Kush R. Varshney is a distinguished research staff member at IBM Research – T. J. Watson Research Center where he leads the machine learning group in the Foundations of Trustworthy AI department and co-directs the IBM Science for Social Good initiative. He has invented several new methods in the fairness, interpretability, robustness, transparency, and safety of machine learning systems and applied them with numerous private corporations and social change organizations. His team developed the AI Fairness 360, AI Explainability 360, and Uncertainty Quantification 360 open-source toolkits.

Trustworthy Machine Learning

Accuracy is not enough when you're developing machine learning systems for consequential application domains. You also need to make sure that your models are fair, have not been tampered with, will not fall apart in different conditions, and can be understood by people. Your design and development process has to be transparent and inclusive. You don't want the systems you create to be harmful, but to help people flourish in ways they consent to. All of these considerations beyond accuracy that make machine learning safe, responsible, and worthy of our trust have been described by many experts as the biggest challenge of the next five years. I hope this book equips you with the thought process to meet this challenge.

This book is most appropriate for project managers, data scientists, and other practitioners in high-stakes domains who care about the broader impact of their work, have the patience to think about what they're doing before they jump in, and do not shy away from a little math.

In writing the book, I have taken advantage of the dual nature of my job as an applied data scientist part of the time and a machine learning researcher the other part of the time. Each chapter focuses on a different use case that technologists tend to face when developing algorithms for financial services, health care, workforce management, social change, and other areas. These use cases are fictionalized versions of real engagements I've worked on. The contents bring in the latest research from trustworthy machine learning, including some that I've personally conducted as a machine learning researcher.

—Kush

Trustworthy Machine Learning • Varshney

Trustworthy Machine Learning

concepts for developing accurate, fair, robust, explainable, transparent, inclusive, empowering, and beneficial machine learning systems

Kush R. Varshney

Trustworthy Machine Learning

Kush R. Varshney

ISBN 979-8-41-190395-9

Kush R. Varshney / Trustworthy Machine Learning
Chappaqua, New York, USA

Cover image by W. T. Lee, United States Geological Survey, circa 1925. The photograph shows a person standing on top of a horse, which is standing on a precarious section of a desolate rock formation. The horse must really be worthy of the person's trust.

How to cite:

- APA: Varshney, K. R. (2022). *Trustworthy Machine Learning*. Independently Published. http://www.trustworthymachinelearning.com.
- IEEE: K. R. Varshney, *Trustworthy Machine Learning*. Chappaqua, NY, USA: Independently Published, 2022.
- MLA: Varshney, Kush R. *Trustworthy Machine Learning*. Independently Published, 2022.
- Bibtex:

```
@book{Varshney2022,
        author="Kush R. Varshney",
        title="Trustworthy Machine Learning",
        publisher="Independently Published",
        address="Chappaqua, NY, USA",
        year="2022"
}
```

“For it is in giving that we receive.”

—St. Francis

Contents

Preface

Decision making in high-stakes applications, such as educational assessment, credit, employment, health care, and criminal justice, is increasingly data-driven and supported by machine learning models. Machine learning models are also enabling critical cyber-physical systems such as self-driving automobiles and robotic surgery. Recommendations of content and contacts on social media platforms are determined by machine learning systems.

Advancements in the field of machine learning over the last several years have been nothing short of amazing. Nonetheless, even as these technologies become increasingly integrated into our lives, journalists, activists, and academics uncover characteristics that erode the trustworthiness of these systems. For example, a machine learning model that supports judges in pretrial detention decisions was reported to be biased against black defendants. Similarly, a model supporting resume screening for employment at a large technology company was reported to be biased against women. Machine learning models for computer-aided diagnosis of disease from chest x-rays were shown to give importance to markers contained in the image, rather than details of the patients' anatomy. Self-driving car fatalities have occurred in unusual confluences of conditions that the underlying machine learning algorithms had not been trained on. Social media platforms have knowingly and surreptitiously promoted harmful content. In short, while each day brings a new story of a machine learning algorithm achieving superhuman performance on some task, these marvelous results are only in the *average* case. The reliability, safety, security, and transparency required for us to trust these algorithms in *all* cases remains elusive. As a result, there is growing popular will to have more fairness, robustness, interpretability, and transparency in these systems.

They say "history doesn't repeat itself, but it often rhymes." We have seen the current state of affairs many times before with technologies that were new to their age. The 2016 book *Weapons of Math Destruction* by Cathy O'Neil, catalogs numerous examples of machine learning algorithms gone amok. In the conclusion, O'Neil places her work in the tradition of Progressive Era muckrakers Upton Sinclair and Ida Tarbell. Sinclair's classic 1906 book *The Jungle* tackled the processed food industry. It helped spur the passage of the Federal Meat Inspection Act and the Pure Food and Drug Act, which together regulated that all foods must be cleanly prepared and free from adulteration.

In the 1870s, Henry J. Heinz started one of the largest food companies in the world today. At a time when food companies were adulterating their products with wood fibers and other fillers, Heinz started selling horseradish, relishes, and sauces made of natural and organic ingredients. Heinz offered these products in transparent glass containers when others were using dark containers. His company innovated processes for sanitary food preparation, and was the first to offer factory tours that were open to the public. The H. J. Heinz Company lobbied for the passage of the aforementioned Pure Food and Drug Act, which became the precursor to regulations for food labels and tamper-resistant packaging. These practices increased trust and adoption of the products. They provided Heinz a competitive advantage, but also advanced industry standards and benefited society.

And now to the rhyme. What is the current state of machine learning and how do we make it more trustworthy? What are the analogs to natural ingredients, sanitary preparation, and tamper-resistant

packages? What are machine learning's transparent containers, factory tours, and food labels? What is the role of machine learning in benefiting society?

The aim of this book is to answer these questions and present a unified perspective on trustworthy machine learning. There are several excellent books on machine learning in general from various perspectives. There are also starting to be excellent texts on individual topics of trustworthy machine learning such as fairness[1] and explainability.[2] However, to the best of my knowledge, there is no single self-contained resource that defines trustworthy machine learning and takes the reader on a tour of all the different aspects it entails.

I have tried to write the book I would like to read if I were an advanced technologist working in a high-stakes domain who does not shy away from some applied mathematics. The goal is to impart a *way of thinking* about putting together machine learning systems that regards safety, openness, and inclusion as first-class concerns. We will develop a *conceptual foundation* that will give you the confidence and a starting point to dive deeper into the topics that are covered.

> "Many people see computer scientists as builders, as engineers, but I think there's a deeper intellectual perspective that many CS people share, which sees computation as a metaphor for how we think about the world."
>
> —Suresh Venkatasubramanian, computer scientist at Brown University

We will neither go into extreme depth on any one topic nor work through software code examples, but will lay the groundwork for how to approach real-world development. To this end, each chapter contains a realistic, but fictionalized, scenario drawn from my experience that you might have already faced or will face in the future. The book contains a mix of narrative and mathematics to elucidate the increasingly sociotechnical nature of machine learning and its interactions with society. The contents rely on some prior knowledge of mathematics at an undergraduate level as well as statistics at an introductory level.[3]

> "If you want to make a difference, you have to learn how to operate within imperfect systems. Burning things down rarely works. It may allow for personal gains. But if you care about making the system work for many, you have to do it from the inside."
>
> —Nadya Bliss, computer scientist at Arizona State University

The topic of the book is intimately tied to social justice and activism, but I will primarily adopt the Henry Heinz (developer) standpoint rather than the Upton Sinclair (activist) standpoint. This choice is

---

[1]Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. URL: https://fairmlbook.org, 2020.

[2]Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. URL: https://christophm.github.io/interpretable-ml-book, 2019.

[3]A good reference for mathematical background is: Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge, England, UK: Cambridge University Press, 2020.

not meant to disregard or diminish the essential activist perspective, but represents my perhaps naïve technological solutionist ethos and optimism for improving things from the inside. Moreover, most of the theory and methods described herein are only small pieces of the overall puzzle for making machine learning worthy of society's trust; there are procedural, systemic, and political interventions in the sociotechnical milieu that may be much more powerful.

This book stems from my decade-long professional career as a researcher working on high-stakes applications of machine learning in human resources, health care, and sustainable development as well as technical contributions to fairness, explainability, and safety in machine learning and decision theory. It draws on ideas from a large number of people I have interacted with over many years, filtered through my biases. I take responsibility for all errors, omissions, and misrepresentations. I hope you find it useful in your work and life.

1

Establishing Trust

Artificial intelligence is the study of machines that exhibit traits associated with a human mind such as perception, learning, reasoning, planning, and problem solving. Although it had a prior history under different names (e.g. cybernetics and automata studies), we may consider the genesis of the field of artificial intelligence to be the Dartmouth Summer Research Project on Artificial Intelligence in the summer of 1956. Soon thereafter, the field split into two camps: one focused on symbolic systems, problem solving, psychology, performance, and serial architectures, and the other focused on continuous systems, pattern recognition, neuroscience, learning, and parallel architectures.[1] This book is primarily focused on the second of the two partitions of artificial intelligence, namely machine learning.

The term *machine learning* was popularized in Arthur Samuel's description of his computer system that could play checkers,[2] not because it was explicitly programmed to do so, but because it learned from the experiences of previous games. In general, machine learning is the study of algorithms that take data and information from observations and interactions as input and *generalize* from specific inputs to exhibit traits of human thought. Generalization is a process by which specific examples are abstracted to more encompassing concepts or decision rules.

One can subdivide machine learning into three main categories: (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning. In supervised learning, the input data includes observations and labels; the labels represent some sort of true outcome or common human practice in reacting to the observation. In unsupervised learning, the input data includes only observations. In reinforcement learning, the inputs are interactions with the real world and rewards accrued through those actions rather than a fixed dataset.

---

[1]Allen Newell. "Intellectual Issues in the History of Artificial Intelligence." In: *The Study of Information: Interdisciplinary Messages*. Ed. by Fritz Machlup and Una Mansfield. New York, New York, USA: John Wiley & Sons, 1983, pp. 187–294.
[2]A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers." In: *IBM Journal of Research and Development* 3.3 (Jul. 1959), pp. 210–229.

The applications of machine learning may be divided into three broad categories: (1) cyber-physical systems, (2) decision sciences, and (3) data products. Cyber-physical systems are engineered systems that integrate computational algorithms and physical components, e.g. surgical robots, self-driving cars, and the smart grid. Decision sciences applications use machine learning to aid people in making important decisions and informing strategy, e.g. pretrial detention, medical treatment, and loan approval. Data products applications are the use of machine learning to automate informational products, e.g. web advertising placement and media recommendation. These settings vary widely in terms of their interaction with people, the scale of data, the time scale of operation and consequence, and the severity of consequences. Trustworthy machine learning is important in all three application categories, but is typically more pronounced in the first two categories: cyber-physical systems and decision sciences. In data products applications, trustworthy machine learning contributes to a functioning non-violent society.

Just a few years ago, the example applications in all of the categories would have been unheard of. In recent years, however, machine learning has achieved superlative performance on several narrowly-defined tasks across domains (often surpassing the abilities of human experts on those same tasks) and invaded the popular imagination due to the confluence of three factors: data, algorithms, and computation. The amount of data that is captured digitally and thus available to machine learning algorithms has increased exponentially. Algorithms such as deep neural networks have been developed to generalize well from that data. Computational paradigms such as graphical processing units and cloud computing have allowed machine learning algorithms to tractably learn from very large datasets.

The end result is that machine learning has become a general purpose technology that can be used in many different application domains for many different uses. Like other general purpose technologies before it,[3] such as the domestication of plants, the wheel, and electricity, machine learning is starting to remake all parts of society. In some parts of the world, machine learning already has an incipient role in every part of our lives, including health and wellness, law and order, commerce, entertainment, finance, human capital management, communication, transportation, and philanthropy.

Despite artificial intelligence's promise to reshape different sectors, there has not yet been wide adoption of the technology except in certain pockets such as electronic commerce and media. Like other general purpose technologies, there are many short-term costs to the changes required in infrastructure, organizations, and human capital.[4] In particular, it is difficult for many businesses to collect and curate data from disparate sources. Importantly, corporations do not trust artificial intelligence and machine learning in critical enterprise workflows because of a lack of transparency into the inner workings and a potential lack of reliability. For example, a recent study of business

[3]List of general purpose technologies: domestication of plants, domestication of animals, smelting of ore, wheel, writing, bronze, iron, waterwheel, three-masted sailing ship, printing, steam engine, factory system, railway, iron steamship, internal combustion engine, electricity, motor vehicle, airplane, mass production, computer, lean production, internet, biotechnology, nanotechnology. Richard G. Lipsey, Kenneth I. Carlaw, and Clifford T. Bekar. *Economic Transformations*. Oxford, England, UK: Oxford University Press, 2005.

[4]Brian Bergstein. "This Is Why AI Has Yet to Reshape Most Businesses." In: *MIT Technology Review* (Feb. 2019). URL: https://www.technologyreview.com/s/612897/this-is-why-ai-has-yet-to-reshape-most-businesses.

decision makers found that only 21% of them have a high level of trust in different types of analytics;[5] the number is likely smaller for machine learning, which is a part of analytics in business parlance.

> "A decision aid, no matter how sophisticated or 'intelligent' it may be, may be rejected by a decision maker who does not trust it, and so its potential benefits to system performance will be lost."
>
> —Bonnie M. Muir, psychologist at University of Toronto

This book is being written at a juncture in time when there is a lot of enthusiasm for machine learning. It is also a time when many societies are reckoning with social justice. Many claim that it is the beginning of the age of artificial intelligence, but others are afraid of impending calamity. The technology is poised to graduate from the experimental sandboxes of academic and industrial laboratories to truly widespread adoption across domains, but only if the gap in trust can be overcome.

I restrain from attempting to capture the zeitgeist of the age, but provide a concise and self-contained treatment of the technical aspects of machine learning. The goal is not to mesmerize you, but to get you to think things through.[6] There is a particular focus on mechanisms for increasing the trustworthiness of machine learning systems. As you'll discover throughout the journey, there is no single best approach toward trustworthy machine learning applicable across all applications and domains. Thus, the text focuses on helping you develop the thought process for weighing the different considerations rather than giving you a clear-cut prescription or recipe to follow. Toward this end, I provide an operational definition of trust in the next section and use it as a guide on our conceptual development of trustworthy machine learning. I tend to present evergreen concepts rather than specific tools and tricks that may soon become dated.

1.1 Defining Trust

What is trust and how do we operationalize it for machine learning?

> "What is trust? I could give you a dictionary definition, but you know it when you feel it. Trust happens when leaders are transparent, candid, and keep their word. It's that simple."
>
> —Jack Welch, CEO of General Electric

It is not enough to simply be satisfied by: 'you know it when you feel it.' The concept of trust is defined and studied in many different fields including philosophy, psychology, sociology, economics, and organizational management. Trust is the relationship between a *trustor* and a *trustee*: the trustor trusts the trustee. A definition of trust from organizational management is particularly appealing and

[5]Maria Korolov. "Explainable AI: Bringing Trust to Business AI Adoption." In: *CIO* (Sep. 2019). URL: https://www.cio.com/article/3440071/explainable-ai-bringing-trust-to-business-ai-adoption.html.
[6]The curious reader should research the etymology of the word 'mesmerize.'

relevant for defining trust in machine learning because machine learning systems in high-stakes applications are typically used within organizational settings. *Trust is the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party.*[7] This definition can be put into practice as a foundation for desiderata of machine learning systems.

1.1.1 Trusted vs. Trustworthy

Embedded within this definition is the idea that the trustee has certain properties that make it *trustworthy*, i.e. the qualities by which the trustor can expect the trustee to perform the important action referred to in the definition of trust. Being trustworthy does not automatically imply that the trustee is trusted. The trustor must consciously make a decision to be vulnerable to the trustee based on its trustworthiness and other factors including cognitive biases of the trustor. Understandably, potential trustors who are already vulnerable as members of marginalized groups may not want to become even more vulnerable. A system may not be trusted no matter how worthy of trust it is.

> "The toughest thing about the power of trust is that it's very difficult to build and very easy to destroy."
>
> —Thomas J. Watson, Sr., CEO of IBM

Moreover, the trustor's expectation of the trustee can evolve over time, even if the trustworthiness of the trustee remains constant. A typical dynamic of increasing trust over time begins with the trustor's expectation of performance being based on (1) the *predictability* of individual acts, moves onto (2) expectation based on *dependability* captured in summary statistics, finally culminating in (3) the trustor's expectation of performance based on *faith* that dependability will continue in the future.[8] Predictability could arise from some sort of understanding of the trustee by the trustor (for example their motivations or their decision-making procedure) or by low variance in the trustee's behavior. The expectation referred to in dependability is the usual notion of expectation in probability and statistics.

In much of the literature on the topic, both the trustor and the trustee are people. For our purposes, however, an end-user or other person is the trustor and the machine learning system is the trustee. Although the specifics may differ, there are not many differences between a trustworthy person and a trustworthy machine learning system. However, the final trust of the trustor, subject to cognitive biases, may be quite different for a human trustee and machine trustee depending on the task.[9]

1.1.2 Attributes of Trustworthiness

Building upon the above definition of trust and trustworthiness, you can list many different attributes of a trustworthy person: availability, competence, consistency, discreetness, fairness, integrity, loyalty,

---

[7]Roger C. Mayer, James H. Davis, and F. David Schoorman. "An Integrative Model of Organizational Trust." In: *Academy of Management Review* 20.3 (Jul. 1995), pp. 709–734.
[8]John K. Rempel, John G. Holmes, and Mark P. Zanna. "Trust in Close Relationships." In: *Journal of Personality and Social Psychology* 49.1 (Jul. 1985), pp. 95–112.
[9]Min Kyung Lee. "Understanding Perception of Algorithmic Decisions: Fairness, Trust, and Emotion in Response to Algorithmic Management." In: *Big Data & Society* 5.1 (Jan.–Jun. 2018).

openness, promise fulfilment, and receptivity to name a few.[10] Similarly, you can list several attributes of a trustworthy information system, such as: correctness, privacy, reliability, safety, security, and survivability.[11] The 2019 International Conference on Machine Learning (ICML) listed the following topics under trustworthy machine learning: adversarial examples, causality, fairness, interpretability, privacy-preserving statistics and machine learning, and robust statistics and machine learning. The European Commission's High Level Expert Group on Artificial Intelligence listed the following attributes: lawful, ethical, and robust (both technically and socially).

Such long and disparate lists give us some sense of what people deem to be trustworthy characteristics, but are difficult to use as anything but a rough guide. However, we can distill these attributes into a set of *separable* sub-domains that provide an organizing framework for trustworthiness. Several pieces of work converge onto a nearly identical set of four such separable attributes; a selected listing is provided in Table 1.1. The first three rows of Table 1.1 are attributes of trustworthy people. The last two rows are attributes of trustworthy artificial intelligence. Importantly, through separability, it is implied that each of the qualities is conceptually different and we can examine each of them in isolation of each other.

Table 1.1. *Attributes of trustworthy people and artificial intelligence systems.*

	Source	Attribute 1	Attribute 2	Attribute 3	Attribute 4
trustworthy people	Mishra[12]	competent	reliable	open	concerned
	Maister et al.[13]	credibility	reliability	intimacy	low self-orientation
	Sucher and Gupta[14]	competent	use fair means to achieve its goals	take responsibility for all its impact	motivated to serve others' interests as well as its own
trustworthy artificial intelligence	Toreini et al.[15]	ability	integrity	predictability	benevolence
	Ashoori and Weisz[16]	technical competence	reliability	understandability	personal attachment

[10]Graham Dietz and Deanne N. Den Hartog. "Measuring Trust Inside Organisations." In: *Personnel Review* 35.5 (Sep. 2006), pp. 557–588.
[11]Fred B. Schneider, ed. *Trust in Cyberspace*. Washington, DC, USA: National Academy Press, 1999.
[12]Aneil K. Mishra. "Organizational Responses to Crisis: The Centrality of Trust." In: *Trust in Organizations*. Ed. by Roderick M. Kramer and Thomas Tyler. Newbury Park, California, USA: Sage, 1996, pp. 261–287.
[13]David H. Maister, Charles H. Green, and Robert M. Galford. *The Trusted Advisor*. New York, New York, USA: Touchstone, 2000.
[14]Sandra J. Sucher and Shalene Gupta. "The Trust Crisis." In: *Harvard Business Review* (Jul. 2019). URL: https://hbr.org/cover-story/2019/07/the-trust- crisis.
[15]Ehsan Toreini, Mhairi Aitken, Kovila Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad van Moorsel. "The Relationship Between Trust in AI and Trustworthy Machine Learning Technologies." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 272–283.
[16]Maryam Ashoori and Justin D. Weisz. "In AI We Trust? Factors That Influence Trustworthiness of AI-Infused Decision-Making Processes." arXiv:1912.02675, 2019.

1.1.3 Mapping Trustworthy Attributes to Machine Learning

Interpreting the attributes of trustworthiness from the table in the context of machine learning is the primary thread of this book. In particular, we take Attribute 1 (competence) to be basic performance such as the accuracy of a machine learning model. Good performance, appropriately quantified based on the specifics of the problem and application,[17] is a necessity to be used in any real-world task.

We take Attribute 2 to include the reliability, safety, security and fairness of machine learning models and systems. Machine learning systems need to maintain good and correct performance across varying operating conditions. Different conditions could come from natural changes in the world or from malevolent or benevolent human-induced changes.

We take Attribute 3 to consist of various aspects of openness and human interaction with the machine learning system. This includes communication from the machine to the human through comprehensibility of models by people as well as transparency into overall machine learning system pipelines and lifecycles. It also includes communication from the human to the machine to supply personal and societal desires and values.

We take Attribute 4 (selflessness) to be the alignment of the machine learning system's purpose with a society's wants. The creation and development of machine learning systems is not independent of its creators. It is possible for machine learning development to go in a dystopian direction, but it is also possible for machine learning development to be intertwined with matters of societal concern and applications for social good, especially if the most vulnerable members of society are empowered to use machine learning to meet their own goals.

Although each of the four attributes are conceptually distinct, they may have complex interrelationships. We return to this point later in the book, especially in Chapter 14. There, we describe relationships among the different attributes (some are tradeoffs, some are not) that policymakers must reason about to decide a system's intended operations.

We use the following working definition of trustworthy machine learning in the remainder of the book. **A trustworthy machine learning system is one that has sufficient:**

1. **basic performance,**
2. **reliability,**
3. **human interaction, and**
4. **aligned purpose.**

We keep the focus on making machine learning systems worthy of trust rather than touching on other (possibly duplicitous) ways of making them trusted.

1.2 Organization of the Book

The organization of the book closely follows the four attributes in the definition of trustworthy machine learning. I am purposefully mindful in developing the concepts slowly rather than jumping ahead quickly to the later topics that may be what are needed in immediate practice. This is because

[17] Kiri L. Wagstaff. "Machine Learning that Matters." In: *Proceedings of the International Conference on Machine Learning*. Edinburgh, Scotland, UK, Jun.–Jul. 2012, pp. 521–528.

the process of creating trustworthy machine learning systems, given the high consequence of considerations like safety and reliability, should also be done in a thoughtful manner without overzealous haste. Taking shortcuts can come back and bite you.

> "Slow down and let your System 2 take control."[18]
>
> —Daniel Kahneman, behavioral economist at Princeton University
>
> "Worry about rhythm rather than speed."
>
> —Danil Mikhailov, executive director of data.org

Highlighted in Figure 1.1, the remainder of Part 1 discusses the book's limitations and works through a couple of preliminary topics that are important for understanding the concepts of trustworthy machine learning: the personas and lifecycle of developing machine learning systems in practice, and quantifying the concept of safety in terms of uncertainty.

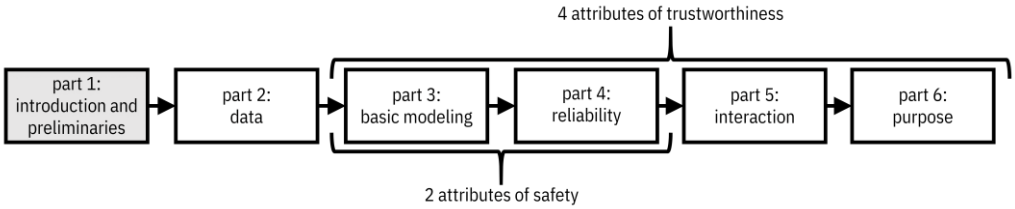


Figure 1.1. *Organization of the book. This first part focuses on introducing the topic of trustworthy machine learning and covers a few preliminary topics.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 1 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

Part 2 is a discussion of data, the prerequisite for doing machine learning. In addition to providing a short overview of different data modalities and sources, the part touches on three topics relevant for trustworthy machine learning: biases, consent, and privacy.

Part 3 relates to the first attribute of trustworthy machine learning: basic performance. It describes optimal detection theory and different formulations of supervised machine learning. It teaches several different learning algorithms such as discriminant analysis, naïve Bayes, k-nearest neighbor, decision

[18]Kahneman and Tversky described two ways in which the brain forms thoughts, which they call 'System 1' and 'System 2.' System 1 is fast, automatic, emotional, stereotypic and consciousness. System 2 is slow, effortful, logical, calculating, and conscious. Please engage the 'System 2' parts of your thought processes and be deliberate when you develop trustworthy machine learning systems.

trees and forests, logistic regression, support vector machines, and neural networks. The part concludes with methods for causal discovery and causal inference.

Part 4 is about the second attribute of trustworthy machine learning: reliability. This attribute is discussed through three specific topics: distribution shift, fairness, and adversarial robustness. The descriptions of these topics not only define the problems, but also provide solutions for detecting and mitigating the problems.

Part 5 is about the third attribute: human interaction with machine learning systems in both directions—understanding the system and giving it instruction. The part begins with interpretability and explainability of models. It moves onto methods for testing and documenting aspects of machine learning algorithms that can then be transparently reported, e.g. through factsheets. The final topic of this part is on the machine eliciting the policies and values of people and society to govern its behavior.

Part 6 discusses the fourth attribute: what those values of people and society may be. It begins by covering the ethics principles assembled by different parties as their paradigms for machine learning. Next, it discusses how the inclusion of creators of machine learning systems with diverse lived experiences broadens the values, goals, and applications of machine learning, leading in some cases to the pursuit of social good through the technology. Finally, it shows how the prevailing paradigm of machine learning in information recommendation platforms leads to filter bubbles and disinformation, and suggests alternatives. The final chapter about platforms is framed in terms of trustworthy institutions, which have different attributes than individual trustworthy people or individual trustworthy machine learning systems.

1.3 Limitations

Machine learning is an increasingly vast topic of study that requires several volumes to properly describe. The elements of trust in machine learning are also now becoming quite vast. In order to keep this book manageable for both me (the author) and you (the reader) it is limited in its depth and coverage of topics. Parts of the book are applicable both to simpler data analysis paradigms that do not involve machine learning and to explicitly programmed computer-based decision support systems, but for the sake of clarity and focus, they are not called out separately.

Significantly, despite trustworthy machine learning being a topic at the intersection of technology and society, the focus is heavily skewed toward technical definitions and methods. I recognize that philosophical, legal, political, sociological, psychological, and economic perspectives may be even more important to understanding, analyzing, and affecting machine learning's role in society than the technical perspective. Nevertheless, these topics are outside the scope of the book. Insights from the field of human-computer interaction are also extremely relevant to trustworthy machine learning; I discuss these to a limited extent at various points in the book, particularly Part 5.

Within machine learning, I focus on supervised learning at the expense of unsupervised and reinforcement learning. I do, however, cover graphical representations of probability and causality as well as their inference. Within supervised learning, the primary focus is on classification problems in which the labels are categorical. Regression, ordinal regression, ranking, anomaly detection, recommendation, survival analysis, and other problems without categorical labels are not the focus. The depth in describing various classification algorithms is limited and focused on high-level concepts rather than more detailed accounts or engineering tricks for using the algorithms.

Several different forms and modalities of data are briefly described in Part 2, such as time series, event streams, graphs, and parsed natural language. However, the primary focus of subsequent chapters is on forms of data represented as feature vectors.[19] Structured, tabular data as well as images are naturally represented as feature vectors. Natural language text is also often represented by a feature vector for further analysis.

An important ongoing direction of machine learning research is transfer learning, a paradigm in which previously learned models are repurposed for new uses and contexts after some amount of fine-tuning with data from the new context. A related concept for causal models is statistical transportability. Nonetheless, this topic is beyond the scope of the book except in passing in a couple of places. Similarly, the concepts of multi-view machine learning and causal data fusion, which involve the modeling of disparate sets of features are not broached. In addition, the paradigm of active learning, in which the labeling of data is done sequentially rather than in batch before modeling, is not discussed in the book.

As a final set of technical limitations, the depth of the mathematics is limited. For example, I do not present the concepts of probability at a depth requiring measure theory. Moreover, I stop at the posing of optimization problems and do not go into specific algorithms for conducting the optimization.[20] Discussions of statistical learning theory, such as generalization bounds, are also limited.

1.4 Positionality Statement

It is highly atypical for a computer science or engineering book to consider the influence of the author's personal experiences and background on its contents. Such a discussion is known as a *reflexivity statement* or *positionality statement* in the social sciences. I do so here since power and privilege play a key role in how machine learning is developed and deployed in the real-world. This recognition is increasing because of a current increase in attention to social justice in different societies. Therefore, it is important to be transparent about me so that you can assess potential biases against marginalized individuals and groups in the contents of the book. I'll evaluate myself using the four dimensions of trustworthiness detailed earlier in the chapter (competence, reliability, interaction, and purpose).

> "Science currently is taught as some objective view from nowhere (a term I learned about from reading feminist studies works), from no one's point of view."
>
> —Timnit Gebru, research scientist at Google

I encourage you, the reader, to create your own positionality statement as you embark on your journey to create trustworthy machine learning systems.

---

[19]A feature is an individual measurable attribute of an observed phenomenon. Vectors are mathematical objects that can be added together and multiplied by numbers.

[20]Mathematical optimization is the selection of a best element from some set of alternatives based on a desired criterion.

1.4.1 Competence and Credibility

I completed a doctorate in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT). My dissertation included a new kind of supervised machine learning method and a decision-theoretic model of human decision making that quantitatively predicts racial bias. I have been a research staff member at IBM Research – Thomas J. Watson Research Center since 2010 conducting research on statistical signal processing, data mining, and machine learning. The results have been published in various reputed workshops, conferences, journals, and magazines including ICML, the Conference on Neural Information Processing Systems (NeurIPS), the International Conference on Learning Representations (ICLR), the ACM Conference on Knowledge Discovery and Data Mining (KDD), the AAAI/ACM Conference on Artificial Intelligence, Ethics and Society (AIES), the Journal of Machine Learning Research (JMLR), the IEEE Transactions on Signal Processing, the IEEE Transactions on Information Theory, and the Proceedings of the IEEE. I have defined a large part of the strategy for trustworthy machine learning at IBM Research and a large subset of my own work has been on interpretability, safety, fairness, transparency, value alignment, and social good in machine learning and artificial intelligence.

I have developed real-world solutions that have been deployed in high-stakes applications of machine learning and data science during engagements with IBM business units, various clients of IBM, and social change organizations. I have led teams that developed the comprehensive open source toolkits and resources on fairness, explainability and uncertainty quantification named AI Fairness 360, AI Explainability 360 and Uncertainty Quantification 360, and transitioned some of their capabilities into the IBM Watson Studio product. I have spoken at various industry-oriented meetups and conventions such as the O'Reilly AI Conference, Open Data Science Conference, and IBM Think.

I have been an adjunct faculty member at New York University (NYU) and a guest lecturer in courses at Cornell, Georgetown, NYU, Princeton, Rutgers, and Syracuse. I organized the Workshop on Human Interpretability in Machine Learning at ICML annually from 2016 to 2020 as well as several other workshops and symposia related to trustworthy machine learning. I served as a track chair for the practice and experience track of the 2020 ACM Conference on Fairness, Accountability and Transparency and was a member of the Partnership on AI's Safety-Critical AI expert group.

To compose this book, I am channeling all these past experiences along with the interactions with students, lifelong learners, and colleagues that these experiences have afforded. Of course, I have less depth of knowledge about the topics of some of the chapters than others, but have some level of both practical/applied and theoretical knowledge on all of them.

1.4.2 Reliability and Biases

Reliability stems from the ability to work in different contexts and conditions. I have only had one employer, which limits this ability. Nevertheless, by working at IBM Research and volunteering with DataKind (an organization that helps professional data scientists conduct projects with social change organizations), my applied data science work has engaged with a variety of for-profit corporations, social enterprises, and non-profit organizations on problems in human resources and workforce analytics, health systems and policy, clinical health care, humanitarian response, international development, financial inclusion, and philanthropic decision making. Moreover, my research contributions have been disseminated not only in machine learning research venues, but also

statistics, operations research, signal processing, information theory, and information systems venues, as well as the industry-oriented venues I mentioned earlier.

More importantly for trustworthy machine learning, I would like to mention my privileges and personal biases. I was born and raised in the 1980s and 1990s in predominantly white upper middle-class suburbs of Syracuse, a medium-sized city in upstate New York located on the traditional lands of the Onöñda'gaga' people, that is one of the most racially-segregated in the United States. Other places I have lived for periods of three months or longer are Ithaca, Elmsford, Ossining, and Chappaqua in New York; Burlington and Cambridge in Massachusetts; Livermore, California; Ludhiana, New Delhi, and Aligarh in northern India; Manila, Philippines; Paris, France; and Nairobi, Kenya. I am a cis male, second-generation American of South Asian descent. To a large extent, I am an adherent of dharmic religious practices and philosophies. One of my great-great-grandfathers was the first Indian to study at MIT in 1905. My father and his parents lived hand-to-mouth at times, albeit with access to the social capital of their forward caste group. My twin brother, father, and both grandfathers are or were professors of electrical engineering. My mother was a public school teacher. I studied in privileged public schools for my primary and secondary education and an Ivy League university for my undergraduate education. My employer, IBM, is a powerful and influential corporation. As such, I have been highly privileged in understanding paths to academic and professional success and having an enabling social network. Throughout my life, however, I have been a member of a minority group with limited political power. I have had some visibility into hardship beyond the superficial level, but none of this experience has been *lived experience*, where I would not have a chance to leave if I wanted to.

1.4.3 Interaction

I wrote the book with some amount of transparency. While I was writing the first couple of chapters in early 2020, anyone could view them through Overleaf (https://v2.overleaf.com/read/bzbzymggkbzd). After I signed a book contract with Manning Publications, chapters were posted to the Manning Early Access Program as I wrote them, with readers having an opportunity to engage via the Manning liveBook Discussion Forum. After the publisher and I parted ways in September 2021, I posted chapters of the in-progress manuscript to http://www.trustworthymachinelearning.com. I received several useful comments from various individuals throughout the drafting process via email (krvarshn@us.ibm.com), Twitter direct message (@krvarshney), telephone (+1-914-945-1628), and personal meetings. When I completed version 0.9 of the book at the end of December 2021, I posted it at the same site. On January 28, 2022, I convened a panel of five people with lived experiences different from mine to provide their perspectives on the content contained in version 0.9 using a modified Diverse Voices method.[21] An electronic version of this edition of the book will continue to be available at no cost at the same website: http://www.trustworthymachinelearning.com.

[21]Lassana Magassa, Meg Young, and Batya Friedman. "Diverse Voices: A How-To Guide for Facilitating Inclusiveness in Tech Policy." Tech Policy Lab, University of Washington, 2017. The panelists who provided impartial input were Mashael Alzaid, Kenya Andrews, Noah Chasek-Macfoy, Scott Fancher, and Timothy Odonga. As a central part of the Diverse Voices method, they were offered honoraria, which some declined. The funds came from an honorarium I received for participating in an AI Documentation Summit convened by The Data Nutrition Project in January 2022.

1.4.4 Motivation and Values

My motivations begin with family values. The great-great-grandfather I mentioned above returned to India with knowledge of industrial-scale glassmaking from MIT and made social impact by establishing a factory in service of *swaraj*, self-governance in India, and the training of local workers. One of my grandfathers applied his knowledge of systems and control theory to problems in agriculture and also worked toward social justice in India through non-technological means. My other grandfather joined UNESCO to establish engineering colleges in developing Iraq and Thailand. My mother taught science in an inner-city school district's special program for students caught with weapons in their regular middle and high schools.

In the same way, consistent with family values as well as external ethics (*yama*),[22] internal ethics (*niyama*),[23] and the ethos of the American dream, my personal motivation is to advance today's most societally-impactful technology (machine learning), mitigate its harmfulness, apply it to uplift humanity, and train others to do the same. I co-founded the IBM Science for Social Good fellowship program in 2015–2016 and direct it toward these aims.

The reason I wrote this book is many-fold. First, I feel that although many of the topics that are covered in the book, like fairness, explainability, robustness, and transparency are often talked about together, there is no source that unifies them in a coherent thread. With this book, there is such a resource for technologists, developers, and researchers to learn from. Second, I feel that in industry practice, the unbridled success of deep learning has led to too much emphasis on engineers squeezing out a little more accuracy with little conceptual understanding and little regard to considerations beyond accuracy (the other three attributes of trust). The aim of the book is to fill the conceptual understanding gap for the practitioners who wish to do so, especially those working in high-stakes application domains. (Cai and Guo find that many software engineers fundamentally desire guidance on understanding and applying the conceptual underpinnings of machine learning.[24]) The inclusion of considerations beyond predictive accuracy cannot be an afterthought; it must be part of the plan from the beginning of any new project. Third, I would like to empower others who share my values and ethics to pursue a future in which there is a virtuous cycle of research and development in which technology helps society flourish and society helps technology flourish.

1.5 Summary

- Machine learning systems are influencing critical decisions that have consequences to our daily lives, but society lacks trust in them.
- Trustworthiness is composed of four attributes: competence, reliability, openness, and selflessness.

---

[22]List of *yamas*: *ahiṃsā* (non-harm), *satya* (benevolence and truthfulness), *asteya* (responsibility and non-stealing), *brahmacarya* (good direction of energy), and *aparigraha* (simplicity and generosity).

[23]List of *niyamas*: *śauca* (clarity and purity), *santoṣa* (contentment), *tapas* (sacrifice for others), *svādhyayā* (self-study), and *īsvara-praṇidhāna* (humility and service to something bigger).

[24]Carrie J. Cai and Philip J. Guo. "Software Developers Learning Machine Learning: Motivations, Hurdles, and Desires." In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*. Memphis, Tennessee, USA, Oct. 2019, pp. 25–34.

- The book is organized to match this decomposition of the four components of trust.
- Despite my limitations and the limitations of the contents, the book endeavors to develop a conceptual understanding not only of the principles and theory behind how machine learning systems can achieve these goals to become more trustworthy, but also develop the algorithmic and non-algorithmic methods to pursue them in practice.
- By the end of the book, your thought process should naturally be predisposed to including elements of trustworthiness throughout the lifecycle of machine learning solutions you develop.

2

Machine Learning Lifecycle

Imagine that you are a project manager on the innovation team of m-Udhār Solar, a (fictional) pay-as-you-go solar energy provider to poor rural villages that is struggling to handle a growing load of applications. The company is poised to expand from installing solar panels in a handful of pilot districts to all the districts in the state, but only if it can make loan decisions for 25 times as many applications per day with the same number of loan officers. You think machine learning may be able to help.

Is this a problem to address with machine learning? How would you begin the project? What steps would you follow? What roles would be involved in carrying out the steps? Which stakeholders' buy-in would you need to win? And importantly, what would you need to do to ensure that the system is *trustworthy*? Making a machine learning system trustworthy should not be an afterthought or add-on, but should be part of the plan from the beginning.

The end-to-end development process or *lifecycle* involves several steps:

1. problem specification,
2. data understanding,
3. data preparation,
4. modeling,
5. evaluation, and
6. deployment and monitoring.

Narrow definitions consider only the modeling step to be the realm of machine learning. They consider the other steps to be part of the broader endeavor of data science and engineering. Most books and research on machine learning are similarly focused on the modeling stage. However, you cannot really execute the development and deployment of a trustworthy machine learning system without focusing on all parts of the lifecycle. There are no shortcuts. This chapter sketches out the master plan.

2.1 A Mental Model for the Machine Learning Lifecycle

The six steps of the machine learning lifecycle given above, also illustrated in Figure 2.1, are codified in the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. This is the mental model to keep in mind of how machine learning systems should be developed and deployed. Although the flow largely proceeds sequentially through the steps, there are several opportunities to go back and redo earlier steps. This description is stylized; even good examples of real-world lifecycles are messier, but the main idea continues to hold.

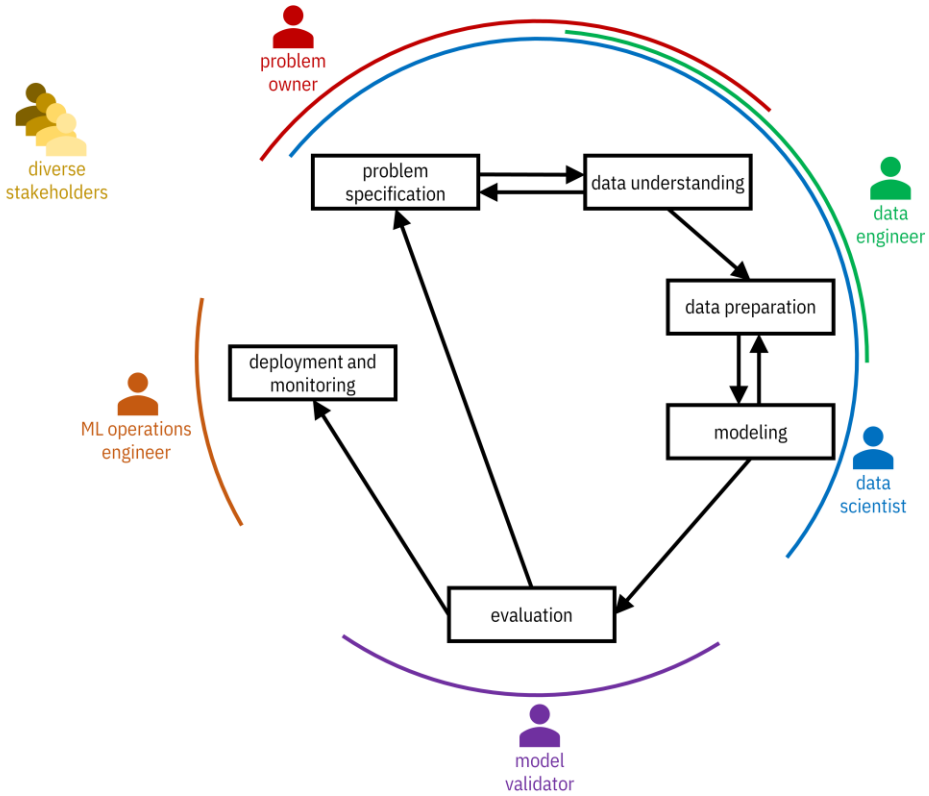


Figure 2.1. *Steps of the machine learning lifecycle codified in CRISP-DM. Different personas participate in different phases of the lifecycle.* Accessible caption. A series of six steps arranged in a circle: (1) problem specification; (2) data understanding; (3) data preparation; (4) modeling; (5) evaluation; (6) deployment and monitoring. There are some backward paths: from data understanding to problem specification; from modeling to data preparation; from evaluation to problem specification. Five personas are associated with different steps: problem owner with steps 1–2; data engineer with steps 2–3; data scientist with steps 1–4; model validator with step 5; ML operations engineer with step 6. A diverse stakeholders persona is on the side overseeing all steps.

Because the modeling stage is often put on a pedestal, there is a temptation to use the analogy of an onion in working out the project plan: start with the core modeling, work your way out to data

understanding/preparation and evaluation, and then further work your way out to problem specification and deployment/monitoring. This analogy works well for a telecommunications system for example,[1] both pedagogically and in how the technology is developed, but a sequential process is more appropriate for a trustworthy machine learning system. Always start with understanding the use case and specifying the problem.

> "People are involved in every phase of the AI lifecycle, making decisions about which problem to address, which data to use, what to optimize for, etc."
>
> —Jenn Wortman Vaughan, research scientist at Microsoft

The different steps are carried out by different parties with different personas including problem owners, data engineers, data scientists, model validators, and machine learning (ML) operations engineers. Problem owners are primarily involved with problem specification and data understanding. Data engineers work on data understanding and data preparation. Data scientists tend to play a role in all of the first four steps. Model validators perform evaluation. ML operations engineers are responsible for deployment and monitoring.

Additional important personas in the context of trustworthiness are the potential trustors of the system: human decision makers being supported by the machine learning model (m-Udhār loan officers), affected parties about whom the decisions are made (rural applicants; they may be members of marginalized groups), regulators and policymakers, and the general public. Each stakeholder has different needs, concerns, desires, and values. Systems must meet those needs and align with those values to be trustworthy. Multi-stakeholder engagement is essential and cannot be divorced from the technical aspects of design and development. Documenting and transparently reporting the different steps of the lifecycle help build trust among stakeholders.

2.2 Problem Specification

The first step when starting the development of a machine learning system is to define the problem. What is the problem owner hoping to accomplish and why? The director of m-Udhār Solar wishes to automate a task that is cumbersome and costly for people to do. In other scenarios, problem owners may want to augment the capabilities of human decision makers to improve the quality of their decisions. They may have other goals altogether. In some cases, the problem should not even be solved to begin with, because doing so may cause or exacerbate societal harms and breach the lines of ethical behavior.[2] A *harm* is an outcome with a severe unwanted effect on a person's life. This definition of harm is made more precise in Chapter 3. Let's repeat this important point: do not solve problems that would lead to harms for someone or some group.

---

[1] C. Richard Johnson, Jr. and William A. Sethares. *Telecommunication Breakdown: Concepts of Communication Transmitted via Software-Defined Radio*. Upper Saddle River, New Jersey, USA: Prentice Hall, 2003.

[2] Andrew D. Selbst, danah boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. "Fairness and Abstraction in Sociotechnical Systems." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 59–68.

> "We all have a responsibility to ask not just, 'can we do this?', but 'should we do this?'"
>
> —Kathy Baxter, ethical AI practice architect at Salesforce

Problem identification and understanding is best done as a dialogue between problem owners and data scientists because problem owners might not have the imagination of what is possible through machine learning and data scientists do not have a visceral understanding of the pain points that problem owners are facing. Problem owners should also invite representatives of marginalized groups for a seat at the problem understanding table to voice their pain points.[3] Problem identification is arguably the most important and most difficult thing to do in the entire lifecycle. An inclusive design process is imperative. Finding the light at the end of the tunnel is actually not that hard, but finding the tunnel can be very hard. The best problems to tackle are ones that have a benefit to humanity, like helping light up the lives and livelihoods of rural villagers.

Once problem owners have identified a problem worth solving, they need to specify metrics of success. Being a social enterprise, the metric for m-Udhār Solar is number of households served with acceptable risk of defaulting. In general, these metrics should be in real-world terms relevant to the use case, such as lives saved, time reduced, or cost avoided.[4] The data scientist and problem owner can then map the real-world problem and metrics to machine learning problems and metrics. This specification should be as crisp as possible, including both the quantities to be measured and their acceptable values.

The goals need not be specified only as traditional key performance indicators, but can also include objectives for maintenance of performance across varying conditions, fairness of outcomes across groups and individuals, resilience to threats, or number of insights provided to users. Defining what is meant by fairness and specifying a threat model are part of this endeavor. For example, m-Udhār aims not to discriminate by caste or creed. Again, these real-world goals must be made precise through a conversation between problem owners, diverse voices, and data scientists. The process of eliciting objectives is known as *value alignment*.

One important consideration in problem scoping is resource availability, both in computing and human resources. A large national or multinational bank will have many more resources than m-Udhār Solar. A large technology company will have the most of all. What can reasonably be accomplished is gated by the skill of the development team, the computational power for training models and evaluating new samples, and the amount of relevant data.

Machine learning is not a panacea. Even if the problem makes sense, machine learning may not be the most appropriate solution to achieve the metrics of success. Oftentimes, back-of-the-envelope calculations can indicate the lack of fit of a machine learning solution before other steps are undertaken. A common reason for machine learning to not be a viable solution is lack of appropriate data, which brings us to the next step: data understanding.

