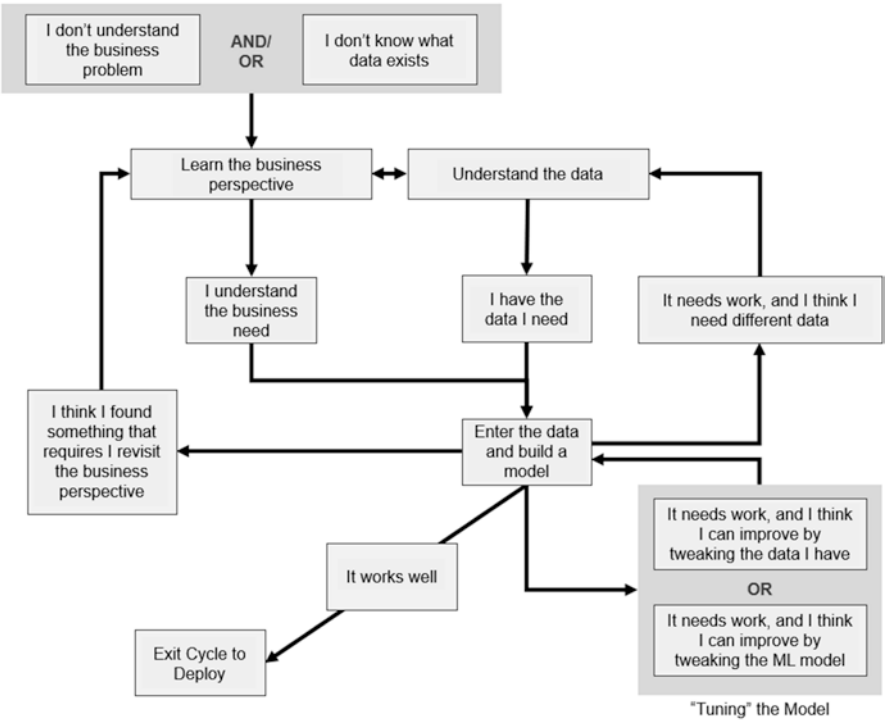
This iterative concept is clearly articulated in the visual of the CRISP-DM project management model:

[1] "Post Hoc" literally translates to "after this" and refers to all the "analysis after the analysis." This term is often used to identify digging into results and even doing more research after the initial analysis is complete.

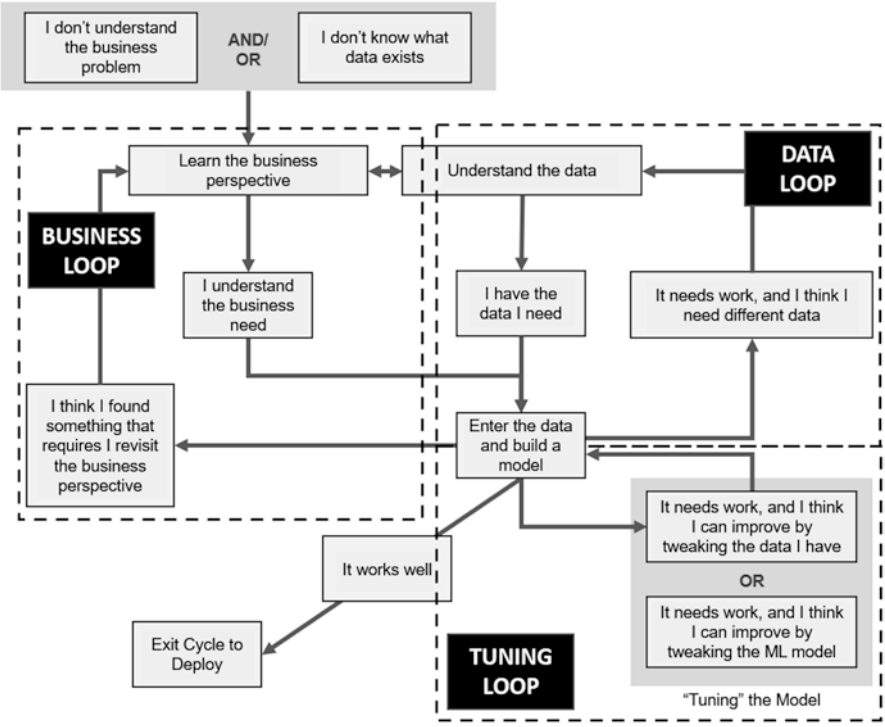


The bidirectional arrows are intended to show these fundamental feedback loops between understanding the business and understanding the data as well as between having the data you *think* you need and producing a model that works. The final loop between Evaluation and Business Understanding adds a final checkpoint to enable movement from a model that is not working yet back to Business Understanding.

Because of all these loops, it makes more sense to think about model development as being executed in a continuous circle until requirements are met. In the first step of the loop, the goal is to *appreciate* the problem and the data. Second, you must *assemble* a model by building and tuning it. All the information and learnings you get along the way lead to new approaches, new features, new model configurations and they can all feedback into future iterations of the development phase, restarting the cycle. With each iteration, the intent is that the model will improve or move closer to the goals established for the business. To illustrate more specifically, we have taken the general 5-stage approach of CRISP-DM and added our own slightly more detailed interpretation:



Looking closely at this flow chart shows that all of machine learning model development boils down to just three feedback loops, with one exit:



	Starts with	Go back if
Business loop	Appreciating the business perspective and stakeholder needs.	Something during the build shows you that you need additional information, or you do not understand the business perspective as well as you thought you did.
Data loop	Appreciating the data landscape and capabilities.	You realize during model build that you have insufficient data, unclean data, or additional sourcing needs in order to create an effective model.
Tuning loop	Building a prototype model.	You realize that the model is not performing well enough, but you have what you need; you just need to iterate with what you have got.