---

[3]Meg Young, Lassana Magassa, and Batya Friedman. "Toward Inclusive Tech Policy Design: A Method for Underrepresented Voices to Strengthen Tech Policy Documents." In: *Ethics and Information Technology* 21.2 (Jun. 2019), pp. 89–103. The input of diverse stakeholders, especially those from marginalized groups, should be monetarily compensated.

[4]Kiri L. Wagstaff. "Machine Learning that Matters." In: *Proceedings of the International Conference on Machine Learning*. Edinburgh, Scotland, UK, Jun.–Jul. 2012, pp. 521–528.

2.3 Data Understanding

Once the problem has been identified and specified, a relevant dataset must be collected. In instances where the problem is to automate an existing decision-making process, identifying the relevant dataset is fairly straightforward. M-Udhār's dataset consists of attributes and other inputs that loan officers used to make decisions in the past, along with their decisions. The inputs constitute the *features* and the historical decisions constitute the *labels* for a supervised machine learning task. But there may also be data that loan officers did not use that could be leveraged by a machine learning system. A promise of so-called 'big data' is the inclusion of large sets of attributes, many weakly correlated to the label, that would overwhelm a person but not a machine. For the machine learning system to make even better decisions than people, true outcomes rather than decisions should ideally be the labels, e.g. whether an applicant defaulted on their loan in the future rather than the approval decision.

Machine learning can also be applied in use cases that are new processes for an organization and no exact historical data exists. Here, proxy data must be identified. For example, a health system may wish to start offering home nursing care to indisposed individuals proactively, but may not have data directly applicable for understanding this decision. Data from previous interactions of patients with the health system may be used as a proxy. In other cases, it may be that new data must be collected. In yet other cases, it may be that relevant data neither exists nor can be collected, and the problem must be specified differently.

Once a dataset has been identified or collected, it is critical for the data scientist and data engineer to understand the semantics of the various features and their values by consulting the problem owner and other subject matter experts as well as by consulting a *data dictionary* (a document describing the features) if one exists. They should also conduct exploratory data analysis and visualization. This understanding can help identify problems in the data such as *leakage*, the presence of information in the features helpful in predicting the label that would not be available in new inputs to a deployed system, and various forms of *bias*. One important form of bias is *social bias* in which a proxy for the label does not well-reflect the true label of interest. For example, using past number of doctor visits may not be a good proxy of how sick an individual is if there are socioeconomic reasons why some people visit the doctor more than others at the same level of ill health. A similar social bias stems from prejudice: labels from historical human decisions contain systematic differences across groups. Other important biases include *population bias*: the dataset underrepresents certain inputs and overrepresents others, and *temporal bias*: issues stemming from the timing of data collection.

The data understanding stage also requires the development team to consider data usage issues. Just because features are available (and may even improve the performance of a model), that does not mean that they can and should be used. Use of certain features may be prohibited by law, be unethical, or may not have appropriate consent in place. For example, m-Udhār Solar may have the surname of the applicant available, which indicates the applicant's caste and religion and may even help a model achieve better performance, but it is unethical to use. The use of other features may pose privacy risks. A more detailed treatment of data-related issues is presented in Part 2 of the book.

2.4 Data Preparation

Data integration, data cleaning, and feature engineering constitute the data preparation step of the lifecycle. The end goal of this stage is a final training dataset to be used in modeling. Starting from the insights gleaned in the data understanding phase, *data integration* starts by extracting, transforming, and loading (ETL) data from disparate relevant databases and other data sources. Next, the data from the disparate sources is joined into a single dataset that is maintained in a format amenable to downstream modeling. This step is most challenging when dealing with humongous data sources.

Data cleaning is also based on data understanding from the previous stage. Some of the key components of data cleaning are:

- filling in missing values (known as *imputation*) or discarding them,
- binning continuous feature values to account for outliers,
- grouping or recoding categorical feature values to deal with rarely occurring values or to combine semantically similar values, and
- dropping features that induce leakage or should not be used for legal, ethical, or privacy reasons.

Feature engineering is mathematically transforming features to derive new features, including through interactions of several raw features. Apart from the initial problem specification, feature engineering is the point in the lifecycle that requires the most creativity from data scientists. Data cleaning and feature engineering require the data engineer and data scientist to make many choices that have no right or wrong answer. Should m-Udhār's data engineer and data scientist group together any number of motorcycles owned by the household greater than zero? How should they encode the profession of the applicant? The data scientist and data engineer should revisit the project goals and continually consult with subject matter experts and stakeholders with differing perspectives to help make appropriate choices. When there is ambiguity, they should work towards safety, reliability, and aligning with elicited values.

2.5 Modeling

The modeling step receives a clear problem specification (including metrics of success) and a fixed, clean training dataset. A mental model for trustworthy modeling includes three main parts:

1. pre-processing the training data,
2. training the model with a machine learning algorithm, and
3. post-processing the model's output predictions.

This idea is diagrammed in Figure 2.2. Details of this step will be covered in depth throughout the book, but an overview is provided here.

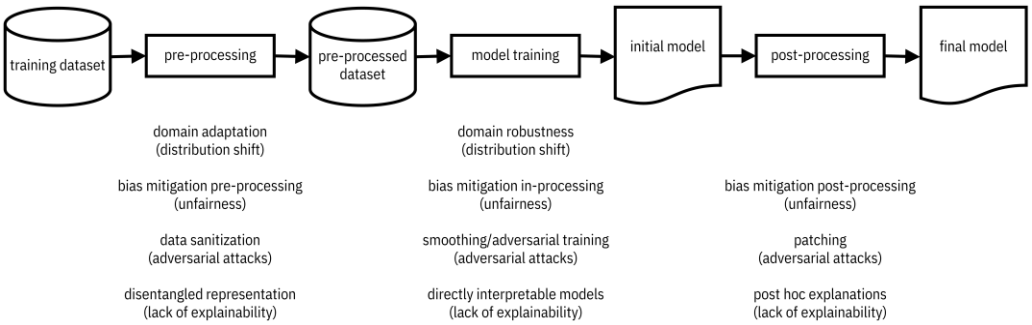


Figure 2.2. *Main parts of trustworthy machine learning modeling. Distribution shift, unfairness, adversarial attacks, and lack of explainability can be mitigated using the various techniques listed below each part. Details of these methods are presented in the remainder of the book.* Accessible caption. A block diagram with a training dataset as input to a pre-processing block with a pre-processed dataset as output. The pre-processed dataset is input to a model training block with an initial model as output. The initial model is input to a post-processing block with a final model as output. The following techniques are examples of pre-processing: domain adaptation (distribution shift); bias mitigation pre-processing (unfairness); data sanitization (adversarial attacks); disentangled representation (lack of explainability). The following techniques are examples of model training: domain robustness (distribution shift); bias mitigation in-processing (unfairness); smoothing/adversarial training (adversarial attacks); directly interpretable models (lack of explainability). The following techniques are examples of post-processing: bias mitigation post-processing (unfairness); patching (adversarial attacks); post hoc explanations (lack of explainability).

Different from data preparation, data *pre-processing* is meant to alter the statistics or properties of the dataset to achieve certain goals. *Domain adaptation* overcomes a lack of robustness to changing environments, including temporal bias. *Bias mitigation pre-processing* changes the dataset to overcome social bias and population bias. *Data sanitization* aims to remove traces of data poisoning attacks by malicious actors. Learning *disentangled representations* overcomes a lack of human interpretability of the features. All should be performed as required by the problem specification.

The main task in the modeling step is to use an algorithm that finds the patterns in the training dataset and generalizes from them to fit a model that will predict labels for new unseen data points with good performance. (The term *predict* does not necessarily imply forecasting into the future, but simply refers to providing a guess for an unknown value.) There are many different algorithms for fitting models, each with a different *inductive bias* or set of assumptions it uses to generalize. Many machine learning algorithms explicitly minimize the objective function that was determined in the problem specification step of the lifecycle. Some algorithms minimize an approximation to the specified objective to make the mathematical optimization easier. This common approach to machine learning is known as *risk minimization*.

The *no free lunch theorem* of machine learning says that there is no one best machine learning algorithm for all problems and datasets.[5] Which one works best depends on the characteristics of the

[5]David H. Wolpert. "The Lack of A Priori Distinctions Between Learning Algorithms." In: *Neural Computation* 8.7 (Oct. 1996), pp. 1341–1390.

dataset. Data scientists try out several different methods, tune their parameters, and see which one performs best empirically. The empirical comparison is conducted by randomly splitting the training dataset into a training partition on which the model is fit and a testing partition on which the model's performance is validated. The partitioning and validation can be done once, or they can be done several times. When done several times, the procedure is known as *cross-validation*; it is useful because it characterizes the stability of the results. Cross-validation should be done for datasets with a small number of samples.

The basic machine learning algorithm can be enhanced in several ways to satisfy additional objectives and constraints captured in the problem specification. One way to increase reliability across operating conditions is known as *domain robustness*. Machine learning algorithms that reduce unwanted biases are known as *bias mitigation in-processing*. One example category of methods for defending against data poisoning attacks is known as *smoothing*. A defense against a different kind of adversarial attack, the evasion attack, is *adversarial training*. Certain machine learning algorithms produce models that are simple in form and thus *directly interpretable* and understandable to people. Once again, all of these enhancements should be done according to the problem specification.

Post-processing rules change the predicted label of a sample or compute additional information to accompany the predicted label. Post-processing methods can be divided into two high-level categories: *open-box* and *closed-box*. Open-box methods utilize information from the model such as its parameters and functions of its parameters. Closed-box methods can only see the output predictions arising from given inputs. Open-box methods should be used if possible, such as when there is close integration of model training and post-processing in the system. In certain scenarios, post-processing methods, also known as *post hoc* methods, are isolated from the model for logistical or security reasons. In these scenarios, only closed-box methods are tenable. Post-processing techniques for increasing reliability, mitigating unwanted biases, defending against adversarial attacks, and generating explanations should be used judiciously to achieve the goals of the problem owner. For example, post hoc explanations are important to provide to m-Udhār Solar's loan officers so that they can better discuss the decision with the applicant.

The specification of certain use cases calls for *causal* modeling: finding generalizable instances of cause-and-effect from the training data rather than only correlative patterns. These are problems in which input interventions are meant to change the outcome. For example, when coaching an employee for success, it is not good enough to identify the pattern that putting in extra hours is predictive of a promotion. Good advice represents a causal relationship: if the employee starts working extra hours, then they can expect to be promoted. It may be that there is a common cause (e.g. conscientiousness) for both doing quality work and working extra hours, but it is only doing quality work that causes a promotion. Working long hours while doing poor quality work will not yield a promotion; causal modeling will show that.

2.6 Evaluation

Once m-Udhār's data scientists have a trained and tested model that they feel best satisfies the problem owner's requirements, they pass it on to model validators. A model validator conducts further independent testing and evaluation of the model, often with a completely separate *held-out* dataset that the data scientist did not have access to. It is important that the held-out set not have any leakage from

the training set. To stress-test the model's safety and reliability, the model validator can and should evaluate it on data collected under various conditions and data generated to simulate unlikely events.

The model validator persona is part of *model risk management*. Model risk is the chance of decisions supported by statistical or machine learning models yielding gross harms. Issues can come from any of the preceding lifecycle steps: from bad problem specification to data quality problems to bugs in the machine learning algorithm software. Even this late in the game, it is possible that the team might have to start over if issues are discovered. It is only after the model validator signs off on the model that it is put into production. Although not standard practice yet in machine learning, this 'signing off' can be construed as a *declaration of conformity*, a document often used in various industries and sectors certifying that a product is operational and safe.

2.7 Deployment and Monitoring

The solar panels are loaded on the truck and the electricians are just waiting to find out which households to install them at. The last step on the long road to the productive use of the machine learning system is finally here! The ML operations engineer takes center stage to deploy the model. Starting with a certified model, there are still questions to be answered. What infrastructure will bring new data to the model? Will predictions be made in batch or one-by-one? How much latency is allowable? How will the user interact with the system? The engineer works with different stakeholders to answer the questions and implements the infrastructure to meet the needs, resulting in a deployed model.

Important for making the model trustworthy, the ML operations engineer must also implement tools to monitor the model's performance to ensure it is operating as expected. As before, performance includes all relevant metrics of success in the problem specification, not only traditional key performance indicators. The performance of trained models can degrade over time as the incoming data statistically drifts away from the training data. If drift is detected, the monitoring system should notify the development team and other relevant stakeholders. All four attributes of trustworthiness (basic performance, reliability, human interaction, and aligned purpose) permeate throughout the machine learning lifecycle and must be accounted for in the development, deployment, and monitoring plan from the beginning to the end. M-Udhār Solar has now deployed its loan origination automation system and is able to easily serve applicants not just in one entire state, but a few neighboring ones as well.

2.8 Summary

- The machine learning lifecycle consists of six main sequential steps: (1) problem specification, (2) data understanding, (3) data preparation, (4) modeling, (5) evaluation, and (6) deployment and monitoring, performed by people in different roles.
- The modeling step has three parts: (1) pre-processing, (2) model training, and (3) post-processing.
- To operationalize a machine learning system, plan for the different attributes of trustworthiness starting from the first step of problem specification. Considerations beyond basic performance should not be sprinkled on at the end like pixie dust, but developed at every step of the way with input from diverse stakeholders, including affected users from marginalized groups.

3

Safety

Imagine that you are a data scientist at the (fictional) peer-to-peer lender ThriveGuild. You are in the problem specification phase of the machine learning lifecycle for a system that evaluates and approves borrowers. The problem owners, diverse stakeholders, and you yourself want this system to be trustworthy and not cause harm to people. Everyone wants it to be safe. But what is *harm* and what is *safety* in the context of a machine learning system?

Safety can be defined in very domain-specific ways, like safe toys not having lead paint or small parts that pose choking hazards, safe neighborhoods having low rates of violent crime, and safe roads having a maximum curvature. But these definitions are not particularly useful in helping define safety for machine learning. Is there an even more basic definition of safety that could be extended to the machine learning context? Yes, based on the concepts of (1) *harm*, (2) *aleatoric uncertainty* and *risk*, and (3) *epistemic uncertainty*.[1] (These terms are defined in the next section.)

This chapter teaches you how to approach the problem specification phase of a trustworthy machine learning system from a safety perspective. Specifically, by defining safety as minimizing two different types of uncertainty, you can collaborate with problem owners to crisply specify safety requirements and objectives that you can then work towards in the later parts of the lifecycle.[2] The chapter covers:

- Constructing the concept of *safety* from more basic concepts applicable to machine learning: *harm*, *aleatoric uncertainty*, and *epistemic uncertainty*.
- Charting out how to distinguish between the two types of uncertainty and articulating how to quantify them using probability theory and possibility theory.
- Specifying problem requirements in terms of summary statistics of uncertainty.

---

[1]Niklas Möller and Sven Ove Hansson. "Principles of Engineering Safety: Risk and Uncertainty Reduction." In: *Reliability Engineering and System Safety* 93.6 (Jun. 2008), pp. 798–805.
[2]Kush R. Varshney and Homa Alemzadeh. "On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products." In: *Big Data* 5.3 (Sep. 2017), pp. 246–255.

- Sketching how to update probabilities in light of new information.
- Applying ideas of uncertainty to understand the relationships among different attributes and figure out what is independent of what else.

3.1 Grasping Safety

Safety is the reduction of both aleatoric uncertainty (or risk) and *epistemic uncertainty associated with harms.* First, let's talk about harm. All systems, including the lending system you're developing for ThriveGuild, yield outcomes based on their state and the inputs they receive. In your case, the input is the applicant's information and the outcome is the decision to approve or deny the loan. From ThriveGuild's perspective (and from the applicant's perspective, if we're truly honest about it), a desirable outcome is approving an applicant who will be able to pay back their loan and denying an applicant who will not be able to pay back their loan. An undesirable outcome is the opposite. Outcomes have associated costs, which could be in monetary or other terms. An undesired outcome is a *harm* if its cost exceeds some threshold. Unwanted outcomes of small severity, like getting a poor movie recommendation, are not counted as harms.

In the same way that harms are undesired outcomes whose cost exceeds some threshold, trust only develops in situations where the stakes exceed some threshold.[3] Remember from Chapter 1 that the trustor has to be vulnerable to the trustee for trust to develop, and the trustor does not become vulnerable if the stakes are not high enough. Thus safety-critical applications are not only the ones in which trust of machine learning systems is most relevant and important, they are also the ones in which trust can actually be developed.

Now, let's talk about aleatoric and epistemic uncertainty, starting with uncertainty in general. Uncertainty is the state of current knowledge in which something is not known. ThriveGuild does not know if borrowers will or will not default on loans given to them. All applications of machine learning have some form of uncertainty. There are two main types of uncertainty: *aleatoric uncertainty* and *epistemic uncertainty*.[4]

Aleatoric uncertainty, also known as statistical uncertainty, is inherent randomness or stochasticity in an outcome that cannot be further reduced. Etymologically derived from dice games, aleatoric uncertainty is used to represent phenomena such as vigorously flipped coins and vigorously rolled dice, thermal noise, and quantum mechanical effects. Incidents that will befall a ThriveGuild loan applicant in the future, such as the roof of their home getting damaged by hail, may be subject to aleatoric uncertainty. *Risk* is the average outcome under aleatoric uncertainty.

On the other hand, epistemic uncertainty, also known as systematic uncertainty, refers to knowledge that is not known in practice, but could be known in principle. The acquisition of this knowledge would

[3]Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. "Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Mar. 2021, pp. 624–635.

[4]Eyke Hüllermeier and Willem Waegeman. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods." In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506.

reduce the epistemic uncertainty. ThriveGuild's epistemic uncertainty about an applicant's loan-worthiness can be reduced by doing an employment verification.

> "Not knowing the chance of mutually exclusive events and knowing the chance to be equal are two quite different states of knowledge."
>
> —Ronald A. Fisher, statistician and geneticist

Whereas aleatoric uncertainty is inherent, epistemic uncertainty depends on the observer. Do all observers have the same amount of uncertainty? If yes, you are dealing with aleatoric uncertainty. If some observers have more uncertainty and some observers have less uncertainty, then you are dealing with epistemic uncertainty.

The two uncertainties are quantified in different ways. Aleatoric uncertainty is quantified using *probability* and epistemic uncertainty is quantified using *possibility*. You have probably learned probability theory before, but it is possible that possibility theory is new to you. We'll dive into the details in the next section. To repeat the definition of safety in other words: *safety is the reduction of the probability of expected harms and the possibility of unexpected harms*. Problem specifications for trustworthy machine learning need to include both parts, not just the first part.

The reduction of aleatoric uncertainty is associated with the first attribute of trustworthiness (basic performance). The reduction of epistemic uncertainty is associated with the second attribute of trustworthiness (reliability). A summary of the characteristics of the two types of uncertainty is shown in Table 3.1. Do not take the shortcut of focusing only on aleatoric uncertainty when developing your machine learning model; make sure that you focus on epistemic uncertainty as well.

Table 3.1. *Characteristics of the two types of uncertainty.*

Type	Definition	Source	Quantification	Attribute of Trustworthiness
aleatoric	randomness	inherent	probability	basic performance
epistemic	lack of knowledge	observer-dependent	possibility	reliability

3.2 Quantifying Safety with Different Types of Uncertainty

Your goal in the problem specification phase of the machine learning lifecycle is to work with the ThriveGuild problem owner to set quantitative requirements for the system you are developing. Then in the later parts of the lifecycle, you can develop models to meet those requirements. So you need a quantification of safety and thus quantifications of costs of outcomes (are they harms or not), aleatoric uncertainty, and epistemic uncertainty. Quantifying these things requires the introduction of several concepts, including: sample space, outcome, event, probability, random variable, and possibility.

3.2.1 Sample Spaces, Outcomes, Events, and Their Costs

The first concept is the *sample space*, denoted as the set Ω, that contains all possible *outcomes*. ThriveGuild's lending decisions have the sample space $\Omega = \{\text{approve}, \text{deny}\}$. The sample space for one of the applicant features, employment status, is $\Omega = \{\text{employed}, \text{unemployed}, \text{other}\}$.

Toward quantification of sample spaces and safety, the *cardinality* or *size* of a set is the number of elements it contains, and is denoted by double bars $\|\cdot\|$. A *finite* set contains a natural number of elements. An example is the set $\{12, 44, 82\}$ which contains three elements, so $\|\{12, 44, 82\}\| = 3$. An *infinite* set contains an infinite number of elements. A *countably* infinite set, although infinite, contains elements that you can start counting, by calling the first element 'one,' the second element 'two,' the third element 'three,' and so on indefinitely without end. An example is the set of integers. *Discrete* values are from either finite sets or countably infinite sets. An *uncountably* infinite set is so dense that you can't even count the elements. An example is the set of real numbers. Imagine counting all the real numbers between 2 and 3—you cannot ever enumerate all of them. *Continuous* values are from uncountably infinite sets.

An *event* is a set of outcomes (a subset of the sample space Ω). For example, one event is the set of outcomes $\{\text{employed}, \text{unemployed}\}$. Another event is the set of outcomes $\{\text{employed}, \text{other}\}$. A set containing a single outcome is also an event. You can assign a cost to either an outcome or to an event. Sometimes these costs are obvious because they relate to some other quantitative loss or gain in units such as money. Other times, they are more subjective: how do you really quantify the cost of the loss of life? Getting these costs can be very difficult because it requires people and society to provide their value judgements numerically. Sometimes, relative costs rather than absolute costs are enough. Again, only undesirable outcomes or events with high enough costs are considered to be harms.

3.2.2 Aleatoric Uncertainty and Probability

Aleatoric uncertainty is quantified using a numerical assessment of the likelihood of occurrence of event A, known as the *probability* $P(A)$. It is the ratio of the cardinality of the event A to the cardinality of the sample space Ω:[5]

$$P(A) = \frac{\|A\|}{\|\Omega\|}.$$

Equation 3.1

The properties of the probability function are:

1. $P(A) \geq 0$,
2. $P(\Omega) = 1$, and
3. if A and B are disjoint events (they have no outcomes in common; $A \cap B = \emptyset$), then $P(A \cup B) = P(A) + P(B)$.

[5]Equation 3.1 is only valid for finite sample spaces, but the same high-level idea holds for infinite sample spaces.

These three properties are pretty straightforward and just formalize what we normally mean by probability. A probability of an event is a number between zero and one. The probability of one event *or* another event happening is the sum of their individual probabilities as long as the two events don't contain any of the same outcomes.

The *probability mass function* (pmf) makes life easier in describing probability for discrete sample spaces. It is a function p that takes outcomes ω as input and gives back probabilities for those outcomes. The sum of the pmf across all outcomes in the sample space is one, $\sum_{\omega \in \Omega} p(\omega) = 1$, which is needed to satisfy the second property of probability.

The probability of an event is the sum of the pmf values of its constituent outcomes. For example, if the pmf of employment status is $p(\text{employed}) = 0.60$, $p(\text{unemployed}) = 0.05$, and $p(\text{other}) = 0.35$, then the probability of event $\{\text{employed}, \text{other}\}$ is $P(\{\text{employed}, \text{other}\}) = 0.60 + 0.35 = 0.95$. This way of adding pmf values to get an overall probability works because of the third property of probability.

Random variables are a really useful concept in specifying the safety requirements of machine learning problems. A random variable X takes on a specific numerical value x when X is measured or observed; that numerical value is random. The set of all possible values of X is $\mathcal{X}$. The probability function for the random variable X is denoted P_X. Random variables can be discrete or continuous. They can also represent categorical outcomes by mapping the outcome values to a finite set of numbers, e.g. mapping {employed, unemployed, other} to $\{0, 1, 2\}$. The pmf of a discrete random variable is written as $p_X(x)$.

Pmfs don't exactly make sense for uncountably infinite sample spaces. So the *cumulative distribution function* (cdf) is used instead. It is the probability that a continuous random variable X takes a value less than or equal to some sample point x, i.e. $F_X(x) = P(X \leq x)$. An alternative representation is the *probability density function* (pdf) $p_X(x) = \frac{d}{dx} F_X(x)$, the derivative of the cdf with respect to x.[6] The value of a pdf is not a probability, but integrating a pdf over a set yields a probability.

To better understand cdfs and pdfs, let's look at one of the ThriveGuild features you're going to use in your machine learning lending model: the income of the applicant. Income is a continuous random variable whose cdf may be, for example:[7]

$$F_X(x) = \begin{cases} 1 - e^{-0.5x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}.$$

Equation 3.2

Figure 3.1 shows what this distribution looks like and how to compute probabilities from it. It shows that the probability that the applicant's income is less than or equal to 2 (in units such as ten thousand dollars) is $1 - e^{-0.5 \cdot 2} = 1 - e^{-1} \approx 0.63$. Most borrowers tend to earn less than 2. The pdf is the derivative of the cdf:

---

[6] I overload the notation p_X; it should be clear from the context whether I'm referring to a pmf or pdf.

[7] This specific choice is an exponential distribution. The general form of an exponential distribution is: $p_X(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$, for any $\lambda > 0$.

$$p_X(x) = \begin{cases} 0.5e^{-0.5x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}.$$

Equation 3.3

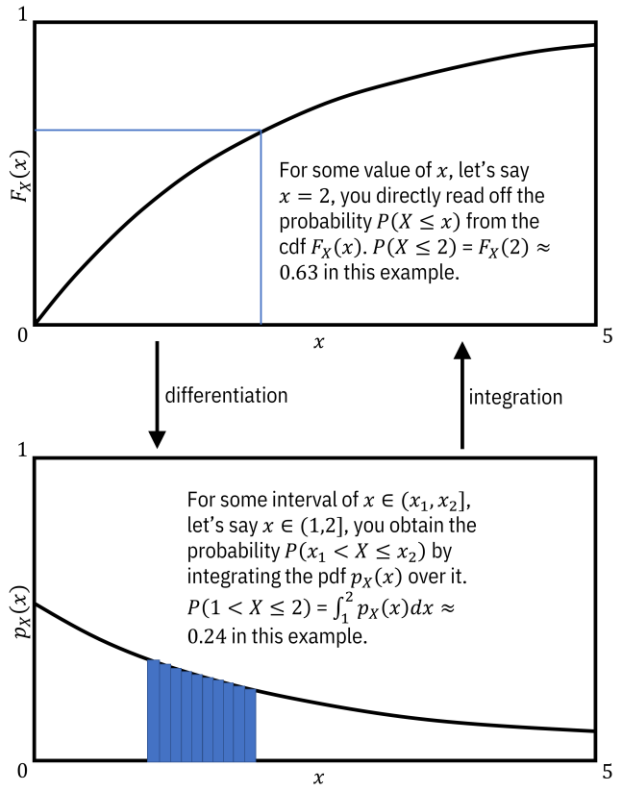


Figure 3.1. *An example cdf and corresponding pdf from the ThriveGuild income distribution example.* Accessible caption. A graph at the top shows the cdf and a graph at the bottom shows its corresponding pdf. Differentiation is the operation to go from the top graph to the bottom graph. Integration is the operation to go from the bottom graph to the top graph. The top graph shows how to read off a probability directly from the value of the cdf. The bottom graph shows that obtaining a probability requires integrating the pdf over an interval.

Joint pmfs, cdfs, and pdfs of more than one random variable are *multivariate* functions and can contain a mix of discrete and continuous random variables. For example, $p_{X,Y,Z}(x, y, z)$ is the notation for the pdf of three random variables X, Y, and Z. To obtain the pmf or pdf of a subset of the random variables, you sum the pmf or integrate the pdf over the rest of the variables outside of the subset you want to keep. This act of summing or integrating is known as *marginalization* and the resulting probability distribution is called the *marginal* distribution. You should contrast the use of the term

'marginalize' here with the social marginalization that leads individuals and groups to be made powerless by being treated as insignificant.

The employment status feature and the loan approval label in the ThriveGuild model are random variables that have a joint pmf. For example, this multivariate function could be $p(\text{employed, approve}) = 0.20$, $p(\text{employed, deny}) = 0.40$, $p(\text{unemployed, approve}) = 0.01$, $p(\text{unemployed, deny}) = 0.04$, $p(\text{other, approve}) = 0.10$, and $p(\text{other, deny}) = 0.25$. This function is visualized as a table of probability values in Figure 3.2. Summing loan approval out from this joint pmf, you recover the marginal pmf for employment status given earlier. Summing employment status out, you get the marginal pmf for loan approval as $p(\text{approve}) = 0.31$ and $p(\text{deny}) = 0.69$.

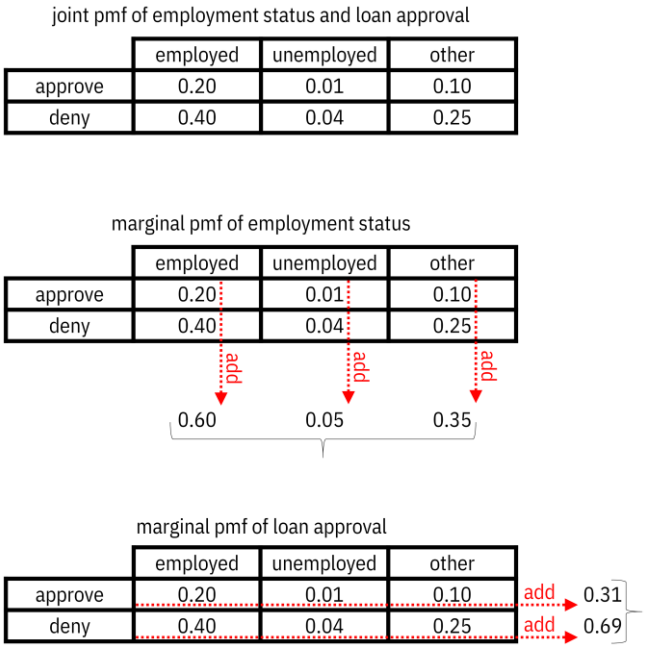


Figure 3.2. *Examples of marginalizing a joint distribution by summing out one of the random variables.* Accessible caption. A table of the joint pmf has employment status as the columns and loan approval as the rows. The entries are the probabilities. Adding the numbers in the columns gives the marginal pmf of employment status. Adding the numbers in the rows gives the marginal pmf of loan approval.

Probabilities, pmfs, cdfs, and pdfs are all tools for quantifying aleatoric uncertainty. They are used to specify the requirements for the accuracy of models, which is critical for the first of the two parts of safety: risk minimization. A correct prediction is an event and the probability of that event is the accuracy. For example, working with the problem owner, you may specify that the ThriveGuild lending model must have at least a 0.92 probability of being correct. The accuracy of machine learning models and other similar measures of basic performance are the topic of Chapter 6 in Part 3 of the book.

3.2.3 Epistemic Uncertainty and Possibility

Aleatoric uncertainty is concerned with chance whereas epistemic uncertainty is concerned with imprecision, ignorance, and lack of knowledge. Probabilities are good at capturing notions of randomness, but betray us in representing a lack of knowledge. Consider the situation in which you have no knowledge of the employment and unemployment rates in a remote country. It is not appropriate for you to assign any probability distribution to the outcomes employed, unemployed, and other, not even equal probabilities to the possible outcomes because that would express a precise knowledge of equal chances. The only thing you can say is that the outcome will be from the set $\Omega = \{\text{employed, unemployed, other}\}$.

Thus, epistemic uncertainty is best represented using sets without any further numeric values. You might be able to specify a smaller subset of outcomes, but not have precise knowledge of likelihoods within the smaller set. In this case, it is not appropriate to use probabilities. The subset distinguishes between outcomes that are possible and those that are impossible.

Just like our friend, the real-valued probability function $P(A)$ for aleatoric uncertainty, there is a corresponding *possibility function* $\Pi(A)$ for epistemic uncertainty which takes either the value 0 or the value 1. A value 0 denotes an impossible event and a value 1 denotes a possible event. In a country in which the government offers employment to anyone who seeks it, the possibility of unemployment $\Pi(\text{unemployed})$ is zero. The possibility function satisfies its own set of three properties, which are pretty similar to the three properties of probability:

1. $\Pi(\emptyset) = 0$,
2. $\Pi(\Omega) = 1$, and
3. if A and B are disjoint events (they have no outcomes in common; $A \cap B = \emptyset$), then $\Pi(A \cup B) = \max\big(\Pi(A), \Pi(B)\big)$.

One difference is that the third property of possibility contains maximum, whereas the third property of probability contains addition. Probability is *additive*, but possibility is *maxitive*. The probability of an event is the sum of the probabilities of its constituent outcomes, but the possibility of an event is the maximum of the possibilities of its constituent outcomes. This is because possibilities can only be zero or one. If you have two events, both of which have possibility equal to one, and you want to know the possibility of one or the other occurring, it does not make sense to add one plus one to get two, you should take the maximum of one and one to get one.

You should use possibility in specifying requirements for the ThriveGuild machine learning system to address the epistemic uncertainty (reliability) side of the two-part definition of safety. For example, there will be epistemic uncertainty in what the best possible model parameters are if there is not enough of the right training data. (The data you ideally want to have is from the present, from a fair and just world, and that has not been corrupted. However, you're almost always out of luck and have data from the past, from an unjust world, or that has been corrupted.) The data that you have can bracket the possible set of best parameters through the use of the possibility function. Your data tells you that one set of model parameters is possibly the best set of parameters, and that it is impossible for other different sets of model parameters to be the best. Problem specifications can place limits on the cardinality of the possibility set. Dealing with epistemic uncertainty in machine learning is the topic of Part 4 of the book in the context of generalization, fairness, and adversarial robustness.

3.3 Summary Statistics of Uncertainty

Full probability distributions are great to get going with problem specification, but can be unwieldy to deal with. It is easier to set problem specifications using *summary statistics* of probability distributions and random variables.

3.3.1 Expected Value and Variance

The most common statistic is the *expected value* of a random variable. It is the *mean* of its distribution: a typical value or long-run average outcome. It is computed as the integral of the pdf multiplied by the random variable:

$$E[X] = \int_{-\infty}^{\infty} x p_X(x) dx.$$

Equation 3.4

Recall that in the example earlier, ThriveGuild borrowers had the income pdf $0.5e^{-0.5x}$ for $x \geq 0$ and zero elsewhere. The expected value of income is thus $\int_0^{\infty} x 0.5e^{-0.5x}\, dx = 2$.[8] When you have a bunch of samples drawn from the probability distribution of X, denoted $\{x_1, x_2, \ldots, x_n\}$, then you can compute an empirical version of the expected value, the *sample mean*, as $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$. Not only can you compute the expected value of a random variable alone, but also the expected value of any function of a random variable. It is the integral of the pdf multiplied by the function. Through expected values of performance, also known as *risk*, you can specify average behaviors of systems being within certain ranges for the purposes of safety.

How much variability in income should you plan for among ThriveGuild applicants? An important expected value is the *variance* $E[(X - E[X])^2]$, which measures the spread of a distribution and helps answer the question. Its sample version, the *sample variance* is computed as $\frac{1}{n-1}\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2$. The *correlation* between two random variables X (e.g., income) and Y (e.g., loan approval) is also an expected value, $E[XY]$, which tells you whether there is some sort of statistical relationship between the two random variables. The *covariance*, $E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]$, tells you whether if one random variable increases, the other will also increase, and vice versa. These different expected values and summary statistics give different insights about aleatoric uncertainty that are to be constrained in the problem specification.

3.3.2 Information and Entropy

Although means, variances, correlations, and covariances capture a lot, there are other kinds of summary statistics that capture different insights needed to specify machine learning problems. A different way to summarize aleatoric uncertainty is through the *information* of random variables. Part of information theory, the information of a discrete random variable X with pmf $p_X(x)$ is $I(x) = -\log\left(p_X(x)\right)$. This logarithm is usually in base 2. For very small probabilities close to zero, the information is very large. This makes sense since the occurrence of a rare event (an event with small probability) is deemed

[8] The expected value of a generic exponentially-distributed random variable is $1/\lambda$.

very informative. For probabilities close to one, the information is close to zero because common occurrences are not informative. Do you go around telling everyone that you did not win the lottery? Probably not, because it is not informative. The expected value of the information of X is its *entropy*:

$$H(X) = E[I(X)] = -\sum_{x \in \mathcal{X}} p_X(x) \log\big(p_X(x)\big).$$

Equation 3.5

Uniform distributions with equal probability for all outcomes have maximum entropy among all possible distributions. The difference between the maximum entropy achieved by the uniform distribution and the entropy of a given random variable is the *redundancy*. It is known as the *Theil index* when used to summarize inequality in a population. For a discrete random variable X taking non-negative values, which is usually the case when measuring assets, income, or wealth of individuals, the Theil index is:

$$\text{Theil index} = \sum_{x \in \mathcal{X}} p_X(x) \frac{x}{E[X]} \log\left(\frac{x}{E[X]}\right),$$

Equation 3.6

where $\mathcal{X} = \{0,1, \ldots, \infty\}$ and the logarithm is the natural logarithm. The index's values range from zero to one. The entropy-maximizing distribution in which all members of a population have the same value, which is the mean value, has zero Theil index and represents the most equality. A Theil index of one represents the most inequality. It is achieved by a pmf with one non-zero value and all other zero values. (Think of one lord and many serfs.) In Chapter 10, you'll see how to use the Theil index to specify machine learning systems in terms of their individual fairness and group fairness requirements together.

3.3.3 Kullback-Leibler Divergence and Cross-Entropy

The *Kullback-Leibler (K-L) divergence* compares two probability distributions and gives a different avenue for problem specification. For two discrete random variables defined on the same sample space with pmfs $p(x)$ and $q(x)$, the K-L divergence is:

$$D(p \parallel q) = -\sum_{x \in \mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right).$$

Equation 3.7

It measures how similar or different two distributions are. Similarity of one distribution to a reference distribution is often a requirement in machine learning systems.

The *cross-entropy* is another quantity defined for two random variables on the same sample space that represents the average information in one random variable with pmf $p(x)$ when described using a different random variable $q(x)$:

$$H(p \parallel q) = -\sum_{x \in \mathcal{X}} p(x) \log\big(q(x)\big).$$

Equation 3.8

As such, it is the entropy of the first random variable plus the K-L divergence between the two variables:

$$H(p \parallel q) = H(p) + D(p \parallel q).$$

Equation 3.9

When $p = q$, then $H(p \parallel q) = H(p)$ because the K-L divergence term goes to zero and there is no remaining mismatch between p and q. Cross-entropy is used as an objective for training neural networks as you'll see in Chapter 7.

3.3.4 Mutual Information

As the last summary statistic of aleatoric uncertainty in this section, let's talk about *mutual information*. It is the K-L divergence between a joint distribution $p_{X,Y}(x, y)$ and the product of its marginal distributions $p_X(x)p_Y(y)$:

$$I(X, Y) = D\left(p_{X,Y}(x, y) \parallel p_X(x)p_Y(y)\right).$$

Equation 3.10

It is symmetric in its two arguments and measures how much information is shared between X and Y. In Chapter 5, mutual information is used to set a constraint on privacy: the goal of not sharing information. It crops up in many other places as well.

3.4 Conditional Probability

When you're looking at all the different random variables available to you as you develop ThriveGuild's lending system, there will be many times that you get more information by measuring or observing some random variables, thereby reducing your epistemic uncertainty about them. Changing the possibilities of one random variable through observation can in fact change the probability of another random variable. The random variable Y given that the random variable X takes value x is not the same as just the random variable Y on its own. The probability that you would approve a loan application without knowing any specifics about the applicant is different from the probability of your decision if you knew, for example, that the applicant is employed.

This updated probability is known as a *conditional probability* and is used to quantify a probability when you have additional information that the outcome is part of some event. The conditional probability of event A given event B is the ratio of the cardinality of the joint event A and B, to the cardinality of the event B:[9]

$$P(A \mid B) = \frac{\|A \cap B\|}{\|B\|} = \frac{P(A \cap B)}{P(B)}.$$

Equation 3.11

In other words, the sample space changes from Ω to B, so that is why the denominator of Equation 3.1 ($\|A\|/\|\Omega\|$) changes from Ω to B in Equation 3.11. The numerator $\|A \cap B\|$ captures the part of the event A that is within the new sample space B. There are similar conditional versions of pmfs, cdfs, and pdfs defined for random variables.

Through conditional probability, you can reason not only about distributions and summaries of uncertainty, but also how they change when observations are made, outcomes are revealed, and evidence is collected. Using a machine learning model is similar to getting the conditional probability of the label given the feature values of an input data point. The probability of loan approval given the features for one specific applicant being employed with an income of 15,000 dollars is a conditional probability.

In terms of summary statistics, the *conditional entropy* of Y given X is:

$$H(Y \mid X) = -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p_{Y,X}(y,x) \log\left(\frac{p_{Y,X}(y,x)}{p_X(x)}\right).$$

Equation 3.12

It represents the average information remaining in Y given that X is observed.

Mutual information can also be written using conditional entropy as:

$$I(X,Y) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y).$$

Equation 3.13

In this form, you can see that mutual information quantifies the reduction in entropy in a random variable by conditioning on another random variable. In this role, it is also known as *information gain*, and used as a criterion for learning decision trees in Chapter 7. Another common criterion for learning decision trees is the *Gini index*:

[9]Event B has to be non-empty and the sample space has to be finite for this definition to be applicable.

$$\text{Gini index} = 1 - \sum_{x \in \mathcal{X}} p_X^2(x).$$

Equation 3.14

3.5 Independence and Bayesian Networks

Understanding uncertainty of random variables becomes easier if you can determine that some of them are unlinked. For example, if certain features are unlinked to other features and also to the label, then they do not have to be considered in a machine learning problem specification.

3.5.1 Statistical Independence

Towards the goal of understanding unlinked variables, let's define the important concept called *statistical independence*. Two events are mutually independent if one outcome is not informative of the other outcome. The statistical independence between two events is denoted $A \perp\!\!\!\perp B$ and is defined by

$$A \perp\!\!\!\perp B \Leftrightarrow P(A \mid B) = P(A).$$

Equation 3.15

Knowledge of the tendency of A to occur given that B has occurred is not changed by knowledge of B. If in ThriveGuild's data, $P(\text{employed} \mid \text{deny}) = 0.50$ and $P(\text{employed}) = 0.60$, then since the two numbers 0.50 and 0.60 are not the same, employment status and loan approval are not independent, they are dependent. Employment status *is* used in loan approval decisions. The definition of conditional probability further implies that:

$$A \perp\!\!\!\perp B \Leftrightarrow P(A, B) = P(A)P(B).$$

Equation 3.16

The probability of the joint event is the product of the marginal probabilities. Moreover, if two random variables are independent, their mutual information is zero.

The concept of independence can be extended to more than two events. Mutual independence among several events is more than simply a collection of pairwise independence statements; it is a stronger notion. A set of events is mutually independent if any of the constituent events is independent of all subsets of events that do not contain that event. The pdfs, cdfs, and pmfs of mutually independent random variables can be written as the products of the pdfs, cdfs, and pmfs of the individual constituent random variables. One commonly used assumption in machine learning is of *independent and identically distributed* (i.i.d.) random variables, which in addition to mutual independence, states that all of the random variables under consideration have the same probability distribution.

A further concept is *conditional independence*, which involves at least three events. The events A and B are conditionally independent given C, denoted $A \perp\!\!\!\perp B \mid C$, when knowledge of the tendency of A to occur given that B has occurred is not changed by knowledge of B precisely when it is known that C

occurred. Similar to the unconditional case, the probability of the joint conditional event is the product of the marginal conditional probabilities under conditional independence.

$$A \perp\!\!\!\perp B \mid C \Leftrightarrow P(A \cap B \mid C) = P(A \mid C)P(B \mid C).$$

Equation 3.17

Conditional independence also extends to random variables and their pmfs, cdfs, and pdfs.

3.5.2 Bayesian Networks

To get the full benefit of the simplifications from independence, you should trace out all the different dependence and independence relationships among the applicant features and the loan approval decision. *Bayesian networks*, also known as directed probabilistic *graphical models*, serve this purpose. They are a way to represent a joint probability of several events or random variables in a structured way that utilizes conditional independence. The name graphical model arises because each event or random variable is represented as a node in a graph and edges between nodes represent dependencies, shown in the example of Figure 3.3, where A_1 is income, A_2 is employment status, A_3 is loan approval, and A_4 is gender. The edges have an orientation or direction: beginning at *parent* nodes and ending at *child* nodes. Employment status and gender have no parents; employment status is the parent of income; both income and employment status are the parents of loan approval. The set of parents of the argument node is denoted $pa(\cdot)$.

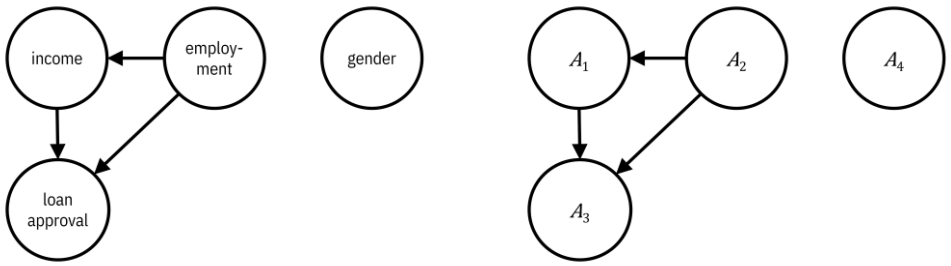


Figure 3.3. *An example graphical model consisting of four events. The employment status and gender nodes have no parents; employment status is the parent of income, and thus there is an edge from employment status to income; both income and employment status are the parents of loan approval, and thus there are edges from income and from employment status to loan approval. The graphical model is shown on the left with the names of the events and on the right with their symbols.*

The statistical relationships are determined by the graph structure. The probability of several events $A_1, \ldots, A_n$ is the product of all the events conditioned on their parents:

$$P(A_1, \ldots, A_n) = \prod_{j=1}^{n} P\left(A_j \mid pa(A_j)\right).$$

Equation 3.18

As a special case of Equation 3.18 for the graphical model in Figure 3.3, the corresponding probability may be written as $P(A_1, A_2, A_3, A_4) = P(A_1 \mid A_2)P(A_2)P(A_3 \mid A_1, A_2)P(A_4)$. Valid probability distributions lead to directed *acyclic* graphs. Graphs are acyclic if you follow a path of arrows and can never return to nodes you started from. An *ancestor* of a node is any node that is its parent, parent of its parent, parent of its parent of its parent, and so on recursively.

From the small and simple graph structure in Figure 3.3, it is clear that the loan approval depends on both income and employment status. Income depends on employment status. Gender is independent of everything else. Making independence statements is more difficult in larger and more complicated graphs, however. Determining all of the different independence relationships among all the events or random variables is done through the concept of *d-separation*: a subset of nodes S_1 is independent of another subset of nodes S_2 conditioned on a third subset of nodes S_3 if S_3 d-separates S_1 and S_2. One way to explain d-separation is through the three different motifs of three nodes each shown in Figure 3.4, known as a *causal chain*, *common cause*, and *common effect*. The differences among the motifs are in the directions of the arrows. The configurations on the left have no node that is being conditioned upon, i.e. no node's value is observed. In the configurations on the right, node A_3 is being conditioned upon and is thus shaded. The causal chain and common cause motifs without conditioning are *connected*. The causal chain and common cause with conditioning are *separated*: the path from A_1 to A_2 is *blocked* by the knowledge of A_3. The common effect motif without conditioning is separated; in this case, A_3 is known as a *collider*. Common effect with conditioning is connected; moreover, conditioning on any descendant of A_3 yields a connected path between A_1 and A_2. Finally, a set of nodes S_1 and S_2 is d-separated conditioned on a set of nodes S_3 if and only if each node in S_1 is separated from each node in S_2.[10]

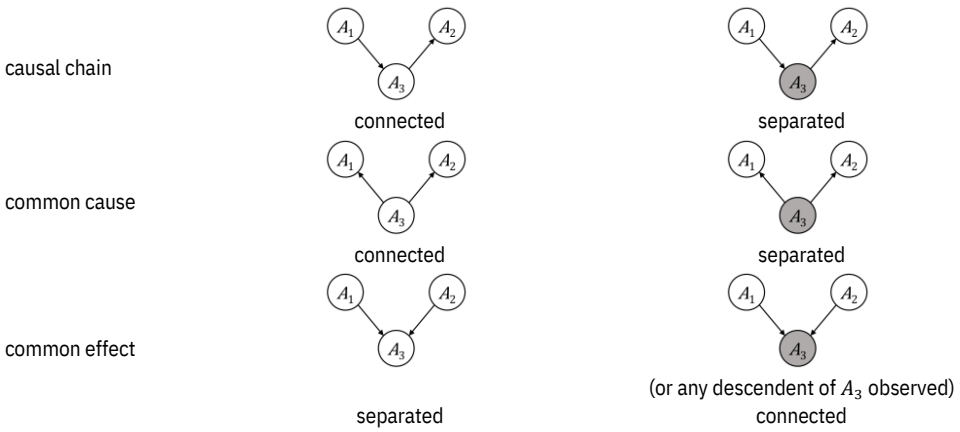


Figure 3.4. *Configurations of nodes and edges that are connected and separated. Nodes colored gray have been observed.* Accessible caption. The causal chain is $A_1 \rightarrow A_3 \rightarrow A_2$; it is connected when A_3 is unobserved and separated when A_3 is observed. The common cause is $A_1 \leftarrow A_3 \rightarrow A_2$; it is connected when A_3 is unobserved and separated when A_3 is observed. The common effect is $A_1 \rightarrow A_3 \leftarrow A_2$; it is separated when A_3 is unobserved and connected when A_3 or any of its descendants are observed.