By leveraging these three loops an analyst will eventually arrive at the part in the lifecycle when it is time to end model development. This is as much art as it is science—on one hand, if the business' targets are technically met, the team may be tempted to mark the model as complete and move on to deployment. On the other hand, based on what has been discovered along the way the team may strive to produce the most effective model that it can within the prescribed schedule and budget constraints. In each case, this will be a balance of producing what was asked yet considering the factors which may influence the long-term sustainability of the model's validity, value, and the effort to maintain validity and value. The ideal is

rarely either extreme. A good data team will inform the business leaders of the new things they discover that will impact the business problem and the model's sustainability and reach a compromise of what additional steps should be taken compared to the risks of not taking those steps. This may even require an update to the business case (see Chap. 12) so that new insights and risks are well-documented and communicated.

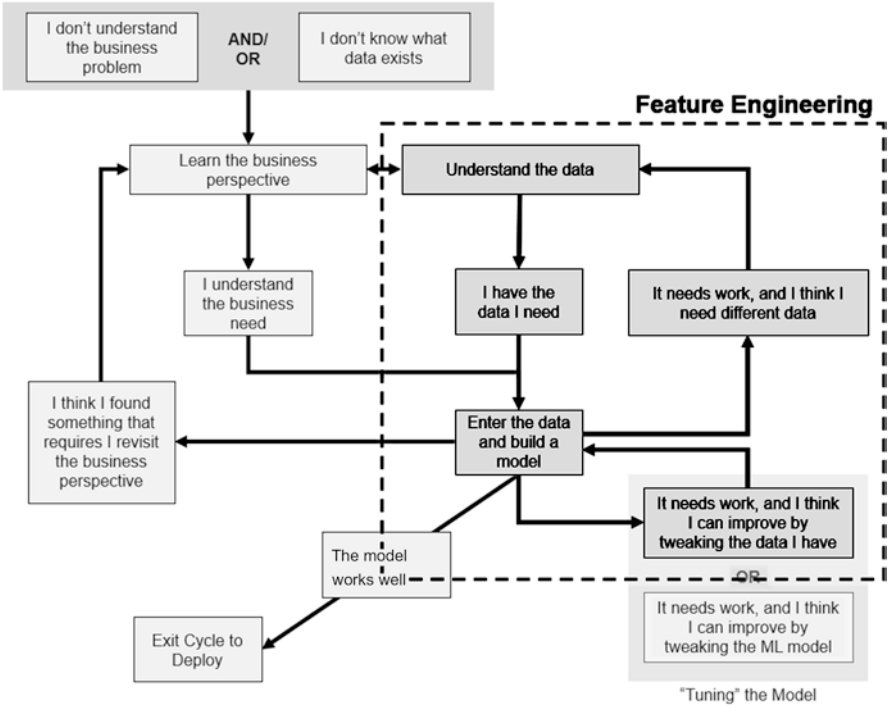
Sometimes the team will get stuck in the loop because they are struggling to produce a model that meets requirements. Just like knowing when good enough is good enough, knowing when to shut an effort down is important too. Having invested significantly in the development of a model can make it difficult for the team to hedge or pivot away from the original goal. However, it is important to identify the limitations of the data or algorithms and realize that sometimes an effort will not produce sufficient results. As we said earlier, failing to produce a result that meets expectations is not unheard of for machine learning efforts, although in almost all cases the team and business partners can use the effort to inform future projects and can often even use pieces of the data and business case to apply to future work.

In this same vein, it is very important to not force deployment on a model that does not meet accuracy or quality standards. If the model is not performing and the business is willing to lower the model accuracy standards, the team should evaluate the differentiated impact on the business. This revisiting of the business case should be thoroughly discussed and well-communicated with all stakeholders. This will prevent delivery of something that meets the expectations of other stakeholders, but unintentionally violates other stakeholders' expectations.

14.2 Feature Engineering

Throughout the book, we have mainly used the term "predictors" when talking about the input data for a model. At this point, we would like to transition to the more commonly used term in machine learning, "features." The difference between predictor variables and features is essentially semantic. That said, it does make sense to point out that when working in machine learning "features" will more typically be used in lieu of "independent variable" or "predictor variable."

With that difference in mind, "feature engineering" becomes a much more understandable term. Feature engineering is essentially the art and science of designing what data goes into a machine learning model. From the previous section, the data loop with some help from the tuning loop is primarily dedicated to this concept:



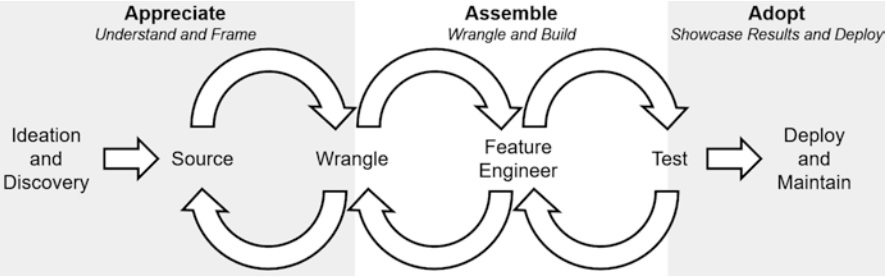
Drawing the lines here is an oversimplification because in order to do feature engineering well, a data scientist must understand all the feedback loops discussed earlier. This means that early homework to gain insight through a combination of business acumen and understanding people/behavioral data, and application of those insights during the technical build will always accompany these data-oriented loops.