[10]There may be dependence not captured in the structure if one random variable is a deterministic function of another.

Although d-separation among two sets of nodes can be checked by checking all three-node motifs along all paths between the two sets, there is a more constructive algorithm to check for d-separation.

1. Construct the *ancestral graph* of S_1, S_2, and S_3. This is the subgraph containing the nodes in S_1, S_2, and S_3 along with all of their ancestors and all of the edges among these nodes.
2. For each pair of nodes with a common child, draw an undirected edge between them. This step is known as *moralization*.[11]
3. Make all edges undirected.
4. Delete all S_3 nodes.
5. If S_1 and S_2 are separated in the undirected sense, then they are d-separated.

An example is shown in Figure 3.5.

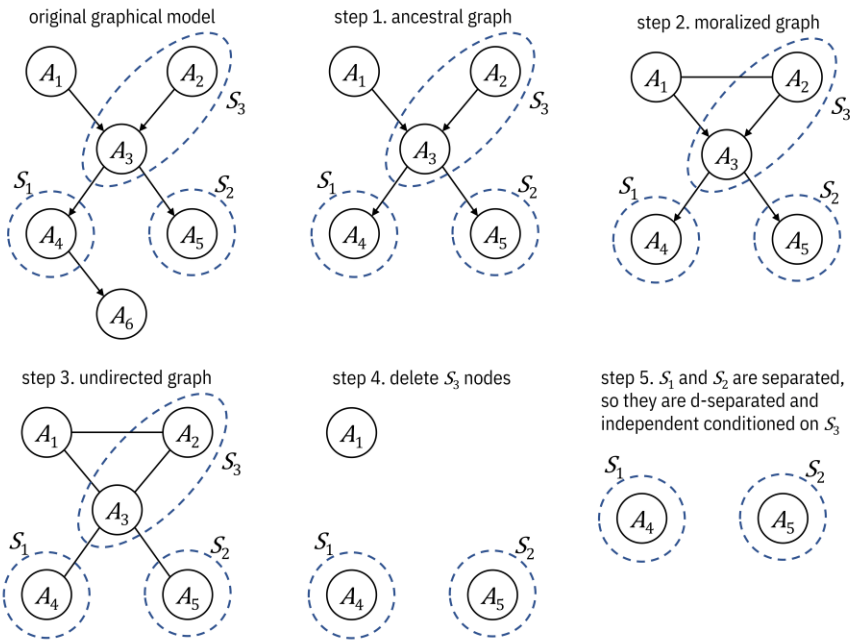


Figure 3.5. *An example of running the constructive algorithm to check for d-separation.* Accessible caption. The original graph has edges from A_1 and A_2 to A_3, from A_3 to A_4 and A_5, and from A_4 to A_6. S_1 contains only A_4, S_2 contains only A_5, and S_3 contains A_2 and A_3. After step 1, A_6 is removed. After step 2, an undirected edge is drawn between A_1 and A_2. After step 3, all edges are undirected. After step 4, only A_1, A_4, and A_5 remain and there are no edges. After step 5, only A_4 and A_5, and equivalently S_1 and S_2, remain and there is no edge between them. They are separated, so S_1 and S_2 are d-separated conditioned on S_3.

[11]The term moralization reflects a value of some but not all societies: that it is moral for the parents of a child to be married.

3.5.3 Conclusion

Independence and conditional independence allow you to know whether random variables affect one another. They are fundamental relationships for understanding a system and knowing which parts can be analyzed separately while determining a problem specification. One of the main benefits of graphical models is that statistical relationships are expressed through structural means. Separations are more clearly seen and computed efficiently.

3.6 Summary

- The first two attributes of trustworthiness, accuracy and reliability, are captured together through the concept of safety.
- Safety is the minimization of the aleatoric uncertainty and the epistemic uncertainty of undesired high-stakes outcomes.
- Aleatoric uncertainty is inherent randomness in phenomena. It is well-modeled using probability theory.
- Epistemic uncertainty is lack of knowledge that can, in principle, be reduced. Often in practice, however, it is not possible to reduce epistemic uncertainty. It is well-modeled using possibility theory.
- Problem specifications for trustworthy machine learning systems can be quantitatively expressed using probability and possibility.
- It is easier to express these problem specifications using statistical and information-theoretic summaries of uncertainty than full distributions.
- Conditional probability allows you to update your beliefs when you receive new measurements.
- Independence and graphical models encode random variables not affecting one another.

4

Data Sources and Biases

The mission of the (fictional) non-profit organization Unconditionally is charitable giving. It collects donations and distributes unconditional cash transfers—funds with no strings attached—to poor households in East Africa. The recipients are free to do whatever they like with the money. Unconditionally is undertaking a new machine learning project to identify the poorest of the poor households to select for the cash donations. The faster they can complete the project, the faster and more efficiently they can move much-needed money to the recipients, some of whom need to replace their thatched roofs before the rainy season begins.

The team is in the data understanding phase of the machine learning lifecycle. Imagine that you are a data scientist on the team pondering which data sources to use as features and labels to estimate the wealth of households. You examine all sorts of data including daytime satellite imagery, nighttime illumination satellite imagery, national census data, household survey data, call detail records from mobile phones, mobile money transactions, social media posts, and many others. What will you choose and why? Will your choices lead to unintended consequences or to a trustworthy system?

The data understanding phase is a really exciting time in the lifecycle. The problem goals have been defined; working with the data engineers and other data scientists, you cannot wait to start acquiring data and conducting exploratory analyses. Having data is a prerequisite for doing machine learning, but not any data will do. It is important for you and the team to be careful and intentional at this point. Don't take shortcuts. Otherwise, before you know it, you will have a glorious edifice built upon a rocky foundation.

> "Garbage in, garbage out."
>
> —Wilf Hey, computer scientist at IBM

This chapter begins Part 2 of the book focused on all things data (remember the organization of the book shown in Figure 4.1).

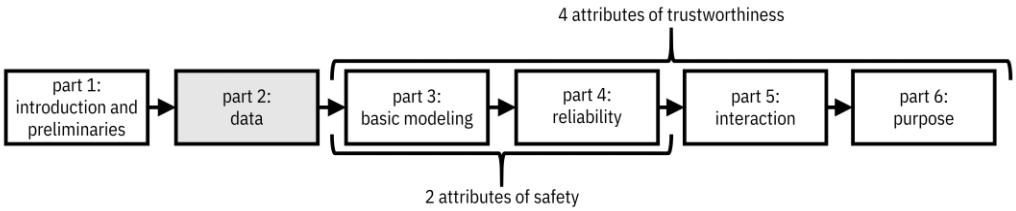


Figure 4.1. *Organization of the book. This second part focuses on different considerations of trustworthiness when working with data.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 2 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

The chapter digs into how you and Unconditionally's data engineers and other data scientists should:

- use knowledge of characteristics of different data modalities to evaluate datasets,
- select among different sources of data, and
- appraise datasets for biases and validity.

Appraising data sets for biases is critical for trustworthiness and is the primary focus of the chapter. The better job done at this stage, the less correction and mitigation of harms needs to be done in later stages of the lifecycle. Bias evaluation should include input from affected individuals of the planned machine learning system. If all possible relevant data is deemed too biased, a conversation with the problem owner and other stakeholders on whether to even proceed with the project is a must. (Data privacy and consent are investigated in Chapter 5.)

4.1 Modalities

Traditionally, when most people imagine data, they imagine tables of numbers in an accounting spreadsheet coming out of some system of record. However, data for machine learning systems can include digital family photographs, surveillance videos, tweets, legislative documents, DNA strings, event logs from computer systems, sensor readings over time, structures of molecules, and any other information in digital form. In the machine learning context, data is assumed to be a finite number of samples drawn from any underlying probability distribution.

The examples of data given above come from different *modalities* (images, text, time series, etc.). A modality is a category of data defined by how it is received, represented, and understood. Figure 4.2 presents a mental model of different modalities. There are of course others that are missing from the figure.

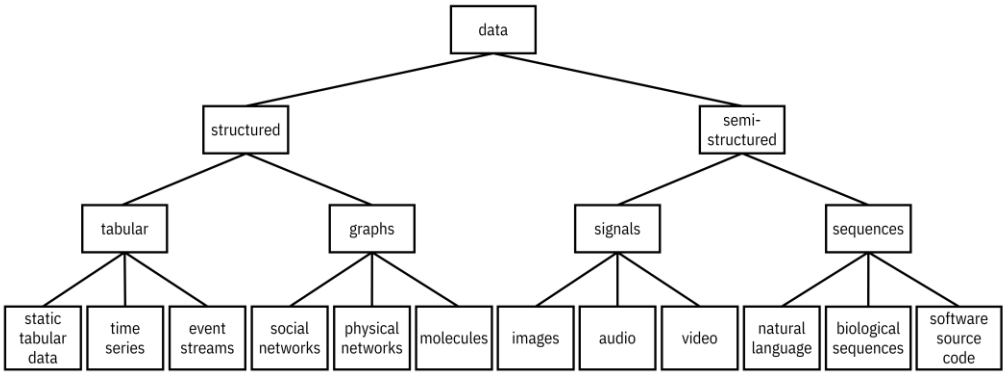


Figure 4.2. *A mental model of different modalities of data.* Accessible caption. A hierarchy diagram with data at its root. Data has children structured and semi-structured. Structured has children tabular and graphs. Tabular has children static tabular data, time series, and event streams. Graphs has children social networks, physical networks, and molecules. Semi-structured has children signals and sequences. Signals has children images, audio, and video. Sequences has children natural language, biological sequences, and software source code.

One of Unconditionally's possible datasets is from a household survey. It is an example of the *static tabular data* modality and part of the *structured* data category of modalities.[1] It is static because it is not following some time-varying phenomenon. The *columns* are different attributes that can be used as features and labels, and the *rows* are different records or sample points, i.e. different people and households. The columns contain numeric values, ordinal values, categorical values, strings of text, and special values such as dates. Although tabular data might look official, pristine, and flawless at first glance due to its nice structure, it can hide all sorts of false assumptions, errors, omissions, and biases.

Time series constitute another modality that can be stored in tabular form. As measurements at regular intervals in time (usually of numeric values), such data can be used to model trends and forecast quantities in time. *Longitudinal* or *panel* data, repeated measurements of the same individuals over time, are often time series. Household surveys are rarely longitudinal however, because they are logistically difficult to conduct. *Cross-sectional* surveys, simply several tabular datasets taken across time but without any linking, are logistically much easier to collect because the same individuals do not have to be tracked down.

Another of Unconditionally's possible datasets is mobile money transactions. Time stamps are a critical part of transactions data, but are not time series because they do not occur at regular intervals. Every mobile money customer asynchronously generates an event whenever they receive or disburse funds, not mediated by any common clock across customers. Transaction data is an example of the *event stream* modality. In addition to a time stamp, event streams contain additional values that are measured such as monetary amount, recipient, and items purchased. Other event streams include clinical tests conducted in a hospital and social services received by clients.

[1]There are modalities with even richer structure than tabular data, such as graphs that can represent social networks and the structure of chemical molecules.

Unconditionally can estimate poverty using satellite imagery. Digital *images* are the modality that spurred a lot of the glory showered upon machine learning in the past several years. They are part of the *semi-structured* branch of modalities. In general, images can be regular optical images or ones measured in other ranges of the electromagnetic spectrum. They are composed of numeric pixel values across various channels in their raw form and tend to contain a lot of spatial structure. *Video*, an image sequence over time, has a lot of spatiotemporal structure. Modern machine learning techniques learn these spatial and spatiotemporal representations by being trained on vast quantities of data, which may themselves contain unwanted biases and unsuitable content. (The model for the specified problem is a fine-tuned version of the model pre-trained on the large-scale, more generic dataset. These large pre-trained models are referred to as *foundation models*.) Videos may also contain *audio* signals.

One of your colleagues at Unconditionally imagines that although less likely, the content of text messages and social media posts might predict a person's poverty level. This modality is *natural language* or *text*. Longer documents, including formal documents and publications, are a part of the same modality. The syntax, semantics, and pragmatics of human language is complicated. One way of dealing with text includes parsing the language and creating a syntax tree. Another way is representing text as sparse structured data by counting the existence of individual words, pairs of words, triplets of words, and so on in a document. These *bag-of-words* or *n-gram* representations are currently being superseded by a third way: sophisticated *large language models*, a type of foundation model, trained on vast corpora of documents. Just like in learning spatial representations of images, the learning of language models can be fraught with many different biases, especially when the norms of the language in the training corpus do not match the norms of the application. A language model trained on a humongous pile of newspaper articles from the United States will typically not be a good foundation for a representation for short, informal, code-mixed text messages in East Africa.[2]

Typically, structured modalities are their own representations for modeling and correspond to deliberative decision making by people, whereas semi-structured modalities require sophisticated transformations and correspond to instinctive perception by people. These days, the sophisticated transformations for semi-structured data tend to be learned using deep neural networks that are trained on unimaginably large datasets. This process is known as *representation learning*. Any biases present in the very large background datasets carry over to models fine-tuned on a problem-specific dataset because of the originally opaque and uncontrollable representation learning leading to the foundation model. As such, with semi-structured data, it is important that you not only evaluate the problem-specific dataset, but also the background dataset. With structured datasets, it is more critical that you analyze data preparation and feature engineering.[3]

4.2 Data Sources

Not only do the various kinds of data being considered by Unconditionally vary by modality, they also vary by how and where they come from, i.e., their *provenance*. As part of data understanding, your team

[2]Other strings with language-like characteristics such as DNA or amino acid sequences and software source code are currently being approached through techniques similar to natural language processing.

[3]There are new foundation models for structured modalities. Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. "Tabular Transformers for Modeling Multivariate Time Series." In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Jun. 2021, pp. 3565–3569.

at Unconditionally must evaluate your *data sources* carefully to separate the wheat from the chaff: only including the good stuff. There are many different categories of data sources, which imply different considerations in assessing their quality.

4.2.1 Purposefully Collected Data

You may think that most data used in creating machine learning systems is expressly and carefully collected for the purpose of the problem, but you would be blissfully wrong. In fact, most data used in machine learning systems is repurposed. Some of the exceptions include data collected through surveys and censuses. These sources have the veneer of being well-designed and with minimal bias, but this might not always be the case. For example, if you rely on the recently completed national census in Kenya for either features or labels in Unconditionally's poverty prediction problem, you may suffer from non-responses and malicious data manipulation.

Another kind of purposefully collected data is generated from scientific experiments. Again, well-designed and well-conducted experiments should yield reliable data. However, there is a prevalent lack of trust in the scientific method due to practices such as misuse of data analysis to find patterns in data that can be selectively presented as statistically significant (p-hacking and the file drawer problem), lack of reproducibility, and outright fraud.

4.2.2 Administrative Data

Administrative data is the data collected by organizations about their routine operations for non-statistical reasons. Among Unconditionally's list of possible datasets, call detail records and mobile money transactions fit the bill. Data scientists and engineers frequently repurpose administrative data to train models because it is there and they can. Sometimes, it even makes sense to do so.

Since administrative data is a record of operations, which might have a direct bearing on an organization's bottom line or be subject to audit, it is usually quite correct. There are often difficulties in attempting to integrate different sources of administrative data within an organization due to their being siloed across teams. Such data can also contain traces of historical prejudices as well.

The most important thing for you to be aware of with administrative data is that it might not exactly match the predictive problem you are trying to solve. The machine learning problem specification may ask for a certain label, but the administrative data may contain columns that can only be proxies for that desired label. This mismatch can be devastating for certain individuals and groups, even if it is a decent proxy on average. For example, recall that in Chapter 2, we discussed how the number of doctor visits might not be a good proxy for how sick a patient is if there are external factors that prevent some groups from accessing health care. Also, the granularity of the records might be different than what is needed in the problem, e.g. individual phone numbers in call detail records instead of all activity by a household.

4.2.3 Social Data

Social data is data about people or created by people, and includes user-generated content, relationships between people, and traces of behavior.[4] Postings of text and images on social media platforms are a perfect example. Friendship networks and search histories are other examples. Similar to

[4]Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. "Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries." In: *Frontiers in Big Data* 2.13 (Jul. 2019).

administrative data, these sources are not produced for the problem specification, but are repurposed for predictive or causal modeling. Many a time, just like administrative data, social data is only a proxy for what the problem specification requires and can be misleading or even outright wrong. The social media content of potential recipients of Unconditionally's cash transfer you analyze may be like this.

Since social data is created for purposes like communicating, seeking jobs, and maintaining friendships, the quality, accuracy, and reliability of this data source may be much less than administrative data. Text may include various slang, non-standard dialects, misspellings, and biases. Other modalities of social data are riddled with vagaries of their own. The information content of individual data points might not be very high. Also, there can be large amounts of sampling biases because not all populations participate in social platforms to the same extent. In particular, marginalized populations may be invisible in some types of social data.

4.2.4 Crowdsourcing

Supervised learning requires both features and labels. Unlabeled data is much easier to acquire than labeled data. *Crowdsourcing* is a way to fill the gap: crowd workers label the sentiment of sentences, determine whether a piece of text is hate speech, draw boxes around objects in images, and so on.[5] They evaluate explanations and the trustworthiness of machine learning systems. They help researchers better understand human behavior and human-computer interaction. Unconditionally contracted with crowd workers to label the type of roof of homes in satellite images.

In many crowdsourcing platforms, the workers are low-skill individuals whose incentive is monetary. They sometimes communicate with each other outside of the crowdsourcing platform and behave in ways that attempt to game the system to their benefit. The wages of crowd workers may be low, which raises ethical concerns. They may be unfamiliar with the task or the social context of the task, which may yield biases in labels. For example, crowd workers may not have the context to know what constitutes a household in rural East Africa and may thus introduce biases in roof labeling. (More details on this example later.) Gaming the system may also yield biases. Despite some platforms having quality control mechanisms, if you design the labeling task poorly, you will obtain poor quality data. In some cases, especially those involving applications with a positive social impact, the crowdworkers may have higher skill and be intrinsically motivated to do a conscientious job. Nevertheless, they may still be unfamiliar with the social context or have other biases.

4.2.5 Data Augmentation

Sometimes, especially in specialized problem domains, the amount of available data is not sufficient to learn high-performing models. *Data augmentation*—performing various transformations of the given dataset—may be used to increase data set size without actually collecting additional data. In image data, transformations for augmentation include rotations, flips, shifts, warps, additions of noise, and so on. In natural language data, transformations can include replacing words with synonyms. These sorts of heuristic transformations introduce some level of your subjectivity, which may yield certain biases.

---

[5]Jennifer Wortman Vaughan. "Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research." In: *Journal of Machine Learning Research* 18.193 (May 2018).

Another way to perform data augmentation is through *generative machine learning*: using the given data to train a data generator that outputs many more samples to then be used for training a classifier. Ideally, these generated data points should be as diverse as the given dataset. However, a big problem known as *mode collapse*, which produces samples from only one part of the probability distribution of the given data, can yield severe biases in the resulting dataset.

4.2.6 Conclusion

Different data sources are useful in addressing various problem specifications, but all have biases of one kind or the other. Most data sources are repurposed. You must take care when selecting among data sources by paying attention to the more prevalent biases for any given data source. The next section describes biases from the perspective of their different kinds and where in the lifecycle they manifest.

4.3 Kinds of Biases

Your team is chugging along in the data understanding phase of the machine learning lifecycle. You know how different data modalities and data sources can go awry. These issues are your focus while appraising data for biases and lack of *validity* as it passes through various *spaces* in the machine learning lifecycle. A model of biases, validity, and spaces for you to keep in mind is given in Figure 4.3.

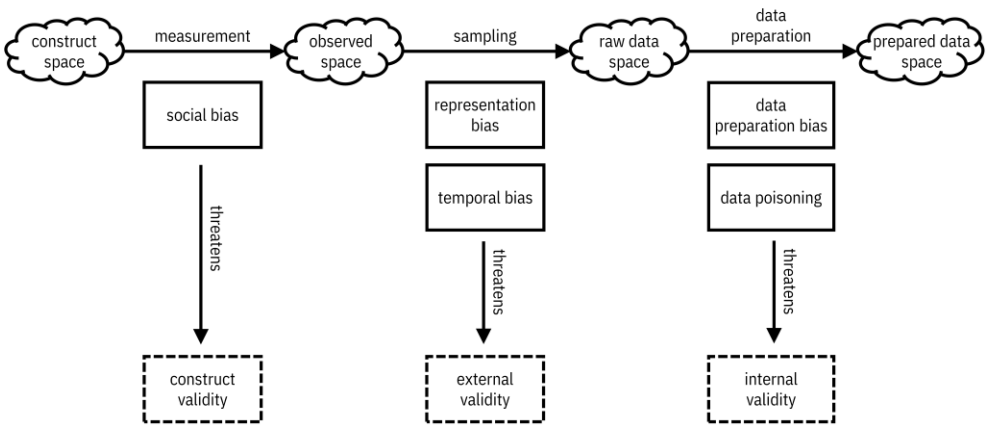


Figure 4.3. *A mental model of spaces, validities, and biases.* Accessible caption. A sequence of four spaces, each represented as a cloud. The construct space leads to the observed space via the measurement process. The observed space leads to the raw data space via the sampling process. The raw data space leads to the prepared data space via the data preparation process. The measurement process contains social bias, which threatens construct validity. The sampling process contains representation bias and temporal bias, which threatens external validity. The data preparation process contains data preparation bias and data poisoning, which threaten internal validity.

There are three main kinds of validity: (1) *construct validity*, (2) *external validity*, and (3) *internal validity*.[6] Construct validity is whether the data really measures what it ought to measure. External validity is whether analyzing data from a given population generalizes to other populations. Internal validity is whether there are any errors in the data processing.

The various kinds of validity are threatened by various kinds of bias. There are many categorizations of types of bias, but for simplicity, let's focus on just five.[7] *Social bias* threatens construct validity, *representation bias* and *temporal bias* threaten external validity, and *data preparation bias* and *data poisoning* threaten internal validity. These biases are detailed throughout this section.

It is useful to also imagine different spaces in which various abstract and concrete versions of the data exist: a *construct space*, an *observed space*, a *raw data space*, and a *prepared data space*. The construct space is an abstract, unobserved, theoretical space in which there are no biases. *Hakuna matata*, the East African problem-free philosophy, reigns in this ideal world. The construct space is operationalized to the observed space through the measurement of features and labels.[8] Data samples collected from a specific population in the observed space live in the raw data space. The raw data is processed to obtain the final prepared data to train and test machine learning models.

4.3.1 Social Bias

Whether it is experts whose decision making is being automated or it is crowd workers, people's judgement is involved in going from labels in the construct space to labels in the observed space. These human judgements are subject to human cognitive biases which can lead to implicit social biases (associating stereotypes towards categories of people without conscious awareness) that yield systematic disadvantages to unprivileged individuals and groups.[9] If decision makers are prejudiced, they may also exert explicit social bias. These biases are pernicious and reinforce deep-seated structural inequalities. Human cognitive biases in labeling can yield other sorts of systematic errors as well.

There can also be structural inequalities in features too. If an aptitude test asks questions that rely on specific cultural knowledge that not all test-takers have, then the feature will not, in fact, be a good representation of the test-taker's underlying aptitude. And most of the time, this tacit knowledge will favor privileged groups. Historical underinvestment and lack of opportunity among marginalized social groups also yield similar bias in features.

Your team found an interesting case of social bias when appraising your crowdsourced labels of roofs seen in satellite images in East African villages. The crowd workers had marked and labeled not only the roof of the main house of a household compound, but also separate structures of the same household such as a free-standing kitchen and free-standing sleeping quarters for young men. They had no idea that this is how households are laid out in this part of the world. The bias, if not caught, would have led to incorrect inferences of poverty.

---

[6]Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. "Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries." In: *Frontiers in Big Data* 2.13 (Jul. 2019).

[7]Harini Suresh and John Guttag. "A Framework for Understanding Sources of Harm Throughout the Machine Learning Life Cycle." In: *Proceedings of the ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. Oct. 2021, p. 17.

[8]Abigail Z. Jacobs and Hanna Wallach. "Measurement and Fairness." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Mar. 2021, pp. 375–385.

[9]Lav R. Varshney and Kush R. Varshney. "Decision Making with Quantized Priors Leads to Discrimination." In: *Proceedings of the IEEE* 105.2 (Feb. 2017), pp. 241–255.

4.3.2 Representation Bias

Once operating in the observation space of features and labels, the data engineers on your team must actually acquire sample data points. Ideally, this sampling should be done in such a way that the acquired data set is representative of the underlying population. Often however, there is *selection bias* such that the probability distribution in the observed space does not match the distribution of the data points. External validity may be affected. A specific example of selection bias is unprivileged groups being either underrepresented or overrepresented in the dataset, which leads to machine learning models either ignoring their special characteristics to satisfy an average performance metric or focusing too much on them leading to systematic disadvantage. Upon appraisal of one of Unconditionally's mobile phone datasets, the data engineers found that senior citizens were underrepresented because mobile phone ownership was lower in that subpopulation. They also found that the national census may have been undercounting people in some counties because the statistics authority had not provisioned enough census takers there.

Representation bias need not only be selection bias. Even if present, the characteristics of the features and labels that come from one subpopulation may be different than those from another. Representativeness is not only a question of the presence and absence of data points, but is a broader concept that includes, among others, systematic differences in data quality.

4.3.3 Temporal Bias

Temporal bias is another bias that happens when the observation space is sampled to collect the raw data. It also puts external validity at stake. Once a dataset has been collected, it can get out of sync with the distribution in the observation space if the observation space drifts and shifts over time. *Covariate shift* refers to the distribution of the features, *prior probability shift* refers to the distribution of the labels, and *concept drift* refers to the conditional distribution of the labels given the features. These drifts and shifts can be gradual or sudden. (Distribution shift is covered in greater detail in Chapter 9.) An example of covariate shift in Unconditionally's satellite image dataset is that some locations were observed in the rainy season and some were observed in the dry season.

4.3.4 Data Preparation Bias

The data preparation phase follows the data understanding phase in the machine learning lifecycle. Many biases can be introduced in data preparation that limit internal validity. For example, the data engineers on your team must do something to rows containing missing values. If they follow the common practice of dropping these rows and the missingness is correlated with a sensitive feature, like a debt feature being missing more often for certain religious groups, they have introduced a new bias. Other biases can enter in data preparation through other data cleaning, data enrichment, and data aggregation steps, as well as in data augmentation (see Section 4.2.5).

A sometimes overlooked bias is the use of proxies in the labels. For example, arrests are a problematic proxy for committing crimes. Innocent people are sometimes arrested and more arrests happen where there is more police presence (and police are deployed unevenly). Health care utilization is a problematic proxy for an individual's health status because groups utilize health care systems unevenly. Data preparation biases are often subtle and involve some choices made by the data engineer and data scientist, who are influenced by their own personal and social biases. You can help mitigate

some of these social biases by taking input from a diverse panel of informants in the data understanding phase of the lifecycle. (The role of a diverse team in data understanding is covered in greater depth in Chapter 16.)

4.3.5 Data Poisoning

Finally, a malicious actor can introduce unwanted biases into a dataset, unbeknown to you. This kind of adversarial attack is known as *data poisoning*. Data poisoning is accomplished through different means, including *data injection* and *data manipulation*. Data injection is adding additional data points with characteristics desired by the adversary. Data manipulation is altering the data points already in the dataset. (Data poisoning is covered in more detail in Chapter 11.)

Most of the biases introduced in this chapter can be implemented deliberately to reduce the performance of a machine learning system or otherwise degrade it. Security analysts commonly focus on attacks on accuracy, but other considerations like fairness can also be attacked. In addition to degrading performance, data poisoning can introduce a so-called backdoor for the adversary to exploit later. For example, someone trying to swindle Unconditionally might introduce satellite images of households next to rivers always labeled as severe poverty to trick your model into giving more cash transfers to riverside communities.

4.3.6 Conclusion

The different categories of bias neutralize different types of validity. Appraising data and preparing data are difficult tasks that must be done comprehensively without taking shortcuts. More diverse teams may be able to brainstorm more threats to validity than less diverse teams. Assessing data requires a careful consideration not only of the modality and source, but also of the measurement, sampling, and preparation. The mental model of biases provides you with a checklist to go through before using a dataset to train a machine learning model. Have you evaluated social biases? Is your dataset representative? Could there be any temporal dataset shifts over time? Have any data preparation steps accidently introduced any subtle biases? Has someone snuck in, accessed the data, and changed it for their malicious purpose?

What should you do if any bias is found? Some biases can be overcome by collecting better data or redoing preparation steps better. Some biases will slip through and contribute to epistemic uncertainty in the modeling phase of the machine learning lifecycle. Some of the biases that have slipped through can be mitigated in the modeling step explicitly through defense algorithms or implicitly by being robust to them. You'll learn how in Part 4 of the book.

4.4 Summary

- Data is the prerequisite for modeling in machine learning systems. It comes in many forms from various sources and can pick up many different biases along the way.
- It is critical to ascertain which biases are present in a dataset because they jeopardize the validity of the system solving the specified problem.
- Evaluating structured datasets involves evaluating the dataset itself, including a focus on data preparation. Evaluating semi-structured datasets that are represented by foundation models and

learned representations additionally involves evaluating large-scale background datasets.

- The source of most data for machine learning is repurposed data created and collected for other reasons. Evaluating the original reason for data creation provides insight into a dataset's bias.
- No matter how careful one is, there is no completely unbiased dataset. Nevertheless, the more effort put in to catching and fixing biases before modeling, the better.
- Trustworthy machine learning systems should be designed to mitigate biases that slip through the data understanding and data preparation phases of the lifecycle.

5

Privacy and Consent

A global virus pandemic is starting to abate, and different organizations are scrambling to put together 'back-to-work' plans to allow employees to return to their workplace after several months in lockdown at home. Toward this end, organizations are evaluating a (fictional) machine learning-based mobile app named TraceBridge. It supports the return to the office by collecting and modeling location traces, health-related measurements, other social data (e.g. internal social media and calendar invitations among employees), and administrative data (e.g. space planning information and org charts), to facilitate digital contact tracing: the process of figuring out disease-spreading interactions between an infected person and others. Is TraceBridge the right solution? Will organizations be able to re-open safely or will the employees be homebound for even more seemingly unending months?

The data that TraceBridge collects, even if free from many biases investigated in Chapter 5, is not free from concern. Does TraceBridge store the data from all employees in a centralized database? Who has access to the data? What would be revealed if there were a data breach? Have the employees been informed about possible uses of the data and agreed to them? Does the organization have permission to share their data with other organizations? Can employees opt out of the app or would that jeopardize their livelihood? Who gets to know that an employee has tested positive for the disease? Who gets to know their identity and their contacts?

The guidance to data scientists in Chapter 4 was to be wary of biases that creep into data and problem formulations because of the harms they can cause. In this chapter, the thing to be wary about is whether it is even right to use certain data for reasons of consent, power, and privacy.[1] Employers are now evaluating the app. However, when the problem owners, developers, and data scientists of TraceBridge were creating the app, they had to:

- weigh the need for consent, diffusion of power, and privacy,

[1]Eun Seo Jo and Timnit Gebru. "Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 306–316.

- differentiate different kinds of anonymity for privacy, and
- question whether anonymity is the only way to achieve privacy.

Let's critique their choices.

5.1 Consent, Power, and Privacy

The design of TraceBridge utilizes purposefully collected data, social data, and administrative data because using all of these data sources as features increases the performance of the underlying machine learning models. The app does not inform employees that it accesses internal social media, calendar invitations, and administrative data. In the app design, the employer's organizational leadership has full control over the data.

As the designers, the TraceBridge team thought they were taking a simple and effective approach, but they did not understand that they were introducing problems of consent and power. The employer holds all the power in the deployment of the app because it can require the usage of the app as a condition of employment without any opportunity for the employee to give consent. Employees also have no opportunity to provide informed consent to the use of specific parts of their data. The employer holds the power to use the data not only for contact tracing of the viral infection, but also to track worker movements and interactions for other reasons like noting too many breaks and unwanted gatherings. Nothing prevents them from selling the data to other interested parties, leaving the employees powerless over their data. Overall, the design favors the powerful employer and fails the vulnerable employees.

Furthermore, the TraceBridge system design stores all personally-identifiable data it uses centrally without encryption or other safeguards for security, and makes it available without obfuscation to the organization's management as the default behavior. When an infection is detected, an alert goes out to all people in the organization. Details of the identity of the infected person are transmitted to management and all inferred contacts.

The TraceBridge team may think they are providing a turnkey solution that does not overcomplicate things on the backend, but their design choices sacrifice *privacy*, the ability of individuals to withhold information about themselves. Privacy is considered an essential human right in many value systems and legal frameworks. The central repository of personally-identifiable information provides no protections to maintain anonymity in the employee's data. The health status and movement of employees throughout the day is completely obvious by name. Furthermore, by revealing identifying information through alerts, there is no maintenance of anonymity. The TraceBridge team has been quite negligent of privacy considerations and any organization using the app will likely be on the wrong side of the law.

In a broad sense, data is a valuable commodity. It reveals a lot about human behavior at a gross level, but also about the behavior of individual people. Just like other natural resources, it can be extracted from the vulnerable without their consent and furthermore be exploited for their subjugation.[2] Some

[2]Shakir Mohamed, Marie-Therese Png, and William Isaac. "Decolonial AI: Decolonial Theory as Sociotechnical Foresight in Artificial Intelligence." In: *Philosophy and Technology* 33 (Jul. 2020), pp. 659–684.

even argue that people should be compensated for their personal data because they are selling their privacy.[3] In short, *data is power*.

> "Data is the new oil."
>
> —Clive Humby, data science entrepreneur at dunnhumby

Data used in machine learning is often fraught with power and consent issues because it is often repurposed from other uses or is so-called *data exhaust*: byproducts from people's digital activities. For example, many large-scale image datasets used for training computer vision models are scraped from the internet without explicit consent from the people who posted the images.[4] Although there may be implicit consent through vehicles such as Creative Commons licenses, a lack of explicit consent can nevertheless be problematic. Sometimes copyright laws are violated in scraped and repurposed data.

Why does this happen? It is almost always due to system designers taking shortcuts to gather large datasets and show value quickly without giving thought to power and consent. And it is precisely the most powerful who tend to be least cognizant of issues of power. People from marginalized, minoritized, and otherwise less powerful backgrounds tend to have more knowledge of the perspectives of both the powerful and the powerless.[5] This concept, known as the *epistemic advantage* of people with lived experience of marginalization, is covered in greater detail in Chapter 16. Similarly, except in regulated application domains such as health care, privacy issues have usually been an afterthought due to convenience. Things have started to change due to comprehensive laws such as the General Data Protection Regulation enacted in the European Economic Area in 2018.

In summary, problem owners and data scientists should not have any calculus to weigh issues of power, consent and privacy against conveniences in data collection. For the fourth attribute of trust (aligned purpose), trustworthy machine learning systems require that data be used consensually, especially from those who could be subject to exploitation. No ifs, ands, or buts!

5.2 Achieving Privacy through Anonymization

After receiving unfavorable feedback from organizations that they risk breaking privacy laws, the TraceBridge development team is back to the drawing board. They must figure out what the heck privacy is all about, pick among competing frameworks, and then incorporate them into their system.

In preserving privacy, there are two main use cases: (1) data publishing and (2) data mining. *Privacy-preserving data publishing* is anonymizing data to fully disclose it without violating privacy. *Privacy-preserving data mining* is querying data while controlling the disclosure of information at the individual level. Privacy-preserving data publishing is also known as *non-interactive anonymization* and privacy-

---

[3]Nicholas Vincent, Yichun Li, Renee Zha, and Brent Hecht. "Mapping the Potential and Pitfalls of 'Data Dividends' as a Means of Sharing the Profits of Artificial Intelligence." arXiv:1912.00757, 2019.

[4]Abeba Birhane and Vinay Uday Prabhu. "Large Image Datasets: A Pyrrhic Win for Computer Vision?" In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. Jan. 2021, pp. 1536–1546.

[5]Miliann Kang, Donovan Lessard, and Laura Heston. *Introduction to Women, Gender, Sexuality Studies*. Amherst, Massachusetts, USA: University of Massachusetts Amherst Libraries, 2017.

preserving data mining is also known as *interactive anonymization*. TraceBridge may want to do either or both: publishing datasets for examination by organizational leaders or state regulators, and issuing contact tracing alerts without exposing individually-identifiable data. They have to go down both paths and learn about the appropriate technical approaches in each case: *syntactic anonymity* for data publishing and *differential privacy* for data mining.[6]

There are three main categories of variables when dealing with privacy: (1) identifiers, (2) quasi-identifiers, and (3) sensitive attributes. *Identifiers* directly reveal the identity of a person. Examples include the name of the person, national identification numbers such as the social security number, or employee serial numbers. Identifiers should be dropped from a dataset to achieve privacy, but such dropping is not the entire solution. In contrast, *quasi-identifiers* do not uniquely identify people on their own, but can reveal identity when linked together through a process known as *re-identification*. Examples are gender, birth date, postal code, and group membership. *Sensitive attributes* are features that people do not want revealed. Examples are health status, voting record, salary, and movement information. Briefly, syntactic anonymity works by modifying quasi-identifiers to reduce their information content, including suppressing them, generalizing them, and shuffling them. Differential privacy works by adding noise to sensitive attributes. A mental model for the two modes of privacy is given in Figure 5.1.

To make this mental model more concrete, let's see how it applies to an actual sample dataset of employees and their results on a diagnostic test for the virus (specifically the cycle threshold (CT) value of a polymerase chain reaction test), which we treat as sensitive. The original dataset, the transformed dataset after k-anonymity with $k = 3$, and the transformed dataset after differential privacy are shown in Table 5.1, Table 5.2, and Table 5.3 (details on k-anonymity and differential privacy are forthcoming).

Table 5.1. *A sample original dataset.*

Name	Department	CT Value
Joseph Cipolla	Trustworthy AI	12
Kweku Yefi	Neurosymbolic AI	20
Anjali Singh	AI Applications	35
Celia Sontag	Compute Acceleration	31
Phaedra Paragios	Software-Defined Architecture	19
Chunhua Chen	Thermal Packaging	27

Table 5.2. *The sample original dataset under k-anonymity with* $k = 3$.

Organization	CT Value
AI	12
AI	20
AI	35
Hybrid Cloud	31
Hybrid Cloud	19
Hybrid Cloud	27

[6]John S. Davis II and Osonde A. Osoba. "Privacy Preservation in the Age of Big Data: A Survey." *RAND Justice, Infrastructure, and Environment* Working Paper WR-1161, 2016.

Table 5.3. *The values returned for queries under differential privacy with Laplace noise added to the sensitive attribute in the sample original dataset.*

Name	Department	CT Value
Joseph Cipolla	Trustworthy AI	13.5
Kweku Yefi	Neurosymbolic AI	12.8
Anjali Singh	AI Applications	32.7
Celia Sontag	Compute Acceleration	35.9
Phaedra Paragios	Software-Defined Architecture	22.1
Chunhua Chen	Thermal Packaging	13.4

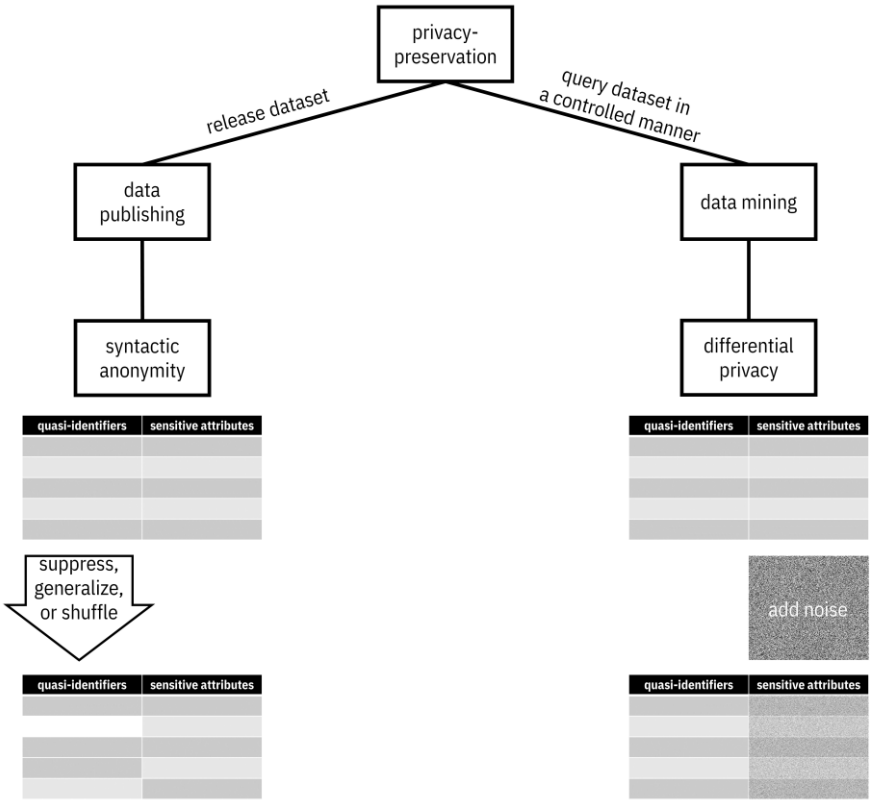


Figure 5.1. *A mental model of privacy-preservation broken down into two branches: data publishing with syntactic anonymity and data mining with differential privacy.* Accessible caption. A hierarchy diagram with privacy-preservation at its root. One child is data publishing, which is done when you release dataset. The only child of data publishing is syntactic anonymity. Syntactic anonymity is illustrated by a table with columns for quasi-identifiers and sensitive attributes. By suppressing, generalizing, or shuffling quasi-identifiers, some rows have been reordered and others have taken on a different value. The other child of privacy-preservation is data mining, which is done when you query dataset in a controlled manner. The only child of data mining is differential privacy. Differential privacy is also illustrated by a table with columns for quasi-identifiers and sensitive attributes. By adding noise to sensitive attributes, all the rows are noisy.

5.2.1 Data Publishing and Syntactic Anonymity

The simplest form of syntactic anonymity is *k-anonymity*.[7] By means of suppressing values of quasi-identifiers (replacing the value with a null value) or generalizing their values (for example replacing 5-digit zip codes with only their first three digits), the idea of k-anonymity is to create groups of records of cardinality at least k that have exactly the same modified quasi-identifier values. All the records within a group or cluster become equivalent and cannot be distinguished. Randomly shuffling identifiers within a quasi-identifier group achieves the same effect. If there are n data points in the original dataset, then there should be about n/k groups in the anonymized dataset, each of approximately the same cardinality.

Weaknesses of k-anonymity include susceptibility to the *homogeneity attack* and the *background knowledge attack*. The homogeneity attack takes advantage of many records within a k-member cluster having the same sensitive attributes, which means that even without precise re-identification, the sensitive information of individuals is still revealed. The background knowledge attack takes advantage of side information of subgroups having specific distributions of sensitive attributes to home in on likely sensitive attribute values of individuals. An extension of k-anonymity known as is *l-diversity* overcomes these vulnerabilities.[8] It further requires each k-member group to have at least l distinct values of sensitive attributes.

A further enhancement of k-anonymity and l-diversity is *t-closeness*.[9] Starting with the basic definition of k-anonymity, t-closeness further requires that the suitably-defined distance between the sensitive attribute probability distribution of each k-member group and the global sensitive attribute probability distribution of all records in the dataset is less than or equal to t. Simply put, all the groups should be similar in their distribution of sensitive attributes. Finding a t-closeness transformation of a given dataset is computationally difficult.

The re-identification risks of k-anonymity, l-diversity, and t-closeness have interpretations in terms of mutual information, which was introduced in Chapter 3. If X is the random variable for quasi-identifiers in the original dataset, $\tilde{X}$ is the random variable for quasi-identifiers in the anonymized dataset, and W is the random variable for sensitive attributes, then we have the following quantitative problem specifications:

- $I(X, \tilde{X}) \leq log \frac{n}{k}$ (k-anonymity),
- $I(W, \tilde{X}) \leq H(W) - log\, l$ (l-diversity), and
- $I(W, \tilde{X}) \leq t$ (t-closeness).[10]

Through k-anonymity, the reidentification risk is reduced down from that of the full dataset to the number of clusters. With l-diversity or t-closeness added on top of k-anonymity, the predictability of the

[7]Latanya Sweeney. "k-Anonymity: A Model for Protecting Privacy." In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (Oct. 2002), pp. 557–570.
[8]Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. "l-Diversity: Privacy Beyond k-Anonymity." In: *ACM Transactions on Knowledge Discovery from Data* 1.1 (Mar. 2007), p. 3.
[9]Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity." In: *Proceedings of the IEEE International Conference on Data Engineering*. Istanbul, Turkey, Apr. 2007, pp. 106–115.
[10]Michele Bezzi. "An Information Theoretic Approach for Privacy Metrics." In: *Transactions on Data Privacy* 3.3 (Dec. 2010), pp. 199–215.

sensitive attributes from the anonymized quasi-identifiers is constrained. These relationships are valuable ways of reasoning about what information is and is not revealed due to anonymization. By expressing them in the common statistical language of information theory, they can be examined and studied alongside other problem specifications and requirements of trustworthiness in broader contexts.

5.2.2 Data Mining and Differential Privacy

The other branch of anonymization is differential privacy and applies to use cases involving querying a dataset, not simply releasing it. The setup is that an organization has a dataset and knows exactly which query it will be dealing with. Some example queries are to return the count of a certain value in the dataset or the average value of a column in the dataset. A query could even be as complicated as returning a machine learning classifier trained on the dataset. Some queries are easier and some queries are harder to anonymize. In the differential privacy setup, the organization has to maintain control over the dataset and all queries to it. This is in contrast to syntactic anonymity where once the dataset has been anonymized, it is footloose and free. The basic method of differential privacy is adding noise to sensitive attributes.

Getting down into a little more detail, let's say that TraceBridge has a dataset W_1 of all employees positive for the viral disease. Another employee is detected to be positive and is added to the dataset giving us a new dataset W_2 that only differs from W_1 by the addition of one row. Let's say that the query function $y(W)$ is the number of employees who have cancer, which is important for better understanding the relationship between cancer and the viral disease.[11] Cancer diagnosis status is considered sensitive. Instead of actually returning $y(W)$, a differentially-private system gives back a noisy version of $y(W)$ by adding a random value to it. The value it returns is $\tilde{Y}(W) = y(W) + \text{noise}$. $\tilde{Y}$ is a random function which we can think of as a random variable that takes sample value $\tilde{y}$. The goal of differential privacy is expressed by the following bound involving the probabilities of queries from the original and new datasets:

$$P\big(\tilde{Y}(W_1) = \tilde{y}\big) \leq e^{\epsilon} P\big(\tilde{Y}(W_2) = \tilde{y}\big), \text{for all } \tilde{y}.$$

Equation 5.1

The ϵ is a tiny positive parameter saying how much privacy we want. The value of e^{ϵ} becomes closer and closer to one as ϵ gets closer and closer to zero. When ϵ is zero, the two probabilities are required to be equal and thus the two datasets have to be indistinguishable, which is exactly the sense of anonymity that differential-privacy is aiming for.[12] You can't tell the difference in the query result when you add the new person in, so you can't figure out their sensitive attribute any more than what you could have figured out in general from the dataset.

---

[11] https://rebootrx.org/covid-cancer

[12] We should really write $P\big(\tilde{Y}(W_1) \in S\big) \leq e^{\epsilon} P\big(\tilde{Y}(W_2) \in S\big)$ for some interval or other set S because if $\tilde{Y}$ is a continuous random variable, then its probability of taking any specific value is always zero. It only has a specific probability when defined over a set.

The main trick in differential privacy is solving for the kind of noise and its strength to add to $y(W)$. For lots of query functions, the best kind of noise comes from the Laplace distribution.[13] As stated earlier, some queries are easier than others. This easiness is quantified using *global sensitivity*, which measures how much a single row of a dataset impacts the query value. Queries with smaller global sensitivity need lower strength noise to achieve ϵ-differential privacy.

Just like with syntactic privacy, it can be easier to think about differential privacy alongside other problem specifications in trustworthy machine learning like accuracy, fairness, robustness, and explainability when expressed in terms of information theory rather than the more specialized terminology used in defining it earlier. To do so, we also need to say that the dataset W is a random variable, so the probabilities that we want to be close to each other are the noised query results conditioned on the dataset realizations $P(\tilde{Y} \mid W = w_1)$ and $P(\tilde{Y} \mid W = w_2)$. Then we can pose our objective of differential privacy as wanting the mutual information between the dataset and noisy query result $I(W, \tilde{Y})$ to be minimized. With some more specifics added to minimizing the mutual information, we can get back a relationship in terms of the ϵ of ϵ-differential privacy.[14] The idea of examining the mutual information between the dataset and the query is as follows. Since mutual information measures the reduction in uncertainty about the dataset by the knowledge of the query, having zero (or small) mutual information indicates that we don't learn anything about the dataset's composition from the query result, which is exactly the idea of differential privacy.

Differential privacy is intended to prevent attributing the change of the query's value to any one person's row of data. Nevertheless, one criticism of differential privacy as imagined in its information theory presentation is that it can allow the value of sensitive attributes to be inferred if there are correlations or associations among the rows. Stricter information-theoretic conceptions of differential privacy have been developed that require the rows of the dataset to be independent.

5.2.3 Conclusion

TraceBridge has a few different use cases as part of their app and contact tracing system. One is publishing sensitive data for the leaders of the organization, external regulators, or auditors to look at. A second is to interactively query statistics from the sensitive data without revealing things about individuals. Each use has different appropriate approaches: syntactic anonymity for the first and differential privacy for the second, along with different requirements on the system design and the infrastructure required. Existing legal protections in various jurisdictions and application domains are mostly for the first use case (data publishing), but the regulations themselves are usually unclear on their precise notion of privacy. TraceBridge may have to go with both approaches to privacy in developing a trusted system.

We've reached the end of this section and haven't talked about the tradeoff of privacy with utility. All measures and approaches of providing privacy should be evaluated in conjunction with how the data is going to be used. It is a balance. The tradeoff parameters are there for a reason. The usefulness of a dataset after k-anonymization is usually pretty good for a decent-sized k, but might not be so great after

---

[13]The pdf of the Laplace distribution is $p_X(x) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$, where μ is the mean and b is a scale parameter such that the variance is $2b^2$.

[14]Darakshan J. Mir. "Information-Theoretic Foundations of Differential Privacy." In: *Foundations and Practice of Security*. Montreal, Canada, Oct. 2012, pp. 374–381.

achieving t-closeness for a decent t. Similarly, a criticism of differential privacy for typical queries is that the usefulness of the query results is not that great for a small ϵ (adding a large magnitude of noise). However, there are no blanket statements to be made: these intuitions have to be appraised for specific scenarios and datasets, and privacy parameters have to be chosen carefully without taking shortcuts and incorporating input from multiple stakeholders.

5.3 Other Ways of Achieving Privacy

The two technical approaches that yield anonymized data (syntactic anonymity for data publishing and differential privacy for data mining) are not the only ways for TraceBridge to achieve privacy. Here's a really easy way for them to achieve privacy: lock up the data and throw away the key. If they don't publish the data and provide no ability to query it, they have perfect privacy. But they don't have any utility from the data either. So what else can they do?

One answer is to set up *institutional controls* and procedures so that only qualified individuals have access to data, and only for specific approved uses. Cleared data scientists may only be allowed to access the data on closed computing systems with strong enforcement to prevent data breaches. Keeping data in a decentralized manner rather than all in one centralized place can also help prevent breaches.

A second answer is to bring the algorithm to the data rather than the other way around. Working through a decentralized system where different pieces of data are kept in different places, *secure multi-party computation* allows a value to be computed using data from different sources without revealing the inputs sent by each data source to other data sources.

A third answer is encryption. TraceBridge can use *fully homomorphic encryption* to compute things on encrypted data and get the answer they would have gotten if the data hadn't been encrypted. This approach can be a computational beast, but is getting more and more computationally tractable every day.

With all three of these approaches: institutional controls, secure multi-party computation, and fully homomorphic encryption, the question of *what* is computed remains open. People and organizations can be using these techniques and still be outputting some value or summary statistic that discloses sensitive individual information. It may thus make sense to combine these methods with, for example, differential privacy.

5.4 Summary

- Data is a valuable resource that comes from people. The use of this data should be consensually obtained. If there is no consent, do not proceed.
- It is easy for data scientists to set up machine learning systems that exploit and subjugate vulnerable individuals and groups. Do not do it. Instead, be careful, thoughtful, and take input from powerless groups.
- By consenting to the use of their data, people give up their privacy. Various methods can be used to preserve their privacy.
- Syntactic anonymization methods group together individuals with similar quasi-identifiers and then obfuscate those quasi-identifiers. These methods are useful when publishing individual-

level data.

- Differential privacy methods add noise to queries about sensitive attributes when users can only interact with the data through known and fixed queries. These methods are useful when statistically analyzing the data or computing models from the data.
- Securing access to data provides an alternative to data anonymization.

6

Detection Theory

Let's continue from Chapter 3, where you are the data scientist building the loan approval model for the (fictional) peer-to-peer lender ThriveGuild. As then, you are in the first stage of the machine learning lifecycle, working with the problem owner to specify the goals and indicators of the system. You have already clarified that safety is important, and that it is composed of two parts: basic performance (minimizing aleatoric uncertainty) and reliability (minimizing epistemic uncertainty). Now you want to go into greater depth in the problem specification for the first part: basic performance. (Reliability comes in Part 4 of the book.)

What are the different quantitative metrics you could use in translating the problem-specific goals (e.g. expected profit for the peer-to-peer lender) to machine learning quantities? Once you've reached the modeling stage of the lifecycle, how would you know you have a good model? Do you have any special considerations when producing a model for risk assessment rather than simply offering an approve/deny output?

Machine learning models are *decision functions*: based on the borrower's features, they decide a response that may lead to an autonomous approval/denial action or be used to support the decision making of the loan officer. The use of decision functions is known as statistical discrimination because we are distinguishing or differentiating one class label from the other. You should contrast the use of the term 'discrimination' here with *unwanted* discrimination that leads to systematic advantages to certain groups in the context of algorithmic fairness in Chapter 10. Discrimination here is simply telling the difference between things. Your favorite wine snob talking about their discriminative palate is a distinct concept from racial discrimination.

This chapter begins Part 3 of the book on basic modeling (see Figure 6.1 to remind yourself of the lay of the land) and uses *detection theory*, the study of *optimal* decision making in the case of categorical output responses,[1] to answer the questions above that you are struggling with.

[1]Estimation theory is the study of optimal decision making in the case of continuous output responses.

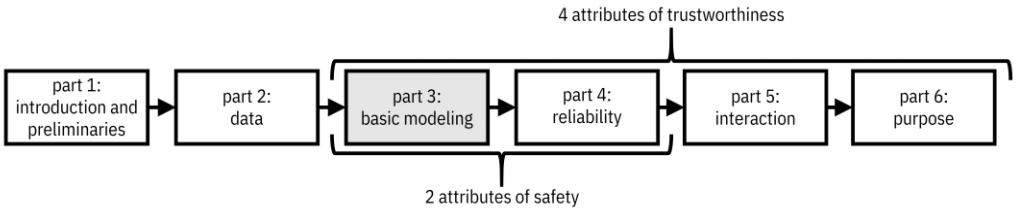


Figure 6.1. *Organization of the book. This third part focuses on the first attribute of trustworthiness, competence and credibility, which maps to machine learning models that are well-performing and accurate.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 3 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

Specifically, this chapter focuses on:

- selecting metrics to quantify the basic performance of your decision function (including ones that summarize performance across operating conditions),
- testing whether your decision function is as good as it could ever be, and
- differentiating performance in risk assessment problems from performance in binary decision problems.

6.1 Selecting Decision Function Metrics

You, the ThriveGuild data scientist, are faced with the *binary detection* problem, also known as the *binary hypothesis testing* problem, of predicting which loan applicants will default, and thereby which applications to deny.[2] Let Y be the loan approval decision with label $y = 0$ corresponding to deny and label $y = 1$ corresponding to approve. Feature vector X contains employment status, income, and other attributes. The value $y = 0$ is called a *negative* and the value $y = 1$ is called a *positive*. The random variables for the features and label are governed by the pmfs given the special name *likelihood functions* $p_{X|Y}(x \mid y = 0)$ and $p_{X|Y}(x \mid y = 1)$, as well as by *prior probabilities* $p_0 = P(Y = 0)$ and $p_1 = P(Y = 1) = 1 - p_0$. The basic task is to find a *decision function* $\hat{y}: \mathcal{X} \to \{0,1\}$ that predicts a label from the features.[3]

6.1.1 Quantifying the Possible Events

There are four possible events in the binary detection problem:

1. the decision function predicts 0 and the true label is 0,
2. the decision function predicts 0 and the true label is 1,

[2]For ease of explanation in this chapter and in later parts of the book, we mostly stick with the case of two label values and do not delve much into the case with more than two label values.

[3]This is also the basic task of supervised machine learning. In supervised learning, the decision function is based on data samples from (X, Y) rather than on the distributions; supervised learning is coming up soon enough in the next chapter, Chapter 7.

3. the decision function predicts 1 and the true label is 1, and
4. the decision function predicts 1 and the true label is 0.

These are known as *true negatives* (TN), *false negatives* (FN), *true positives* (TP), and *false positives* (FP), respectively. A true negative is denying an applicant who should be denied according to some ground truth, a false negative is denying an applicant who should be approved, a true positive is approving an applicant who should be approved, and a false positive is approving an applicant who should be denied. Let's organize these events in a table known as the *confusion matrix*:

	$Y = 1$	$Y = 0$
$\hat{y}(X) = 1$	TP	FP
$\hat{y}(X) = 0$	FN	TN

Equation 6.1

The probabilities of these events are:

$p_{\text{TP}} = P(\hat{y}(X) = 1 \mid Y = 1)$	$p_{\text{FP}} = P(\hat{y}(X) = 1 \mid Y = 0)$
$p_{\text{FN}} = P(\hat{y}(X) = 0 \mid Y = 1)$	$p_{\text{TN}} = P(\hat{y}(X) = 0 \mid Y = 0)$

Equation 6.2

These conditional probabilities are nothing more than a direct implementation of the definitions of the events. The probability p_{TN} is known as the *true negative rate* as well as the specificity and the selectivity. The probability p_{FN} is known as the *false negative rate* as well as the probability of missed detection and the miss rate. The probability p_{TP} is known as the *true positive rate* as well as the probability of detection, the recall, the sensitivity, and the power. The probability p_{FP} is known as the *false positive rate* as well as the probability of false alarm and the fall-out. The probabilities can be organized in a slightly different table as well:

$P(\hat{y}(X) \mid Y)$	$Y = 1$	$Y = 0$
$\hat{y}(X) = 1$	p_{TP}	p_{FP}
$\hat{y}(X) = 0$	p_{FN}	p_{TN}

Equation 6.3

These probabilities give you some quantities by which to understand the performance of the decision function $\hat{y}$. Selecting one over the other involves thinking about the events themselves and how they relate to the real-world problem. A false positive, approving an applicant who should be denied, means that a ThriveGuild lender has to bear the cost of a default, so it should be kept small. A false negative, denying an applicant who should be approved, is a lost opportunity for ThriveGuild to make a profit through the interest they charge.

The events above are conditioned on the true label. Conditioning on the predicted label also yields events and probabilities of interest in characterizing performance:

$P\big(Y \mid \hat{y}(X)\big)$	$Y = 1$	$Y = 0$
$\hat{y}(X) = 1$	p_{PPV}	p_{FDR}
$\hat{y}(X) = 0$	p_{FOR}	p_{NPV}

Equation 6.4

These conditional probabilities are reversed from Equation 6.2. The probability p_{NPV} is known as the *negative predictive value*. The probability p_{FOR} is known as the *false omission rate*. The probability p_{PPV} is known as the *positive predictive value* as well as the precision. The probability p_{FDR} is known as the *false discovery rate*. If you care about the quality of the decision function, focus on the first set (p_{TN}, p_{FN}, p_{TP}, p_{FP}). If you care about the quality of the predictions, focus on the second set (p_{NPV}, p_{FOR}, p_{PPV}, p_{FDR}).

When you need to numerically compute these probabilities, apply the decision function to several i.i.d. samples of (X, Y) and denote the number of TN, FN, TP, and FP events as n_{TN}, n_{FN}, n_{TP}, and n_{FP}, respectively. Then use the following estimates of the probabilities:

$p_{\text{TP}} \approx \dfrac{n_{\text{TP}}}{n_{\text{TP}} + n_{\text{FN}}}$	$p_{\text{FP}} \approx \dfrac{n_{\text{FP}}}{n_{\text{FP}} + n_{\text{TN}}}$
$p_{\text{FN}} \approx \dfrac{n_{\text{FN}}}{n_{\text{FN}} + n_{\text{TP}}}$	$p_{\text{TN}} \approx \dfrac{n_{\text{TN}}}{n_{\text{TN}} + n_{\text{FP}}}$

$p_{\text{PPV}} \approx \dfrac{n_{\text{TP}}}{n_{\text{TP}} + n_{\text{FP}}}$	$p_{\text{FDR}} \approx \dfrac{n_{\text{FP}}}{n_{\text{FP}} + n_{\text{TP}}}$
$p_{\text{FOR}} \approx \dfrac{n_{\text{FN}}}{n_{\text{FN}} + n_{\text{TN}}}$	$p_{\text{NPV}} \approx \dfrac{n_{\text{TN}}}{n_{\text{TN}} + n_{\text{FN}}}$

Equation 6.5

As an example, let's say that ThriveGuild makes the following number of decisions: $n_{\text{TN}} = 1234$, $n_{\text{FN}} = 73$, $n_{\text{TP}} = 843$, and $n_{\text{FP}} = 217$. You can estimate the various performance probabilities by plugging these numbers into the respective expressions above. The results are $p_{\text{TN}} \approx 0.85$, $p_{\text{FN}} \approx 0.08$, $p_{\text{TP}} \approx 0.92$, $p_{\text{FP}} \approx$

0.15, $p_{\text{NPV}} \approx 0.94$, $p_{\text{FOR}} \approx 0.06$, $p_{\text{PPV}} \approx 0.80$, and $p_{\text{FDR}} \approx 0.20$. These are all reasonably good values, but must ultimately be judged according to the ThriveGuild problem owner's goals and objectives.

6.1.2 Summary Performance Metrics

Collectively, false negatives and false positives are *errors*. The *probability of error*, also known as the error rate, is the sum of the false negative rate and false positive rate weighted by the prior probabilities:

$$p_{\text{E}} = p_0 p_{\text{FP}} + p_1 p_{\text{FN}}.$$

Equation 6.6

The *balanced* probability of error, also known as the balanced error rate, is the unweighted average of the false negative rate and false positive rate:

$$p_{\text{BE}} = \frac{1}{2} p_{\text{FP}} + \frac{1}{2} p_{\text{FN}}.$$

Equation 6.7

They summarize the basic performance of the decision function. Balancing is useful when there are a lot more data points with one label than the other, and you care about each type of error equally. *Accuracy*, the complement of the probability of error: $1 - p_{\text{E}}$, and *balanced accuracy*, the complement of the balanced probability of error: $1 - p_{\text{BE}}$, are sometimes easier for problem owners to appreciate than error rates.

The F_1-score, the harmonic mean of p_{TP} and p_{PPV}, is an accuracy-like summary measure to characterize the quality of a prediction rather than the decision function:

$$F_1 = 2 \frac{p_{\text{TP}} p_{\text{PPV}}}{p_{\text{TP}} + p_{\text{PPV}}}.$$

Equation 6.8

Continuing the example from before with $p_{\text{TP}} \approx 0.92$ and $p_{\text{PPV}} \approx 0.80$, let ThriveGuild's prior probability of receiving applications to be denied according to some ground truth be $p_0 = 0.65$ and applications to be approved be $p_1 = 0.35$. Then, plugging in to the relevant equations above, you'll find ThriveGuild to have $p_{\text{E}} \approx 0.13$, $p_{\text{BE}} \approx 0.11$, and $F_1 \approx 0.86$. Again, these are reasonable values that may be deemed acceptable to the problem owner.

As the data scientist, you can get pretty far with these abstract TN, FN, TP, and FP events, but they have to be put in the context of the problem owner's goals. ThriveGuild cares about making good bets on borrowers so that they are profitable. More generally across real-world applications, error events yield significant consequences to affected people including loss of life, loss of liberty, loss of livelihood, etc. Therefore, to truly characterize the performance of a decision function, it is important to consider the *costs* associated with the different events. You can capture these costs through a cost function $c\big(Y, \hat{y}(X)\big)$ and denote the costs as $c(0,0) = c_{00}$, $c(1,0) = c_{10}$, $c(1,1) = c_{11}$, and $c(0,1) = c_{01}$, respectively.

Taking costs into account, the characterization of performance for the decision function is known as the Bayes risk R:

$$R = (c_{10} - c_{00})p_0 p_{FP} + (c_{01} - c_{11})p_1 p_{FN} + c_{00}p_0 + c_{11}p_1.$$

Equation 6.9

Breaking the equation down, you'll see that the two error probabilities, p_{FP} and p_{FN} are the main components, multiplied by their relevant prior probabilities and costs. The costs of the non-error events appear just multiplied by their costs. The Bayes risk is the performance metric most often used in finding optimal decision functions. Actually finding the decision function is known as solving the *Bayesian detection* problem. Eliciting the cost function $c(\cdot,\cdot)$ for a given real-world problem from the problem owner is part of value alignment, described in Chapter 14.

A mental model or roadmap, shown in Figure 6.2, to hold throughout the rest of the chapter is that the Bayes risk and the Bayesian detection problem are the central concept, and all other concepts are related to the central concept in various ways and for various purposes. The terms and concepts that have not yet been defined and evaluated are coming up soon.

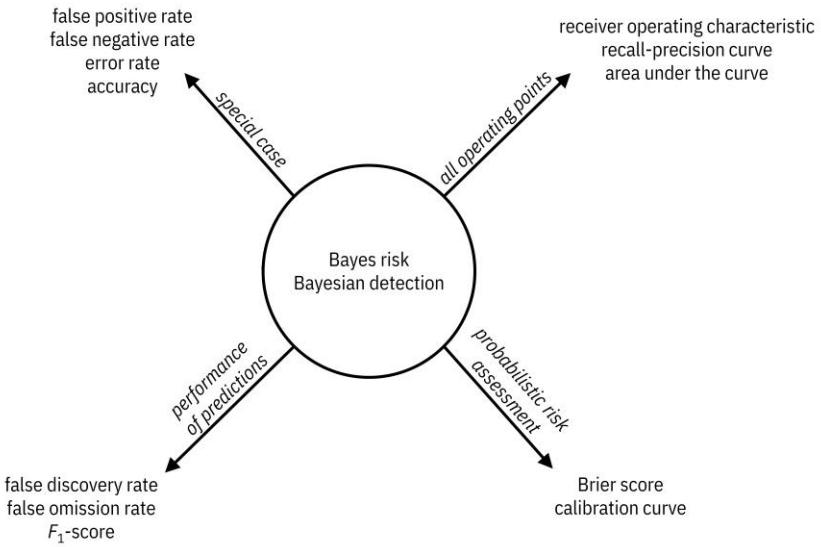


Figure 6.2. *A mental model for different concepts in detection theory surrounding the central concept of Bayes risk and Bayesian detection.* A diagram with Bayes risk and Bayesian detection at the center and four other groups of concepts radiating outwards. False positive rate, false negative rate, error rate, and accuracy are special cases. Receiver operating characteristic, recall-precision curve, and area under the curve arise when examining all operating points. Brier score and calibration curve arise in probabilistic risk assessment. False discover rate, false omission rate, and F_1-score relate to performance of predictions.

Because getting things right is a good thing, it is often assumed that there is no cost to correct decisions, i.e., $c_{00} = 0$ and $c_{11} = 0$, which is also assumed in this book going forward. In this case, the Bayes risk simplifies to:

$$R = c_{10}p_0p_{FP} + c_{01}p_1p_{FN}.$$

Equation 6.10

To arrive at this simplified equation, just insert zeros for c_{00} and c_{11} in Equation 6.9. The Bayes risk with $c_{10} = 1$ and $c_{01} = 1$ is the probability of error.

We are implicitly assuming that $c(\cdot,\cdot)$ does not depend on X except through $\hat{y}(X)$. This assumption is not required, but made for simplicity. You can easily imagine scenarios in which the cost of a decision depends on the feature. For example, if one of the features used in the loan approval decision by ThriveGuild is the value of the loan, the cost of an error (monetary loss) depends on that feature. Nevertheless, for simplicity, we usually make the assumption that the cost function does not explicitly depend on the feature value. For example, under this assumption, the cost of a false negative may be $c_{10} = 100{,}000$ dollars and the cost of a false positive $c_{01} = 50{,}000$ dollars for all applicants.

6.1.3 Accounting for Different Operating Points

The Bayes risk is all well and good if there is a fixed set of prior probabilities and a fixed set of costs, but things change. If the economy improves, potential borrowers might become more reliable in loan repayment. If a different problem owner comes in and has a different interpretation of opportunity cost, then the cost of false negatives c_{10} changes. How should you think about the performance of decision functions across different sets of those values, known as different *operating points*?

Many decision functions are parameterized by a threshold η (including the optimal decision function that will be demonstrated in Section 6.2). You can change the decision function to be more or less forgiving of false positives or false negatives, but not both at the same time. Varying η explores this tradeoff and yields different error probability pairs $(p_{\text{FP}}, p_{\text{FN}})$, i.e. different operating points. Equivalently, different operating points correspond to different false positive rate and true positive rate pairs $(p_{\text{FP}}, p_{\text{TP}})$. The curve traced out on the p_{FP}–p_{TP} plane as the parameter η is varied from zero to infinity is the *receiver operating characteristic* (ROC). The ROC takes values $(p_{\text{FP}} = 0, p_{\text{TP}} = 0)$ when $\eta \to \infty$ and $(p_{\text{FP}} = 1, p_{\text{TP}} = 1)$ when $\eta \to 0$. You can understand this because at one extreme, the decision function always says $\hat{y} = 0$; in this case there are no FPs and no TPs. At the other extreme, the decision function always says $\hat{y} = 1$; in this case all decisions are either FPs or TPs.

The ROC is a concave, nondecreasing function illustrated in Figure 6.3. The closer to the top left corner it goes, the better. The best ROC for discrimination goes straight up to (0,1) and then makes a sharp turn to the right. The worst ROC is the diagonal line connecting (0,0) and (1,1) achieved by random guessing. The area under the ROC, also known as the *area under the curve* (AUC) synthesizes performance across all operating points and should be selected as a metric when it is likely that the same threshold-parameterized decision function will be applied in very different operating conditions. Given the shapes of the worst (diagonal line) and best (straight up and then straight to the right) ROC curves, you can see that the AUC ranges from 0.5 (area of bottom right triangle) to 1 (area of entire square).[4]

[4]The recall-precision curve is an alternative to understand performance across operating points. It is the curve traced out on the p_{PPV}–p_{TP} plane starting at $(p_{PPV} = 0, p_{TP} = 1)$ and ending at $(p_{PPV} = 1, p_{TP} = 0)$. It has a one-to-one mapping with the ROC and is more easily understood by some people. Jesse Davis and Mark Goadrich. "The Relationship Between Precision-Recall

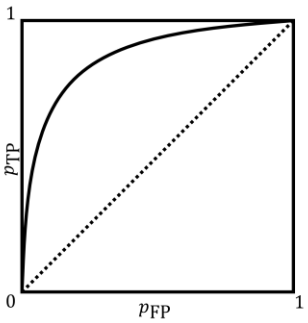


Figure 6.3. *An example receiver operating characteristic (ROC).* Accessible caption. A plot with p_{TP} on the vertical axis and p_{FP} on the horizontal axis. Both axes range from 0 to 1. A dashed diagonal line goes from (0,0) to (1,1) and corresponds to random guessing. A solid concave curve, the ROC, goes from (0,0) to (1,1) staying above and to the left of the diagonal line.

6.2 The Best That You Can Ever Do

As the ThriveGuild data scientist, you have given the problem owner an entire menu of basic performance measures to select from and indicated when different choices are more and less appropriate. The Bayes risk is the most encompassing and most often used performance characterization for a decision function. Let's say that Bayes risk was chosen in the problem specification stage of the machine learning lifecycle, including selecting the costs. Now you are in the modeling stage and need to figure out if the model is performing well. The best way to do that is to optimize the Bayes risk to obtain the best possible decision function with the smallest Bayes risk and compare the current model's Bayes risk to it.

> "The predictability ceiling is often ignored in mainstream ML research. Every prediction problem has an upper bound for prediction—the Bayes-optimal performance. If you don't have a good sense of what it is for your problem, you are in the dark."
>
> —Mert R. Sabuncu, computer scientist at Cornell University

Let us denote the best possible decision function as $\hat{y}^*(\cdot)$ and its corresponding Bayes risk as R^*. They are specified using the minimization of the expected cost:

---

and ROC Curves." In: *Proceedings of the International Conference on Machine Learning*. Pittsburgh, Pennsylvania, USA, Jun. 2006, pp. 233–240.

$$\hat{y}^*(\cdot) = \arg\min_{\hat{y}(\cdot)} E\big[c\big(Y, \hat{y}(X)\big)\big],$$

Equation 6.11

where the expectation is over both X and Y. Because it achieves the minimal cost, the function $\hat{y}^*(\cdot)$ is the best possible $\hat{y}(\cdot)$ by definition. Whatever Bayes risk R^* it has, no other decision function can have a lower Bayes risk R.

We aren't going to work it out here, but the solution to the minimization problem in Equation 6.11 is the Bayes optimal decision function, and takes the following form:

$$\hat{y}^*(\cdot) = \begin{cases} 0, & \Lambda(x) \leq \eta \\ 1, & \Lambda(x) > \eta \end{cases}$$

Equation 6.12

where $\Lambda(x)$, known as the *likelihood ratio*, is defined as:

$$\Lambda(x) = \frac{p_{X|Y}(\,x \mid Y = 1\,)}{p_{X|Y}(\,x \mid Y = 0\,)}$$

Equation 6.13

and η, known as the *threshold*, is defined as:

$$\eta = \frac{c_{10} p_0}{c_{01} p_1}.$$

Equation 6.14

The likelihood ratio is as its name says: it is the ratio of the likelihood functions. It is a scalar value even if the features X are multivariate. As the ratio of two non-negative pdf values, it has the range $[0, \infty)$ and can be viewed as a random variable. The threshold is made up of both costs and prior probabilities. This optimal decision function $\hat{y}^*(\cdot)$ given in Equation 6.12 is known as the *likelihood ratio test*.

6.2.1 Example

As an example, let ThriveGuild's loan approval decision be determined solely by one feature X: the income of the applicant. Recall that we modeled income to be exponentially-distributed in Chapter 3. Specifically, let $p_{X|Y}(\,x \mid Y = 1\,) = 0.5e^{-0.5x}$ and $p_{X|Y}(\,x \mid Y = 0\,) = e^{-x}$, both for $x \geq 0$. Like earlier in this chapter, $p_0 = 0.65$, $p_1 = 0.35$, $c_{10} = 100000$, and $c_{01} = 50000$. Then simply plugging in to Equation 6.13, you'll get:

$$\Lambda(x) = \frac{0.5e^{-0.5x}}{e^{-x}} = 0.5e^{0.5x}, \qquad x \geq 0$$

Equation 6.15

and plugging in to Equation 6.14, you'll get:

$$\eta = \frac{100000}{50000}\frac{0.65}{0.35} = 3.7.$$

Equation 6.16

Plugging these expressions into the Bayes optimal decision function given in Equation 6.12, you'll get:

$$\hat{y}^*(x) = \begin{cases} 0, & 0.5e^{0.5x} \leq 3.7 \\ 1, & 0.5e^{0.5x} > 3.7 \end{cases}$$

Equation 6.17

which can be simplified to:

$$\hat{y}^*(x) = \begin{cases} 0, & x \leq 4 \\ 1, & x > 4 \end{cases}$$

Equation 6.18

by multiplying both sides of the inequalities in both cases by 2, taking the natural logarithm, and then multiplying by 2 again. Applicants with an income less than or equal to 4 are denied and applicants with an income greater than 4 are approved. The expected value of $X \mid Y = 1$ is 2 and the expected value of $X \mid Y = 0$ is 1. Thus in this example, an applicant's income has to be quite a bit higher than the mean to be approved.

You should use the Bayes-optimal risk R^* to lower bound the performance of any machine learning classifier that you might try for a given data distribution.[5] No matter how hard you work or how creative you are, you can never overcome the Bayes limit. So you should be happy if you get close. If the Bayes-optimal risk itself is too high, then the thing to do is to go back to the data understanding and data preparation stages of the machine learning lifecycle and get more informative data.

6.3 Risk Assessment and Calibration

To approve or to deny, that is the question for ThriveGuild. Or is it? Maybe the question is actually: what is the probability that the borrower will default? Maybe the problem is not binary classification, but probabilistic risk assessment. It is certainly an option for you, the data scientist, and the problem owner to consider during problem specification. Thresholding a probabilistic risk assessment yields a classification, but there are a few subtleties for you to weigh.

[5]There are techniques for estimating the Bayes risk of a dataset without having access to its underlying probability distribution. Ryan Theisen, Huan Wang, Lav R. Varshney, Caiming Xiong, and Richard Socher. "Evaluating State-of-the-Art Classification Models Against Bayes Optimality" In: *Advances in Neural Information Processing Systems* 34 (Dec. 2021).

The likelihood ratio ranges from zero to infinity and the threshold value $\eta = 1$ is optimal for equal priors and equal costs. Applying any monotonically increasing function to both the likelihood ratio and the threshold still yields a Bayes optimal decision function with the same risk R^*. That is,

$$\hat{y}^*(\cdot) = \begin{cases} 0, & g\big(\Lambda(x)\big) \leq g(\eta) \\ 1, & g\big(\Lambda(x)\big) > g(\eta) \end{cases}$$

Equation 6.19

for any monotonically increasing function $g(\cdot)$ is still optimal.

It is somewhat more natural to think of a *score* $s(x)$ to be in the range [0,1] because it corresponds to the label values $y \in \{0,1\}$ and could also potentially be interpreted as a probability. The score, a continuous-valued output of the decision function, can then be thought of as a confidence in the prediction and be obtained by applying a suitable g function to the likelihood ratio. In this case, 0.5 is the threshold for equal priors and costs. Intermediate score values are less confident and extreme score values (towards 0 and 1) are more confident. Just as the likelihood ratio may be viewed as a random variable, the score may also be viewed as a random variable S. The *Brier score* is an appropriate performance metric for the continuous-valued output score of the decision function:

$$\text{Brier score} = E[(S - Y)^2].$$

Equation 6.20

It is the mean-squared error of the score S with respect to the true label Y. For a finite number of samples $\{(s_1, y_1), \ldots, (s_n, y_n)\}$, you can compute it as:

$$\text{Brier score} = \frac{1}{n}\sum_{j=1}^{n}\big(s_j - y_j\big)^2.$$

Equation 6.21

The Brier score decomposes into the sum of two separable components: *calibration* and *refinement*.[6] The concept of calibration is that the predicted score corresponds to the proportion of positive true labels. For example, a bunch of data points all having a calibrated score of $s = 0.7$ implies that 70% of them have true label $y = 1$ and 30% of them have true label $y = 0$. Said another way, perfect calibration implies that the probability of the true label Y being 1 given the predicted score S being s is the value s itself: $P(Y = 1 \mid S = s) = s$. Calibration is important for probabilistic risk assessments: a perfectly calibrated score can be interpreted as a probability of predicting one class or the other. It is also an important concept for evaluating causal inference methods, described in Chapter 8, for algorithmic fairness, described in Chapter 10, and for communicating uncertainty, described in Chapter 13.

[6]José Hernández-Orallo, Peter Flach, and Cèsar Ferri. "A Unified View of Performance Metrics: Translating Threshold Choice into Expected Classification Loss." In: *Journal of Machine Learning Research* 13 (Oct. 2012), pp. 2813–2869.

Since any monotonically increasing transformation $g(\cdot)$ can be applied to a decision function without changing its ability to discriminate, you can improve the calibration of a decision function by finding a better $g(\cdot)$. The calibration loss quantitatively captures how close a decision function is to perfect calibration. The refinement loss is a sort of variance of how tightly the true labels distribute around a given score. For $\{(s_1, y_1), \ldots, (s_n, y_n)\}$ that have been sorted by their score values and binned into k groups $\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$ with average values $\{(\bar{s}_1, \bar{y}_1), \ldots, (\bar{s}_k, \bar{y}_k)\}$ within the bins

$$\text{calibration loss} = \frac{1}{n}\sum_{i=1}^{k} \|\mathcal{B}_i\| \, (\bar{s}_i - \bar{y}_i)^2$$

$$\text{refinement loss} = \frac{1}{n}\sum_{i=1}^{k} \|\mathcal{B}_i\| \, \bar{y}_i(1 - \bar{y}_i).$$

Equation 6.22

As stated earlier, the sum of the calibration loss and refinement loss is the Brier score.

A *calibration curve*, also known as a reliability diagram, shows the $(\bar{s}_k, \bar{y}_k)$ values as a plot. One example is shown in Figure 6.4. The closer to a straight diagonal from (0,0) to (1,1), the better. Plotting this curve is a good diagnostic tool for you to understand the calibration of a decision function.

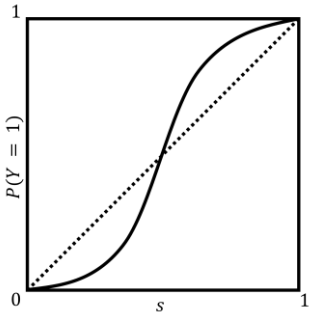


Figure 6.4. *An example calibration curve.* Accessible caption. A plot with $P(Y = 1)$ on the vertical axis and s on the horizontal axis. Both axes range from 0 to 1. A dashed diagonal line goes from (0,0) to (1,1) and corresponds to perfect calibration. A solid S-shaped curve, the calibration curve, goes from (0,0) to (1,1) starting below and to the right of the diagonal line before crossing over to being above and to the left of the diagonal line.

6.4 Summary

- Four possible events result from binary decisions: false negatives, true negatives, false positives, and true positives.
- Different ways to combine the probabilities of these events lead to classifier performance metrics

appropriate for different real-world contexts.

- One important one is Bayes risk: the combination of the false negative probability and false positive probability weighted by both the costs of those errors and the prior probabilities of the labels. It is the basic basic performance measure for the first attribute of safety and trustworthiness.
- Detection theory, the study of optimal decisions, which provides fundamental limits to how well machine learning models may ever perform is a tool for you to assess the basic performance of your models.
- Decision functions may output continuous-valued scores rather than only hard, zero or one, decisions. Scores indicate confidence in a prediction. Calibrated scores are those for which the score value is the probability of a sample belonging to a label class.

7

Supervised Learning

The (fictional) information technology company JCN Corporation is reinventing itself and changing its focus to artificial intelligence and cloud computing. As part of managing its talent during this enterprise transformation, it is conducting a machine learning project to estimate the expertise of its employees from a variety of data sources such as self-assessments of skills, work artifacts (patents, publications, software documentation, service claims, sales opportunities, etc.), internal non-private social media posts, and tabular data records including the employee's length of service, reporting chain, and pay grade. A random subset of the employees has been explicitly evaluated on a binary yes/no scale for various AI and cloud skills, which constitute the labeled training data for machine learning. JCN Corporation's data science team has been given the mission to predict the expertise evaluation for all the other employees in the company. For simplicity, let's focus on only one of the expertise areas: serverless architecture.

Imagine that you are on JCN Corporation's data science team and have progressed beyond the problem specification, data understanding, and data preparation phases of the machine learning lifecycle and are now at the modeling phase. By applying detection theory, you have chosen an appropriate quantification of performance for predicting an employee's skill in serverless architecture: the error rate—the Bayes risk with equal costs for false positives and false negatives—introduced in Chapter 6.

It is now time to get down to the business of learning a decision function (a classifier) from the training data that generalizes well to predict expertise labels for the unlabeled employees. Deep learning is one family of classifiers that is on the tip of everyone's tongue. It is certainly one option for you, but there are many other kinds of classifiers too. How will you evaluate different classification algorithms to select the best one for your problem?

> "My experience in industry strongly confirms that deep learning is a narrow sliver of methods needed for solving complex automated decision making problems."
>
> —Zoubin Ghahramani, chief scientist at Uber

A very important concept in practicing machine learning, first mentioned in Chapter 2, is the *no free lunch theorem*. There is no one single machine learning method that is best for all datasets.[1] What is a good choice for one dataset might not be so great for another dataset. It all depends on the characteristics of the dataset and the *inductive bias* of the method: the assumptions on how the classifier should *generalize* outside the training data points. The challenge in achieving good generalization and a small error rate is protecting against *overfitting* (learning a model that too closely matches the idiosyncrasies of the training data) and *underfitting* (learning a model that does not adequately capture the patterns in the training data). The goal is to get to the Goldilocks point where things are not too hot (overfitting) and not too cold (underfitting), but *just right*.

An implication of the no free lunch theorem is that you must try several different methods for the JCN Corporation expertise problem and see how they perform empirically before deciding on one over another. Simply brute forcing it—training all the different methods and computing their test error to see which one is smallest—is common practice, but you decide that you want to take a more refined approach and analyze the inductive biases of different classifiers. Your analysis will determine the *domains of competence* of various classifiers: what types of datasets do they perform well on and what type of datasets do they perform poorly on.[2] Recall that competence or basic accuracy is the first attribute of trustworthy machine learning as well as the first half of safety.

Why would you want to take this refined approach instead of simply applying a bunch of machine learning methods from software packages such as scikit-learn, tensorflow, and pytorch without analyzing their inductive biases? First, you have heeded the admonitions from earlier chapters to be safe and to not take shortcuts. More importantly, however, you know you will later be creating new algorithms that respect the second (reliability) and third (interaction) attributes of trustworthiness. You must not only be able to apply algorithms, you must be able to analyze and evaluate them before you can create. Now go forth and analyze classifiers for inventorying expertise in the JCN Corporation workforce.

7.1 Domains of Competence

Different classifiers work well on different datasets depending on their characteristics.[3] But what characteristics of a dataset matter? What are the parameters of a domain of competence? A key concept to answer those questions is the *decision boundary*. In Chapter 6, you learned that the Bayes optimal decision function is a likelihood ratio test which is a threshold of the one-dimensional likelihood ratio. If you invert the likelihood ratio, you can go back to the feature space with d feature dimensions $x^{(1)}, \ldots, x^{(d)}$ and trace out surfaces to which that single threshold value maps. The collection of these surfaces is a *level set* of the likelihood ratio function and is known as the decision boundary. Imagine the likelihood ratio function being like the topography and bathymetry of the Earth. Anything underwater receives the classification $\hat{y} = 0$ (employee is unskilled in serverless architecture) and anything above

[1]David H. Wolpert. "The Lack of A Priori Distinctions Between Learning Algorithms." In: *Neural Computation* 8.7 (Oct. 1996), pp. 1341–1390.

[2]Tin Kam Ho and Ester Bernadó-Mansilla. "Classifier Domains of Competence in Data Complexity Space." In: *Data Complexity in Pattern Recognition*. Ed. by Mitra Basu and Tin Kam Ho. London, England, UK: Springer, 2006, pp. 135–152.

[3]Maniel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" In: *Journal of Machine Learning Research* 15 (Oct. 2014), pp. 3133–3181.

water receives the classification $\hat{y} = 1$ (employee is skilled in serverless architecture). Sea level is the threshold value and the coastline is the level set or decision boundary. An example of a decision boundary for a two-dimensional feature space is shown in Figure 7.1.

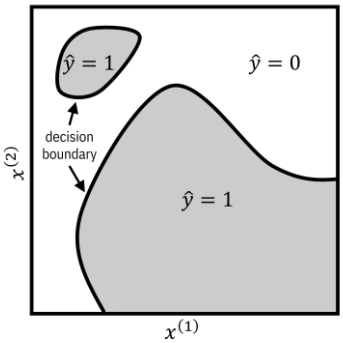


Figure 7.1. *An example of a decision boundary in a feature space. The gray regions correspond to feature values for which the decision function predicts employees are skilled in serverless architecture. The white regions correspond to features for which the decision function predicts employees are unskilled in serverless architecture. The black lines are the decision boundary.* Accessible caption. A stylized plot with the first feature dimension $x^{(1)}$ on the horizontal axis and the second feature dimension $x^{(2)}$ on the vertical axis. The space is partitioned into a couple of blob-like gray regions labeled $\hat{y} = 1$ and a white region labeled $\hat{y} = 0$. The boundary between the regions is marked as the decision boundary. Classifier regions do not have to be all one connected component.

Three key characteristics of a dataset determine how well the inductive biases of a classifier match the dataset:

1. overlap of data points from the two class labels near the decision boundary,
2. linearity or nonlinearity of the decision boundary, and
3. number of data points, their density, and amount of clustering.

Classifier domains of competence are defined in terms of these three considerations.[4] Importantly, domains of competence are relative notions: does one classification algorithm work better than others?[5] They are not absolute notions, because at the end of the day, the absolute performance is limited by the Bayes optimal risk defined in Chapter 6. For example, one classification method that you tried may work better than others on datasets with a lot of class overlap near the decision boundary, nearly linear shape of the decision boundary, and not many data points. Another classification method may work better than

[4]In the scope of this chapter, the JCN team use these characteristics qualitatively as a means of gaining intuition. Quantitative measures for these characteristics are described by Tin Kam Ho and Mitra Basu. "Complexity Measures of Supervised Classification Problems." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.3 (Mar. 2002), pp. 289–300.

[5]For the purposes of this chapter, 'work better' is only in terms of basic performance (the first attribute of trustworthiness), not reliability or interaction (the second and third attributes of trustworthiness). Classifier domains of reliability and domains of quality interaction can also be defined.

others on datasets without much class overlap near a tortuously-shaped decision boundary. Yet another classification method may work better than others on very large datasets. In the remainder of this chapter, you will analyze many different supervised learning algorithms. The aim is not only describing how they work, but analyzing their inductive biases and domains of competence.

7.2 Two Ways to Approach Supervised Learning

Let's begin by cataloging what you and the team of JCN Corporation data scientists have at your disposal. Your training dataset consists of n samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ independently drawn from the probability distribution $p_{X,Y}(x, y)$. The features x_j sampled from the random variable X are numerical or categorical quantities derived from skill self-assessments, work artifacts, and so on. There are d features, so x_j is a d-dimensional vector. The labels y_j, sampled from the random variable Y, are the binary (zero or one) expertise evaluations on serverless architecture. Importantly, you do not have access to the precise distribution $p_{X,Y}(x, y)$, but only to the finite number of samples contained in the training dataset drawn from the distribution. This is the key difference between the supervised machine learning problem and the Bayesian detection problem introduced in Chapter 6. The goal is the same in both the machine learning and detection problems: find a decision function $\hat{y}$ that predicts labels from features.

What are your options to find the classifier $\hat{y}$ based on the training data? You cannot simply minimize the Bayes risk functional or the probability of error directly, because that would rely on full knowledge of the probability distribution of the features and labels, which you do not have. You and the team have two options:

1. *plug-in approach*: estimate the likelihood functions and prior probabilities from the training data, and plug them into the Bayes optimal likelihood ratio test described in Chapter 6, or
2. *risk minimization*: optimize a classifier over an empirical approximation to the error rate computed on the training data samples.

There are specific methods within these two broad categories of supervised classification algorithms. A mental model for different ways of doing supervised machine learning is shown in Figure 7.2.

7.3 Plug-In Approach

First, you and the rest of the JCN Corporation data science team try out plug-in methods for supervised classification. The main idea is to use the training data to estimate the likelihood functions $p_{X|Y}(x \mid y = 0)$ and $p_{X|Y}(x \mid y = 1)$, and then plug them into the likelihood ratio to obtain the classifier.

7.3.1 Discriminant Analysis

One of the most straightforward plug-in methods, *discriminant analysis*, assumes a parametric form for the likelihood functions and estimates their parameters. Then just like in Chapter 6, it obtains a decision function by taking the ratio of these likelihood functions and comparing them to a threshold η. The actual underlying likelihood functions do not have to be exactly their assumed forms and usually aren't in practice. If they are somewhat close, that is good enough. The assumed parametric form is precisely the inductive bias of discriminant analysis.

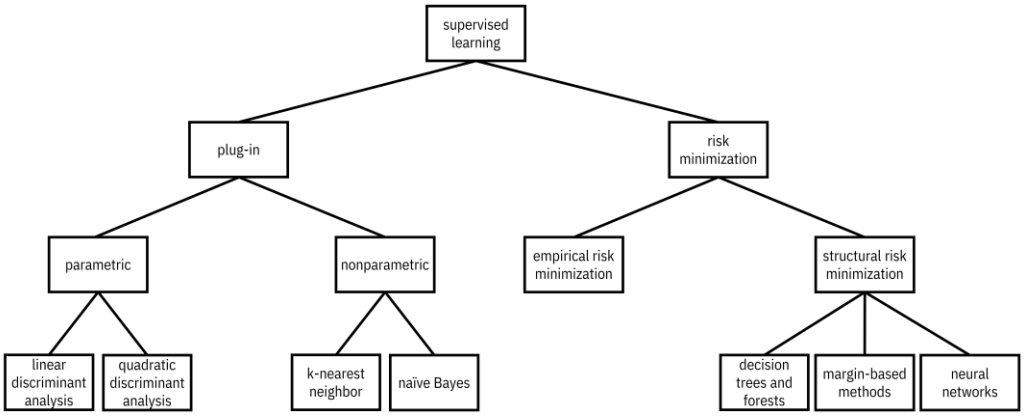


Figure 7.2. *A mental model for different ways of approaching supervised machine learning.* A hierarchy diagram with supervised learning at its root. Supervised learning has children plug-in and risk minimization. Plug-in has children parametric and nonparametric. Parametric has children linear discriminant analysis and quadratic discriminant analysis. Nonparametric has children k-nearest neighbor and naïve Bayes. Risk minimization has children empirical risk minimization and structural risk minimization. Structural risk minimization has children decision trees and forests, margin-based methods, and neural networks.

If the assumed parametric form for the likelihood functions is multivariate Gaussian in d dimensions with mean parameters μ_0 and μ_1 and covariance matrix parameters Σ_0 and Σ_1,[6] then the first step is to compute their empirical estimates $\hat{\mu}_0$, $\hat{\mu}_1$, $\hat{\Sigma}_0$, and $\hat{\Sigma}_1$ from the training data, which you know how to do from Chapter 3. The second step is to plug those estimates into the likelihood ratio to get the classifier decision function. Under the Gaussian assumption, the method is known as *quadratic discriminant analysis* because after rearranging and simplifying the likelihood ratio, the quantity compared to a threshold turns out to be a quadratic function of x. If you further assume that the two covariance matrices Σ_0 and Σ_1 are the same matrix Σ, then the quantity compared to a threshold is even simpler: it is a linear function of x, and the method is known as *linear discriminant analysis*.

Figure 7.3 shows examples of linear and quadratic discriminant analysis classifiers in $d = 2$ dimensions trained on the data points shown in the figure. The red diamonds are the employees in the training set unskilled at serverless architecture. The green squares are the employees in the training set skilled at serverless architecture. The domain of competence for linear and quadratic discriminant analysis is datasets whose decision boundary is mostly linear, with a dense set of data points of both class labels near that boundary.