That said, feature engineering is the most critical place for all four parts of the Analytics Ikigai to work in harmony. It starts with business acumen—understanding the problem, the audience, and the different potentials for solutions. It then moves to statistics and research methods—how to turn what is known about the business into an answerable, researchable question and execute with a sound methodology. These two steps are overlaid with people/behavior: how does behavioral theory influence how the team attacks the problem and potential solutions, and how does sound HR practice contribute to ensuring the business problem is solved in an appropriate way? And then finally, the computer science: how does all of that translate into the ingestion, validation, and engineering of good data that can go into a model?

Of course, we know it is not serial—there is much iteration and feedback looping between the domains. But when deciding what is going into a model, the analyst and team will have to draw on all four areas to do it optimally.

Admittedly, the first iteration of feature engineering may take a very broad-brush approach (i.e., get as much data as is reasonable). Most often the team goes this way during data wrangling in order to cover all bases. This is like saying, "I'm not sure what I want to cook yet, so I'll just make sure my fridge is full." It is significantly easier to decide what you *do not* want to cook with, than having to go back out to the grocery store when you realize you do not have an ingredient you need.

That said, there are still many steps between fridge and plate—feature engineering is all about this work. If this sounds like data wrangling, in some ways it is. Data wrangling and feature engineering go hand in hand. Data wrangling is the data-sourcing side of getting data ready for a model whereas feature engineering is the building, tuning, and testing side. It can be thought of like this:



Wrangling and feature engineering are just the front and back ends of model assembly. Yin and yang working together to go backward to data sources and then forward to testable models. And similar to cooking, feature engineering can be thought of as two basic types:

- Selection (what ingredients to use)
- Transforming (what do you need to do to an ingredient before it can be used)

Selecting which features to include is probably the single most influential part of whether a machine learning algorithm is going to provide the insight intended. That said, it is also nuanced and requires iteration—it is unlikely the first attempt or first combination of features selected will yield the best possible outcome. Also, it would take a whole book to go through all the aspects of selection during feature engineering (and indeed, there are entire books written on the subject). For the purposes of this introductory text, we would simply like to provide three tips when beginning this journey.

Tip 1: Reduce Noise

Earlier in this book we talked about concepts like collinearity, dimensionality reduction, principal components, latent variables, and the general concept that not every data point provides unique differentiated value to your model. This is the part of the process where that theory becomes reality. When you are wrangling your data and choosing your features, you must get intimately familiar with what data you are going to feed your model, which things are telling a unique story, and which are introducing noise or confusion to the model. This art is a combination of (1) understanding the business and behavioral theory and (2) understanding the stats.

The business and behavioral aspects are critical because they will tell you what things make sense to go together. If you are looking at turnover, a good understanding of business might tell you to look at performance metrics or compensation. An understanding of behavioral theory would point you toward manager quality and organizational cultural metrics. Neither will tell the whole story on its own but understanding what drives an outcome of interest will tell you where to start.

After pulling features in and doing preliminary analysis, the statistics will tell you more about how they relate. During EDA (Exploratory Data Analysis) you and the data team can see which features complement each other to drive more predictive validity compared with features that overlap and do not add incremental value. The advanced statistics behind this exploration is not fit for this text but is something a data scientist can help with. Methods like factor analytics, analysis of covariance, and even machine learning in an exploratory way (e.g., clustering) can help here.

Tip 2: Be Smart about the Number of Features

How many features you ultimately select for your model will be a function of (1) how much data you have available, (2) how explainable your model needs to be, and (3) which machine learning methods you employ. Some algorithms are not negatively impacted by a large number of features, whereas more traditional models like logistic regression can suffer from too many features and features that are related or correlated.

A related question when considering this tip is "how explainable does my model need to be?" As a general rule, less transparent methods (e.g., neural networks) tend to be the methods more comfortable with larger amounts of features. This relates back to whether you will be able to explain how a model works, which is sometimes a critical attribute of success, especially in HR. Keep in mind the balance between how many features are in your model and the risk of overfitting or making the model too opaque to explain.

Tip 3: Sometimes you must go backward to go forward

As you engineer features, you may discover that your data is not structured in a way that will quantify what you need. In this case, sometimes a solution can be to (1) create new features from existing data, (2) go back to find data that you did not originally source, or (3) generate data that does not currently exist.

For example, in our previous example of business context informing attrition reasons, the business may feel manager instability[2] is a driving factor. Raw transaction data is not set up to tell us this information, so an entirely new feature must be created to quantify this concept. Creating a feature that identifies the number of unique managers that an employee has had in a given period of time can then be analyzed to see how it can contribute to a predictive model.