[6]The mathematical form of the likelihood functions is: $p_{X|Y}(x \mid y = 0) = \frac{1}{\sqrt{(2\pi)^d det(\Sigma_0)}} e^{-\frac{1}{2}(x-\mu_0)^T \Sigma_0^{-1}(x-\mu_0)}$ and $p_{X|Y}(x \mid y = 1) = \frac{1}{\sqrt{(2\pi)^d det(\Sigma_1)}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1)}$, where det is the matrix determinant function.

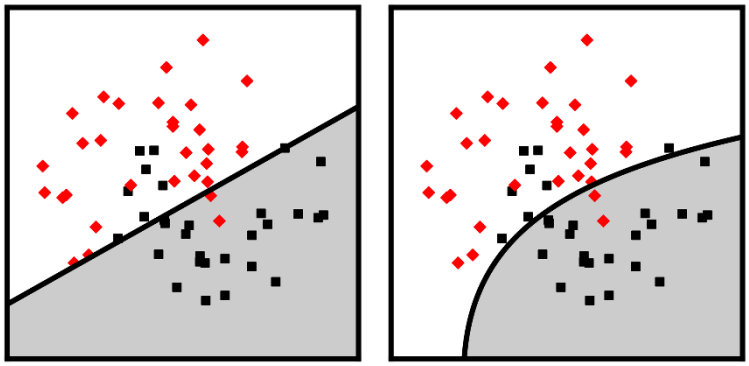


Figure 7.3. *Examples of linear discriminant analysis (left) and quadratic discriminant analysis (right) classifiers.* Accessible Caption. Stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The linear discriminant decision boundary is a straight line that cuts through the middle of the two classes. The quadratic discriminant decision boundary is a smooth curve that turns a little to enclose one of the classes.

7.3.2 Nonparametric Density Estimation

You continue your quest to analyze different classifiers for estimating the expertise of JCN employees. Instead of assuming a parametric form for the likelihood functions like in discriminant analysis, you try to estimate the likelihood functions in a nonparametric fashion. The word *nonparametric* is a misnomer. It does not mean that there are no parameters in the estimated likelihood function at all; it means that the number of parameters is on par with the number of training data points.

A common way of estimating a likelihood function nonparametrically is *kernel density estimation*. The idea is to place a smooth function like a Gaussian pdf centered on each of the training data points and take the normalized sum of those functions as the estimate of the likelihood function. In this case, the parameters are the centers of the smooth functions, so the number of parameters equals the number of data points. Doing this for both likelihood functions separately, taking their ratio, and comparing to a threshold yields a valid classifier. However, it is a pretty complicated classifier. You would need a lot of data to get a good kernel density estimate, especially when the data has a lot of feature dimensions d.

Instead of doing the full density estimate, a simplification is to assume that all the feature dimensions of X are mutually independent. Under this assumption, the likelihood functions factor into products of one-dimensional pdfs that can be estimated separately with much less data. If you take the ratio of these products of one-dimensional pdfs (a likelihood ratio) and compare to a threshold, voilà, you have a *naïve Bayes* classifier. The name of this method contains 'naïve' because it is somewhat naïve to assume that all feature dimensions are independent—they never are in real life. It contains 'Bayes' because of plugging in to the Bayes-optimal likelihood ratio test. Often, this classifier does not outperform other classifiers in terms of accuracy, so its domain of competence is often non-existent.

A different nonparametric method is the *k-nearest neighbor* classifier. The idea behind it is very simple. Look at the labels of the k closest training data points and predict whichever label is more common in those nearby points. A distance metric is needed to measure 'close' and 'near.' Typically, Euclidean distance (the normal straight-line distance) is used, but other distance metrics could be used

instead. The k-nearest neighbor method works better than other classifiers when the decision boundary is very wiggly and broken up into lots of components, and when there is not much overlap in the classes. Figure 7.4 shows examples of naïve Bayes and k-nearest neighbor classifiers in two dimensions. The k-nearest neighbor classifier in the figure has $k = 5$.

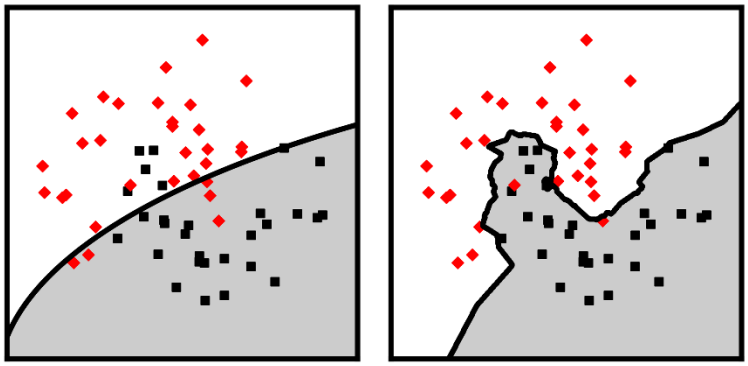


Figure 7.4. *Examples of naïve Bayes (left) and k-nearest neighbor (right) classifiers.* Accessible caption. Stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The naïve Bayes decision boundary is a smooth curve that turns a little to enclose one of the classes. The k-nearest neighbor decision boundary is very jagged and traces out the positions of the classes closely.

7.4 Risk Minimization Basics

You and the JCN team have tried out a few plug-in methods for your task of predicting which employees are skilled in serverless architecture and are ready to move on to a different category of machine learning methods: *risk minimization*. Whereas plug-in methods take one step back from the Bayes-optimal likelihood ratio test by estimating the likelihood functions from data, risk minimization takes two steps back and directly tries to find decision functions or decision boundaries that minimize an estimate of the Bayes risk.

7.4.1 Empirical Risk Minimization

Remember from Chapter 6 that the probability of error p_E, the special case of the Bayes risk with equal costs that you've chosen as the performance metric, is:

$$p_E = p_0 P(\hat{y}(X) = 1 \mid Y = 0) + p_1 P(\hat{y}(X) = 0 \mid Y = 1).$$

Equation 7.1

The prior probabilities of the class labels p_0 and p_1 multiply the probabilities of the events when the decision function is wrong $P(\hat{y}(X) = 1 \mid Y = 0)$ and $P(\hat{y}(X) = 0 \mid Y = 1)$. You cannot directly compute the error probability because you do not have access to the full underlying probability distribution. But is there an approximation to the error probability that you can compute using the training data?

First, because the training data is sampled i.i.d. from the underlying distribution, the proportion of employees in the training data set skilled and unskilled at serverless architecture will approximately match the prior probabilities p_0 and p_1, so you do not have to worry about them explicitly. Second, the probabilities of both the false positive event $P(\hat{y}(X) = 1 \mid Y = 0)$ and false negative event $P(\hat{y}(X) = 0 \mid Y = 1)$ event can be expressed collectively as $P(\hat{y}(X) \neq Y)$, which corresponds to $\hat{y}(x_j) \neq y_j$ for training data samples. The *zero-one loss* function $L\left(y_j, \hat{y}(x_j)\right)$ captures this by returning the value 1 for $\hat{y}(x_j) \neq y_j$ and the value 0 for $\hat{y}(x_j) = y_j$. Putting all these things together, the empirical approximation to the error probability, known as the *empirical risk* R_{emp}, is:

$$R_{\text{emp}} = \frac{1}{n}\sum_{j=1}^{n} L\left(y_j, \hat{y}(x_j)\right).$$

Equation 7.2

Minimizing the empirical risk over all possible decision functions $\hat{y}$ is a possible classification algorithm, but not one that you and the other JCN Corporation data scientists evaluate just yet. Let's understand why not.

7.4.2 Structural Risk Minimization

Without any constraints, you can find a decision function that brings the empirical risk to zero but does not generalize well to new unseen data points. At the extreme, think about a classifier that memorizes the training data points and gets them perfectly correct, but always predicts $\hat{y} = 0$ (unskilled at serverless architecture) everywhere else. This is not the desired behavior—the classifier has overfit. So just minimizing the empirical risk does not yield a competent classifier. This memorizing classifier is pretty complex. There's nothing smooth or simple about it because it has as many discontinuities as there are training set employees skilled at serverless architecture.

Constraining the complexity of the classifier forces it to not overfit. To be a bit more precise, if you constrain the decision function $\hat{y}$ to be an element of some class of functions or *hypothesis space* $\mathcal{F}$ that only includes low-complexity functions, then you will prevent overfitting. But you can go too far with the constraints as well. If the hypothesis space is too small and does not contain any functions with the capacity to capture the important patterns in the data, it may underfit the data and not generalize either. It is important to control the hypothesis space to be just right. This idea is known as the *structural risk minimization principle*.

Figure 7.5 shows the idea in pictorial form using a sequence of nested hypothesis spaces $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_6$. As an example of nested hypothesis spaces, $\mathcal{F}_1$ could be all constant functions, $\mathcal{F}_2$ could be all linear functions, $\mathcal{F}_3$ could be all quadratic functions, and $\mathcal{F}_6$ could be all polynomial functions. $\mathcal{F}_1$ contains the least complex $\hat{y}$ functions while $\mathcal{F}_6$ also contains more complex $\hat{y}$ functions. $\mathcal{F}_1$ underfits as it has large values for both the empirical risk R_{emp} calculated on the training data and the probability of error p_{E}, which measures generalization. $\mathcal{F}_6$ overfits as it has zero R_{emp} and a large value for p_{E}. $\mathcal{F}_2$ achieves a good balance and is just right.

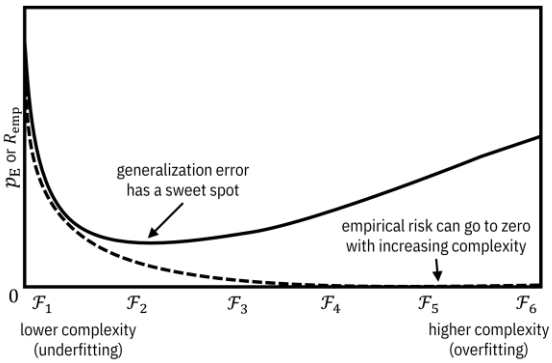


Figure 7.5. *Illustration of the structural risk minimization principle.* Accessible caption. A plot with p_E or R_{emp} on the vertical axis and increasing complexity of hypothesis spaces on the horizontal axis. The empirical risk decreases all the way to zero with increasing complexity. The generalization error first decreases and then increases. It has a sweet spot in the middle.

The hypothesis space $\mathcal{F}$ is the inductive bias of the classifier. Thus, within the paradigm of the structural risk minimization principle, different choices of hypothesis spaces yield different domains of competence. In the next section, you and your team of JCN data scientists analyze several different risk minimization classifiers popularly used in practice, including decision trees and forests, margin-based classifiers (logistic regression, support vector machines, etc.), and neural networks.

7.5 Risk Minimization Algorithms

You are now analyzing the competence of some of the most popular classifiers used today that fit into the risk minimization paradigm. The basic problem is to find the function f within the hypothesis space $\mathcal{F}$ that minimizes the average loss function $L\left(y_j, f\left(x_j\right)\right)$:

$$\hat{y}(\cdot) = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} L\left(y_j, f\left(x_j\right)\right).$$

Equation 7.3

This equation may look familiar because it is similar to the Bayesian detection problem in Chapter 6. The function in the hypothesis space that minimizes the sum of the losses on the training data is $\hat{y}$. Different methods have different hypothesis spaces $\mathcal{F}$ and different loss functions $L(\cdot,\cdot)$. An alternative way to control the complexity of the classifier is not through changing the hypothesis space $\mathcal{F}$, but through a complexity penalty or *regularization term* $J(\cdot)$ weighted by a *regularization parameter* λ:

$$\hat{y}(\cdot) = arg\ \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} L\left(y_j, f\left(x_j\right)\right) + \lambda J(f).$$

Equation 7.4

The choice of regularization term J also yields an inductive bias for you to analyze.

7.5.1 Decision Trees and Forests

One of the simplest hypothesis spaces is the set of *decision stumps* or *one-rules*. These classifiers create a single split along a single feature dimension like a numerical expertise self-assessment feature or a length of service feature. Any data point whose value is on one side of a threshold gets classified as skilled in serverless architecture, and on the other side as unskilled in serverless architecture. For categorical features, a split is just a partitioning of the values into two groups. The other features besides the one participating in the decision stump can be anything. An example of a decision stump is shown in Figure 7.6 as a node with two branches and also through its decision boundary.

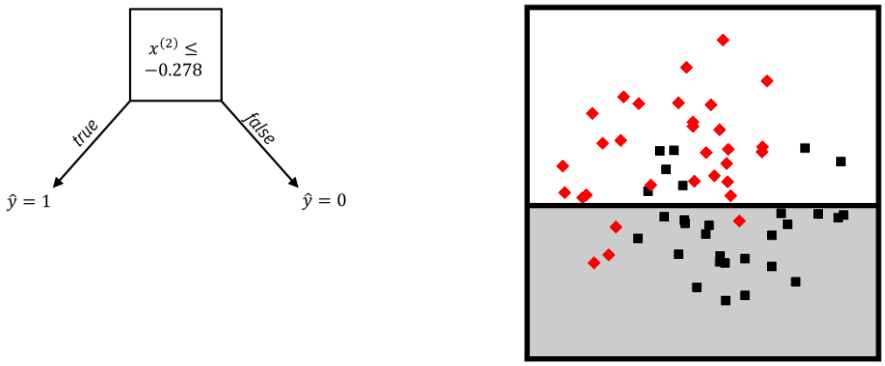


Figure 7.6. *An example of a decision stump classifier.* Accessible caption. On the left is a decision node $x^{(2)} \leq -0.278$. When it is true, $\hat{y} = 1$ and when it is false, $\hat{y} = 0$. On the right is a stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The decision boundary is a horizontal line.

The hypothesis space of *decision trees* includes decision functions with more complexity than decision stumps. A decision tree is created by splitting on single feature dimensions within each branch of the decision stump, splitting within those splits, and so on. An example of a decision tree with two levels is shown in Figure 7.7. Decision trees can go much deeper than two levels to create fairly complex decision boundaries. An example of a complex decision boundary from a decision tree classifier is shown in Figure 7.8.

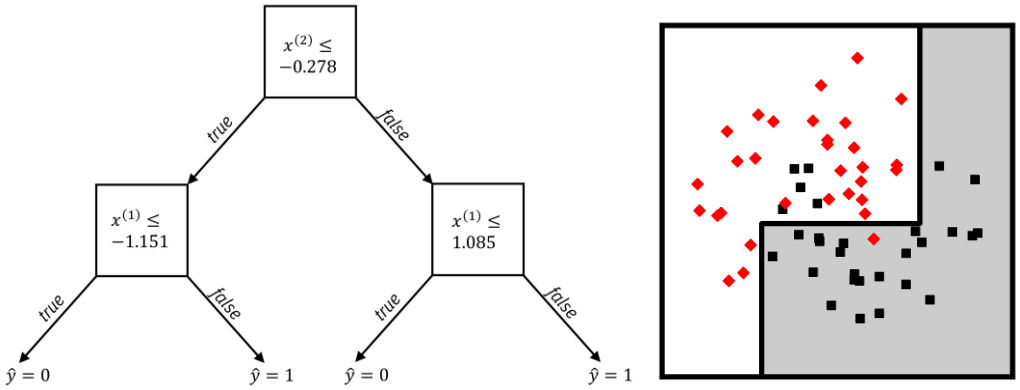


Figure 7.7. *An example of a two-level decision tree classifier.* Accessible caption. On the left is a decision node $x^{(2)} \leq -0.278$. When it is true, there is another decision node $x^{(1)} \leq -1.151$. When this decision node is true, $\hat{y} = 0$ and when it is false, $\hat{y} = 1$. When the top decision node is false, there is another decision node $x^{(1)} \leq 1.085$. When this decision node is true, $\hat{y} = 0$ and when it is false, $\hat{y} = 1$. On the right is a stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The decision boundary is a made up of three line segments: the first segment is vertical, it turns right into a horizontal segment, and then up into another vertical segment.

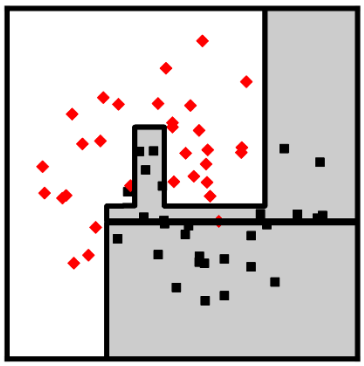


Figure 7.8. *An example decision tree classifier with many levels.* Accessible caption. A stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The decision boundary is a made up of several vertical and horizontal segments.

The hypothesis space of *decision forests* is made up of *ensembles* of decision trees that vote for their prediction, possibly with an unequal weighting given to different trees. The weighted majority vote from the decision trees is the overall classification. The mental model for a decision forest is illustrated in Figure 7.9 and an example decision boundary is given in Figure 7.10.

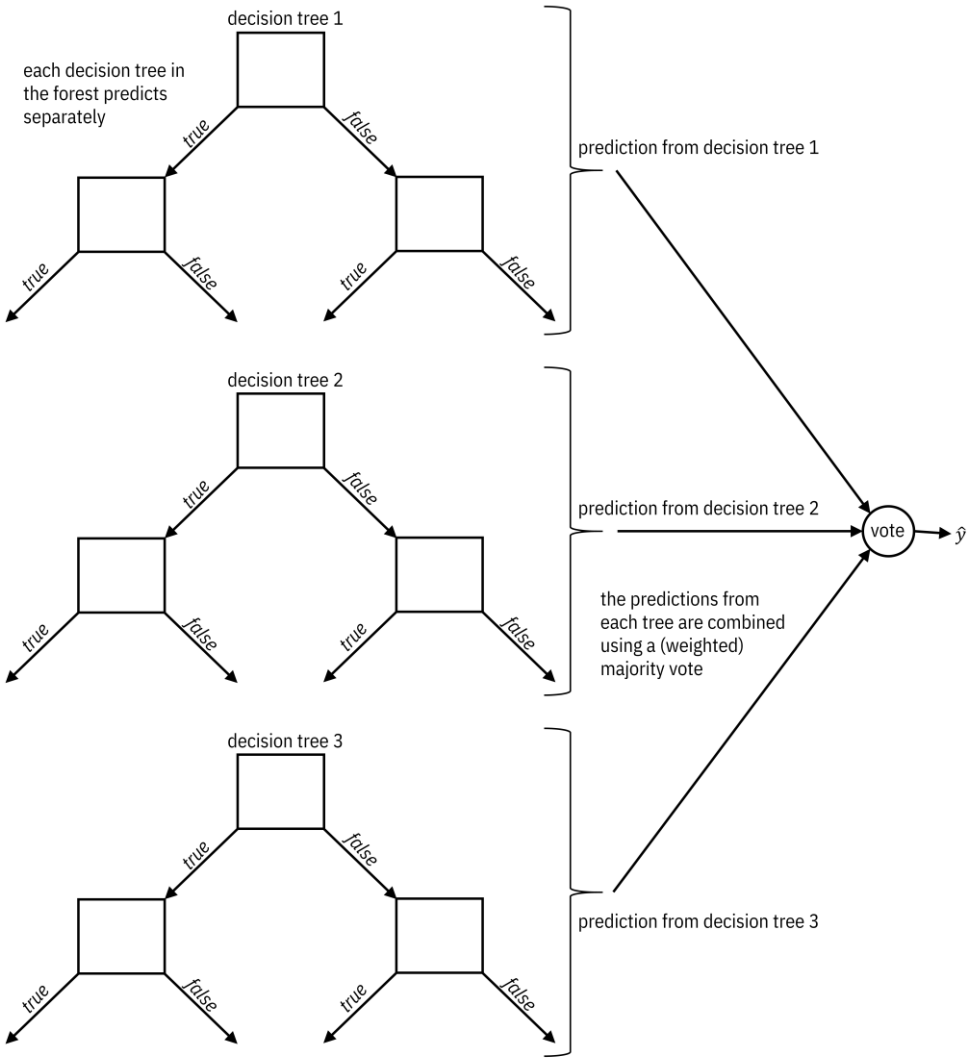


Figure 7.9. *A mental model for a decision forest.* Accessible caption. Three individual decision trees each predict separately. Their predictions feed into a vote node which outputs $\hat{y}$. The predictions from each tree are combined using a (weighted) majority vote.

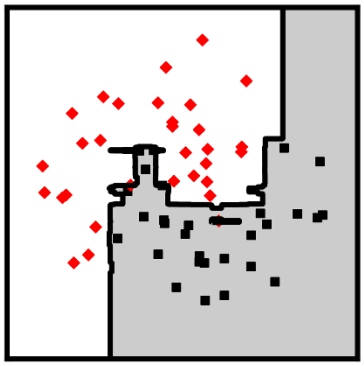


Figure 7.10. A*n example decision forest classifier*. Accessible caption. Stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The decision boundary is fairly jagged with mostly axis-aligned segments and traces out the positions of the classes closely.

Decision stumps and decision trees can be directly optimized for the zero-one loss function that appears in the empirical risk.[7] More commonly, however, greedy heuristic methods are employed for learning decision trees in which the split for each node is done one at a time, starting from the root and progressing to the leaves. The split is chosen so that each branch is as pure as can be for the two classes: mostly just employees skilled at serverless architecture on one side of the split, and mostly just employees unskilled at serverless architecture on the other side of the split. The purity can be quantified by two different information-theoretic measures, information gain and Gini index, which were introduced in Chapter 3. Two decision tree algorithms are popularly-used: the *C5.0 decision tree* that uses information gain as its splitting criterion, and the *classification and regression tree (CART)* that uses Gini index as its splitting criterion. The depth of decision trees is controlled to prevent overfitting. The domain of competence of C5.0 and CART decision trees is tabular datasets in which the phenomena represented in the features tend to have threshold and clustering behaviors without much class overlap.

Decision forests are made up of a lot of decision trees. C5.0 or CART trees are usually used as these base classifiers. There are two popular ways to train decision forests: *bagging* and *boosting*. In bagging, different subsets of the training data are presented to different trees and each tree is trained separately. All trees have equal weight in the majority vote. In boosting, a sequential procedure is followed. The first tree is trained in the standard manner. The second tree is trained to focus on the training samples that the first tree got wrong. The third tree focuses on the errors of the first two trees, and so on. Earlier trees receive greater weight. Decision forests have good competence because of the diversity of their base classifiers. As long as the individual trees are somewhat competent, any unique mistake that any one tree makes is washed out by the others for an overall improvement in generalization.

The *random forest* classifier is the most popular bagged decision forest and the *XGBoost* classifier is the most popular boosted decision forest. Both have very large domains of competence. They are robust

[7]Oktay Günlük, Jayant Kalagnanam, Minhan Li, Matt Menickelly, and Katya Scheinberg. "Optimal Generalized Decision Trees via Integer Programming." arXiv:1612.03225, 2019.

and work extremely well for almost all kinds of structured datasets. They are the first-choice algorithms for practicing data scientists to achieve good accuracy models with little to no tuning of parameters.

7.5.2 Margin-Based Methods

Margin-based classifiers constitute another popular family of supervised learning algorithms. This family includes *logistic regression* and *support vector machines* (SVMs). The hypothesis space of margin-based classifiers is more complex than decision stumps, but in a different way than decision trees. Margin-based classifiers allow any linear decision boundary rather than only ones parallel to single feature dimensions. Going even further, margin-based classifiers can have nonlinear decision boundaries in the original feature space by applying nonlinear functions to the features and finding linear decision boundaries in that transformed space.[8]

The main concept of these algorithms is the *margin*, the distance of data points to the decision boundary. With a linear decision boundary, the form of the classifier is $\hat{y}(x_j) = \text{step}(w^T x_j) = (\text{sign}(w^T x_j) + 1)/2$,[9] where w is a *weight vector* or *coefficient vector* that is learned from the training data. The absolute value of $w^T x_j$ is the distance of the data point to the decision boundary and is thus the margin of the point. The quantity $w^T x_j$ is positive if x_j is on one side of the hyperplane defined by w and negative if x_j is on the other side. The step function gives a classification of 0 (unskilled at serverless architecture) for negative margin and a classification of 1 (skilled at serverless architecture) for positive margin. The stuff with the sign function (adding one and dividing by two) is just a way to recreate the behavior of the step function.

Surrogates for the zero-one loss function L are used in the risk minimization problem. Instead of taking two arguments, these margin-based loss functions take the single argument $(2y_j - 1)w^T x_j$. When $w^T x_j$ is multiplied by $(2y_j - 1)$, the result is positive for a correct classification and negative for an incorrect classification.[10] The loss is large for negative inputs and small or zero for positive inputs. In logistic regression, the loss function is the logistic loss: $L\left((2y_j - 1)w^T x_j\right) = \log(1 + e^{-(2y_j-1)w^T x_j})$ and in SVMs, the loss function is the hinge loss: $L\left((2y_j - 1)w^T x_j\right) = \max\{0, 1 - (2y_j - 1)w^T x_j\}$. The shape of these loss function curves is shown in Figure 7.11.

The regularization term J for the standard forms of logistic regression and SVMs is $\|w\|^2$, the length-squared of the coefficient vector (also known as the ℓ_2-norm squared). The loss function, regularization term, and nonlinear feature mapping together constitute the inductive bias of the classifier. An alternative regularization term is the sum of the absolute values of the coefficients in w (also known as the ℓ_1-norm), which provides the inductive bias for w to have many zero-valued coefficients. Example linear and nonlinear logistic regression and SVM classifiers are shown in Figure 7.12. The domain of competence for margin-based classifiers is fairly broad: structured datasets of moderate size. SVMs work a little better than logistic regression when the features are noisy.

---

[8]The nonlinear functions of the features are usually *kernel functions*, which satisfy certain mathematical properties that allow for efficient optimization during training.

[9]In the nonlinear case, replace $w^T x_j$ with $w^T k(x_j)$ for a kernel function k. To avoid cluttering up the mathematical notation, always assume that x_j or $k(x_j)$ has a column of all ones to allow for a constant shift.

[10]Computing $(2y_j - 1)$ is the inverse of applying the sign function, adding one, and dividing by two. It is performed to get values -1 and $+1$ from the class labels. When a classification is correct, $w^T x_j$ and $(2y_j - 1)$ have the same sign. Multiplying two numbers with the same sign results in a positive number. When a classification is incorrect, $w^T x_j$ and $(2y_j - 1)$ have different signs. Multiplying two numbers with different signs results in a negative number.

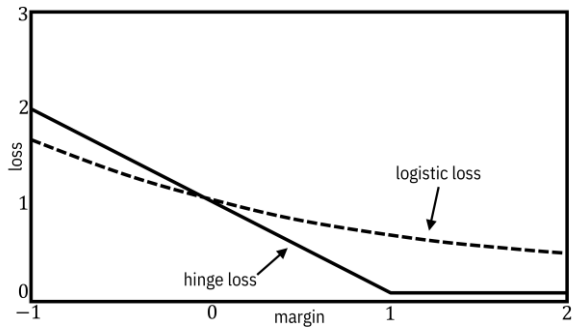


Figure 7.11. *Margin-based loss functions.* Accessible caption. A plot with loss on the vertical axis and margin on the horizontal axis. The logistic loss decreases smoothly. The hinge loss decreases linearly until the point (1,0), after which it is 0 for all larger values of the margin.

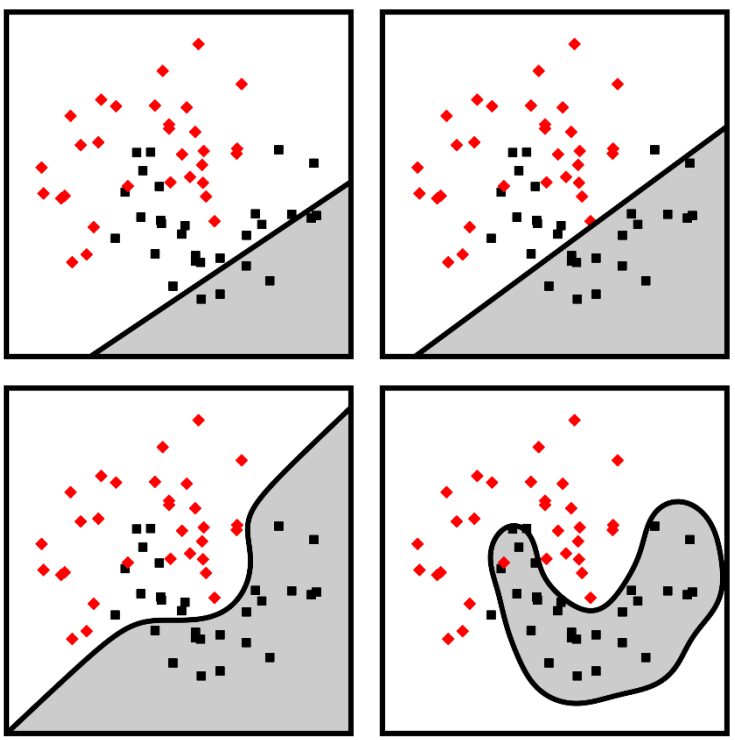


Figure 7.12. *Example linear logistic regression (top left), linear SVM (top right), nonlinear polynomial SVM (bottom left), and nonlinear radial basis function SVM (bottom right) classifiers.* Accessible caption. Stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The linear logistic regression and linear SVM decision boundaries are diagonal lines through the middle of the moons. The polynomial SVM decision boundary is a diagonal line with a smooth bump to better follow the classes. The radial basis function SVM decision boundary smoothly encircles one of the classes with a blob-like region.

7.5.3 Neural Networks

The final family of classifiers that you and the other JCN Corporation data scientists analyze is *artificial neural networks*. Neural networks are all the rage these days because of their superlative performance on high-profile tasks involving large-scale semi-structured datasets (image classification, speech recognition, natural language processing, bioinformatics, etc.), which is their domain of competence. The hypothesis space of neural networks includes functions that are compositions of compositions of compositions of simple functions known as *neurons*. The best way to understand the hypothesis space is graphically as *layers* of neurons, represented as nodes, connected to each other by weighted edges. There are three types of layers: an input layer, possibly several hidden layers, and an output layer. The basic picture to keep in mind is shown in Figure 7.13. The term *deep learning* which is bandied about quite a bit these days refers to *deep neural networks*: architectures of neurons with many many hidden layers.

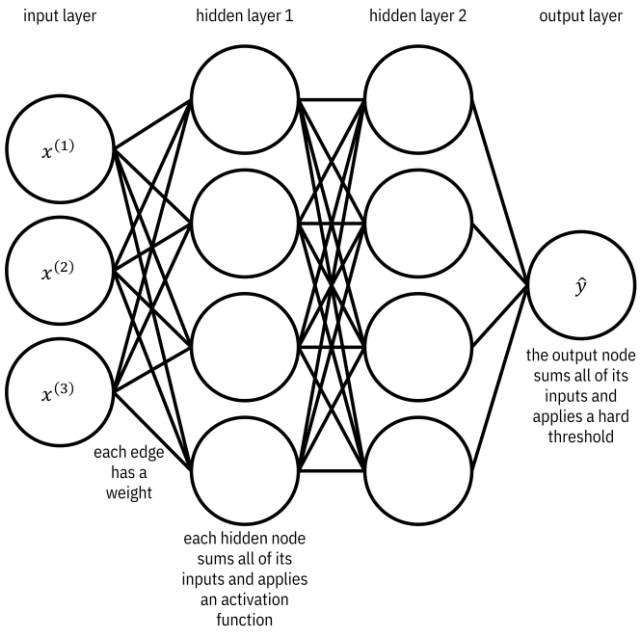


Figure 7.13. *Diagram of a neural network.* Accessible caption. Three nodes on the left form the input layer. They are labeled $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$. To the right of the input layer is hidden layer 1 with four nodes. To the right of hidden layer 1 is hidden layer 2 with four hidden nodes. To the right of hidden layer 2 is one node labeled $\hat{y}$ constituting the output layer. There are edges between each node of one layer and each node of the adjacent layer. Each edge has a weight. Each hidden node sums all of its inputs and applies an activation function. The output node sums all of its inputs and applies a hard threshold.

Logistic regression is actually a very simple neural network with just an input layer and an output node, so let's start there. The input layer is simply a set of nodes, one for each of the d feature dimensions $x^{(1)}, \ldots, x^{(d)}$ relevant for predicting the expertise of employees. They have weighted edges coming out of them, going into the output node. The weights on the edges are the coefficients in w, i.e. $w^{(1)}, \ldots, w^{(d)}$. The

output node sums the weighted inputs, so computes $w^T x$, and then passes the sum through the step function. This overall procedure is exactly the same as logistic regression described earlier, but described in a graphical way.

In the regular case of a neural network with one or more hidden layers, nodes in the hidden layers also start with a weighted summation. However, instead of following the summation with an abrupt step function, hidden layer nodes use softer, more gently changing *activation functions*. A few different activation functions are used in practice, whose choice contributes to the inductive bias. Two examples, the sigmoid or logistic activation function $1/(1+e^{-z})$ and the rectified linear unit (ReLU) activation function $\max\{0, z\}$, are shown in Figure 7.14. The ReLU activation is typically used in all hidden layers of deep neural networks because it has favorable properties for optimization techniques that involve the gradient of the activation function.

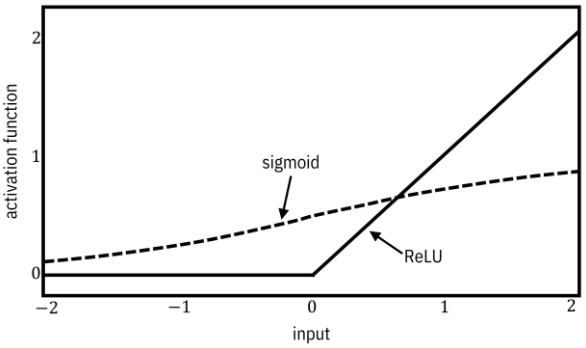


Figure 7.14. *Activation functions.* Accessible caption. A plot with activation function on the vertical axis and input on the horizontal axis. The sigmoid function is a gently rolling S-shaped curve that equals 0.5 at input value 0, approaches 0 as the input goes to negative infinity, and approaches 1 as the input goes to positive infinity. The ReLU function is 0 for all negative inputs and increases linearly starting at 0.

When there are several hidden layers, the outputs of nodes in one hidden layer feed into the nodes of the next hidden layer. Thus, the neural network's computation is a sequence of compositions of weighted sum, activation function, weighted sum, activation function, and so on until reaching the output layer, which finally applies the step function. The number of nodes per hidden layer and the number of hidden layers is a design choice for JCN Corporation's data scientists to make.

You and your team have analyzed the hypothesis space. Cool beans. The next thing for you to analyze is the loss function of neural networks. Recall that margin-based loss functions multiply the true label y_j by the distance $w^T x_j$ (not by the predicted label $\hat{y}(x_j)$), before applying the step function. The *cross-entropy loss*, the most common loss function used in neural networks, does kind of the same thing. It compares the true label y_j to a soft prediction $\varphi(x_j)$ in the range [0,1] computed in the output node before the step function has been applied to it. The cross-entropy loss function is:

$$L\left(y_j, \varphi(x_j)\right) = -\left(y_j \log\left(\varphi(x_j)\right) + (1-y_j)\log\left(1-\varphi(x_j)\right)\right).$$

Equation 7.5

The form of the expression comes from cross-entropy, the average information in the true label random variable y when described using the predicted distance random variable φ, introduced in Chapter 3. Cross-entropy should be minimized because you want the description in terms of the prediction to be matched to the ground truth. It turns out that the cross-entropy loss is equivalent to the margin-based logistic loss function in binary classification problems, but it is pretty involved to show it mathematically because the margin-based loss function is a function of one variable that multiplies the prediction and the true label, whereas the two arguments are kept separate in cross-entropy loss.[11]

The last question to ask is about regularization. Although ℓ_1-norm, ℓ_2-norm, or other penalties can be added to the cross-entropy loss, the most common way to regularize neural networks is *dropout*. The idea is to randomly remove some nodes from the network on each iteration of an optimization procedure during training. Dropout's goal is somewhat similar to bagging, but instead of creating an ensemble of several neural networks explicitly, dropout makes each iteration appear like a different neural network of an ensemble, which helps diversity and generalization. An example neural network classifier with one hidden layer and ReLU activation functions is shown in Figure 7.15. Repeating the statement from the beginning of this section, the domain of competence for artificial neural networks is semi-structured datasets with a large number of data points.

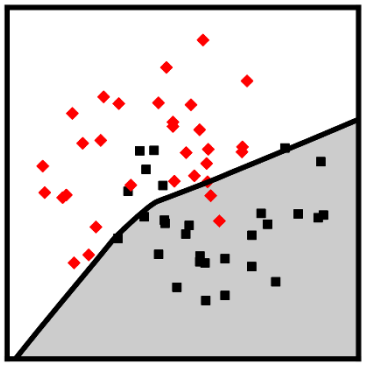


Figure 7.15. *Example neural network classifier.* Accessible caption. Stylized plot showing two classes of data points arranged in a noisy yin yang or interleaving moons configuration. The decision boundary is mostly smooth and composed of two almost straight diagonal segments that form a slightly bent elbow in the middle of the two moons.

7.5.4 Conclusion

You have worked your way through several different kinds of classifiers to compare and contrast their domains of competence and evaluate their appropriateness for your expertise assessment prediction task. Your dataset consists of mostly structured data, is of moderate size, and has a lot of feature axis-aligned separations between employees skilled and unskilled at serverless architecture. For these

[11]Tyler Sypherd, Mario Diaz, Lalitha Sankar, and Peter Kairouz. "A Tunable Loss Function for Binary Classification." In: *Proceedings of the IEEE International Symposium on Information Theory*. Paris, France, Jul. 2019, pp. 2479–2483.

reasons, you can expect that XGBoost will be a competent classifier for your problem. But you should nevertheless do some amount of empirical testing of a few different methods.

7.6 Summary

- There are many different methods for finding decision functions from a finite number of training samples, each with their own inductive biases for how they generalize.
- Different classifiers have different domains of competence: what kinds of datasets they have lower generalization error on than other methods.
- Parametric and non-parametric plug-in methods (discriminant analysis, naïve Bayes, k-nearest neighbor) and risk minimization methods (decision trees and forests, margin-based methods, neural networks) all have a role to play in practical machine learning problems.
- It is important to analyze their inductive biases and domains of competence not only to select the most appropriate method for a given problem, but also to be prepared to extend them for fairness, robustness, explainability, and other elements of trustworthiness.

8

Causal Modeling

In cities throughout the United States, the difficulty of escaping poverty is exacerbated by the difficulty in obtaining social services such as job training, mental health care, financial education classes, legal advice, child care support, and emergency food assistance. They are offered by different agencies in disparate locations with different eligibility requirements. It is difficult for poor individuals to navigate this perplexity and avail themselves of services that they are entitled to. To counteract this situation, the (fictional) integrated social service provider ABC Center takes a holistic approach by housing many individual social services in one place and having a centralized staff of social workers guide their clients. To better advise clients on how to advance themselves in various aspects of life, the center's director and executive staff would like to analyze the data that the center collects on the services used by clients and the life outcomes they achieved. As problem owners, they do not know what sort of data modeling they should do. Imagine that you are a data scientist collaborating with the ABC Center problem owners to analyze the situation and suggest appropriate problem specifications, understand and prepare the data available, and finally perform modeling. (This chapter covers a large part of the machine learning lifecycle whereas other chapters so far have mostly focused on smaller parts.)

Your first instinct may be to suggest that ABC Center take a machine learning approach that predicts life outcomes (education, housing, employment, etc.) from a set of features that includes classes taken and sessions attended. Examining the associations and correlations in the resulting trained model may yield some insights, but misses something very important. Do you know what it is? It's *causality*! If you use a standard machine learning formulation of the problem, you can't say that taking an automobile repair training class *causes* an increase in the wages of the ABC Center client. When you want to understand the effect of *interventions* (specific actions that are undertaken) on outcomes, you have to do more than machine learning, you have to perform causal modeling.[1] Cause and effect are central to understanding the world, but standard supervised learning is not a method for obtaining them.

---

[1]Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. "A Survey of Learning Causality from Data: Problems and Methods." In: *ACM Computing Surveys* 53.4 (Jul. 2020), p. 75.

Toward the goal of suggesting problem formulations to ABC Center, understanding the relevant data, and creating models for them, in this chapter you will:

- distinguish a situation as requiring a causal model or a typical predictive machine learning model,
- discover the graph structure of causal relations among all random variables in a system, and
- compute the quantitative causal effect of an intervention on an outcome, including from observational data.

8.1 Contrasting Causal Modeling and Predictive Modeling

If an ABC Center client completes a one-on-one counseling session, it may *cause* a change in their level of anxiety. In contrast, completing the sessions does not *cause* an increase in, say, the price of eggs even if the price of eggs suddenly jumps the day after every client's counseling session as the price is unrelated to ABC Center. In addition, two different things can cause the same result: both counseling sessions and an increase in wages can *cause* a reduction in anxiety. You can also be fooled by the common cause fallacy: a client secures stable housing and then purchases a used car. The stable housing does not *cause* the car purchase, but both are *caused* by a wage increase.

But what is this elusive notion called causality? It is not the same as correlation, the ability to predict, or even statistical dependence. Remember how we broke down the meanings of *trustworthiness* and *safety* into smaller components (in Chapter 1 and Chapter 3, respectively)? Unfortunately, we cannot do the same for causality since it is an elementary concept that cannot be broken down further. The basic definition of causality is: if *doing* something makes something else happen, then the something we *did* is a *cause* of the something that happened. The key word in the statement is *do*. Causation requires doing. The actions that are done are known as interventions or *treatments*. Interventions can be done by people or by nature; the focus in this chapter is on interventions done consciously by people.

> "While probabilities encode our beliefs about a static world, causality tells us whether and how probabilities change when the world changes, be it by intervention or by act of imagination."
>
> —Judea Pearl, computer scientist at University of California, Los Angeles

8.1.1 Structural Causal Models

A causal model is a quantitative attempt at capturing notions of causality among random variables that builds upon probability theory. *Structural causal models* are one key approach for causal modeling. They contain two parts: a *causal graph* (a graphical model like the Bayesian networks we went over in Chapter 3) and *structural equations*. As shown in Figure 8.1, the graph for both counseling sessions and a change in wages causing a change in anxiety is made up of three nodes arranged in a common effect motif: counseling → anxiety ← wages. The graph for increased wages causing stable housing and car purchase is also made up of three nodes, but arranged in the common cause motif: housing ← wages → car. The graph in the figure puts both subgraphs together along with another common cause: having access to

child care causing both wages and stable housing. (If a client has child care, they can more easily search for jobs and places to live since they don't have to take their child around with them.)

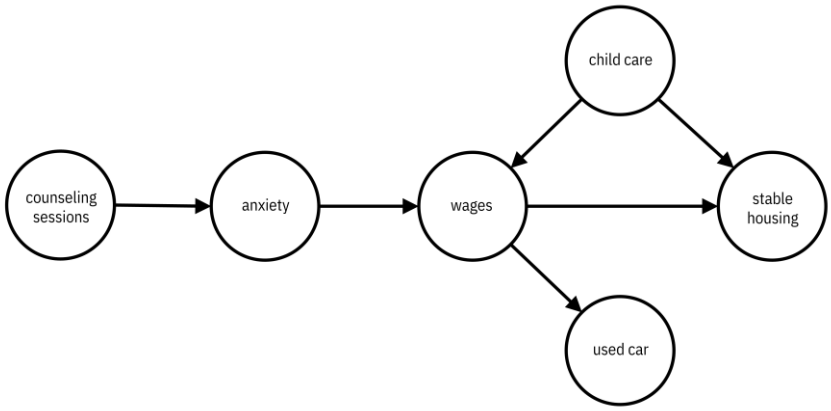


Figure 8.1. *An example causal graph of how the clients of ABC Center respond to interventions and life changes.* Accessible caption. A graph with six nodes: counseling sessions, anxiety, wages, child care, used car, and stable housing. There are edges from counseling sessions to anxiety, anxiety to wages, wages to used car, wages to stable housing, and child care to both wages and stable housing.

Nodes represent random variables in structural causal models just as they do in Bayesian networks. However, edges don't just represent statistical dependencies, they also represent causal relationships. A directed edge from random variable T (e.g. counseling sessions) to Y (e.g. anxiety) indicates a causal effect of T (counseling sessions) on Y (anxiety). Since structural causal models are a generalization of Bayesian networks, the Bayesian network calculations for representing probabilities in factored form and for determining conditional independence through d-separation continue to hold. However, structural causal models capture something more than the statistical relationships because of the structural equations.

Structural equations, also known as *functional models*, tell you what happens when you *do* something. Doing or intervening is the act of forcing a random variable to take a certain value. Importantly, it is not just passively observing what happens to all the random variables when the value of one of them has been revealed—that is simply a conditional probability. Structural causal modeling requires a new operator $do(\cdot)$, which indicates an intervention. The interventional distribution $P\big(Y \mid do(t)\big)$ is the distribution of the random variable Y when the random variable T is forced to take value t. For a causal graph with only two nodes T (counseling session) and Y (anxiety) with a directed edge from T to Y, $T \rightarrow Y$, the structural equation takes the functional form:

$$P\big(Y \mid do(t)\big) = f_Y(t, noise_Y),$$

Equation 8.1

where $noise_Y$ is some noise or randomness in Y and f_Y is any function. There is an exact equation relating an intervention on counseling sessions, like starting counseling sessions (changing the variable from 0 to 1), to the probability of a client's anxiety. The key point is that the probability can truly be expressed as an equation with the treatment as an argument on the right-hand side. Functional models for variables with more parents would have those parents as arguments in the function f_Y, for example $P(Y \mid do(t)) = f_Y(t_1, t_2, t_3, noise_Y)$ if Y has three parents. (Remember from Chapter 3 that directed edges begin at parent nodes and end at child nodes.)

8.1.2 Causal Model vs. Predictive Model

How do you tell that a problem is asking for a causal model rather than a predictive model that would come from standard supervised machine learning? The key is identifying whether something is actively changing one or more of the features. The act of classifying borrowers as good bets for a loan does not change anything about the borrowers at the time, and thus calls for a predictive model (also known as an *associational* model) as used by ThriveGuild in Chapter 3 and Chapter 6. However, wanting to understand if providing job training to a client of ABC Center (actively changing the value of a feature on job preparedness) results in a change to their probability of being approved by ThriveGuild is a causal modeling question.

Using predictive models to form causal conclusions can lead to great harms. Changes to input features of predictive models do not necessarily lead to desired changes of output labels. All hell can break loose if a decision maker is expecting a certain input change to lead to a certain output change, but the output simply does not change or changes in the opposite direction. Because ABC Center wants to model what happens to clients upon receiving social services, you should suggest to the director that the problem specification be focused on causal models to understand the center's set of interventions (various social services) and outcomes that measure how well a client is progressing out of poverty.

An important point is that even if a model is only going to be used for prediction, and not for making decisions to change inputs, causal models help sidestep issues introduced in Chapter 4—construct validity (the data really measures with it should), internal validity (no errors in data processing), and external validity (generalization to other settings)—because it forces the predictions to be based on real, salient phenomena rather than spurious phenomena that just happen to exist in a given dataset. For this reason, causality is an integral component of trustworthy machine learning, and comes up in Part 4 of the book that deals with reliability. Settling for predictive models is a shortcut when just a little more effort to pursue causal models would make a world of difference.

8.1.3 Two Problem Formulations

There are two main problem formulations in causal modeling for ABC Center to consider in the problem specification phase of the development lifecycle. The first is obtaining the structure of the causal graph, which will allow them to understand which services yield effects on which outcomes. The second problem formulation is obtaining a number that quantifies the causal effect between a given treatment variable T (maybe it is completing the automobile repair class) and a given outcome label Y (maybe it is wages). This problem is described further in Section 8.2.

Proceeding to the data understanding and data preparation phases of the lifecycle, there are two types of data, *interventional data* and *observational data*, that may come up in causal modeling. They are detailed in Section 8.3. Very briefly, interventional data comes from a purposefully designed experiment

and observational data does not. Causal modeling with interventional data is usually straightforward and causal modeling with observational data is much more involved.

In the modeling phase when dealing with observational data, the two problem formulations correspond to two different categories of methods. *Causal discovery* is to learn the structural causal model. *Causal inference* is to estimate the causal effect. Specific methods for conducting causal discovery and causal inference from observational data are the topic of Sections 8.4 and 8.5, respectively. A mental model of the modeling methods for the two formulations is given in Figure 8.2.