From a sourcing perspective, exploration into a model may illuminate gaps in the data that, if filled, will make your model more effective. For example, it is quite common that while doing exploratory analysis to discover a trend for a certain

[2] In this case, "manager instability" means a given employee has had a large number of managers in a short period of time.

group (like a geography or job code) you will discover the need for a follow-up analysis. This insight then leads to new data requirements to fill and zoom in on the initial insight, and this sends the data team back into data sourcing to bring forward the new required information. Sometimes this is a matter of additional data mining but can even enter into the realm of data creation through methods like surveying or data transformation.

Transforming features is the other major aspect of feature engineering. In Chap. 13, we talked briefly about different techniques which could be used to transform data. As we said, we will not get into the details of all the possible data transformation techniques in this book—that is for more in-depth texts on data science or computer science.

However, we do want to provide an overview. From the machine learning perspective, engineering your features with transformation is when you are changing the structure or format of data so that it fits better in your model. During wrangling, you transform to make the data initially usable. The goal may be cleaning, validating, imputing, standardizing, or another outcome so that your data is ready for the initial modeling steps. During feature engineering, the techniques are the same, but the goal is typically more nuanced. Your data should be clean and valid at this point, but now you must make more specific decisions about the structure of your data. Here are some of the examples from earlier which compare wrangling and engineering:

Technique	Wrangle example	Feature engineering example
Mathematical Transformation	My performance data is seasonal, so I would rather see how far above average sales each month is. Therefore I will standardize the sales metric using z-scores.	It looks like my study is well fit for linear regression. However, sales dollars across tenure are not linear. I will apply a log transformation to make my data more linear.
Binning	I need to use engagement as an independent variable, so I will transform survey scores into three groups: high, medium, and low.	It appears that three categories do not work too well—too many outliers. I think I will bin using five categories instead: Very high, high, medium, low, and very low.
Free-form cleanup	I changed the values "RU," "Rutgers," "Rutgers U," and "Rutgers New Brunswick" to "Rutgers University," so they can all be tabulated together.	Seems I do not have enough people from every school to analyze school-by-school. Let's transform all schools based on zip code. Everyone in Rutgers is now coded as "Northeast Universities."
Append, Merge, Filter, or Join	I need to create a dataset with data from (1) employee records, (2) the performance management system, and (3) the learning management system. I will use their Personnel ID to bring the correct fields in from each and create a dataset with one record per employee.	Turns out the model did not work with what I pulled in initially. I wonder if the model would work if I only look at training from the last year? Or if I only look at a particular region?

Tweaking and tuning your features in this exploratory way can surface data and trends that would not normally be available by simply putting wrangled data into a machine learning algorithm.

14.3 Testing Your Model: Training Sets and Cross-Validation

The final piston in the iteration engine of model development is testing for performance. As with all parts of development, models should be evaluated regularly throughout the development cycle—so it is not really a serial process. But as we talked about, the iterative give and take overlaps with its previous section, feature engineering.

The first step in assessing is model fitting. Model fitting is the process of feeding data to the model to train the algorithm. In supervised models, the data should include both the features (predictors) and the outcome (labels in classification models or values in regression models).

When training a model with the data you have available, you want to take one of two approaches to training. The first approach can be thought of as a Train/Test Approach. In this approach, you only feed the model part of your data—this is called the Training Dataset. The model learns from these data how to make the predictions you want to make.

The rest of the data is the Test Dataset. Once the model has learned how to make predictions from the Training Dataset, you feed it data it has never seen before *without* the outcome data and see if it can predict accurately. The typical Train/Test data split is 80/20 with 80% reserved for testing.

This approach is used because when training a model, it will do its best to predict with what you give it and as a result almost always predicts the training data better than the test data. This is because the model has already seen the training data and has been tuned to predict it. The Test Dataset is so important since it represents "the real world" - the model has never seen it before and therefore can be used as a proxy for how the model will perform with data in the future.

The second approach to fitting is called Cross-Validation. In Cross-Validation, the data is divided into "folds," which are simply slices of data. The most common number of folds used is 10. The model trains itself on nine folds and tests itself on the remaining fold. Then, the model cycles through iteratively and trains through all the data, holding out a different fold each time. This allows the model to use "more data" to train on, but still assess on novel data each time around.

14.4 Beginning Adoption: The Results Review

Once the problem has been Appreciated, and an effective model has been Assembled, the next step is to Adopt the model. Adoption of a machine learning model has similar steps to the implementation and change management strategies of other sorts of projects and processes you may have experienced in HR. As such, one of the most common first steps is communicating results to stakeholders. This most usually takes the form of a formal presentation where the team reviews (1) what was planned, (2) the activities, and (3) the results.

The primary goal during this review is to seek approval of any deliverables developed, which will be different depending on the intention of the project. It could be as limited as a bulleted list of insights with data supporting them, or as comprehensive as a suite of dashboards, reports, and reengineered processes which need to be implemented.

The secondary goal is to align stakeholders on recommendations for next steps. A well-managed project (machine learning or otherwise) will have had transparent and regular communication with stakeholders throughout the process, so there should be no surprises when results are shared.

Third, this may also be one of the very few opportunities to get the working team in front of the most senior stakeholders for the project. Therefore, this meeting can also be thought of as an opportunity to have a comprehensive review (i.e., get the experts in the room to answer all the questions) as well as an opportunity for recognition (i.e., use at least part of the time to show appreciation for the team's efforts).