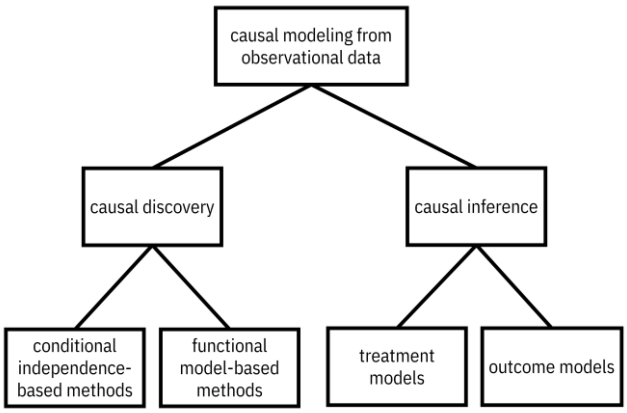


Figure 8.2. *Classes of methods for causal modeling from observational data.* Accessible caption. A hierarchy diagram with causal modeling from observational data at its root with children causal discovery and causal inference. Causal discovery has children conditional independence-based methods and functional model-based methods. Causal inference has children treatment models and outcome models.

8.2 Quantifying a Causal Effect

The second problem specification is computing the *average treatment effect*. For simplicity, let's focus on T being a binary variable taking values in $\{0,1\}$: either a client doesn't get the automobile repair class or they do. Then the average treatment effect τ is:

$$\tau = E[\,Y \mid do(t = 1)\,] - E[\,Y \mid do(t = 0)\,].$$

Equation 8.2

This difference of the expected value of the outcome label under the two values of the intervention precisely shows how the outcome changes due to the treatment. How much do wages change because of the automobile repair class? The terminology contains *average* because of the expected value.

For example, if $Y \mid do(t = 0)$ is a Gaussian random variable with mean 13 dollars per hour and standard deviation 1 dollar per hour,[2] and $Y \mid do(t = 1)$ is a Gaussian random variable with mean 18 dollars per hour and standard deviation 2 dollars per hour, then the average treatment effect is $18 - 13 = 5$ dollars per hour. Being trained in automobile repair increases the earning potential of clients by 5 dollars per hour. The standard deviation doesn't matter here.

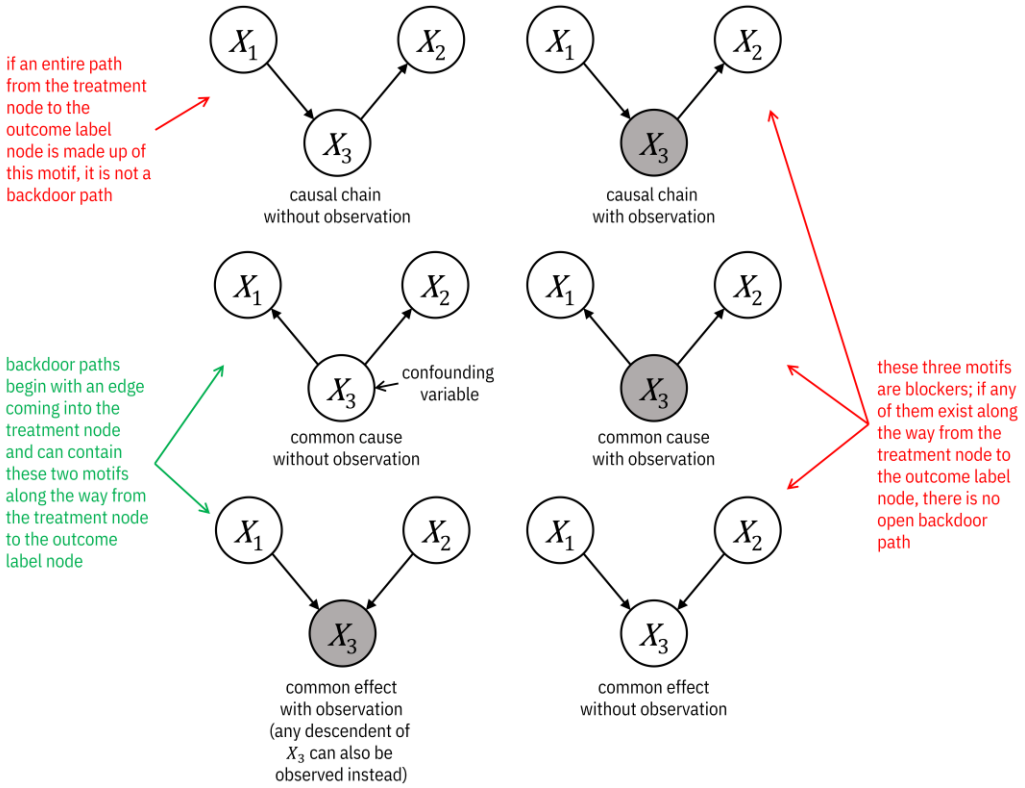


Figure 8.3. *Motifs that block and do not block paths between the treatment node T and the outcome label node Y. Backdoor paths are not blocked.* Accessible caption. If an entire path is made up of causal chains without observation ($X_1 \rightarrow X_3 \rightarrow X_2$), it is not a backdoor path. Backdoor paths begin with an edge coming into the treatment node and can contain common causes without observation ($X_1 \leftarrow X_3 \rightarrow X_2$; X_3 is a confounding variable) and common effects with observation ($X_1 \rightarrow \underline{X_3} \leftarrow X_2$; the underline indicates that X_3 or any of its descendants is observed). In this case, X_3 in the common cause without observation is a confounding variable. The other three motifs—causal chain with observation ($X_1 \rightarrow \underline{X_3} \rightarrow X_2$), common cause with observation ($X_1 \leftarrow \underline{X_3} \rightarrow X_2$), and common effect without observation ($X_1 \rightarrow X_3 \leftarrow X_2$)—are blockers. If any of them exist along the way from the treatment to the outcome label, there is no open backdoor path.

[2]The pdf of a Gaussian random variable X with mean μ and standard deviation σ is $p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$. Its expected value is μ.

8.2.1 Backdoor Paths and Confounders

It is important to note that the definition of the average treatment effect is conditioned on $do(t)$, not on t, and that $P\big(Y \mid do(t)\big)$ and $P(Y \mid t)$ are generally not the same. Specifically, they are not the same when there is a so-called *backdoor path* between T and Y. Remember that a path is a sequence of steps along edges in the graph, irrespective of their direction. A *backdoor path* is any path from T to Y that (1) starts with an edge going into T and (2) is not blocked. The reason for the name 'backdoor' is because the first edge goes backwards into T. (Frontdoor paths have the first edge coming out of T.) Recall from Chapter 3 that a path is blocked if it contains:

1. a causal chain motif with the middle node observed, i.e. the middle node is conditioned upon,
2. a common cause motif with the middle node observed, or
3. a common effect motif with the middle node *not* observed (this is a collider)

anywhere between T and Y. Backdoor paths *can* contain (1) the common cause without observation motif and (2) the common effect with observation motif between T and Y. The motifs that block and do not block a path are illustrated in Figure 8.3.

The lack of equality between the interventional distribution $P\big(Y \mid do(t)\big)$ and the associational distribution $P(Y \mid t)$ is known as *confounding bias*.[3] Any middle nodes of common cause motifs along a backdoor path are *confounding variables* or *confounders*. Confounding is the central challenge to be overcome when you are trying to infer the average treatment effect in situations where intervening is not possible (you cannot $do(t)$). Section 8.5 covers how to mitigate confounding while estimating the average treatment effect.

8.2.2 An Example

Figure 8.4 shows an example of using ABC Center's causal graph (introduced in Figure 8.1) while quantifying a causal effect. The center wants to test whether reducing a client's high anxiety to low anxiety affects their stable housing status. There is a backdoor path from anxiety to stable housing going through wages and child care. The path begins with an arrow going into anxiety. A common cause without observation, wages ← child care → stable housing, is the only other motif along the path to stable housing. It does not block the path. Child care, as the middle node of a common cause, is a confounding variable. If you can do the treatment, that is intervene on anxiety, which is represented diagrammatically with a hammer, the incoming edges to anxiety from counseling sessions and wages are removed. Now there is no backdoor path anymore, and you can proceed with the treatment effect quantification.

Often, however, you cannot do the treatment. These are *observational* rather than *interventional* settings. The observational setting is a completely different scenario than the interventional setting. Figure 8.5 shows how things play out. Since you cannot make the edge between anxiety and wages go away through intervention, you have to include the confounding variable of whether the client has child care or not in your model, and only then will you be able to do a proper causal effect quantification between anxiety and stable housing. Including, observing, or conditioning upon confounding variables

[3]There can be confounding bias without a backdoor path in special cases involving selection bias. Selection bias is when the treatment variable and another variable are common causes for the outcome label.

is known as *adjusting* for them. Adjusting for wages rather than child care is an alternative way to block the backdoor path in the ABC Center graph.

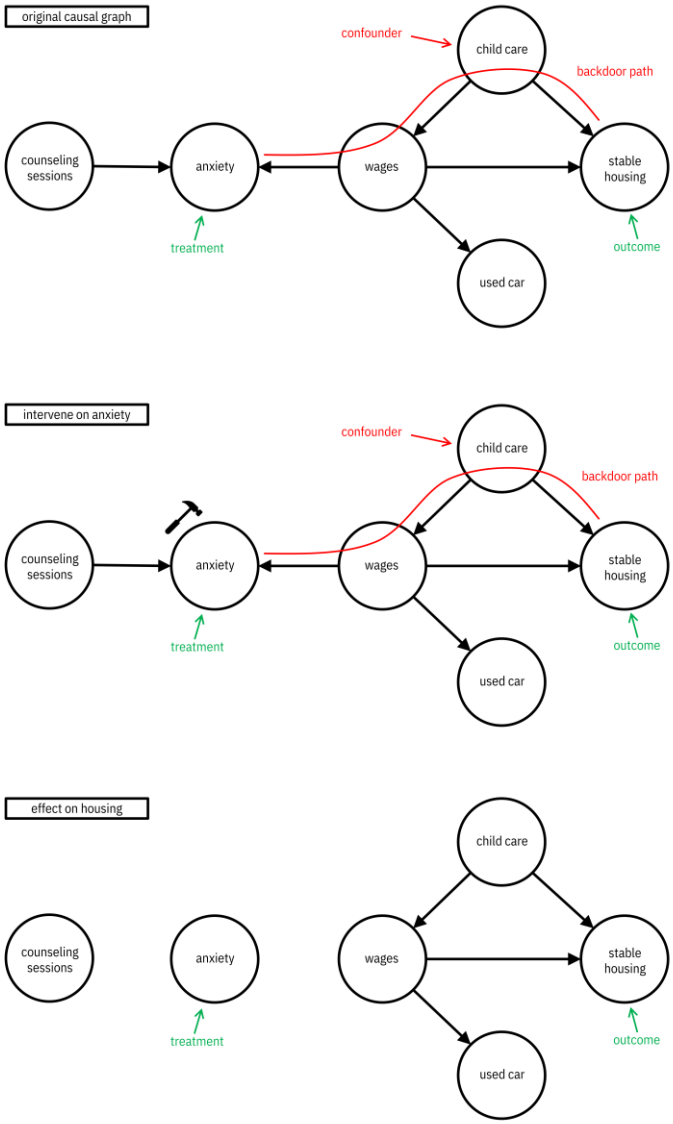


Figure 8.4. *The scenario of causal effect quantification when you can intervene on the treatment.* Accessible caption. The causal graph of Figure 8.1 is marked with anxiety as the treatment and stable housing as the outcome. A backdoor path is drawn between the two passing through wages and child care, which is marked as a confounder. Intervening on anxiety is marked with a hammer. Its effect is the removal of edges into anxiety from counseling sessions and wages, and the removal of the backdoor path.

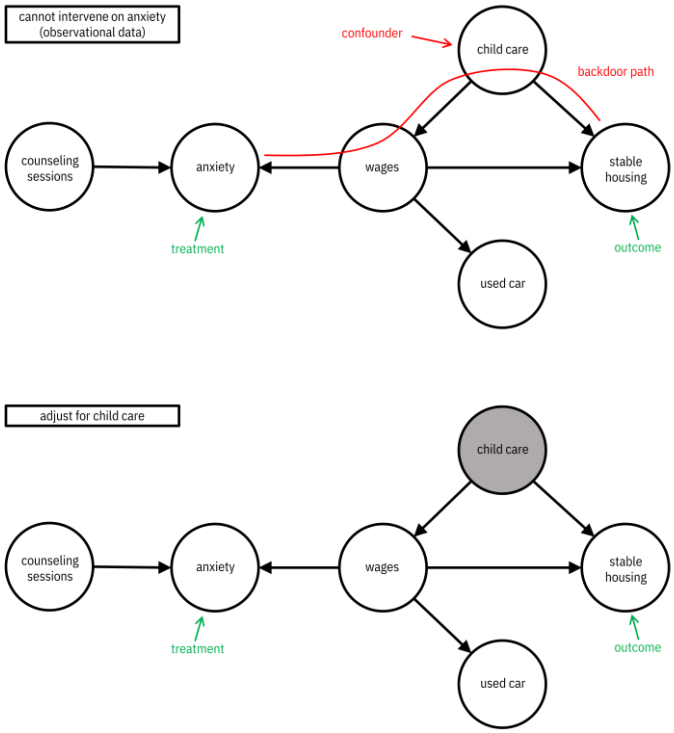


Figure 8.5. *The scenario of causal effect quantification when you cannot intervene on the treatment and thus have to adjust for a variable along a backdoor path.* Accessible caption. The causal graph of Figure 8.1 is marked with anxiety as the treatment and stable housing as the outcome. A backdoor path is drawn between the two passing through wages and child care, which is marked as a confounder. Adjusting for child care colors its node gray and removes the backdoor path.

At the end of the day, the whole point of computing causal effects is to inform decision making. If there are two competing social services that ABC Center can offer a client, causal models should recommend the one with the largest effect on the outcome that they care about. In the next sections, you will proceed to find models for this task from data.

8.3 Interventional Data and Observational Data

You have worked with the director of ABC Center on the problem specification phase and decided on a causal modeling approach rather than a standard machine learning approach for giving insights on interventions for clients to achieve good life outcomes. You have also decided on the specific form of causal modeling: either obtaining the structural causal model or the average treatment effect. The next phases in the machine learning lifecycle are data understanding and data preparation.

There are two types of data in causal modeling: *interventional data* and *observational data*, whose settings you have already been exposed to in the previous section. Interventional data is collected when

you actually do the treatment. It is data collected as part of an experiment that has already been thought out beforehand. An experiment that ABC Center might conduct to obtain interventional data is to enroll one group of clients in a financial education seminar and not enroll another group of clients. The group receiving the treatment of the financial education seminar is the *treatment group* and the group not receiving the seminar is the *control group*. ABC Center would collect data about those clients along many feature dimensions, and this would constitute the dataset to be modeled in the next phase of the lifecycle. It is important to collect data for all features that you think could possibly be confounders.

As already seen in the previous sections and irrespective of whether collected interventionally or observationally, the random variables are: the treatment T that designates the treatment and control groups (anxiety intervention), the outcome label Y (stable housing), and other features X (child care and others). A collection of samples from these random variables constitute the dataset in average treatment effect estimation: $\{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\}$. An example of such a dataset is shown in Figure 8.6. In estimating a structural causal model, you just have random variables X and designate a treatment and outcome label later if needed.

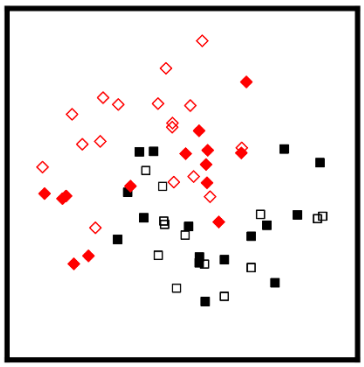


Figure 8.6. *A dataset for treatment effect estimation. The axes are two feature dimensions of x. The unfilled data points are the control group $t = 0$ and the filled data points are the treatment group $t = 1$. The diamond data points have the outcome label $y = 0$ and the square data points have the outcome label $y = 1$.*

The causal graph can be estimated from the entire data if the goal of ABC Center is to get a general understanding of all of the causal relationships. Alternatively, causal effects can be estimated from the treatment, confounding, and outcome label variables if the director wants to test a specific hypothesis such as anxiety reduction seminars having a causal effect on stable housing. A special kind of experiment, known as a *randomized trial*, randomly assigns clients to the treatment group and control group, thereby mitigating confounding within the population being studied.

It is not possible to observe both the outcome label y under the control $t = 0$ and its *counterfactual* y under the treatment $t = 1$ because the same individual cannot both receive and not receive the treatment at the same time. This is known as the *fundamental problem of causal modeling*. The fundamental problem of causal modeling is lessened by randomization because, on average, the treatment group and control group contain matching clients that almost look alike. Randomization does not prevent a lack of external validity however (recall from Chapter 4 that external validity is the ability of a dataset to

generalize to a different population). It is possible that some attribute of all clients in the population is commonly caused by some other variable that is different in other populations.

Randomized trials are considered to be the gold standard in causal modeling that should be done if possible. Randomized trials and interventional data collection more broadly, however, are often prohibited by ethical or logistical reasons. For example, it is not ethical to withhold a treatment known to be beneficial such as a job training class to test a hypothesis. From a logistical perspective, ABC Center may not, for example, have the resources to give half its clients a $1000 cash transfer (similar to Unconditionally's modus operandi in Chapter 4) to test the hypothesis that this intervention improves stable housing. Even without ethical or logistical barriers, it can also be the case that ABC Center's director and executive staff think up a new cause-and-effect relationship they want to investigate after the data has already been collected.

In all of these cases, you are in the setting of observational data rather than interventional data. In observational data, the treatment variable's value has not been forced to a given value; it has just taken whatever value it happened to take, which could be dependent on all sorts of other variables and considerations. In addition, because observational data is often data of convenience that has not been purposefully collected, it might be missing a comprehensive set of possible confounding variables.

The fundamental problem of causal modeling is very apparent in observational data and because of it, testing and validating causal models becomes challenging. The only (unsatisfying) ways to test causal models are (1) through simulated data that can produce both a factual and counterfactual data point, or (2) to collect an interventional dataset from a very similar population in parallel. Regardless, if all you have is observational data, all you can do is work with it in the modeling phase of the lifecycle.

8.4 Causal Discovery Methods

After the data understanding phase comes modeling. How do you obtain a causal graph structure for ABC Center like the one in Figure 8.1? There are three ways to proceed:[4]

1. Enlist subject matter experts from ABC Center to draw out all the arrows (causal relationships) among all the nodes (random variables) manually,
2. Design and conduct experiments to tease out causal relationships, or
3. Discover the graph structure based on observational data.

The first manual option is a good option, but it can lead to the inclusion of human biases and is not scalable to problems with a large number of variables, more than twenty or thirty. The second experimental option is also a good option, but is also not scalable to a large number of variables because interventional experiments would have to be conducted for every possible edge. The third option, known as *causal discovery*, is the most tractable in practice and what you should pursue with ABC Center.[5]

You've probably heard the phrase "those who can, do; those who can't, teach" which is shortened to "those who can't do, teach." In causal modeling from observational data when you can't intervene, the

[4]Clark Glymour, Kun Zhang, and Peter Spirtes. "Review of Causal Discovery Methods Based on Graphical Models." In: *Frontiers in Genetics* 10 (Jun. 2019), p. 524.

[5]There are advanced methods for causal discovery that start with observational data and tell you a few important experiments to conduct to get an even better graph, but they are beyond the scope of the book.

phrase to keep in mind is "those who can't do, assume." Causal discovery has two branches, shown back in Figure 8.2, each with a different assumption that you need to make. The first branch is based on conditional independence testing and relies on the *faithfulness* assumption. The main idea of faithfulness is that the conditional dependence and independence relationships among the random variables encode the causal relationships. There is no coincidental or deterministic relationship among the random variables that masks a causal relationship. The Bayesian network edges are the edges of the structural causal model. Faithfulness is usually true in practice.

One probability distribution can be factored in many ways by choosing different sets of variables to condition on, which leads to different graphs. Arrows pointing in different directions also lead to different graphs. All of these different graphs arising from the same probability distribution are known as a *Markov equivalence class*. One especially informative example of a Markov equivalence class is the setting with just two random variables, say anxiety and wages.[6] The graph with anxiety as the parent and wages as the child and the graph with wages as the parent and anxiety as the child lead to the same probability distribution, but with opposite cause-and-effect relationships. One important point about the conditional independence testing branch of causal discovery methods is that they find *Markov equivalence classes* of graph structures rather than finding single graph structures.

The second branch of causal discovery is based on making assumptions on the form of the structural equations $P\big(Y \mid do(t)\big) = f_Y(t, noise_Y)$ introduced in Equation 8.1. Within this branch, there are several different varieties. For example, some varieties assume that the functional model has a linear function f_Y, others assume that the functional model has a nonlinear function f_Y with additive noise $noise_Y$, and even others assume that the probability distribution of the noise $noise_Y$ has small entropy. Based on the assumed functional form, a best fit to the observational data is made. The assumptions in this branch are much stronger than in conditional independence testing, but lead to single graphs as the solution rather than Markov equivalence classes. These characteristics are summarized in Table 8.1.

Table 8.1. *Characteristics of the two branches of causal discovery methods.*

Branch	Faithfulness Assumption	Assumption on Functional Model	Markov Equivalence Class Output	Single Graph Output
conditional independence	X		X	
functional model		X		X

In the remainder of this section, you'll see an example of each branch of causal discovery in action: the PC algorithm for conditional independence testing-based methods and the additive noise model-based approach for functional model-based methods.

[6]Matthew Ridley, Gautam Rao, Frank Schilbach, and Vikram Patel. "Poverty, Depression, and Anxiety: Causal Evidence and Mechanisms." In: *Science* 370.6522 (Dec. 2020), p. 1289.

8.4.1 An Example Conditional Independence Testing-Based Method

One of the oldest, simplest, and still often-used conditional independence testing-based causal discovery methods is the PC algorithm. Named for its originators, Peter Spirtes and Clark Glymour, the PC algorithm is a greedy algorithm. An ABC Center example of the PC algorithm is presented in Figure 8.7 for the nodes of wages, child care, stable housing, and used car. The steps are as follows:

0. The overall PC algorithm starts with a complete undirected graph with edges between all pairs of nodes.
1. As a first step, the algorithm tests every pair of nodes; if they are independent, it deletes the edge between them. Next it continues to test conditional independence for every pair of nodes conditioning on larger and larger subsets, deleting the edge between the pair of nodes if any conditional independence is found. The end result is the undirected skeleton of the causal graph.

   The reason for this first step is as follows. There is an undirected edge between nodes X_1 and X_2 if and only if X_1 and X_2 are dependent conditioned on every possible subset of all other nodes. (So if a graph has three other nodes X_3, X_4, and X_5, then you're looking for X_1 and X_2 to be (1) unconditionally dependent given no other variables, (2) dependent given X_3, (3) dependent given X_4, (4) dependent given X_5, (5) dependent given X_3, X_4, (6) dependent given X_3, X_5, (7) dependent given X_4, X_5, and (8) dependent given X_3, X_4, X_5.) These conditional dependencies can be figured out using d-separation, which was introduced back in Chapter 3.
2. The second step puts arrowheads on as many edges as it can. The algorithm conducts conditional independence tests between the first and third nodes of three-node chains. If they're dependent conditioned on some set of nodes containing the middle node, then a common cause (collider) motif with arrows is created. The end result is a partially-oriented causal graph. The direction of edges that the algorithm cannot figure out remain unknown. All choices of all of those orientations give you the different graphs that make up the Markov equivalence class.

   The reason for the second step is that an undirected chain of nodes X_1, X_2, and X_3 can be made directed into $X_1 \rightarrow X_2 \leftarrow X_3$ if and only if X_1 and X_3 are dependent conditioned on every possible subset of nodes containing X_2. These conditional dependencies can also be figured out using d-separation.

At the end of the example in Figure 8.7, the Markov equivalence class contains four possible graphs.

In Chapter 3, d-separation was presented in the ideal case when you know the dependence and independence of each pair of random variables perfectly well. But when dealing with data, you don't have that perfect knowledge. The specific computation you do on data to test for conditional independence between random variables is often based on an estimate of the mutual information between them. This seemingly straightforward problem of *conditional independence testing* among continuous random variables has a lot of tricks of the trade that continue to be researched and are beyond the scope of this book.[7]

[7]Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandros G. Dimakis, and Sanjay Shakkottai. "Model-Powered Conditional Independence Test." In: *Advances in Neural Information Processing Systems* 31 (Dec. 2017), pp. 2955–2965.

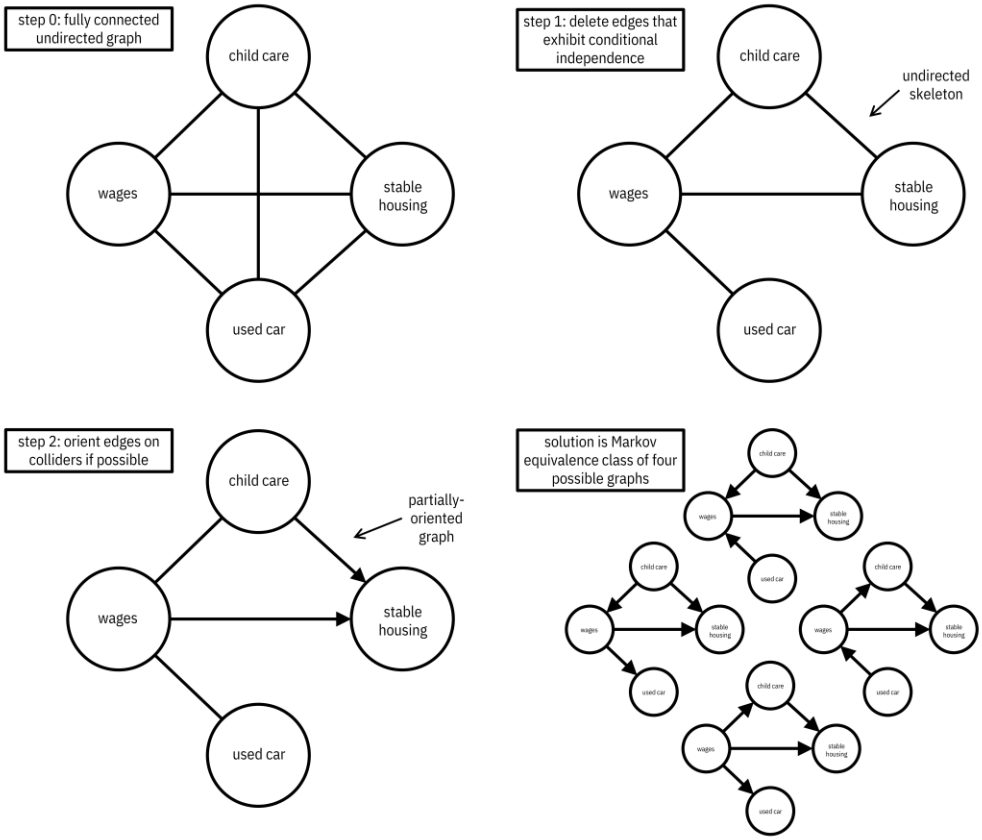


Figure 8.7. *An example of the steps of the PC algorithm.* Accessible caption. In step 0, there is a fully connected undirected graph with the nodes child care, wages, stable housing, and used car. In step 1, the edges between child care and used car, and between stable housing and used car have been removed because they exhibit conditional independence. The undirected skeleton is left. In step 2, the edge between child care and stable housing is oriented to point from child care to stable housing, and the edge between wages and stable housing is oriented to point from wages to stable housing. The edges between child care and wages and between wages and used car remain undirected. This is the partially-oriented graph. There are four possible directed graphs which constitute the Markov equivalence class solution: with edges pointing from child care to wages and used car to wages, with edges pointing from child care to wages and wages to used car, with edges pointing from wages to child care and from used care to wages, and with edges poiting from wages to child care and wages to used car.

8.4.2 An Example Functional Model-Based Method

In the conditional independence test-based methods, no strong assumption is made on the functional form of $P(Y \mid do(t)) = f_Y(t, noise_Y)$. Thus, as you've seen with the PC algorithm, there can remain confusion on the direction of some of the edges. You can't tell which one of two nodes is the cause and which one is the effect. Functional model-based methods do make an assumption on f_Y and are designed to avoid this confusion. They are best understood in the case of just two nodes, say T and Y, or wages and

anxiety. You might think that a change in wages causes a change in anxiety (T causes Y), but it could be the other way around (Y causes T).

One specific method in this functional model-based branch of causal discovery methods is known as the *additive noise model*. It requires that f_Y not be a linear function and that the noise be additive: $P(Y \mid do(t)) = f_Y(t) + noise_Y$; here $noise_Y$ should not depend on t. The plot in Figure 8.8 shows an example nonlinear function along with a quantification of the noise surrounding the function. This noise band is equal in height around the function for all values of t since the noise does not depend on t. Now look at what's going on when t is the vertical axis and y is the horizontal axis. Is the noise band equal in height around the function for all values of y? It isn't, and that's the key observation. The noise has constant height around the function when the cause is the horizontal axis and it doesn't when the effect is the horizontal axis. There are ways to test for this phenomenon from data, but if you understand this idea shown in Figure 8.8, you're golden. If ABC Center wants to figure out whether a decrease in anxiety causes an increase in wages, or if an increase in wages causes a decrease in anxiety, you know what analysis to do.

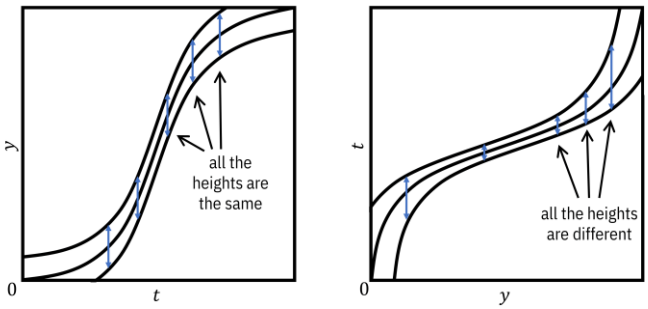


Figure 8.8. *Example of using the additive noise model method to tell cause and effect apart. Since the height of the noise band is the same across all t and different across all y, t is the cause and y is the effect.* Accessible caption. Two plots of the same nonlinear function and noise bands around it. The first plot has y on the vertical axis and t on the horizontal axis; the second has t on the vertical axis and y on the horizontal axis. In the first plot, the height of the noise band is consistently the same for different values of t. In the second plot, the height of the noise band is consistently different for different values of y.

This phenomenon does not happen when the function is linear. Try drawing Figure 8.8 for a linear function in your mind, and then flip it around as a thought experiment. You'll see that the height of the noise is the same both ways and so you cannot tell cause and effect apart.

8.5 Causal Inference Methods

Based on Section 8.4, you have the tools to estimate the structure of the causal relations among random variables collected by ABC Center. But just knowing the relations is not enough for the director. He also wants to quantify the causal effects for a specific treatment and outcome label. Specifically, he wants to know what the effect of anxiety reduction is on stable housing. You now turn to average treatment effect estimation methods to try to answer the question. You are working with observational data because ABC Center has not run a controlled experiment to try to tease out this cause-and-effect relationship. From

Figure 8.5, you know that child care is a confounding variable, and because of proper foresight and not taking shortcuts, you have collected data on it. The t_i values report those clients who received an anxiety reduction treatment, the x_i values report data on clients' child care situation and other possible confounders, and the y_i values are the client's outcome label on stable housing.

Remember our working phrase: "those who can't do, assume." Just like in causal discovery, causal inference from observational data requires assumptions. A basic assumption in causal inference is similar to the independent and identically distributed (i.i.d.) assumption in machine learning, introduced in Chapter 3. This causal inference assumption, the *stable unit treatment value assumption*, simply says that the outcome of one client only depends on the treatment made to that client, and is not affected by treatments to other clients. There are two important assumptions:

1. *No unmeasured confounders* also known as *ignorability*. The dataset needs to contain all the confounding variables within X.
2. *Overlap* also known as *positivity*. The probability of the treatment T given the confounding variables must be neither equal to 0 nor equal to 1. It must take a value strictly greater than 0 and strictly less than 1. This definition explains the name positivity because the probability has to be positive, not identically zero. Another perspective on the assumption is that the probability distribution of X for the treatment group and the probability distribution of X for the control group should overlap; there should not be any support for one of the two distributions where there isn't support for the other. Overlap and the lack thereof is illustrated in Figure 8.9 using a couple of datasets.

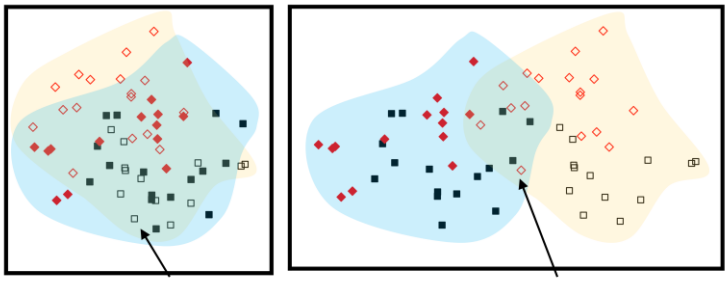


Figure 8.9. *On the left, there is much overlap in the support of the treatment and control groups, so average treatment effect estimation is possible. On the right, there is little overlap, so average treatment effect estimation should not be pursued.* Accessible caption. Plots with data points from the treatment group and control group overlaid with regions indicating the support of their underlying distributions. In the left plot, there is much overlap in the support and in the right plot, there isn't.

Both assumptions together go by the name *strong ignorability*. If the strong ignorability assumptions are not true, you should not conduct average treatment effect estimation from observational data. Why are these two assumptions needed and why are they important? If the data contains all the possible confounding variables, you can adjust for them to get rid of any confounding bias that may exist. If the data exhibits overlap, you can manipulate or balance the data to make it look like the control group and the treatment group are as similar as can be.

If you've just been given a cleaned and prepared version of ABC Center's data to perform average treatment effect estimation on, what are your next steps? There are four tasks for you to do in an iterative manner, illustrated in Figure 8.10. The first task is to specify a causal method, choosing between (1) *treatment models* and (2) *outcome models*. These are the two main branches of conducting causal inference from observational data and were shown back in Figure 8.2. Their details are coming up in the next subsections. The second task in the iterative approach is to specify a machine learning method to be plugged in within the causal method you choose. Both types of causal methods, treatment models and outcome models, are based on machine learning under the hood. The third task is to train the model. The fourth and final task is to evaluate the assumptions to see whether the result can really be viewed as a causal effect.[8] Let's go ahead and run the four tasks for the ABC Center problem.

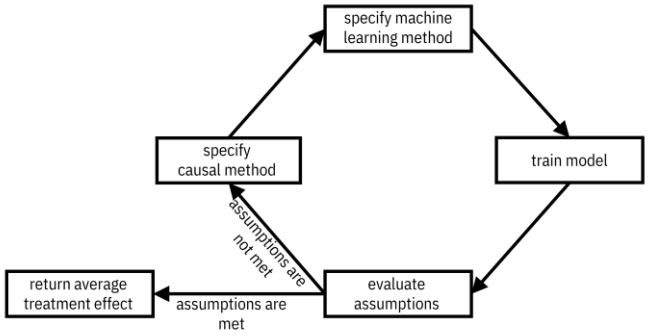


Figure 8.10. *The steps to follow while conducting average treatment effect estimation from observational data. Outside of this loop, you may also go back to the data preparation phase of the machine learning lifecycle if needed.* Accessible caption. A flow diagram starting with specify causal method, leading to specify machine learning method, leading to train model, leading to evaluate assumptions. If assumptions are not met, flow back to specify causal method. If assumptions are met, return average treatment effect.

8.5.1 Treatment Models

The first option for you to consider for your causal method is *treatment models*. Before diving into treatment models, let's define an important concept first: *propensity score*. It is the probability of the treatment (anxiety reduction intervention) conditioned on the possible confounding variables (child care and others), $P(T \mid X)$. Ideally, the decision to give a client an anxiety reduction intervention is independent of anything else, including whether the client has child care. This is true in randomized trials, but tends not to be true in observational data.

The goal of treatment models is to get rid of confounding bias by breaking the dependency between T and X. They do this by assigning weights to the data points so that they better resemble a randomized trial: on average, the clients in the control group and in the treatment group should be similar to each other in their confounding variables. The main idea of the most common method in this branch, *inverse*

[8]Yishai Shimoni, Ehud Karavani, Sivan Ravid, Peter Bak, Tan Hung Ng, Sharon Hensley Alford, Denise Meade, and Ya'ara Goldschmidt. "An Evaluation Toolkit to Guide Model Selection and Cohort Definition in Causal Inference." arXiv:1906.00442, 2019.

probability weighting, is to give more weight to clients in the treatment group that were much more likely to be assigned to the control group and vice versa. Clients given the anxiety reduction treatment $t_j = 1$ are given weight inversely proportional to their propensity score $w_j = 1/P(T = 1 \mid X = x_j)$. Clients not given the treatment $t_j = 0$ are similarly weighted $w_j = 1/P(T = 0 \mid X = x_j)$ which also equals $1/\Big(1 - P(T = 1 \mid X = x_j)\Big)$. The average treatment effect of anxiety reduction on stable housing is then simply the weighted mean difference of the outcome label between the treatment group and the control group. If you define the treatment group as $\mathcal{T} = \{j \mid t_j = 1\}$ and the control group as $\mathcal{C} = \{j \mid t_j = 0\}$, then the average treatment effect estimate is

$$\tau = \frac{1}{\|\mathcal{T}\|}\sum_{j \in \mathcal{T}} w_j\, y_j - \frac{1}{\|\mathcal{C}\|}\sum_{j \in \mathcal{C}} w_j\, y_j.$$

Equation 8.3

Getting the propensity score $P(T \mid X)$ from training data samples $\{(x_1, t_1), \ldots, (x_n, t_n)\}$ is a machine learning task with features x_j and labels t_j in which you want a (calibrated) continuous score as output (the score was called $s(x)$ in Chapter 6). The learning task can be done with any of the machine learning algorithms from Chapter 7. The domains of competence of the different choices of machine learning algorithms for estimating the propensity score are the same as for any other machine learning task, e.g. decision forests for structured datasets and neural networks for large semi-structured datasets.

Once you've trained a propensity score model, the next step is to evaluate it to see whether it meets the assumptions for causal inference. (Just because you can compute an average treatment effect doesn't mean that everything is hunky-dory and that your answer is actually the causal effect.) There are four main evaluations of a propensity score model: (1) covariate balancing, (2) calibration, (3) overlap of propensity distribution, and (4) area under the receiver operating characteristic (AUC). Calibration and AUC were introduced in Chapter 6 as ways to evaluate typical machine learning problems, but covariate balancing and overlap of propensity distribution are new here. Importantly, the use of AUC to evaluate propensity score models is different than its use to evaluate typical machine learning problems.

Since the goal of inverse probability weighting is to make the potential confounding variables X look alike in the treatment and control groups, the first evaluation, *covariate balancing*, tests whether that has been accomplished. This is done by computing the standardized mean difference (SMD) going one-by-one through the X features (child care and other possible confounders). Just subtract the mean value of the feature for the control group data from the mean value for the treatment group data, and divide by the square root of the average variance of the feature for the treatment and control groups. 'Standardized' refers to the division at the end, which is done so that you don't have to worry about the absolute scale of different features. An absolute value of SMD greater than about 0.1 for any feature should be a source of concern. If you see this happening, your propensity score model is not good and you shouldn't draw causal conclusions from the average treatment effect.

The second evaluation is calibration. Since the propensity score model is used as an actual probability in inverse probability weighting, it has to have good calibration to be effective. Introduced in Chapter 6, the calibration loss needs to be small and the calibration curve needs to be as much of a straight line as possible. If they aren't, you shouldn't draw causal conclusions from the average treatment effect you compute and need to go back to step 1.

The third evaluation is based on the distributions of the propensity score for the treatment group and the control group, illustrated in Figure 8.11. Spikes in the distribution near 0 or 1 are bad because they indicate a possible large set of X values that can be almost perfectly classified by the propensity score model. Perfect classification means that there is almost no overlap of the treatment group and control group in that region, which is not desired to meet the positivity assumption. If you see such spikes, you should not proceed with this model. (This evaluation doesn't tell you what the non-overlap region is, but just that it exists.)

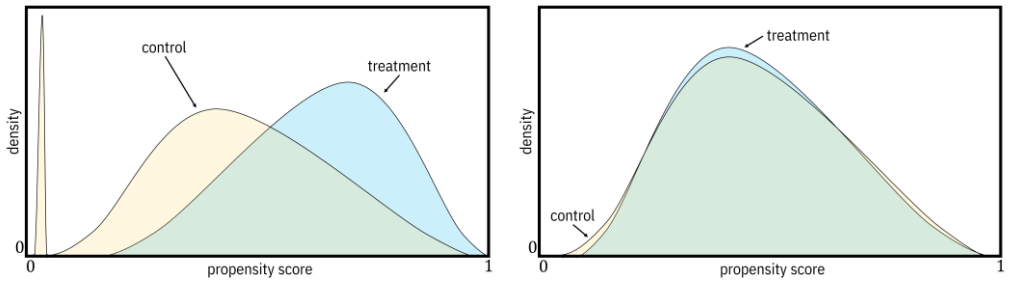


Figure 8.11. *Example propensity score distributions; the one on the left indicates a possible overlap violation, whereas the one on the right does not.* Accessible caption. Plots with density on the vertical axis and propensity score on the horizontal axis. Each plot overlays a control pdf and a treatment pdf. The pdfs in the left plot do not overlap much and the control group distribution has a spike near 0. The pdfs in the right plot are almost completely on top of each other.

The fourth evaluation of a treatment model is the AUC. Although its definition and computation are the same as in Chapter 6, good values of the AUC of a propensity score model are not near the perfect 1.0. Intermediate values of the AUC, like 0.7 or 0.8, are just right for average treatment effect estimation. A poor AUC of nearly 0.5 remains bad for a propensity score model. If the AUC is too high or too low, do not proceed with this model. Once you've done all the diagnostic evaluations and none of them raise an alert, you should proceed with reporting the average treatment effect that you have computed as an actual causal insight. Otherwise, you have to go back and specify different causal methods and/or machine learning methods.

8.5.2 Outcome Models

The other branch of methods you should consider for the causal model in testing whether ABC Center's anxiety reduction intervention has an effect on stable housing is *outcome models*. In treatment models, you use the data $\{(t_1, x_1), \ldots, (t_n, x_n)\}$ to learn the $P(T \mid X = x)$ relationship, but you do something different in outcome models. In this branch, you use the data $\{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\}$ to learn the relationships $E[Y \mid T = 1, X = x]$ and $E[Y \mid T = 0, X = x]$. You're directly predicting the average outcome label of stable housing from the possible confounding variables such as child care along with the anxiety reduction treatment variable. Strong ignorability is required in both treatment models and outcome models. Before moving on to the details of outcome models, the difference between treatment models and outcome models is summarized in Table 8.2.

Table 8.2. *Characteristics of the two branches of causal inference methods.*

Branch	Dataset	What Is Learned	Purpose
treatment models	$\{(t_1, x_1), \ldots, (t_n, x_n)\}$	$P(T \mid X)$	use to weight the data points
outcome models	$\{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\}$	$E[Y \mid T, X]$	use directly for average treatment estimation

Why is learning $E[Y \mid T, X]$ models from data useful? How do you get the average treatment effect of anxiety reduction on stable housing from them? Remember that the definition of the average treatment effect is $\tau = E[Y \mid do(t = 1)] - E[Y \mid do(t = 0)]$. Also, remember that when there is no confounding, the associational distribution and interventional distribution are equal, so $E[Y \mid do(t)] = E[Y \mid T = t]$. Once you have $E[Y \mid T = t, X]$, you can use something known as the law of iterated expectations to adjust for X and get $E[Y \mid T = t]$. The trick is to take an expectation over X because $E_X[E_Y[Y \mid T = t, X]] = E_Y[Y \mid T = t]$. (The subscripts on the expectations tell you which random variable you're taking the expectation with respect to.) To take the expectation over X, you sum the outcome model over all the values of X weighted by the probabilities of each of those values of X. It is clear sailing after that to get the average treatment effect because you can compute the difference $E[Y \mid T = 1] - E[Y \mid T = 0]$ directly.

You have the causal model; now on to the machine learning model. When the outcome label Y takes binary values 0 and 1 corresponding the absence and presence of stable housing, then the expected values are equivalent to the probabilities $P(Y \mid T = 1, X = x)$ and $P(Y \mid T = 0, X = x)$. Learning these probabilities is a job for a calibrated machine learning classifier with continuous score output trained on labels y_i and features (t_j, x_j). You can use any machine learning method from Chapter 7 with the same guidelines for domains of competence. Traditionally, it has been common practice to use linear margin-based methods for the classifier, but nonlinear methods should be tried especially for high-dimensional data with lots of possible confounding variables.

Just like with treatment models, being able to compute an average treatment effect using outcome models does not automatically mean that your result is a causal inference. You still have to evaluate. A first evaluation, which is also an evaluation for treatment models, is calibration. You want small calibration loss and a straight line calibration curve. A second evaluation for outcome models is accuracy, for example measured using AUC. With outcome models, just like with regular machine learning models but different from treatment models, you want the AUC to be as large as possible approaching 1.0. If the AUC is too small, do not proceed with this model and go back to step 1 in the iterative approach to average treatment effect estimation illustrated in Figure 8.10.

A third evaluation for outcome models examines the predictions they produce to evaluate ignorability or no unmeasured confounders. The predicted $Y \mid t = 1$ and $Y \mid t = 0$ values coming out of the outcome models should be similar for clients who were actually part of the treatment group (received the anxiety reduction intervention) and clients who were part of the control group (did not receive the anxiety reduction intervention). If the predictions are not similar, there is still some confounding left over after adjusting for X (child care and other variables), which means that the assumption of no unmeasured confounders is violated. Thus, if the predicted $Y \mid t = 1$ and $Y \mid t = 0$ values for the two groups do not mostly overlap, then do not proceed and go back to the choice of causal model and machine learning model.

8.5.3 Conclusion

You've evaluated two options for causal inference: treatment models and outcome models. Which option is better in what circumstances? Treatment and outcome modeling are inherently different problems with different features and labels. You can just end up having better evaluations for one causal method than the other using the different machine learning method options available. So just try both branches and see how well the results correspond. Some will be better matched to the relevant modeling tasks, depending on the domains of competence of the machine learning methods under the hood.

But do you know what? You're in luck and don't have to choose between the two branches of causal inference. There's a hybrid approach called *doubly-robust* estimation in which the propensity score values are added as an additional feature in the outcome model.[9] Doubly-robust models give you the best of both worlds! ABC Center's director is waiting to decide whether he should invest more in anxiety reduction interventions. Once you're done with your causal modeling analysis, he'll be able to make an informed decision.

8.6 Summary

- Causality is a fundamental concept that expresses how changing one thing (the cause) results in another thing changing (the effect). It is different than correlation, predictability, and dependence.
- Causal models are critical to inform decisions involving interventions and treatments with expected effects on outcomes. Predictive associational models are not sufficient when you are 'doing' something to an input.
- In addition to informing decisions, causal modeling is a way to avoid harmful spurious relationships in predictive models.
- Structural causal models extend Bayesian networks by encoding causal relationships in addition to statistical relationships. Their graph structure allows you to understand what causes what, as well as chains of causation, among many variables. Learning their graph structure is known as causal discovery.
- Causal inference between a hypothesized pair of treatment and outcome is a different problem specification. To validly conduct causal inference from observational data, you must control for confounding.
- Causal modeling requires assumptions that are difficult to validate, but there is a set of evaluations you should perform as part of modeling to do the best that you can.

[9]Miguel A. Hernán and James M. Robins. *Causal Inference: What If*. Boca Raton, Florida, USA: Chapman & Hall/CRC, 2020.

9

Distribution Shift

Wavetel is a leading (fictional) mobile telephony provider in India that expanded its operations to several East and Central African countries in recent years. One of its profit centers in the African markets is credit enabled by mobile money that it runs through partnerships with banks in each of the nations. The most straightforward application of *mobile money* is savings, first started in Kenya in 2007 under the name M-Pesa. With mobile money savings, customers can deposit, withdraw, and transfer funds electronically without a formal bank account, all through their mobile phones. (Remember that these transactions are one of the data sources that Unconditionally evaluated in Chapter 4.) More advanced financial services such as credit and insurance later emerged. In these advanced services, the bank takes on financial risk and can't just hand out accounts without an application process and some amount of due diligence.

Having seen how profitable mobile money-enabled credit can be, Wavetel strongly lobbied for it to be allowed in its home country of India and has just seen the regulations signed into law. Partnering with the (fictional) Bank of Bulandshahr, Wavetel is ready to deploy this new service under the name Phulo. Based on market research, Wavetel and the Bank of Bulandshahr expect Phulo to receive tens of thousands of applications per day when first launched. They have to be ready to approve or deny those applications in near real-time. To deal with this load, imagine that they have hired your data science team as consultants to create a machine learning model that makes the decisions.

The task you face, approving and denying mobile phone-enabled loans for unbanked customers in India has never been done before. The Bank of Bulandshahr's historical loan approval data will not be useful for making decisions on Phulo applicants. However, Wavetel has privacy-preserved data from mobile money-enabled credit systems in several East and Central African countries that it has the rights and consent to use in its India operations. Can you train the Phulo machine learning model using the African datasets? What could go wrong?

If you're not careful, there's a lot that could go wrong. You could end up creating a really harmful and unreliable system, because of *the big lie of machine learning*: the core assumption that training data and testing data is independent and identically distributed (i.i.d.). This is almost never true in the real world,

where there tends to be some sort of difference in the probability distributions of the training data and the data encountered during the model's deployment. This difference in distributions is known as *distribution shift*. A competent model that achieves high accuracy when tested through cross-validation might not maintain that competence in the real world. Too much epistemic uncertainty sinks the ship of even a highly risk-minimizing model.

> "All bets are off if there is a distribution shift when the model is deployed. (There's always a distribution shift.)"
>
> —Arvind Narayanan, computer scientist at Princeton University

This chapter begins Part 4 of the book on reliability and dealing with epistemic uncertainty, which constitutes the second of four attributes of trustworthiness (the others are basic performance, human interaction, and aligned purpose) as well as the second of two attributes of safety (the first is minimizing risk and aleatoric uncertainty). As shown in Figure 9.1, you're halfway home to creating trustworthy machine learning systems!

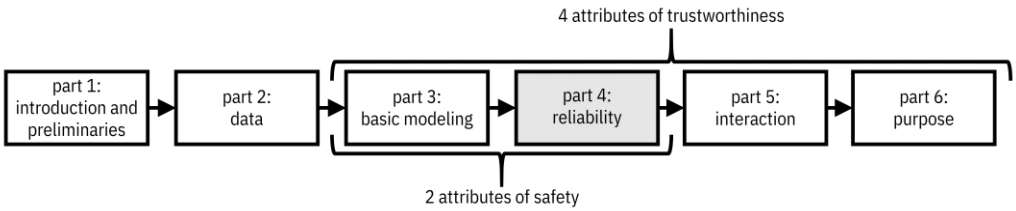


Figure 9.1. *Organization of the book. This fourth part focuses on the second attribute of trustworthiness, reliability, which maps to machine learning models that are robust to epistemic uncertainty.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 4 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

In this chapter, while working through the modeling phase of the machine learning lifecycle to create a safe and reliable Phulo model, you will:

- examine how epistemic uncertainty leads to poor machine learning models, both with and without distribution shift,
- judge which kind of distribution shift you have, and
- mitigate the effects of distribution shift in your model.

9.1 Epistemic Uncertainty in Machine Learning

You have the mobile money-enabled credit approval/denial datasets from East and Central African countries and want to train a reliable Phulo model for India. You know that for safety, you must worry about minimizing epistemic uncertainty: the possibility of unexpected harms. Chapter 3 taught you how

to differentiate aleatoric uncertainty from epistemic uncertainty. Now's the time to apply what you learned and figure out where epistemic uncertainty is rearing its head!

First, in Figure 9.2, let's expand on the picture of the different biases and validities from Figure 4.3 to add a modeling step that takes you from the prepared data space to a prediction space where the output predictions of the model live. As you learned in Chapter 7, in modeling, you're trying to get the classifier to generalize from the training data to the entire set of features by using an inductive bias, without overfitting or underfitting. Working backwards from the prediction space, the modeling process is the first place where epistemic uncertainty creeps in. Specifically, if you don't have the information to select a good inductive bias and hypothesis space, but you could obtain it in principle, then you have epistemic uncertainty.[1] Moreover, if you don't have enough high-quality data to train the classifier even if you have the perfect hypothesis space, you have epistemic uncertainty.

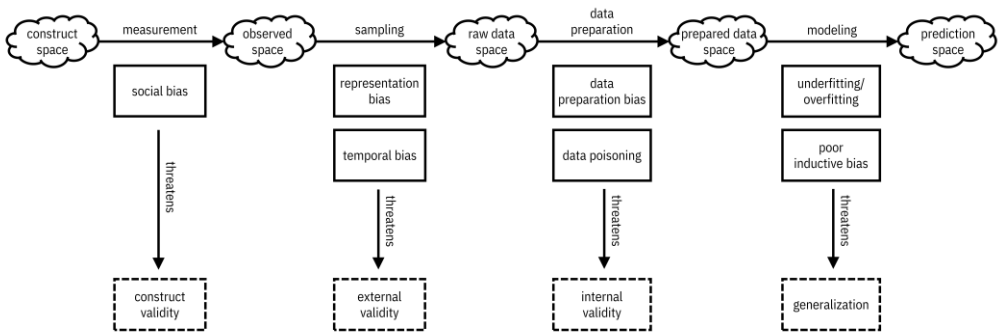


Figure 9.2. *Different spaces and what can go wrong due to epistemic uncertainty throughout the machine learning pipeline.* Accessible caption. A sequence of five spaces, each represented as a cloud. The construct space leads to the observed space via the measurement process. The observed space leads to the raw data space via the sampling process. The raw data space leads to the prepared data space via the data preparation process. The prepared data space leads to the prediction space via the modeling process. The measurement process contains social bias, which threatens construct validity. The sampling process contains representation bias and temporal bias, which threatens external validity. The data preparation process contains data preparation bias and data poisoning, which threaten internal validity. The modeling process contains underfitting/overfitting and poor inductive bias, which threaten generalization.

The epistemic uncertainty in the model has a few different names, including the *Rashomon effect*[2] and *underspecification*.[3] The main idea, illustrated in Figure 9.3, is that a lot of models perform similarly well

[1]Eyke Hüllermeier and Willem Waegeman. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods." In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506.
[2]Aaron Fisher, Cynthia Rudin, and Francesca Dominici. "All Models Are Wrong, but Many Are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneosly." In: *Journal of Machine Learning Research* 20.177 (Dec. 2019). Rashomon is the title of a film in which different witnesses give different descriptions of the same event.
[3]Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan

in terms of aleatoric uncertainty and risk, but have different ways of generalizing because you have not minimized epistemic uncertainty. They all have the *possibility* of being competent and reliable models. They have a possibility value of 1. (Other models that do not perform well have a possibility value of 0 of being competent and reliable models.) However, many of these possible models are unreliable and they take shortcuts by generalizing from spurious characteristics in the data that people would not naturally think are relevant features to generalize from.[4] They are not causal. Suppose one of the African mobile money-enabled credit datasets just happens to have a spurious feature like the application being submitted on a Tuesday that predicts the credit approval label very well. In that case, a machine learning training algorithm will not know any better and will use it as a shortcut to fit the model. And you know that taking shortcuts is no-no for you when building trustworthy machine learning systems, so you don't want to let your models take shortcuts either.

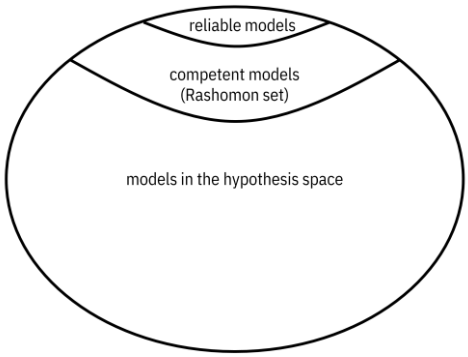


Figure 9.3. *Among all the models you are considering, many of them can perform well in terms of accuracy and related measures; they are competent and constitute the Rashomon set. However, due to underspecification and the epistemic uncertainty that is present, many of the competent models are not safe and reliable.* Accessible caption. A nested set diagram with reliable models being a small subset of competent models (Rashomon set), which are in turn a small subset of models in the hypothesis space.

How can you know that a competent high-accuracy model is one of the reliable, safe ones and not one of the unreliable, unsafe ones? The main way is to stress test it by feeding in data points that are edge cases beyond the support of the training data distribution. More detail about how to test machine learning systems in this way is covered in Chapter 13.

The main way to reduce epistemic uncertainty in the modeling step that goes from the prepared data space to the prediction space is *data augmentation*. If you can, you should collect more data from more environments and conditions, but that is probably not possible in your Phulo prediction task.

Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Nataragan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Mertin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. "Underspecification Presents Challenges for Credibility in Modern Machine Learning." arXiv:2011.03395, 2020.

[4]Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. "Shortcut Learning in Deep Neural Networks." In: *Nature Machine Intelligence* 2.11 (Nov. 2020), pp. 665–673.

Alternatively, you can augment your training dataset by synthetically generating training data points, especially those that push beyond the margins of the data distribution you have. Data augmentation can be done for semi-structured data modalities by flipping, rotating, and otherwise transforming the data points you have. In structured data modalities, like have with Phulo, you can create new data points by changing categorical values and adding noise to continuous values. However, be careful that you do not introduce any new biases in the data augmentation process.

9.2 Distribution Shift is a Form of Epistemic Uncertainty

So far, you've considered the epistemic uncertainty in going from the prepared data space to the prediction space by modeling, but there's epistemic uncertainty earlier in the pipeline too. This epistemic uncertainty comes from all of the biases you don't know about or don't understand very well when going from the construct space to the prepared data space. We'll come back to some of them in Chapter 10 and Chapter 11 when you learn how to make machine learning systems fair and adversarially robust. But for now, when building the Phulo model in India using data from several East and Central African countries, your most obvious source of epistemic uncertainty is sampling. You have sampled your data from a different time (temporal bias) and population (representation bias) than what Phulo is going to be deployed on. You might also have epistemic uncertainty in measurement: it is possible that the way creditworthiness shows up in India is different than in East and Central African countries because of cultural differences (social bias). Putting everything together, the distributions of the training data and the deployed data are not identical: $p_{X,Y}^{(train)}(x, y) \neq p_{X,Y}^{(deploy)}(x, y)$. (Remember that X is features and Y is labels.) You have *distribution shift*.

9.2.1 The Different Types of Distribution Shift

In building out the Phulo model, you know that the way you've measured and sampled the data in historical African contexts is mismatched with the situation in which Phulo will be deployed: present-day India. What are the different ways in which the training data distribution $p_{X,Y}^{(train)}(x, y)$ can be different from the deployment data distribution $p_{X,Y}^{(deploy)}(x, y)$? There are three main ways.

1. *Prior probability shift*, also known as *label shift*, is when the label distributions are different but the features given the labels are the same: $p_Y^{(train)}(y) \neq p_Y^{(deploy)}(y)$[5] and $p_{X|Y}^{(train)}(x \mid y) = p_{X|Y}^{(deploy)}(x \mid y)$.
2. *Covariate shift* is the opposite, when the feature distributions are different but the labels given the features are the same: $p_X^{(train)}(x) \neq p_X^{(deploy)}(x)$ and $p_{Y|X}^{(train)}(y \mid x) = p_{Y|X}^{(deploy)}(y \mid x)$.
3. *Concept drift* is when the labels given the features are different but the features are the same: $p_{Y|X}^{(train)}(y \mid x) \neq p_{Y|X}^{(deploy)}(y \mid x)$ and $p_X^{(train)}(x) = p_X^{(deploy)}(x)$, or when the features given the labels are different but the labels are the same: $p_{X|Y}^{(train)}(x \mid y) \neq p_{X|Y}^{(deploy)}(x \mid y)$ and $p_Y^{(train)}(y) = p_Y^{(deploy)}(y)$.

All other distribution shifts do not have special names like these three types. The first two types of distribution shift come from sampling differences whereas the third type of distribution shift comes

[5]You can also write this as $p_0^{(train)} \neq p_0^{(deploy)}$.

from measurement differences. The three different types of distribution shift are summarized in Table 9.1.

Table 9.1. *The three types of distribution shift.*

Type	What Changes	What is the Same	Source	Threatens	Learning Problem
prior probability shift	Y	$X \mid Y$	sampling	external validity	anticausal learning
covariate shift	X	$Y \mid X$	sampling	external validity	causal learning
concept drift	$Y \mid X$	X	measurement	construct validity	causal learning
	$X \mid Y$	Y			anticausal learning

Let's go through an example of each type to see which one (or more than one) affects the Phulo situation. There will be prior probability shift if there are different proportions of creditworthy people in present-day India and historical African countries, maybe because of differences in the overall economy. There will be covariate shift if the distribution of features is different. For example, maybe people in India have more assets in gold than in East and Central African countries. There will be concept drift if the actual mechanism connecting the features and creditworthiness is different. For example, people in India who talk or SMS with many people may be more creditworthy while in East and Central Africa, people who talk or SMS with few people may be more creditworthy.

One way to describe the different types of distribution shifts is through the context or environment. What does changing the environment in which the data was measured and sampled *do* to the features and label? And if you're talking about doing, you're talking about causality. If you treat the environment as a random variable E, then the different types of distribution shifts have the causal graphs shown in Figure 9.4.[6]

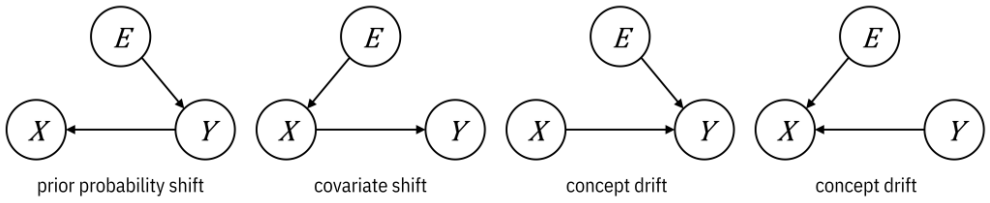


Figure 9.4. *Causal graph representations of the different types of distribution shift.* Accessible caption. Graphs of prior probability shift ($E \rightarrow Y \rightarrow X$), covariate shift ($E \rightarrow X \rightarrow Y$), concept drift ($E \rightarrow Y \leftarrow X$), and concept drift ($E \rightarrow X \leftarrow Y$).

[6]Meelis Kull and Peter Flach. "Patterns of Dataset Shift." In: *Proceedings of the International Workshop on Learning over Multiple Contexts*. Nancy, France, Sep. 2014.

These graphs illustrate a nuanced point. When you have prior probability shift, the label causes the feature and when you have covariate shift, the features cause the label. This is weird to think about, so let's slow down and work through this concept. In the first case, $Y \rightarrow X$, the label is known as an *intrinsic* label and the machine learning problem is known as *anticausal learning*. A prototypical example is a disease with a known pathogen like malaria that causes specific symptoms like chills, fatigue, and fever. The label of a patient having a disease is intrinsic because it is a basic property of the infected patient, which then causes the observed features. In the second case, $X \rightarrow Y$, the label is known as an *extrinsic* label and the machine learning problem is known as *causal learning*. A prototypical example of this case is a syndrome, a collection of symptoms such as Asperger's that isn't tied to a pathogen. The label is just a label to describe the symptoms like compulsive behavior and poor coordination; it doesn't cause the symptoms. The two different versions of concept drift correspond to anticausal and causal learning, respectively. Normally, in the practice of doing supervised machine learning, the distinction between anticausal and causal learning is just a curiosity, but it becomes important when figuring out what to do to mitigate the effect of distribution shift. It is not obvious which situation you're in with the Phulo model, and you'll have to really think about it.

9.2.2 Detecting Distribution Shift

Given that your training data is from East and Central African countries and your deployment data will come from India, you are aware that distribution shift probably exists in your modeling task. But how do you definitively detect it? There are two main ways:

1. *data distribution-based shift detection*, and
2. *classifier performance-based shift detection*,

that are applicable at two different points of the machine learning modeling pipeline, show in Figure 9.5.[7] Data distribution-based shift detection is done on the training data before the model training. Classifier performance-based shift detection is done afterwards on the model. Data distribution-based shift detection, as its name implies, directly compares $p_{X,Y}^{(train)}(x, y)$ and $p_{X,Y}^{(deploy)}(x, y)$ to see if they are similar or different. A common way is to compute their K-L divergence, which was introduced in Chapter 3. If it is too high, then there is distribution shift. Classifier performance-based shift detection examines the Bayes risk, accuracy, F_1-score, or other model performance measure. If it is much poorer than the performance during cross-validation, there is distribution shift.

That is all well and good, but did you notice something about the two methods of distribution shift detection that make them unusable for your Phulo development task? They require the deployed distribution: both its features and labels. But you don't have it! If you did, you would have used it to train the Phulo model. Shift detection methods are really meant for monitoring scenarios in which you keep getting data points to classify over time and you keep getting ground truth labels soon thereafter.

[7] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. "Learning Under Concept Drift: A Review." In: *IEEE Transactions on Knowledge and Data Engineering* 31.12 (Dec. 2019), pp. 2346–2363.

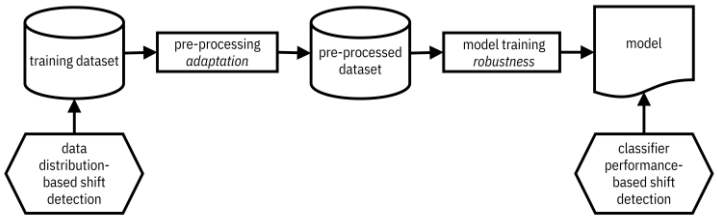


Figure 9.5. *Modeling pipeline for detecting and mitigating distribution shift.* Accessible caption. A block diagram with a training dataset as input to a pre-processing block labeled adaptation with a pre-processed dataset as output. The pre-processed dataset is input to a model training block labeled robustness with a model as output. A data distribution-based shift detection block is applied to the training dataset. A classifier performance-based shift detection block is applied to the model.

If you've started to collect unlabeled feature data from India, you can do *unsupervised* data distribution-based shift detection by comparing the India feature distribution to the Africa feature distributions. But you kind of already know that they'll be different, and this unsupervised approach will not permit you to determine which type of distribution shift you have. Thus, in a Phulo-like scenario, you just have to assume the existence and type of distribution shift based on your knowledge of the problem. (Remember our phrase from Chapter 8: "those who can't do, assume.")

9.2.3 Mitigating Distribution Shift

There are two different scenarios when trying to overcome distribution shift to create a safe and trustworthy machine learning model: (1) you have unlabeled feature data $X^{(deploy)}$ from the deployment distribution, which will let you *adapt* your model to the deployment environment, and (2) you have absolutely no data from the deployment distribution, so you must make the model *robust* to any deployment distribution you might end up encountering.

> "Machine learning systems need to robustly model the range of situations that occur in the real-world."
>
> —Drago Anguelov, computer scientist at Waymo

The two approaches take place in different parts of the modeling pipeline shown in Figure 9.5. Adaptation is done on the training data as a pre-processing step, whereas robustness is introduced as part of the model training process. The two kinds of mitigation are summarized in Table 9.2.

Table 9.2. *The two types of distribution shift mitigation.*

Type	Where in the Pipeline	Known Deployment Environment	Approach for Prior Probability and Covariate Shifts	Approach for Concept Drift
adaptation	pre-processing	yes	sample weights	obtain labels
robustness	model training	no	min-max formulation	invariant risk minimization

The next sections work through adaptation and robustness for the different types of distribution shift. Mitigating prior probability shift and covariate shift is easier than mitigating concept drift because the relationship between the features and labels does not change in the first two types. Thus, the classifier you learned on the historical African training data continues to capture that relationship even on India deployment data; it just needs a little bit of tuning.

9.3 Adaptation

The first mitigation approach, adaptation, is done as a pre-processing of the training data from East and Central African countries using information available in unlabeled feature data $X^{(deploy)}$ from India. To perform adaptation, you must know that India is where you'll be deploying the model and you must be able to gather some features.

9.3.1 Prior Probability Shift

Since prior probability shift arises from sampling bias, the relationship between features and labels, and thus the ROC, does not change. Adapting is simply a matter of adjusting the classifier threshold or operating point based on the confusion matrix, which are all concepts you learned in Chapter 6. A straightforward and effective way to adapt to prior probability shift is a weighting scheme. The algorithm is as follows.[8]

1. Train a classifier on one random split of the training data to get $\hat{y}^{(train)}(x)$ and compute the classifier's confusion matrix on another random split of the training data: $C = \begin{bmatrix} p_{TP} & p_{FP} \\ p_{FN} & p_{TN} \end{bmatrix}$.
2. Run the unlabeled features of the deployment data through the classifier: $\hat{y}^{(train)}\left(X^{(deploy)}\right)$ and compute the probabilities of positives and negatives in the deployment data as a vector: $a = \begin{bmatrix} P\left(\hat{y}^{(train)}\left(X^{(deploy)}\right) = 1\right) \\ P\left(\hat{y}^{(train)}\left(X^{(deploy)}\right) = 0\right) \end{bmatrix}$.
3. Compute weights $w = C^{-1}a$. This is a vector of length two.
4. Apply the weights to the training data points in the first random split and retrain the classifier. The first of the two weights multiplies the loss function of the training data points with label 1. The second of the two weights multiplies the loss function of the training data points with label 0.

The retrained classifier is what you want to use when you deploy Phulo in India under the assumption of prior probability shift.

[8]Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. "Detecting and Correcting for Label Shift with Black Box Predictors." In: *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 3122–3130.

9.3.2 Covariate Shift

Just like adapting to prior probability shift, adapting to covariate shift uses a weighting technique called *importance weighting* to overcome the sampling bias. In an empirical risk minimization or structural risk minimization setup, a weight w_j multiplies the loss function for data point (x_j, y_j):

$$\hat{y}(\cdot) = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} w_j L\left(y_j, f\left(x_j\right)\right).$$

Equation 9.1

(Compare this to Equation 7.3, which is the same thing, but without the weights.) The importance weight is the ratio of the probability density of the features x_j under the deployment distribution and the training distribution: $w_j = p_X^{(deploy)}(x_j)/p_X^{(train)}(x_j)$.[9] The weighting scheme tries to make the African features look more like the Indian features by emphasizing those that are less likely in East and Central African countries but more likely in India.

How do you compute the weight from the training and deployment datasets? You could first try to estimate the two pdfs separately and then evaluate them at each training data point and plug them into the ratio. But that usually doesn't work well. The better way to go is to directly estimate the weight.

> "When solving a problem of interest, do not solve a more general problem as an intermediate step."
>
> —Vladimir Vapnik, computer scientist at AT&T Bell Labs

The most straightforward technique is similar to computing propensity scores in Chapter 8. You learn a classifier with a calibrated continuous score $s(x)$ such as logistic regression or any other classifier from Chapter 7. The dataset to train this classifier is a concatenation of the deployment and training datasets. The labels are 1 for the data points that come from the deployment dataset and the labels are 0 for the data points that come from the training dataset. The features are the features. Once you have the continuous output of the classifier as a score, the importance weight is:[10]

$$w_j = \frac{n^{(train)} s\left(x_j\right)}{n^{(deploy)}\left(1 - s\left(x_j\right)\right)},$$

Equation 9.2

where $n^{(train)}$ and $n^{(deploy)}$ are the number of data points in the training and deployment datasets, respectively.

---

[9]Hidetoshi Shimodaira. "Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function." In: *Journal of Statistical Planning and Inference* 90.2 (Oct. 2000), pp. 227–244.

[10]Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning.* Cambridge, England, UK: Cambridge University Press, 2012.

9.3.3 Concept Drift

Adapting to prior probability shift and covariate shift can be done without labels from the deployment data because they come from sampling bias in which the relationship between features and labels does not change. However, the same thing is not possible under concept drift because it comes from measurement bias: the relationship between features and labels changing from African countries to India. To adapt to concept drift, you must be very judicious in selecting unlabeled deployment data points to (expensively) get labels for. It could mean having a human expert in India look at some applications and provide judgement on whether to approve or deny the mobile money-enabled credit. There are various criteria for choosing points for such annotation. Once you've gotten the annotations, retrain the classifier on the newly labeled data from India (possibly with large weight) along with the older training data from East and Central African countries (possibly with small weight).

9.4 Robustness

Often, you do not have any data from the deployment environment, so adaptation is out of the question. You might not even know what the deployment environment is going to be. It might not even end up being India for all that you know. Robustness does not require you to have any deployment data. It modifies the learning objective and procedure.

9.4.1 Prior Probability Shift

If you don't have any data from the deployment environment (India), you want to make your model robust to whatever prior probabilities of creditworthiness there could be. Consider a situation in which the deployment prior probabilities in India are actually $p_0^{(deploy)}$ and $p_1^{(deploy)} = \left(1 - p_0^{(deploy)}\right)$, but you guess that they are $p_0^{(train)}$ and $\left(1 - p_0^{(train)}\right)$, maybe by looking at the prior probabilities in the different African countries. Your guess is probably a little bit off. If you use the decision function corresponding to your guess with the likelihood ratio test threshold $\frac{p_0^{(train)} c_{10}}{\left(1-p_0^{(train)}\right) c_{01}}$ (recall this was the form of the optimal threshold stated in Chapter 6), it has the following mismatched Bayes risk performance:[11]

$$R\left(p_0^{(deploy)}, p_0^{(train)}\right) = c_{10} p_0^{(deploy)} p_{FP}\left(p_0^{(train)}\right) + c_{01} p_1^{(deploy)} p_{FN}\left(p_0^{(train)}\right).$$

Equation 9.3

You lose out on performance. Your epistemic uncertainty in knowing the right prior probabilities has hurt the Bayes risk.

To be robust to the uncertain prior probabilities in present-day India, choose a value for $p_0^{(train)}$ so that the worst-case performance is as good as possible. Known as a min-max formulation, the problem is to find a min-max optimal prior probability point that you're going to use when you deploy the Phulo model. Specifically, you want:

[11] In Equation 6.10, $R = c_{10} p_0 p_{FP} + c_{01} p_1 p_{FN}$, the dependence of p_{FP} and p_{FN} on p_0 was not explicitly noted, but this dependence exists through the Bayes optimal threshold.

$$\arg \min_{p_0^{(train)}} \max_{p_0^{(deploy)}} R\left(p_0^{(deploy)}, p_0^{(train)}\right).$$

Equation 9.4

Normally in the book, we stop at the posing of the formulation. In this instance, however, since the min-max optimal solution has nice geometric properties, let's carry on. The mismatched Bayes risk function $R\left(p_0^{(deploy)}, p_0^{(train)}\right)$ is a linear function of $p_0^{(deploy)}$ for a fixed value of $p_0^{(train)}$. When $p_0^{(train)} = p_0^{(deploy)}$, the Bayes optimal threshold is recovered and $R\left(p_0^{(deploy)}, p_0^{(deploy)}\right)$ is the optimal Bayes risk R defined in Chapter 6. It is a concave function that is zero at the endpoints of the interval [0,1].[12] The linear mismatched Bayes risk function is tangent to the optimal Bayes risk function at $p_0^{(train)} = p_0^{(deploy)}$ and greater than it everywhere else.[13] This relationship is shown in Figure 9.6.

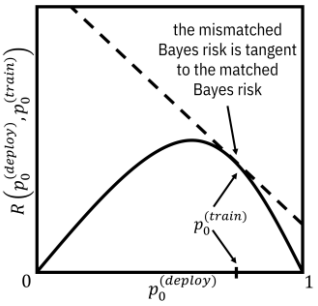


Figure 9.6. *An example mismatched (dashed line) and matched Bayes risk function (solid curve).* Accessible caption. A plot with $R\left(p_0^{(deploy)}, p_0^{(train)}\right)$ on the vertical axis and $p_0^{(deploy)}$ on the horizontal axis. The matched Bayes risk is 0 at $p_0^{(deploy)} = 0$, increases to a peak in the middle and decreases back to 0 at $p_0^{(deploy)} = 1$. Its shape is concave. The mismatched Bayes risk is a line tangent to the matched Bayes risk at the point $p_0^{(deploy)} = p_0^{(train)}$, which in this example is at a point greater than the peak of the matched Bayes risk. There's a large gap between the matched and mismatched Bayes risk, especially towards $p_0^{(deploy)} = 0$.

The solution is the prior probability value at which the matched Bayes risk function has zero slope. It turns out that the correct answer is the place where the mismatched Bayes risk tangent line is flat—at the top of the hump as shown in Figure 9.7. Once you have it, use it in the threshold of the Phulo decision function to deal with prior probability shift.

[12]This is true under the ongoing assumption that the costs of correct classifications $c_{00} = 0$ and $c_{11} = 0$.

[13]Kush R. Varshney. "Bayes Risk Error is a Bregman Divergence." In: *IEEE Transactions on Signal Processing* 59.9 (Sep. 2011), pp. 4470–4472.

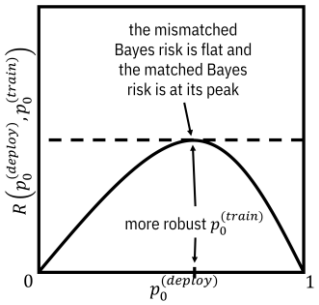


Figure 9.7. *The most robust prior probability to use in the decision function place is where the matched Bayes risk function is maximum.* Accessible caption. A plot with $R\left(p_0^{(deploy)}, p_0^{(train)}\right)$ on the vertical axis and $p_0^{(deploy)}$ on the horizontal axis. The matched Bayes risk is 0 at $p_0^{(deploy)} = 0$, increases to a peak in the middle and decreases back to 0 at $p_0^{(deploy)} = 1$. Its shape is concave. The mismatched Bayes risk is a horizontal line tangent to the matched Bayes risk at the optimal point $p_0^{(deploy)} = p_0^{(train)}$, which is at the peak of the matched Bayes risk. The maximum gap between the matched and mismatched Bayes risk is as small as can be.

9.4.2 Covariate Shift

If you are in the covariate shift setting instead of being in the prior probability shift setting, you have to do something a little differently to make your model robust to the deployment environment. Here too, robustness means setting up a min-max optimization problem so that you do the best that you can in preventing the worst-case behavior. Starting from the importance weight formulation of Equation 9.1, put in an extra maximum loss over the weights:[14]

$$\hat{y}(\cdot) = \arg\min_{f \in \mathcal{F}} \max_{w} \frac{1}{n} \sum_{j=1}^{n} w_j L\left(y_j, f\left(x_j\right)\right),$$

Equation 9.5

where w is the set of possible weights that are non-negative and sum to one. The classifier that optimizes the objective in Equation 9.5 is the robust classifier you'll want to use for the Phulo model to deal with covariate shift.

9.4.3 Concept Drift and Other Distribution Shifts

Just like adapting to concept drift is harder than adapting to prior probability shift and covariate shift because it stems from measurement bias instead of sampling bias, robustness to concept drift is also harder. It is one of the most vexing problems in machine learning. You can't really pose a min-max

[14]Junfeng Wen, Chun-Nam Yu, and Russell Greiner. "Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification." In: *Proceedings of the International Conference on Machine Learning*. Beijing, China, Jun. 2014, pp. 631–639. Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. "Does Distributionally Robust Supervised Learning Give Robust Classifiers?" In: *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 2029–2037.

formulation because the training data from East and Central African countries is not indicating the right relationship between the features and label in India. A model robust to concept drift must extrapolate outside of what the training data can tell you. And that is too open-ended of a task to do well unless you make some more assumptions.

One reasonable assumption you can make is that the set of features split into two types: (1) *causal* or *stable* features, and (2) *spurious* features. You don't know which ones are which beforehand. The causal features capture the intrinsic parts of the relationship between features and labels, and are the same set of features in different environments. In other words, this set of features is *invariant* across the environments. Spurious features might be predictive in one environment or a few environments, but not universally so across environments. You want the Phulo model to rely on the causal features whose predictive relationship with labels holds across Tanzania, Rwanda, Congo, and all the other countries that Wavetel has data from and ignore the spurious features. By doing so, the hope is that the model will not only perform well for the countries in the training set, but also any new country or environment that it encounters, such as India. It will be robust to the environment in which it is deployed.

> "ML enables an increased emphasis on stability and robustness."
>
> —Susan Athey, economist at Stanford University

Invariant risk minimization is a variation on the standard risk minimization formulation of machine learning that helps the model focus on the causal features and avoid the spurious features when there is data from more than one environment available for training. The formulation is:[15]

$$\hat{y}(\cdot) = \arg\min_{f \in \mathcal{F}} \sum_{e \in \mathcal{E}} \frac{1}{n_e} \sum_{j=1}^{n_e} L\left(y_j^{(e)}, f\left(x_j^{(e)}\right)\right).$$

$$\text{such that } f \in \arg\min_{g \in \mathcal{F}} \frac{1}{n_e} \sum_{j=1}^{n_e} L\left(y_j^{(e)}, g\left(x_j^{(e)}\right)\right) \text{ for all } e \in \mathcal{E}.$$

Equation 9.6

Let's break this equation down bit by bit to understand it more. First, the set $\mathcal{E}$ is the set of all environments or countries from which we have training data (Tanzania, Rwanda, Congo, etc.) and each country is indexed by e. There are n_e training samples $\{\left(x_1^{(e)}, y_1^{(e)}\right), \ldots, \left(x_{n_e}^{(e)}, y_{n_e}^{(e)}\right)\}$ from each country. The inner summation in the top line is the regular risk expression that we've seen before in Chapter 7. The outer summation in the top line is just adding up all the risks for all the environments, so that the classifier minimizes the total risk. The interesting part is the constraint in the second line. It is saying that the classifier that is the solution of the top line must simultaneously minimize the risk for each of the environments or countries separately as well. As you know from earlier in the chapter, there can be

[15]Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. "Invariant Risk Minimization." arXiv:1907.02893, 2020.

many different classifiers that minimize the loss—they are the Rashomon set—and that is why the second line has the 'element of' symbol ∈. The invariant risk minimization formulation adds extra specification to reduce the epistemic uncertainty and allow for better out-of-distribution generalization.

You might ask yourself whether the constraint in the second line does anything useful. Shouldn't the first line alone give you the same solution? This is a question that machine learning researchers are currently struggling with.[16] They find that usually, the standard risk minimization formulation of machine learning from Chapter 7 is the most robust to general distribution shifts, without the extra invariant risk minimization constraints. However, when the problem is an anticausal learning problem and the feature distributions across environments have similar support, invariant risk minimization may outperform standard machine learning (remember from earlier in the chapter that the label causes the features in anticausal learning).[17]

In a mobile money-enabled credit approval setting like you have with Phulo, it is not entirely clear whether the problem is causal learning or anticausal learning: do the features cause the label or do the labels cause the features? In a traditional credit scoring problem, you are probably in the causal setting because there are strict parameters on features like salary and assets that cause a person to be viewed by a bank as creditworthy or not. In the mobile money and unbanked setting, you could also imagine the problem to be anticausal if you think that a person is inherently creditworthy or not, and the features you're able to collect from their mobile phone usage are a result of the creditworthiness. As you're developing the Phulo model, you should give invariant risk minimization a shot because you have datasets from several countries, require robustness to concept drift and generalization to new countries, and likely have an anticausal learning problem. You and your data science team can be happy that you've given Wavetel and the Bank of Bulandshahr a model they can rely on during the launch of Phulo.

9.5 Summary

- Machine learning models should not take shortcuts if they are to be reliable. You must minimize epistemic uncertainty in modeling, data preparation, sampling, and measurement.
- Data augmentation is a way to reduce epistemic uncertainty in modeling.
- Distribution shift—the mismatch between the probability distribution of the training data and the data you will see during deployment—has three special cases: prior probability shift, covariate shift, and concept drift. Often, you can't detect distribution shift. You just have to assume it.
- Prior probability shift and covariate shift are easier to overcome than concept drift because they arise from sampling bias rather than measurement bias.
- A pre-processing strategy for mitigating prior probability shift and covariate shift is adaptation, in which sample weights multiply the training loss during the model learning process. Finding

[16]Ishaan Gulrajani and David Lopez-Paz. "In Search of Lost Domain Generalization." In: *Proceedings of the International Conference on Learning Representations*. May 2021. Pritish Kamath, Akilesh Tangella, Danica J. Sutherland, and Nathan Srebro. "Does Invariant Risk Minimization Capture Invariance?" arXiv:2010.01134, 2021.

[17]Kartik Ahuja, Jun Wang, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. "Empirical or Invariant Risk Minimization? A Sample Complexity Perspective." In: *Proceedings of the International Conference on Learning Representations*. May 2021.

the weights requires a fixed target deployment distribution and unlabeled data from it.

- A strategy for mitigating prior probability and covariate shift during model training is min-max robustness, which changes the learning formulation to try to do the best in the worst-case environment that could be encountered during deployment.
- Adapting to concept drift requires the acquisition of some labeled data from the deployment environment.
- Invariant risk minimization is a strategy for mitigating concept drift and achieving distributional robustness that focuses the model's efforts on causal features and ignores spurious features. It may work well in anticausal learning scenarios in which the label causes the features.

10

Fairness

Sospital is a leading (fictional) health insurance company in the United States. Imagine that you are the lead data scientist collaborating with a problem owner in charge of transforming the company's *care management* programs. Care management is the set of services that help patients with chronic or complex conditions manage their health and have better clinical outcomes. Extra care management is administered by a dedicated team composed of physicians, other clinicians, and caregivers who come up with and execute a coordinated plan that emphasizes preventative health actions. The problem owner at Sospital has made a lot of progress in implementing software-based solutions for the care coordination piece and has changed the culture to support them, but is still struggling with the patient intake process. The main struggle is in identifying the members of health plans that need extra care management. This is a mostly manual process right now that the problem owner would like to automate.

You begin the machine learning lifecycle through an initial set of conversations with the problem owner and determine that it is not an exploitative use case that could immediately be an instrument of oppression. It is also a problem in which machine learning may be helpful. You next consult a paid panel of diverse voices that includes actual patients. You learn from them that black Americans have not been served well by the health care system historically and have a deep-seated mistrust of it. Therefore, you should ensure that the machine learning model does not propagate systematic disadvantage to the black community. The system should be *fair* and not contain unwanted biases.

Your task now is to develop a detailed problem specification for a fair machine learning system for allocating care management programs to Sospital members and proceed along the different phases of the machine learning lifecycle without taking shortcuts. In this chapter, you will:

- compare and contrast definitions of fairness in a machine learning context,
- select an appropriate notion of fairness for your task, and
- mitigate unwanted biases at various points in the modeling pipeline to achieve fairer systems.

10.1 The Different Definitions of Fairness

The topic of this chapter, algorithmic fairness, is the most contested topic in the book because it is intertwined with social justice and cannot be reduced to technical-only conceptions. Because of this broader conception of fairness, it may seem odd to you that this chapter is in a part of the book that also contains technical robustness. The reason for including it this way is due to the technical similarities with robustness which you, as a data scientist, can make use of and which are rarely recognized in other literature. This choice was not made to minimize the social importance of algorithmic fairness.

Fairness and justice are almost synonymous, and are political. There are several kinds of justice, including (1) *distributive justice*, (2) *procedural justice*, (3) *restorative justice*, and (4) *retributive justice*.

- Distributive justice is equality in what people receive—the outcomes.
- Procedural justice is sameness in the way it is decided what people receive.
- Restorative justice repairs a harm.
- Retributive justice seeks to punish wrongdoers.

All of the different forms of justice have important roles in society and sociotechnical systems. In the problem specification phase of a model that determines who receives Sospital's care management and who doesn't, you need to focus on distributive justice. This focus on distributive justice is generally true in designing machine learning systems because machine learning itself is focused on outcomes. The other kinds of justice are important in setting the context in which machine learning is and is not used. They are essential in promoting accountability and *holistically* tamping down racism, sexism, classism, ageism, ableism, and other unwanted discriminatory behaviors.

> "Don't conflate CS/AI/tech ethics and social justice issues. They're definitely related, but not interchangeable."
>
> —Brandeis Marshall, computer scientist at Spelman College

Why would different individuals and groups receive an unequal allocation of care management? Since it is a limited resource, not everyone can receive it.[1] The more chronically ill that patients are, the more likely they should be to receive care management. This sort of discrimination is generally acceptable, and is the sort of task machine learning systems are suited for. It becomes unacceptable and unfair when the allocation gives a *systematic* advantage to certain *privileged* groups and individuals and a systematic disadvantage to certain *unprivileged* groups and individuals. Privileged groups and individuals are defined to be those who have historically been more likely to receive the *favorable label* in a machine learning binary classification task. Receiving care management is a favorable label because patients are given extra services to keep them healthy. Other favorable labels include being hired, not being fired, being approved for a loan, not being arrested, and being granted bail. Privilege is a result of power imbalances, and the same groups may not be privileged in all contexts, even within the same society. In some narrow societal contexts, it may even be the elite who are without power.

[1] You can argue that this way of thinking is flawed and society should be doing whatever it takes so that care management is not a limited resource, but it is the reality today.

Privileged and unprivileged groups are delineated by *protected attributes* such as race, ethnicity, gender, religion, and age. There is no one universal set of protected attributes. They are determined from laws, regulations, or other policies governing a particular application domain in a particular jurisdiction. As a health insurer in the United States, Sospital is regulated under Section 1557 of the Patient Protection and Affordable Care Act with the specific protected attributes of race, color, national origin, sex, age, and disability. In health care in the United States, non-Hispanic whites are usually a privileged group due to multifaceted reasons of power. For ease of explanation and conciseness, the remainder of the chapter uses whites as the privileged group and blacks as the unprivileged group.

There are two main types of fairness you need to be concerned about: (1) *group fairness* and (2) *individual fairness*. Group fairness is the idea that the average classifier behavior should be the same across groups defined by protected attributes. Individual fairness is the idea that individuals similar in their features should receive similar model predictions. Individual fairness includes the special case of two individuals who are exactly the same in every respect except for the value of one protected attribute (this special case is known as *counterfactual fairness*). Given the regulations Sospital is operating under, group fairness is the more important notion to include in the care management problem specification, but you should not forget to consider individual fairness in your problem specification.

10.2 Where Does Unfairness Come From?

Unfairness in the narrow scope of allocation decisions (distributive justice) has a few different sources. The most obvious source of unfairness is unwanted bias, specifically social bias in the measurement process (going from the construct space to the observed space) and representation bias in the sampling process (going from the observed space to the raw data space) that you learned about in Chapter 4, shown in Figure 10.1. (This is a repetition of Figure 9.2 and an extension of Figure 4.3 where the concepts of construct space and observed space were first introduced.)

In the data understanding phase, you have figured out that you will use privacy-preserved historical medical claims from Sospital members along with their past selection for care management as the data source. Medical claims data is generated any time a patient sees a doctor, undergoes a procedure, or fills a pharmacy order. It is structured data that includes diagnosis codes, procedure codes, and drug codes, all standardized using the ICD-10, CPT, and NDC schemes, respectively.[2] It also includes the dollar amount billed and paid along with the date of service. It is administrative data used by the healthcare provider to get reimbursed by Sospital.

> "If humans didn't behave the way we do there would be no behavior data to correct. The training data is society."
>
> — M. C. Hammer, musician and technology consultant

[2]See https://www.cms.gov/files/document/blueprint-codes-code-systems-value-sets.pdf for details about these coding schemes.

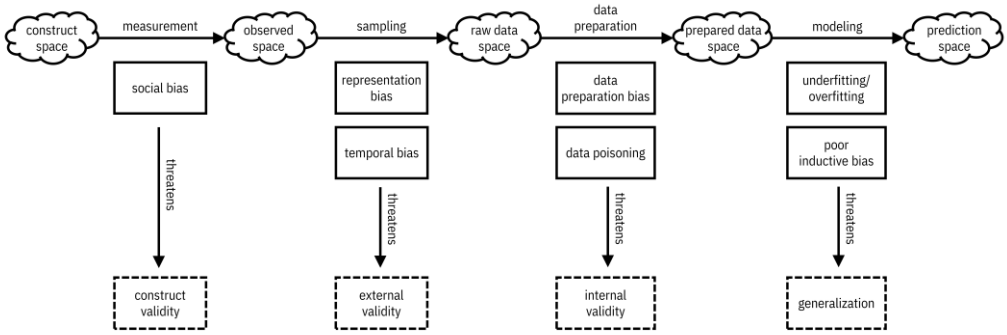


Figure 10.1. *Bias in measurement and sampling are the most obvious sources of unfairness in machine learning, but not the only ones.* Accessible caption. A sequence of five spaces, each represented as a cloud. The construct space leads to the observed space via the measurement process. The observed space leads to the raw data space via the sampling process. The raw data space leads to the prepared data space via the data preparation process. The prepared data space leads to the prediction space via the modeling process. The measurement process contains social bias, which threatens construct validity. The sampling process contains representation bias and temporal bias, which threatens external validity. The data preparation process contains data preparation bias and data poisoning, which threaten internal validity. The modeling process contains underfitting/overfitting and poor inductive bias, which threaten generalization.

Social bias enters claims data in a few ways. First, you might think that patients who visit doctors a lot and get many prescriptions filled, i.e. utilize the health care system a lot, are sicker and thus more appropriate candidates for care management. While it is directionally true that greater health care utilization implies a sicker patient, it is not true when comparing patients across populations such as whites and blacks. Blacks tend to be sicker for an equal level of utilization due to structural issues in the health care system.[3] The same is true when looking at health care cost instead of utilization. Another social bias can be in the codes. For example, black people are less-often treated for pain than white people in the United States due to false beliefs among clinicians that black people feel less pain.[4] Moreover, there can be social bias in the human-determined labels of selection for care management in the past due to implicit cognitive biases or prejudice on the part of the decision maker. Representation bias enters claims data because it is only from Sospital's own members. This population may, for example, undersample blacks if Sospital offers its commercial plans primarily in counties with larger white populations.

Besides the social and representation biases given above that are already present in raw data, you need to be careful that you don't introduce other forms of unfairness in the problem specification and data preparation phases. For example, suppose you don't have the labels from human decision makers in the past. In that case, you might decide to use a threshold on utilization or cost as a proxy outcome

[3]Moninder Singh and Karthikeyan Natesan Ramamurthy. "Understanding Racial Bias in Health Using the Medical Expenditure Panel Survey Data." In: *Proceedings of the NeurIPS Workshop on Fair ML for Health*. Vancouver, Canada, Dec. 2019.
[4]Oluwafunmilayo Akinlade. "Taking Black Pain Seriously." In: *New England Journal of Medicine* 383.e68 (Sep. 2020).

variable, but that would make blacks less likely to be selected for care management at equal levels of infirmity for the reasons described above. Also, as part of feature engineering, you might think to combine individual cost or utilization events into more comprehensive categories, but if you aren't careful you could make racial bias worse. It turns out that combining all kinds of health system utilization into a single feature yields unwanted racial bias, but keeping inpatient hospital nights and frequent emergency room utilization as separate kinds of utilization keeps the bias down in nationally-representative data.[5]

> "As AI is embedded into our day to day lives it's critical that we ensure our models don't inadvertently incorporate latent stereotypes and prejudices."
>
> —Richard Zemel, computer scientist at University of Toronto

You might be thinking that you already know how to measure and mitigate biases in measurement, sampling, and data preparation from Chapter 9, distribution shift. What's different about fairness? Although there is plenty to share between distribution shift and fairness,[6] there are two main technical differences between the two topics. First is access to the construct space. You can get data from the construct space in distribution shift scenarios. Maybe not immediately, but if you wait, collect, and label data from the deployment environment, you will have data reflecting the construct space. However, you never have access to the construct space in fairness settings. The construct space reflects a perfect egalitarian world that does not exist in real life, so you can't get data from it. (Recall that in Chapter 4, we said that *hakuna matata* reigns in the construct space (it means no worries).) Second is the specification of what is sought. In distribution shift, there is no further specification beyond just trying to match the shifted distribution. In fairness, there are precise policy-driven notions and quantitative criteria that define the desired state of data and/or models that are not dependent on the data distribution you have. You'll learn about these precise notions and how to choose among them in the next chapter.

Related to causal and anticausal learning covered in Chapter 9, the protected attribute is like the environment variable. Fairness and distributive justice are usually conceived in a causal (rather than anticausal) learning framework in which the outcome label is extrinsic: the protected attribute may cause the other features, which in turn cause the selection for care management. However, this setup is not always the case.