Finally, unfortunately not all projects end with a usable model. If the team is struggling to build a model to meet the business' needs, or the project requirements are fundamentally shifted or canceled, this meeting may instead be the culmination of previous conversations with stakeholders about the challenges so that the leaders and sponsors of the project can make a formal decision about how to proceed.

To prepare for the results review (or whatever your organization calls it), there may be formal templates that should be used to prepare for or document the results for this meeting. If not, there are general guidelines to keep in mind. And just like the business case, these guidelines follow the project across time and topics to effectively tell the story. However, in this scenario the information is more retrospective than prospective:

	Business case *What we are going to do and why*	Results review *What we did and what happened*
Summary	Executive summary	Executive summary
	Business drivers	Business drivers *Update*
	Scope	Scope *Update*

	Business case *What we are going to do and why*	Results review *What we did and what happened*
Review		Approach
	Assumptions	Assumptions
	Timeline	
	Costs	
	Benefits	
	Risks	
Results		Results
		Final deliverables
		Recommendations/next steps
		Lessons learned

The results review will start with a summary of what was originally agreed to in the business case. In many ways, this communication will have already been done during your business case, but you should take the time to update the materials to reflect the stakeholders' new level of familiarity with the project: industry context, organizational challenges, and other things should be covered at a high level. That said, do not dive too deep or oversell—the project has already been completed and the stakeholders in the room are likely familiar with the bulk of this review material.

The most important part of the summary is to review significant things which have *changed* in the overall business drivers or scope. If something changed in the business or industry which caused the project to pivot its purpose or outcomes, or the scope grew, shrunk, or changed for some reason, these are important details to review as they will have impacted the rest of the model development and project overall.

After Summary, you want to Review. In the business case, this was dedicated to ideas, assumptions, timelines, and watch outs. We design machine learning business cases this way because machine learning projects are often amorphous at this stage—we are not sure we understand all the data or the solution yet.

By the time you get to the results review, many of these categories have been handled, so now is the time to show leaders what you did and what is still outstanding. A good way to think about this part of the presentation is by splitting it into two domains: Approach and Assumptions.

Approach: In your business case, your approach was all prospective ideas and considerations and took the form of proposed activities. Now, you want to tell your stakeholders and leaders how it went. With all those things in mind from your business case, what did you actually do? What steps did you take? How did you mitigate the risks? What were your big exploratory discoveries and how did they shape the ultimate choices for the model? This is the time to sell what you did and why it worked. If you are reviewing an unsuccessful project, this is the time to talk about all of those same things, but instead use the story to frame the challenges, explain how you attempted to mitigate them, and ultimately decided to pause, redirect, or cancel the effort.

In this part of the presentation, it is also important to remember not to get too technical. Discussing specific parameters, tuning methods, specific literature, methodological considerations, and other highly detailed aspects of the project are best left out. Remember to keep the discussion at an appropriate level for the audience but take the time to organize supplemental materials or appendices in case you need them for reference.

Assumptions: Most times, your model will work but only for certain scenarios, for certain populations, or when certain conditions are met. It is important to explain these to the stakeholders and leaders. For example, assume a predictive attrition model is able to predict 80% of people who leave voluntarily within a 6-month window, but it also produces a large number of false positives. This means that if the model says you *will not* leave, it is probably right. But if the model says you *will* leave, sometimes it guesses wrong.

In this scenario, the business will not take aggressive measures on *all* those who the model flags as at risk (since it sometimes gets those wrong). Instead, the business should take an extra step to cull that list down and focus on individuals who are both flagged as at risk and who would have a high impact of loss on the business. And, since there is a higher false positive rate, the intervention should be more general, like taking the form of a career path discussion or stay interview.

There also may still be risks or outstanding concerns which need to be addressed. For example, impending industry or organizational changes, challenges with generalization, or other limitations or uncertainties might be important enough that they need to be reviewed at this time, and likely will be cause for follow-up at a later date when those risks have had an opportunity to reveal themselves and be mitigated.

After Summary and Review, demonstrating the results and effectiveness of the model is the main event. This demonstration is going to vary based on the nature of the deliverables, but regardless of what the outcomes were, it is at this point you must explain in stakeholder-appropriate terms what the results were and how they have been translated into deliverables.

The performance of the model should be compared to the targets outlined in the business case as well as how the model will be used. Specifically, metrics such as accuracy, specificity, sensitivity, confidence intervals, and others should be reviewed and discussed in business terms. It is important to ensure that the business understands the value of the model, but also the realistic limitations of its use.

Though it may not be required for the results review, in many cases the forecasted or realized return on investment may be reviewed. ROI is often expressed alongside metrics like timelines, costs, and risks, so may fit better in those sections, but is often a critical key performance indicator for a model (even if at this point it is still a theoretical or forecasted return).

Next, demonstration of value should naturally transition into a discussion of recommendations, next steps, and lessons learned. If the team was not able to produce a model that meets the business requirements, then this section should still be a

discussion of next steps, but more focused on lessons learned and how the project intends to pivot to provide additional value or shut down.

If the model works well enough to warrant implementation, then part of the discussion should be about the incoming change. Depending on the model's place within the overall project and the nature of the deliverables, this can be anything ranging from a full review of change management and implementation strategy to a simple brief review of technical steps required to operationalize the deliverable. This section may detail how to implement organizational change, training, and process to deploy the model, and/or review sponsorship needs from those in the room. These needs might also be part of a broader implementation plan if the model is part of a larger project. The two major points you must get across to the stakeholders and leadership are (1) what is next and (2) how do we get there?

Finally, sometimes appended to these later sections is a summary of lessons learned. As with the earlier sections, this should be a high-level summary of the key learnings, not an exhaustive list of items. For HR projects, this will often include observations on the following topics:

- Data quality/cleanliness
- Data availability and governance
- Sustainability (how much time and effort it takes to produce useful data)
- Benchmarking insights (internal and external)

This is a rare opportunity to share with leadership a small amount of detail about data infrastructure and governance; however, ensure to keep the insights, learnings, and suggestions audience appropriate. It is a valuable opportunity to help illuminate a sometimes-significant need, but if you go too deep or push too hard, it can have the opposite effect and confuse or frustrate leadership.

14.5 Executing Adoption: Deploying Your Model

Once the results of model development have been presented to stakeholders and approved (and any follow-up questions/issues have been resolved), the model can be deployed to production. As with most adoption steps, planning for this should begin early in the project with essential details worked out before the end of model development. The level of effort required to deploy a model will depend on a variety of factors including the complexity of the deliverables as well as the amount of process and system change needed to implement it.

This book is not intended to be an exhaustive playbook on change management, however we do want to spend time talking about model deployment and major areas to consider when rolling out a model. Implementing a machine learning model is not solely a function of sourcing, cleaning, and analyzing data followed by the communication of insights. As we talked about at length in Chap. 11, the organizational considerations are equally as critical to success. A model that makes accurate predictions is of little value if the results are not used to inform decisions or actions by

the business. Planning for how the model will be used is, in many ways, as important as model development itself. To simplify the consideration landscape, we have boiled it down to two main categories:

- Organizational Considerations
- Technical Considerations

Organizational considerations are essentially principles which can be reduced to the domain of good change management. Technical considerations are principles applied to the technical aspects of project management. Often called "requirements" when working with IT, they include all the considerations required to stand up and maintain any technical effort or piece of software.

Likewise, we have boiled down the types of implementation into two main categories:

- Models creating something new
- Models creating change

Adoption can be thought of in two ways. First, models which create something from nothing. Sometimes a model will help a new process come to life or cause leaders to think and decide on things they have never thought about (at least officially) before. Other times, a model will be aimed at *improving* a business process or decision. And while both are technically a "change," we want to differentiate how a user will see the difference.

Taken together, these four principles can create a 2 × 2 matrix which can generally guide your adoption strategy:

	New	Change
Organizational	"My model is creating insights or a process which people have never seen before. I must help users become comfortable *using something they have never used*"	"My model is changing the way we execute a certain type of decision or process. I must help users become comfortable *transitioning to this new way of doing things*"
Technical	"My model is bringing data together which has not been together before. This creates *novel infrastructure which will require incremental resources and process* to support"	"My model is utilizing existing infrastructure in a new way. We may need small amounts of additional resources, but *almost all of what I need is already available*"

Admittedly, this is an oversimplification of change and project management and if you are working on an effort with significant impact organizationally or technically, we encourage you to do thorough change and project management to ensure all your bases are covered. However, thinking about the implementation this way will help you break down the many moving parts of the work into a few questions you should keep in mind:

- Does my whole project fall into one part of the matrix? If not, which parts fall in which sections?

- How will my end-users and technical partners perceive/receive the different parts of my model and deliverables?
- What is in it for them? Why should they be interested? What, if anything, is creating their desire for change?
- Do they have the information and skills they need to be effective?
- Do I have materials and processes in place to provide support to end-users and to create accountability to ensure adoption?

In a machine learning project, a key place to tie this matrix back is during deliverable design. Especially when the outcome of a model is linked to a dashboard or report, it is important to keep in mind how the model's data and insights will be created by technical partners and then consumed by the end-user. Keeping in mind how this will make their life easier or harder is paramount in doing effective change management.

From the technical perspective, deploying a model can be thought of in two ways:

1. System Integrations
2. Person-Hour Resource Requirements

Models rarely exist in a vacuum without dependencies on other applications or data sources. At minimum, the model will need to be fed data to make predictions. This typically requires that data is extracted from existing systems and loaded to a location where the model can access it. For models that will make regular or continuous predictions, the data pipelining will likely need to be automated or semi-automated.

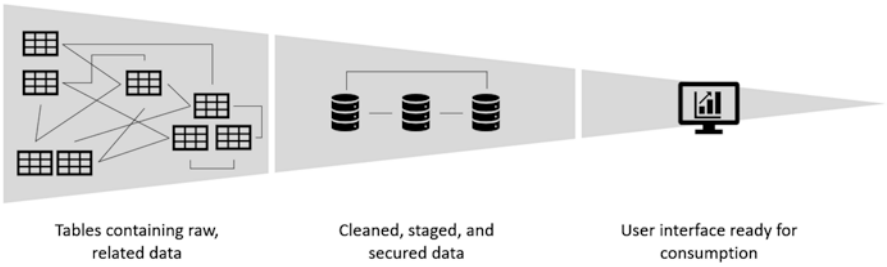
Think about this as "automating the outcomes of data wrangling and feature engineering." When doing these Assemble processes, the steps are creative, exploratory, and iterative. When they are complete, you are left with a discrete set of steps, processes, and logic which transform raw data feeds into ingredients prepped for your model. You must create an infrastructure to maintain this flow for as long as the model will exist.

When you build it, this work is handled by you or the project team. To do this work after deployment, you must obtain person-hour resources. This equates to partnering with other technical teams such as the data owners and/or the IT department. The technical teams can provide resources to build a solution and provide ongoing support to establish any data feeds needed by the model. As with software development, solutions should be evaluated for the balance between cadence, effort, and long-term supportability. The more complex and manual the data, the more costly to support in terms of person-hour resources.

This part of the process should be tied very closely to the Appreciate step where we learned about our data. The feasibility and ultimate longevity of a model will rely on the ability to feed it good data. It is best to partner with data owners and IT support teams early and understand their perspective so that your integration services work for all parties involved.

This logic goes for front-end and back-end systems. Remember back to Chap. 3 when we talked about employee data and where it lives. If your model is producing

a dashboard or report for end-users, you will have IT resource requirements for the back-end, transformation, and presentation layers of the model:



In smaller companies this may be the same team (and your team may even be expected to participate). In larger organizations, these may be entirely different departments! It is important to learn about and engage these teams in the Appreciate stage so that you can create a strong partnership and ensure the model's needs are met without creating undue stress on the teams you partner with.

14.6 Maintain Your Model: Integrity over Time

Machine learning models, like software systems, require care and maintenance to support them after the projects that developed them are completed and closed. The main reason to actively manage models is to maintain the model's accuracy. It is inevitable that over time machine learning models will lose their efficacy and must be retrained or refreshed. Remember from Chap. 8 that a model's ability to make accurate predictions assumes that the past looks like the future. And while all models rely on this to some extent, rarely does the world remain the same for an indefinite period of time.

To do this well, we want to give you two major tenets to keep in mind when maintaining your model, each with a few tips to help you keep your models running smoothly:

- Be Proactive
  - Define and maintain model ownership
  - Have a schedule and expectations
  - Be on the lookout for improvement opportunities
- Watch for Erosion
  - Ensure input integrity
  - Stay current with business expectations
  - Effective models change behavior, but not always for the better

"Be proactive" is good advice in general. In machine learning, it means that you need to stay ahead of the work that needs to be done to keep your model operating smoothly. Just like maintaining a piece of equipment like a car, models need regular attention. And just like physical equipment, if you wait for the signs that it is failing (or about to fail), it will usually be much more expensive (time and money) to fix.

Define and maintain model ownership: It is important to define who will own which pieces of a model. Support of simpler models may come from one team, but complex models with integrations will typically be managed by an HRIS or IT team or even across multiple teams. This is especially true of models that require high availability from users (e.g., reporting updated in "real time") or that support critical decisioning. Due to the technical nature of automated models, round-the-clock support is not typically provided by a workforce analytics team alone, though they may be the first stop for quality assurance from the IT owners.

Clear documentation is an important part of this support system. Data ingestion procedures, delivery dates, auditing, and service level agreements (SLA) for support when things break are all critical to have well-defined. This is most usually a true partnership between traditional IT, analytics/data science, and business owner teams. IT may support the production implementation of a model, but the analytics/data science team would likely be an escalation point along with key business users as subject matter experts. Items like off-hours support will depend on the criticality of the output of the model and should be discussed and agreed upon early in the project.

Having a schedule and expectations is another important way to be proactive. In organizations with mature production environments, support and deployment will likely be closely managed and coordinated with an operations team and with their cadence. Changes to production systems required to deploy or update the model may even be controlled with changes requiring approval from a CAB (Change Approval Board), and/or changes being restricted to specific time windows. Even if your organization does not manage models with this sort of rigor, it is still important to decide when and how often your model can be updated. This ensures that the model is not perpetually "in development" which can be a resource drain with very limited return. Many models will take a traditional software approach and schedule "releases," or specific time windows when the model can be improved. Between release dates, the team will fix "bugs" (i.e., demonstrable ways that the model is broken or not working properly) but will simply log "enhancements" (i.e., things we want to do better but are not truly broken). This helps control workload and testing volume for the maintenance teams.

Along with setting improvement expectations, one must also continually monitor the model for performance by setting metric expectations. Clearly defined metrics and targets should be established for the model as part of an audit process. When metrics deviate too far from targets, the analytics team should get involved to investigate. And from the business perspective, these audits should include testing by business users to ensure the predictions are continuing to meet expectations.

Another way to say this is that model performance should be measured using both traditional technical or statistical model performance terms (e.g., sensitivity,

specificity, etc.) and with business metrics. The stats and technical testing show high-level performance and back-end stability, whereas business metrics tie to the behavior that the model output is intended to influence. This shows continued applicability to the problem. For example, with an attrition model, the business metric could be turnover of high performers. Though the model may perform well by making accurate predictions, it's important to also assess the impact on the business. If the model does not help reduce attrition of the targeted employees, it is not meeting expectations.

All of this expectation management is really an effort to *be on the lookout for improvement opportunities*. Even if a model is meeting its targets and performing to expectations, it should be proactively tuned and nurtured. This can be as simple as a minor tweak or as significant as a complete refresh. That said, model maintenance must also be prudent to avoid fixing things which are not broken or are performing adequately. There is a balance between watching for ways to improve a model, but not wasting resources overengineering it.

Said differently, performance should be monitored for signs of deterioration in accuracy, while also keeping an eye on smart ways to get better. Sometimes minor changes to the data pipeline can have a significant impact on model efficacy. Other times, new data gets captured on employees and those additional factors could improve model performance. Either way, staying up to date on how these changes implicitly impact the model or provide opportunity to improve are great ways to keep a model relevant and performing.

Another smart way to maintain your model is by always *watching for erosion*. In machine learning, a model is only as good as (1) its input data and (2) its ability to represent the reality of the business. If data input is not good, it is cooking with bad ingredients. If it is no longer designed to represent the reality of the business, it no longer knows what its customers want to eat.

HR data are often subject to change, which means you must *ensure input integrity*. For example, if the rating system used for annual employee performance reviews is changed from one year to the next, the model can misinterpret the results. Changing the range of the scale or even the distribution of ratings is often enough to negatively impact a model that relies on performance ratings. The term for this shift in the input data is called data drift and its impact on model efficacy can be dramatic. There is an increasing number of tools and machine learning platforms designed to identify and monitor for data drift.

Other examples are (a) employee surveys, which often change questions, (b) compensation data which are often updated to meet market demands, or (c) headcount, which can fluctuate due to hiring or downsizing. Also, (d) structural changes to HRIS and other HR data systems should be considered. If your organization is moving from one vendor to another, the integrity of the data flow will be disrupted and maybe even compromised if the new system does not store or output data the same way as the old system. The point is, data input to your model is not static, it is always fluctuating with the needs of the business and needs to be watched.

Similarly to the input being dynamic to the needs of the business, the output is also dynamic. Some outcomes have stable definitions (like turnover or sales dollars), but sometimes even seemingly consistent outcomes change. You must *stay*

current with business expectations. This means that what the business finds important may change, and even if it does not, appropriate thresholds or norms can shift. For example, think about "normal" turnover in a strong versus a weak economy. What might be considered a good year in a strong economy (when turnover is typically higher), might be considered terrible in a weak economy (when turnover is traditionally lower). Models are built at points in time, and so the expectations they are designed to meet have to be continually monitored, so they can stay consistent with what the business needs from them.

Finally, when considering consistency, we must remember that *effective models change behavior, but not always for the better*. HR models are especially sensitive to changes because human behavior is complex, difficult to predict, and subject to numerous external influences that are not easily captured.

Sometimes this manifests as a model which improves its environment until it is no longer useful. For example, a model designed to identify struggling performers and help them get on improvement plans might be so impactful that the population it is trying to predict shrinks so low the model can no longer find them. This is a "positive problem" but will be something to consider when maintaining and tuning such a model.

Another concern with model performance is that if the employees that a model is trying to make predictions about become aware of how the model works, they can intentionally influence or "game the system." For example, think of a model which is built that predicts which employees have leadership potential. A key feature in predicting potential turns out to be completion of an overseas assignment. If employees become aware of this, they could seek out an overseas assignment to improve their score.

The key question then becomes, "is encouraging this behavior a good idea?" Specifically, do overseas assignments make employees better leaders OR is there an *underlying factor* that draws employees to overseas assignments that makes them better leaders (e.g., comfort with risk; ability to perform in unfamiliar situations, etc.). If it is the former, the model will be fine and it is wise to encourage overseas assignments as a developmental experience. However, if it is the latter, then as more leaders volunteer, that feature of the model will slowly lose its incremental validity. When maintaining models, these sorts of biases are key to monitor. Even if a model was built to minimize bias, there is always risk that bias may creep into the model later.

Discussion Questions

1. Explain the structure and importance of the Business Loop, the Data Loop, and the Tuning Loop.
2. How do feature engineering and data wrangling work together? Why is this relationship so important?
3. Explain the results review and how it is different from the business case.
4. Choose three of the most important considerations during model deployment. Why are these more important than other considerations?
5. Choose the three model maintenance techniques you think are the most important. Why did you choose them?

Afterward

Thank you for taking the time to read *Introducing HR Analytics with Machine Learning*. Over the three parts of this book, we have framed the need for advanced analytics in HR and demonstrated a model for you to use to maintain balance between the business, human, science, and data demands of making good decisions about employees at work. We have reviewed methods for you to think about and frame your questions to enable data and machine learning to solve problems, as well as introduced the basic statistics and machine learning techniques to use when solving people-data challenges. Finally, we introduced important considerations when starting down an advanced analytics path and provided guidance on how to actually get started using machine learning in an organization.

Machine learning in HR is such an interesting combination of old and new fields, soft and hard science, the subjective and the objective, and many seemingly polar opposites. We really enjoyed having an opportunity to dive into this messiness with you. While we believe that HR has much growth ahead of it to become a truly evidence-based industry, and data science is still learning how to operate in industries where "not everything that counts can be counted" (Albert Einstein), we are excited to continue to help close the gap. Thanks for beginning this journey, and we hope you join us on our mission to use analytics for the betterment of employees and their workplaces.

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1

Index

C. M. Rosett, A. Hagerty, *Introducing HR Analytics with Machine Learning*,
https://doi.org/10.1007/978-3-030-67626-1