10.3 Defining Group Fairness

You've gone back to the problem specification phase after some amount of data understanding because you and the problem owner have realized that there is a strong possibility of unfairness if left unchecked. Given the Section 1557 regulations Sospital is working under as a health insurer, you start by looking

[5]Moninder Singh. "Algorithmic Selection of Patients for Case Management: Alternative Proxies to Healthcare Costs." In: *Proceedings of the AAAI Workshop on Trustworthy AI for Healthcare*. Feb. 2021.
[6]Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. "Exchanging Lessons Between Algorithmic Fairness and Domain Generalization." arXiv:2010.07249, 2020.

deeper into group fairness. Group fairness is about comparing members of the privileged group and members of the unprivileged group on average.

10.3.1 Statistical Parity Difference and Disparate Impact Ratio

One key concept in unwanted discrimination is *disparate impact*: privileged and unprivileged groups receiving different outcomes irrespective of the decision maker's intent and irrespective of the decision-making procedure. Statistical parity difference is a group fairness metric that you can consider in the care management problem specification that quantifies disparate impact by computing the difference in selection rates of the favorable label $P(\hat{y}(X) = \text{fav})$ (rate of receiving extra care) between the privileged ($Z = \text{priv}$; whites) and unprivileged groups ($Z = \text{unpr}$; blacks):

$$\text{statistical parity difference} = P(\hat{y}(X) = \text{fav} \mid Z = \text{unpr}) - P(\hat{y}(X) = \text{fav} \mid Z = \text{priv}).$$

Equation 10.1

A value of 0 means that members of the unprivileged group (blacks) and the privileged group (whites) are getting selected for extra care management at equal rates, which is considered a fair situation. A negative value of statistical parity difference indicates that the unprivileged group is at a disadvantage and a positive value indicates that the privileged group is at a disadvantage. A requirement in a problem specification may be that the learned model must have a statistical parity difference close to 0. An example calculation of statistical parity difference is shown in Figure 10.2.

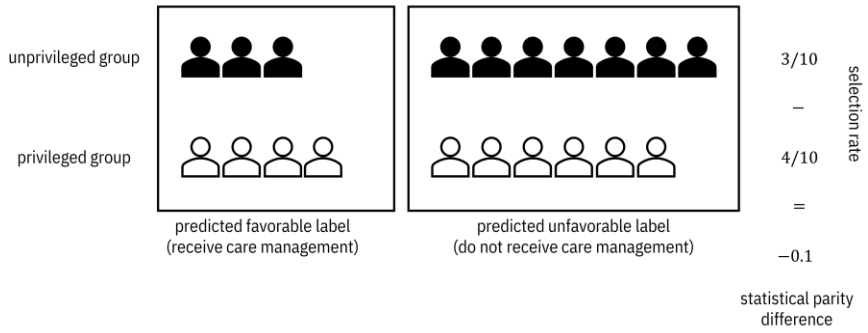


Figure 10.2. *An example calculation of statistical parity difference.* Accessible caption. 3 members of the unprivileged group are predicted with the favorable label (receive care management) and 7 are predicted with the unfavorable label (don't receive care management). 4 members of the privileged group are predicted with the favorable label and 6 are predicted with the unfavorable label. The selection rate for the unprivileged group is 3/10 and for the privileged group is 4/10. The difference, the statistical parity difference is −0.1.

Disparate impact can also be quantified as a ratio:

$$\text{disparate impact ratio} = P(\hat{y}(X) = \text{fav} \mid Z = \text{unpr})/P(\hat{y}(X) = \text{fav} \mid Z = \text{priv}).$$

Equation 10.2

Here, a value of 1 indicates fairness, values less than 1 indicate disadvantage faced by the unprivileged group, and values greater than 1 indicate disadvantage faced by the privileged group. The *disparate impact ratio* is also sometimes known as the *relative risk ratio* or the *adverse impact ratio*. In some application domains such as employment, a value of the disparate impact ratio less than 0.8 is considered unfair and values greater than 0.8 are considered fair. This so-called *four-fifths rule* problem specification is asymmetric because it does not speak to disadvantage experienced by the privileged group. It can be symmetrized by considering disparate impact ratios between 0.8 and 1.25 to be fair. Statistical parity difference and disparate impact ratio can be understood as measuring a form of *independence* between the prediction $\hat{y}(X)$ and the protected attribute Z.[7] Besides statistical parity difference and disparate impact ratio, another way to quantify the independence between $\hat{y}(X)$ and Z is their mutual information.

Both statistical parity difference and disparate impact ratio can also be defined on the training data instead of the model predictions by replacing $\hat{y}(X)$ with Y. Thus, they can be measured and tested (1) on the dataset before model training, as a *dataset fairness metric*, as well as (2) on the learned classifier after model training as a *classifier fairness metric*, shown in Figure 10.3.

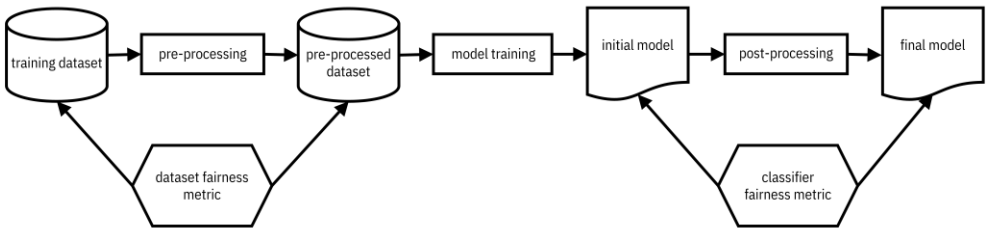


Figure 10.3. *Two types of fairness metrics in different parts of the machine learning pipeline.* Accessible caption. A block diagram with a training dataset as input to a pre-processing block with a pre-processed dataset as output. The pre-processed dataset is input to a model training block with an initial model as output. The initial model is input to a post-processing block with a final model as output. A dataset fairness metric block is applied to the training dataset and pre-processed dataset. A classifier fairness metric block is applied to the initial model and final model.

10.3.2 Average Odds Difference

You've examined disparate impact-based group fairness metrics so far, but want to learn another one before you start comparing and contrasting them as you figure out the problem specification for the care management model. A different group fairness metric is *average odds difference*, which is based on model performance metrics rather than simply the selection rate. (It can thus only be used as a classifier fairness metric, not a dataset fairness metric as shown in Figure 10.3.) The average odds difference involves the two metrics in the ROC: the true favorable label rate (true positive rate) and the false favorable label rate (false positive rate). You take the difference of true favorable rates between the

[7]Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. URL: https://fairmlbook.org, 2020.

unprivileged and privileged groups and the difference of the false favorable rates between the unprivileged and privileged groups, and average them:

$$\begin{aligned}&\text{average odds difference}\\&= \tfrac{1}{2}\big(P(\hat{y}(X) = \text{fav} \mid Y = \text{fav},\ Z = \text{unpr}) - P(\hat{y}(X) = \text{fav} \mid Y = \text{fav},\ Z = \text{priv})\big)\\&+ \tfrac{1}{2}\big(P(\hat{y}(X) = \text{fav} \mid Y = \text{unf},\ Z = \text{unpr}) - P(\hat{y}(X) = \text{fav} \mid Y = \text{unf},\ Z = \text{priv})\big).\end{aligned}$$

Equation 10.3

An example calculation of average odds difference is shown in Figure 10.4.

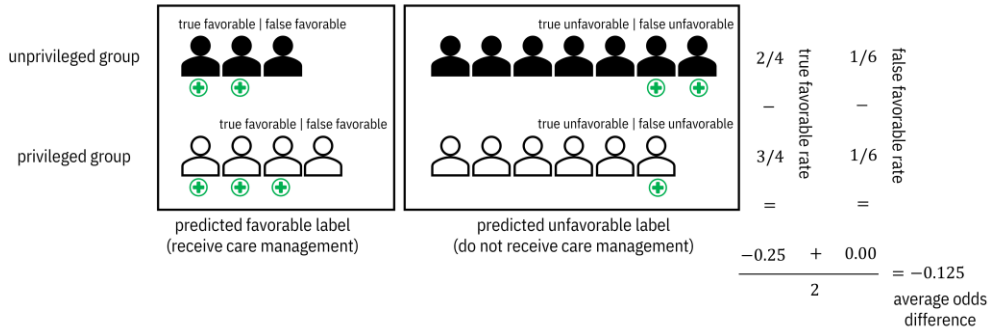


Figure 10.4. *An example calculation of average odds difference. The crosses below the members indicate a true need for care management.* Accessible caption. In the unprivileged group, 2 members receive true favorable outcomes and 2 receive false unfavorable outcomes, giving a 2/4 true favorable rate. In the privileged group, 3 members receive true favorable outcomes and 1 receives a false unfavorable outcome, giving a 3/4 true favorable rate. The true favorable rate difference is −0.25. In the unprivileged group, 1 member receives a false favorable outcome and 5 receive a true unfavorable outcome, giving a 1/6 false favorable rate. In the privileged group, 1 member receives a false favorable outcome and 5 receive a true unfavorable outcome, giving a 1/6 false favorable rate. The false favorable rate difference is 0. Averaging the two differences gives a −0.125 average odds difference.

In the average odds difference, the true favorable rate difference and the false favorable rate difference can cancel out and hide unfairness, so it is better to take the absolute value before averaging:

$$\begin{aligned}&\text{average absolute odds difference}\\&= \tfrac{1}{2}\big|P(\hat{y}(X) = \text{fav} \mid Y = \text{fav},\ Z = \text{unpr}) - P(\hat{y}(X) = \text{fav} \mid Y = \text{fav},\ Z = \text{priv})\big|\\&+ \tfrac{1}{2}\big|P(\hat{y}(X) = \text{fav} \mid Y = \text{unf},\ Z = \text{unpr}) - P(\hat{y}(X) = \text{fav} \mid Y = \text{unf},\ Z = \text{priv})\big|.\end{aligned}$$

Equation 10.3

The average odds difference is a way to measure the *separation* of the prediction $\hat{y}(X)$ and the protected attribute Z by the true label Y in any of the three Bayesian networks shown in Figure 10.5. A value of 0 average absolute odds difference indicates independence of $\hat{y}(X)$ and Z conditioned on Y. This is deemed a fair situation and termed *equality of odds*.

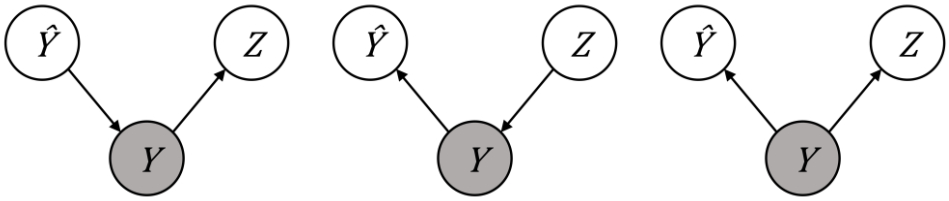


Figure 10.5. *Illustration of the true label Y separating the prediction and the protected attribute in various Bayesian networks.* Accessible caption. Three networks that show separation: $\hat{Y} \rightarrow \underline{Y} \rightarrow Z$, $\hat{Y} \leftarrow \underline{Y} \leftarrow Z$, and $\hat{Y} \leftarrow \underline{Y} \rightarrow Z$.

10.3.3 Choosing Between Statistical Parity and Average Odds Difference

What's the point of these two different group fairness metrics? They don't appear to be radically different. But they actually *are* radically different in an important conceptual way: either you believe there is social bias during measurement or not. These two worldviews have been named (1) "we're all equal" (the privileged group and unprivileged group have the same inherent distribution of health in the construct space, but there is bias during measurement that makes it appear this is not the case) and (2) "what you see is what you get" (there are inherent differences between the two groups in the construct space and this shows up in the observed space without a need for any bias during measurement).[8] Since under the "we're all equal" worldview, there is already structural bias in the observed space (blacks have lower health utilization and cost for the same level of health as whites), it does not really make sense to look at model accuracy rates computed in an already-biased space. Therefore, independence or disparate impact fairness definitions make sense and your problem specification should be based on them. However, if you believe that "what you see is what you get"—the observed space is a true representation of the inherent distributions of the groups and the only bias is sampling bias—then the accuracy-related equality of odds fairness metrics make sense. In this case, your problem specification should be based on equality of odds.

10.3.4 Average Predictive Value Difference

And if it wasn't complicated enough, let's throw one more group fairness definition into the mix: *calibration by group* or *sufficiency*. Recall from Chapter 6 that for continuous score outputs, the predicted score corresponds to the proportion of positive true labels in a *calibrated* classifier, or $P(Y = 1 \mid S = s) = s$. For fairness, you'd like the calibration to be true across the groups defined by protected attributes, so $P(Y = 1 \mid S = s, Z = z) = s$ for all groups z. If a classifier is calibrated by group, it is also *sufficient*, which means that Y and Z conditioned on S (or $\hat{y}(X)$) are independent. The graphical models for sufficiency are shown in Figure 10.6. To allow for better comparison to Figure 10.5 (the graphical models of separation), the predicted score is indicated by $\hat{Y}$ rather than S.

[8]Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. "On the (Im)possibility of Fairness: Different Value Systems Require Different Mechanisms for Fair Decision Making." In: *Communications of the ACM* 64.4 (Apr. 2021), pp. 136–143.

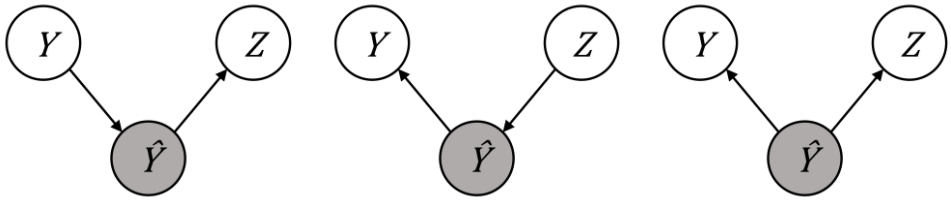


Figure 10.6. *Illustration of the predicted label $\hat{Y}$ separating the true label and the protected attribute in various Bayesian networks, which is known as sufficiency.* Accessible caption. Three networks that show sufficiency: $Y \rightarrow \underline{\hat{Y}} \rightarrow Z$, $Y \leftarrow \underline{\hat{Y}} \leftarrow Z$, and $Y \leftarrow \underline{\hat{Y}} \rightarrow Z$.

Since sufficiency and separation are somewhat opposites of each other with Y and $\hat{Y}$ reversed, their quantifications are also opposites with Y and $\hat{Y}$ reversed. Recall from Chapter 6 that the positive predictive value is the reverse of the true positive rate: $P(Y = \text{fav} \mid \hat{y}(X) = \text{fav})$ and that the false omission rate is the reverse of the false positive rate: $P(Y = \text{fav} \mid \hat{y}(X) = \text{unf})$. To quantify sufficiency unfairness, compute the average difference of the positive predictive value and false omission rate across the unprivileged (black) and privileged (white) groups:

$$\begin{aligned}&\text{average predictive value difference}\\ &= \tfrac{1}{2}\big(P(Y = \text{fav} \mid \hat{y}(X) = \text{fav},\ Z = \text{unpr}) - P(Y = \text{fav} \mid \hat{y}(X) = \text{fav},\ Z = \text{priv})\big)\\ &+ \tfrac{1}{2}\big(P(Y = \text{fav} \mid \hat{y}(X) = \text{unf},\ Z = \text{unpr}) - P(Y = \text{fav} \mid \hat{y}(X) = \text{unf},\ Z = \text{priv})\big).\end{aligned}$$

Equation 10.4

An example calculation for average predictive value difference is shown in Figure 10.7. The example illustrates a case in which the two halves of the metric cancel out because they have opposite sign, so a version with absolute values before averaging makes sense:

$$\begin{aligned}&\text{average absolute predictive value difference}\\ &= \tfrac{1}{2}\left|P(Y = \text{fav} \mid \hat{y}(X) = \text{fav},\ Z = \text{unpr}) - P(Y = \text{fav} \mid \hat{y}(X) = \text{fav},\ Z = \text{priv})\right|\\ &+ \tfrac{1}{2}\left|P(Y = \text{fav} \mid \hat{y}(X) = \text{unf},\ Z = \text{unpr}) - P(Y = \text{fav} \mid \hat{y}(X) = \text{unf},\ Z = \text{priv})\right|.\end{aligned}$$

Equation 10.5

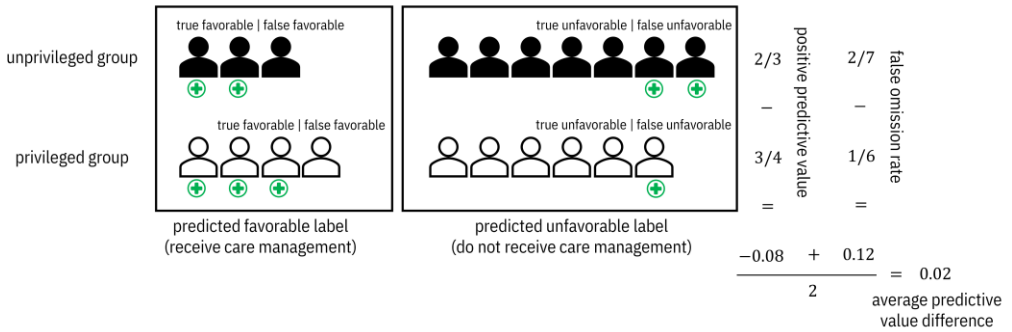


Figure 10.7. *An example calculation of average predictive value difference. The crosses below the members indicate a true need for care management.* Accessible caption. In the unprivileged group, 2 members receive true favorable outcomes and 1 receives a false unfavorable outcome, giving a 2/3 positive predictive value. In the privileged group, 3 members receive true favorable outcomes and 1 receives a false unfavorable outcome, giving a 3/4 positive predictive value. The positive predictive value difference is −0.08. In the unprivileged group, 2 members receive a false unfavorable outcome and 5 receive a true unfavorable outcome, giving a 2/7 false omission rate. In the privileged group, 1 member receives a false unfavorable outcome and 5 receive a true unfavorable outcome, giving a 1/6 false omission rate. The false omission rate difference is 0.12. Averaging the two differences gives a 0.02 average predictive value difference.

10.3.5 Choosing Between Average Odds and Average Predictive Value Difference

What's the difference between separation and sufficiency? Which one makes more sense for the Sospital care management model? This is not a decision based on politics and worldviews like the decision between independence and separation. It is a decision based on what the favorable label grants the affected user: is it assistive or simply non-punitive?[9] Getting a loan is assistive, but not getting arrested is non-punitive. Receiving care management is assistive. In assistive cases like receiving extra care, separation (equalized odds) is the preferred fairness metric because it relates to recall (true positive rate), which is of primary concern in these settings. If receiving care management had been a non-punitive act, then sufficiency (calibration) would have been the preferred fairness metric because precision is of primary concern in non-punitive settings. (Precision is equivalent to positive predictive value, which is one of the two components of the average predictive value difference.).

10.3.6 Conclusion

You can construct all sorts of different group fairness metrics by computing differences or ratios of the various confusion matrix entries and other classifier performance metrics detailed in Chapter 6, but independence, separation, and sufficiency are the three main ones. They are summarized in Table 10.1.

[9]Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. "On the Applicability of ML Fairness Notions." arXiv:2006.16745, 2020. Boris Ruf and Marcin Detyniecki. "Towards the Right Kind of Fairness in AI." arXiv:2102.08453, 2021.

Table 10.1. *The three main types of group fairness metrics.*

Type	Statistical Relationship	Fairness Metric	Can Be A Dataset Metric?	Social Bias in Measurement	Favorable Label
independence	$\hat{Y} \perp\!\!\!\perp Z$	statistical parity difference	yes	yes	assistive or non-punitive
separation	$\hat{Y} \perp\!\!\!\perp Z \mid Y$	average odds difference	no	no	assistive
sufficiency (calibration)	$Y \perp\!\!\!\perp Z \mid \hat{Y}$	average predictive value difference	no	no	non-punitive

Based on the different properties of the three group fairness metrics, and the likely social biases in the data you're using to create the Sospital care management model, you should focus on independence and statistical parity difference.

10.4 Defining Individual and Counterfactual Fairness

An important concept in fairness is *intersectionality*. Things might look fair when you look at different protected attributes separately, but when you define unprivileged groups as the intersection of multiple protected attributes, such as black women, group fairness metrics show unfairness. You can imagine making smaller and smaller groups by including more and more attributes, all the way to a logical end of groups that are just individuals that share all of their feature values. At this extreme, the group fairness metrics described in the previous section are no longer meaningful and a different notion of sameness is needed. That notion is *individual fairness* or *consistency*: that all individuals with the same feature values should receive the same predicted label and that individuals with similar features should receive similar predicted labels.

10.4.1 Consistency

The consistency metric is quantified as follows:

$$\text{consistency} = 1 - \frac{1}{n}\sum_{j=1}^{n}\left|\hat{y}_j - \frac{1}{k}\sum_{j' \in \mathcal{N}_k(x_j)} \hat{y}_{j'}\right|.$$

Equation 10.6

For each of the n Sospital members, the prediction $\hat{y}_j$ is compared to the average prediction of the k nearest neighbors. When the predicted labels of all of the k nearest neighbors match the predicted label of the person themselves, you get 0. If all of the nearest neighbor predicted labels are different from the predicted label of the person, the absolute value is 1. Overall, because of the 'one minus' at the beginning of Equation 10.6, the consistency metric is 1 if all similar points have similar labels and less than 1 if similar points have different labels.

The biggest question in individual fairness is deciding the distance metric by which the nearest neighbors are determined. Which kind of distance makes sense? Should all features be used in the distance computation? Should protected attributes be excluded? Should some feature dimensions be corrected for in the distance computation? These choices are where politics and worldviews come into play.[10] Typically, protected attributes are excluded, but they don't have to be. If you believe there is no bias during measurement (the "what you see is what you get" worldview), then you should simply use the features as is. In contrast, suppose you believe that there are structural social biases in measurement (the "we're all equal" worldview). In that case, you should attempt to undo those biases by correcting the features as they're fed into a distance computation. For example, if you believe that blacks with three outpatient doctor visits are equal in health to whites with five outpatient doctor visits, then your distance metric can add two outpatient visits to the black members as a correction.

10.4.2 Counterfactual Fairness

One special case of individual fairness is when two patients have exactly the same feature values and only differ in one protected attribute. Think of two patients, one black and one white who have an identical history of interaction with the health care system. The situation is deemed fair if both receive the same predicted label—either both are given extra care management or both are not given extra care management—and unfair otherwise. Now take this special case a step further. As a thought experiment, imagine an intervention $do(Z)$ that changes the protected attribute of a Sospital member from black to white or vice versa. If the predicted label remains the same for all members, the classifier is *counterfactually fair*.[11] (Actually intervening to change a member's protected attribute is usually not possible immediately, but this is just a thought experiment.) Counterfactual fairness can be tested using treatment effect estimation methods from Chapter 8.

Protected attributes *causing* different outcomes across groups is an important consideration in many laws and regulations.[12] Suppose you have a full-blown causal graph of all the variables given to you or you discover one from data using the methods of Chapter 8. In that case, you can see which variables have causal paths to the label nodes, either directly or passing through other variables. If any of the variables with causal paths to the label are considered protected attributes, you have a fairness problem to investigate and mitigate.

10.4.3 Theil Index

If you don't want to decide between group and individual fairness metrics as you're figuring out the Sospital care management problem specification, do you have any other options? Yes you do. You can use the Theil index, which was first introduced in Chapter 3 as a summary statistic for uncertainty. It naturally combines both individual and group fairness considerations. Remember from that chapter that the Theil index was originally developed to measure the distribution of wealth in a society. A value

[10]Reuben Binns. "On the Apparent Conflict Between Individual and Group Fairness." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 514–524.
[11]Joshua R. Loftus, Chris Russell, Matt J. Kusner, and Ricardo Silva. "Causal Reasoning for Algorithmic Fairness." arXiv:1805.05859, 2018.
[12]Alice Xiang. "Reconciling Legal and Technical Approaches to Algorithmic Bias." In: *Tennessee Law Review* 88.3 (2021).

of 1 indicates a totally unfair society where one person holds all the wealth and a value of 0 indicates an egalitarian society where all people have the same amount of wealth.

What is the equivalent of wealth in the context of machine learning and distributive justice in health care management? It has to be some sort of non-negative benefit value b_j that you want to be equal for different Sospital members. Once you've defined the benefit b_j, plug it into the Theil index expression and use it as a combined group and individual fairness metric:

$$\text{Theil index } = \frac{1}{n}\sum_{j=1}^{n}\frac{b_j}{\bar{b}}\log\frac{b_j}{\bar{b}}.$$

Equation 10.7

The equation averages the benefit divided by the mean benefit $\bar{b}$, multiplied by its natural log, across all people.

That's all well and good, but benefit to who and under which worldview? The research group that proposed using the Theil index in algorithmic fairness suggested that b_j be 2 for false favorable labels (false positives), 1 for true favorable labels (true positives), 1 for true unfavorable labels (true negatives), and 0 for false unfavorable labels (false negatives).[13] This recommendation is seemingly consistent with the "what you see is what you get" worldview because it is examining model performance, assumes the costs of false positives and false negatives are the same, and takes the perspective of affected members who want to get care management even if they are not truly suitable candidates. More appropriate benefit functions for the problem specification of the Sospital model may be b_j that are (1) 1 for true favorable and true unfavorable labels and 0 for false favorable and false unfavorable labels ("what you see is what you get" while balancing societal needs), or (2) 1 for true favorable and false favorable labels and 0 for true unfavorable and false unfavorable labels ("we're all equal").

10.4.4 Conclusion

Individual fairness consistency and Theil index are both excellent ways to capture various nuances of fairness in different contexts. Just like group fairness metrics, they require you to clarify your worldview and aim for the same goals in a bottom-up way. Since the Sospital care management setting is regulated using group fairness language, it behooves you to use group fairness metrics in your problem specification and modeling. Counterfactual or causal fairness is a strong requirement from the perspective of the philosophy and science of law, but the regulations are only just catching up. So you might need to utilize causal fairness in problem specifications in the future, but not just yet. As you've learned so far, the problem specification and data phases are critical for fairness. But that makes the modeling phase no less important. The next section focuses on bias mitigation to improve fairness as part of the modeling pipeline.

---

[13]Till Speicher, Hoda Heidari, Nina Grgić-Hlača, Krishna P. Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. "A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. London, England, UK, Jul. 2018, pp. 2239–2248.

10.5 Mitigating Unwanted Bias

From the earlier phases of the lifecycle of the Sospital care management model, you know that you must address unwanted biases during the modeling phase. Given the quantitative definitions of fairness and unfairness you've worked through, you know that mitigating bias entails introducing some sort of statistical independence between protected attributes like race and true or predicted labels of needing care management. That sounds easy enough, so what's the challenge? What makes bias mitigation difficult is that other regular predictive features X have statistical dependencies with the protected attributes and the labels (a node for X was omitted from Figure 10.5 and Figure 10.6, but out-of-sight does not mean out-of-mind). The regular features can reconstruct the information contained in the protected attributes and introduce dependencies, even if you do the most obvious thing of dropping the protected attributes from the data. For example, race can be strongly associated both with certain health care providers (some doctors have predominantly black patients and other doctors have predominantly white patients) and with historical selection for extra care management.

Bias mitigation methods *must* be more clever than simply dropping protected attributes. Don't take a shortcut: dropping protected attributes is never the right answer. Remember the two main ways of mitigating the ills of distribution shift in Chapter 9: *adaptation* and *min-max robustness*. When applied to bias mitigation, adaptation-based techniques are much more common than robustness-based ones, but rely on having protected attributes in the training dataset.[14] They are the subject of the remainder of this section. If the protected attributes are not available in the training data, min-max robustness techniques for fairness that mirror those for distribution shift can be used.[15]

Figure 10.8 (a subset of Figure 10.3) shows three different points of intervention for bias mitigation: (1) *pre-processing* which alters the statistics of the training data, (2) *in-processing* which adds extra constraints or regularization terms to the learning process, and (3) *post-processing* which alters the output predictions to make them more fair. Pre-processing can only be done when you have the ability to touch and modify the training data. Since in-processing requires you to mess with the learning algorithm, it is the most involved and least flexible. Post-processing is almost always possible and the easiest to pull off. However, the earlier in the pipeline you are, the more effective you can be.

There are several specific methods within each of the three categories of bias mitigation techniques (pre-processing, in-processing, post-processing). Just like for accuracy, no one best algorithm outperforms all other algorithms on all datasets and fairness metrics (remember the no free lunch theorem). Just like there are differing domains of competence for classifiers covered in Chapter 7, there are differing domains of competence for bias mitigation algorithms. However, fairness is a new field that has not yet been studied extensively enough to have good characterizations of those domains of competence yet. In Chapter 7, it was important to go down into the details of machine learning methods

[14]The assumption that training datasets contain protected attributes can be violated for regulatory or privacy reasons. The situation is known as *fairness under unawareness*. See: Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffry Svacha, and Madeleine Udell. "Fairness Under Unawareness." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Atlanta, Georgia, USA, Jan. 2019, pp. 339–348.

[15]Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. "Fairness Without Demographics in Repeated Loss Minimization." In: *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 1929–1938. Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. "Fairness Without Demographics through Adversarially Reweighted Learning." In: *Advances in Neural Information Processing Systems* 33 (Dec. 2020), pp. 728–740.

because that understanding is used in this and other later chapters. The reason to dive into the details of bias mitigation algorithms is different. In choosing a bias mitigation algorithm, you have to (1) know where in the pipeline you can intervene, (2) consider your worldview, and (3) understand whether protected attributes are allowed as features and will be available in the deployment data when you are scoring new Sospital members.

Figure 10.8. *Three types of bias mitigation algorithms in different parts of the machine learning pipeline.* Accessible caption. A block diagram with a training dataset as input to a bias mitigation pre-processing block with a pre-processed dataset as output. The pre-processed dataset is input to a bias mitigation in-processing block with an initial model as output. The initial model is input to a bias mitigation post-processing block with a final model as output.

10.5.1 Pre-Processing

At the pre-processing stage of the modeling pipeline, you don't have the trained model yet. So pre-processing methods cannot explicitly include fairness metrics that involve model predictions. Therefore, most pre-processing methods are focused on the "we're all equal" worldview, but not exclusively so. There are several ways for pre-processing a training data set: (1) augmenting the dataset with additional data points, (2) applying instance weights to the data points, and (3) altering the labels.

One of the simplest algorithms for pre-processing the training dataset is to append additional rows of made-up members that do not really exist. These imaginary members are constructed by taking existing member rows and flipping their protected attribute values (like counterfactual fairness).[16] The augmented rows are added sequentially based on a distance metric so that 'realistic' data points close to modes of the underlying dataset are added first. This ordering maintains the fidelity of the data distribution for the learning task. A plain uncorrected distance metric takes the "what you see is what you get" worldview and only overcomes sampling bias, not measurement bias. A corrected distance metric like the example described in the previous section (adding two outpatient visits to the black members) takes the "we're all equal" worldview and can overcome both measurement and sampling bias (threats to both construct and external validity). This data augmentation approach needs to have protected attributes as features of the model and they must be available in deployment data.

Another way to pre-process the training data set is through sample weights, similar to inverse probability weighting and importance weighting seen in Chapter 8 and Chapter 9, respectively. The *reweighing* method is geared toward improving statistical parity ("we're all equal" worldview), which can be assessed before the care management model is trained and is a dataset fairness metric.[17] The goal of

[16]Shubham Sharma, Yunfeng Zhang, Jesús M. Ríos Aliaga, Djallel Bouneffouf, Vinod Muthusamy, and Kush R. Varshney. "Data Augmentation for Discrimination Prevention and Bias Disambiguation." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, New York, USA, Feb. 2020, pp. 358–364.

[17]Faisal Kamiran and Toon Calders. "Data Preprocessing Techniques for Classification without Discrimination." In: *Knowledge and Information Systems* 33.1 (Oct. 2012), pp. 1–33.

independence between the label and protected attribute corresponds to their joint probability being the product of their marginal probabilities. This product probability appears in the numerator and the actual observed joint probability appears in the denominator of the weight:

$$w_j = \frac{p_Y(y_j)p_Z(z_j)}{p_{Y,Z}(y_j, z_j)}.$$

Equation 10.8

Protected attributes are required in the training data to learn the model, but they don't have to be part of the model or the deployment data.

Whereas data augmentation and reweighing do not change the training data you have from historical care management decisions, other methods do. One simple method, only for statistical parity and the "we're all equal" worldview, known as *massaging* flips unfavorable labels of unprivileged group members to favorable labels and favorable labels of privileged group members to unfavorable labels.[18] The chosen data points are those closest to the decision boundary that have low confidence. Massaging does not need to have protected attributes in the deployment data.

A different approach, the *fair score transformer*, works on (calibrated) continuous score labels $S = p_{Y|X}(Y = fav \mid x)$ rather than binary labels.[19] It is posed as an optimization in which you find transformed scores S' that have small cross-entropy with the original scores S, i.e. $H(S \parallel S')$, while constraining the statistical parity difference, average odds difference, or other group fairness metrics of your choice to be of small absolute value. You convert the pre-processed scores back into binary labels with weights to feed into a standard training algorithm. You can take the "what you see is what you get" worldview with the fair score transformer because it assumes that the classifier later trained on the pre-processed dataset is competent, so that the pre-processed score it produces is a good approximation to the score predicted by the trained model. Although there are pre-processing methods that alter both the labels and (structured or semi-structured) features,[20] the fair score transformer proves that you only need to alter the labels. It can deal with deployment data that does not come with protected attributes.

Data augmentation, reweighing, massaging, and fair score transformer all have their own domains of competence. Some perform better than others on different fairness metrics and dataset characteristics. You'll have to try different ones to see what happens on the Sospital data.

---

[18]Faisal Kamiran and Toon Calders. "Data Preprocessing Techniques for Classification without Discrimination." In: *Knowledge and Information Systems* 33.1 (Oct. 2012), pp. 1–33.

[19]Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio P. Calmon. "Optimized Score Transformation for Fair Classification." In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Aug. 2020, pp. 1673–1683.

[20]Some examples are the methods described in the following three papers. Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. "Certifying and Removing Disparate Impact." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, Australia, Aug. 2015, pp. 259–268. Flavio P. Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. "Optimized Pre-Processing for Discrimination Prevention." In: *Advances in Neural Information Processing Systems* 30 (Dec. 2017), pp. 3992–4001. Prasanna Sattigeri, Samuel C. Hoffman, Vijil Chenthamarakshan, and Kush R. Varshney. "Fairness GAN: Generating Datasets with Fairness Properties Using a Generative Adversarial Network." In: *IBM Journal of Research and Development* 63.4/5 (Jul./Sep. 2019), p. 3.

10.5.2 In-Processing

In-processing bias mitigation algorithms are straightforward to state, but often more difficult to actually optimize. The statement is as follows: take an existing risk minimization supervised learning algorithm, such as (a repetition of Equation 7.4):

$$\hat{y}(\cdot) = arg\ \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} L\left(y_j, f\left(x_j\right)\right) + \lambda J(f)$$

Equation 10.9

and regularize or constrain it using a fairness metric. The algorithm can be logistic regression and the regularizer can be statistical parity difference, in which case you have the *prejudice remover*.[21] More recent fair learning algorithms are broader and allow for any standard risk minimization algorithm along with a broad set of group fairness metrics as constraints that cover the different types of fairness.[22] A recent in-processing algorithm regularizes the objective function using a causal fairness term. Under strong ignorability assumptions (remember from Chapter 8 that these are no unmeasured confounders and overlap), the regularizer is an average treatment effect-like term $J = E[Y \mid do(z = 1), X] - E[Y \mid do(z = 0), X]$.[23]

Once trained, the resulting models can be used on new unseen Sospital members. These in-processing algorithms do not require the deployment data to contain the protected attribute. The trick with all of them is structuring the bias mitigating regularization term or constraint so that the objective function can tractably be minimized through an optimization algorithm.

10.5.3 Post-Processing

If you're in the situation that the Sospital care management model has already been trained and you cannot change it or touch the training data (for example if you are purchasing a pre-trained model from a vendor to include in your pipeline), then the only option you have is to mitigate unwanted biases using post-processing. You can only alter the output predictions $\hat{Y}$ to meet the group fairness metrics you desire based on your worldview (i.e. flipping the predicted labels from receiving care management to not receiving care management and vice versa). If you have some validation data with labels, you can post-process with the "what you see is what you get" worldview. You can always post-process with the "we're all equal" worldview, with or without validation data.

---

[21]Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. "Fairness-Aware Classifier with Prejudice Remover Regularizer." In: *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Bristol, England, UK, Sep. 2012, pp. 35–50.

[22]Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. "A Reductions Approach to Fair Classification." In: *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 60–69. L. Elisa Celis, Lingxiao Huang, Vijay Kesarwani, and Nisheeth K. Vishnoi. "Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Atlanta, Georgia, USA, Jan. 2019, pp. 319–328. Ching-Yao Chuang and Youssef Mroueh. "Fair Mixup: Fairness via Interpolation." In: *Proceedings of the International Conference on Learning Representations*. May 2021.

[23]Pietro G. Di Stefano, James M. Hickey, and Vlasios Vasileiou. "Counterfactual Fairness: Removing Direct Effects Through Regularization." arXiv:2002.10774, 2020.

Since group fairness metrics are computed on average, flipping any random member's label within a group is the same as flipping any other random member's.[24] A random selection of people, however, seems to be procedurally unfair. To overcome this issue, similar to massaging, you can prioritize flipping the labels of members whose data points are near the decision boundary and are thus low confidence samples.[25] You can also choose people within a group so that you reduce individual counterfactual unfairness.[26] All of these approaches require the protected attribute in the deployment data.

The fair score transformer described in the pre-processing section also has a post-processing version, which does not require the protected attribute and should be considered the first choice algorithm in the category of post-processing bias mitigation if the base classifier outputs continuous scores. It performs well empirically and is not computationally-intensive. Just like the pre-processing version, the idea is to find an optimal transformation of the predicted score output into a new score, which can then be thresholded to a binary prediction for the final care management decision that Sospital makes.

10.5.4 Conclusion

All of the different bias mitigation algorithms are options as you're deciding what to finally do in the care management modeling pipeline. The things you have to think about are:

1. where in the pipeline can you make alterations (this will determine the category pre-, in-, or post-processing)
2. which worldview you've decided with the problem owner (this will disallow some algorithms that don't work for the worldview you've decided)
3. whether the deployment data contains the protected attributes (if not, this will disallow some algorithms that require them).

These different decision points are summarized in Table 10.2. After that, you can just go with the algorithm that gives you the best quantitative results. But what is best? It is simply the pipeline with the best value for the fairness metric you've chosen in your problem specification.

But you might ask, shouldn't I consider a tradeoff of fairness and accuracy when I choose the pipeline? Balancing tradeoffs and relationships among different elements of trustworthy machine learning is more fully covered in Chapter 14, but before getting there, it is important to note one important point. Even though it is a convenient shortcut, measuring classification accuracy on data from the prepared data space, which already contains social bias, representation bias, and data preparation bias is not the right thing to do. Just like you should measure performance of distribution shift adaptation on data from the new environment—its construct space, you should measure accuracy after bias mitigation in its construct space where there is no unfairness. There is a tradeoff between fairness and accuracy measured in the prepared data space, but importantly there is no tradeoff between

---

[24]Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. "On Fairness and Calibration." In: *Advances in Neural Information Processing Systems* 31 (Dec. 2017), pp. 5684–5693.

[25]Faisal Kamiran, Asim Karim, and Xiangliang Zhang. "Decision Theory for Discrimination-Aware Classification." In: *Proceedings of the IEEE International Conference on Data Mining*. Brussels, Belgium, Dec. 2012, pp. 924–929.

[26]Pranay K. Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R. Varshney, and Ruchir Puri. "Bias Mitigation Post-Processing for Individual and Group Fairness." In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Brighton, England, UK, May 2019, pp. 2847–2851.

accuracy and fairness in the construct space.[27] You can approximate a construct space test set by using the data augmentation pre-processing method.

Table 10.2. *Characteristics of the main bias mitigation algorithms.*

Algorithm	Category	Fairness	Protected Attributes in Deployment Data
data augmentation	pre	counterfactual	yes
reweighing	pre	independence	no
massaging	pre	independence	no
fair score transformer	pre, post	independence, separation	no
prejudice remover	in	independence	no
recent in-processing algorithms	in	independence, separation, sufficiency	no
causal regularizer	in	counterfactual	no
group fairness post-processing	post	independence, separation	yes
individual and group fairness post-processing	post	counterfactual, independence, separation	yes

In your Sospital problem, you have almost complete flexibility because you do control the training data and model training, are focused on independence and the "we're all equal" worldview, and are able to include protected attributes for Sospital's members in the deployment data. Try everything, but start with the fair score transformer pre-processing.

10.6 Other Considerations

Before concluding the chapter, let's consider a couple other issues. The first did not come up in the Sospital care management use case, but can come up in other use cases. The Sospital problem lent itself to fairness in the context of direct allocation decisions, but that is not the only possibility. There are also harms in representation or quality-of-service, such as bias in search results. For example, image searches for professions might yield only white people, web search results for personal names overrepresented in the black community might be accompanied by advertisements for criminal defense attorneys, and natural language processing algorithms for language translation or query understanding might associate doctors with men and nurses with women automatically. Some of the bias mitigation algorithms for allocative fairness can be used in representational fairness, but different techniques may be more appropriate.

[27]Michael Wick, Swetasudha Panda, and Jean-Baptiste Tristan. "Unlocking Fairness: A Trade-Off Revisited." In: *Advances in Neural Information Processing Systems* 32 (Dec. 2019), pp. 8783–8792. Kit T. Rodolfa, Hemank Lamba, and Rayid Ghani. "Empirical Observation of Negligible Trade-Offs in Machine Learning for Public Policy." In: *Nature Machine Intelligence* 3 (Oct. 2021), pp. 896–904.

> "Most of this work is narrow in scope, focusing on fine-tuning specific models, making datasets more inclusive/representative, and 'debiasing' datasets. Although such work can constitute part of the remedy, a fundamentally equitable path must examine the wider picture, such as unquestioned or intuitive assumptions in datasets, current and historical injustices, and power asymmetries."
>
> —Abeba Birhane, cognitive scientist at University College Dublin

> "I continue to worry that in CS (as in psychology), debates about bias have become a powerful distraction—drawing attention away from what's most important toward what's more easily measurable."
>
> —J. Nathan Matias, behavioral scientist at Cornell University

The second issue is as follows. Have we too easily swept the important considerations of algorithmic fairness under the rug of mathematics? Yes and no. If you have truly thought through the different sources of inequity arising throughout the machine learning lifecycle utilizing a panel of diverse voices, then applying the quantitative metrics and mitigation algorithms is actually pretty straightforward. It is straightforward because of the hard work you've done before getting to the modeling phase of the lifecycle and you should feel confident in going forward. If you have not done the hard work earlier in the lifecycle (including problem specification), blindly applying bias mitigation algorithms might not reduce harms and can even exacerbate them. So don't take shortcuts.

10.7 Summary

- Fairness has many forms, arising from different kinds of justice. Distributive justice is the most appropriate for allocation decisions made or supported by machine learning systems. It asks for some kind of sameness in the outcomes across individuals and groups.
- Unfairness can arise from problem misspecification (including inappropriate proxy labels), feature engineering, measurement of features from the construct space to the observed space, and sampling of data points from the observed space to the raw data space.
- There are two important worldviews in determining which kind of sameness is most appropriate for your problem.
- If you believe there are social biases in measurement (not only representation biases in sampling), then you have the "we're all equal" worldview; independence and statistical parity difference are appropriate notions of group fairness.
- If you believe there are no social biases in measurement, only representation biases in sampling, then you have the "what you see is what you get" worldview; separation, sufficiency, average odds difference, and average predictive value difference are appropriate notions of group fairness.
- If the favorable label is assistive, separation and average odds difference are appropriate notions of group fairness. If the favorable label is non-punitive, sufficiency and average predictive value difference are appropriate notions of group fairness.

- Individual fairness is a limiting version of group fairness with finer and finer groups. Worldviews play a role in determining distance metrics between individuals.
- Bias mitigation algorithms can be applied as pre-processing, in-processing, or post-processing within the machine learning pipeline. Different algorithms apply to different worldviews. The choice of algorithm should consider the worldview in addition to empirical performance.

11

Adversarial Robustness

Imagine that you are a data scientist at a (fictional) new player in the human resources (HR) analytics space named HireRing. The company creates machine learning models that analyze resumes and metadata in job application forms to prioritize candidates for hiring and other employment decisions. They go in and train their algorithms on each of their corporate clients' historical data. As a major value proposition, the executives of HireRing have paid extra attention to ensuring robustness to distribution shift and ensuring fairness of their machine learning pipelines and are now starting to focus their problem specification efforts on securing models from malicious acts. You have been entrusted to lead the charge in this new area of machine learning security. Where should you begin? What are the different threats you need to be worried about? What can you do to defend against potential adversarial attacks?

Adversaries are people trying to achieve their own goals to the detriment of the goals of HireRing and their clients, usually in a secretive way. For example, they may simply want to make the accuracy of an applicant prioritization model worse. They may be more sophisticated and want to trick the machine learning system into putting some small group of applicants at the top of the priority list irrespective of the employability expressed in their features while leaving the model's behavior unchanged for most applicants.

This chapter teaches you all about defending and certifying the models HireRing builds for its clients by:

- distinguishing different threat models based on what the adversary attacks (training data or models), their goals, and what they are privy to know and change,
- defending against different types of attacks through algorithms that add robustness to models, and
- certifying such robustness of machine learning pipelines.

The topic of adversarial robustness relates to the other two chapters in this part of the book on reliability (distribution shift and fairness) because it also involves a mismatch between the training data and the deployment data. You do not know what that difference is going to be, so you have epistemic

uncertainty that you want to adapt to or be robust against. In distribution shift, the difference in distributions is naturally occurring; in fairness, the difference between a just world and the world we live in is because of encompassing societal reasons; in adversarial robustness, the difference between distributions is because of a sneaky adversary. Another viewpoint on adversarial attacks is not through the lens of malicious actors, but from the lens of probing system reliability—pushing machine learning systems to their extremes—by testing them in worst case scenarios. This alternative viewpoint is not the framing of the chapter, but you should keep it in the back of your mind and we will return to it in Chapter 13.

> "In my view, similar to car model development and manufacturing, a comprehensive 'in-house collision test' for different adversarial threats on an AI model should be the new norm to practice to better understand and mitigate potential security risks."
>
> —Pin-Yu Chen, computer scientist at IBM Research

HireRing has just been selected by a large (fictional) retail chain based in the midwestern United States named Kermis to build them a resume and job application screening model. This is your first chance to work with a real client on the problem specification phase for adversarial robustness and not take any shortcuts. To start, you need to work through the different types of malicious attacks and decide how you can make the HireRing model being developed for Kermis the most reliable and trustworthy it can be. Later you'll work on the modeling phase too.

11.1 The Different Kinds of Adversarial Attacks

As part of the problem specification phase for the machine learning-based job applicant classifier that HireRing is building for Kermis, you have to go in and assess the different threats it is vulnerable to. There are three dimensions by which to categorize adversarial attacks.[1] (1) Which part of the pipeline is being attacked: training or deployment? Attacks on training are known as *poisoning* attacks, whereas attacks on deployment are known as *evasion* attacks. (2) What capabilities does the attacker have? What information about the data and model do they know? What data and models can they change and in what way? (3) What is the goal of the adversary? Do they simply want to degrade performance of the resume screening model in general or do they have more sophisticated and targeted objectives? These three dimensions are similar to the three considerations when picking a bias mitigation algorithm in Chapter 10 (part of pipeline, presence of protected attributes, worldview).

A mental model of the different attack types is shown in Figure 11.1. Let's go through each of the dimensions in turn as a sort of checklist to analyze what Kermis should most be worried about and what the HireRing model should protect against most diligently.

---

[1]Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. "Adversarial Attacks and Defences: A Survey." arXiv:1810.00069, 2018. Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V. Vasilakos. "Privacy and Security Issues in Deep Learning: A Survey." In: *IEEE Access* 9 (Dec. 2021), pp. 4566–4593.

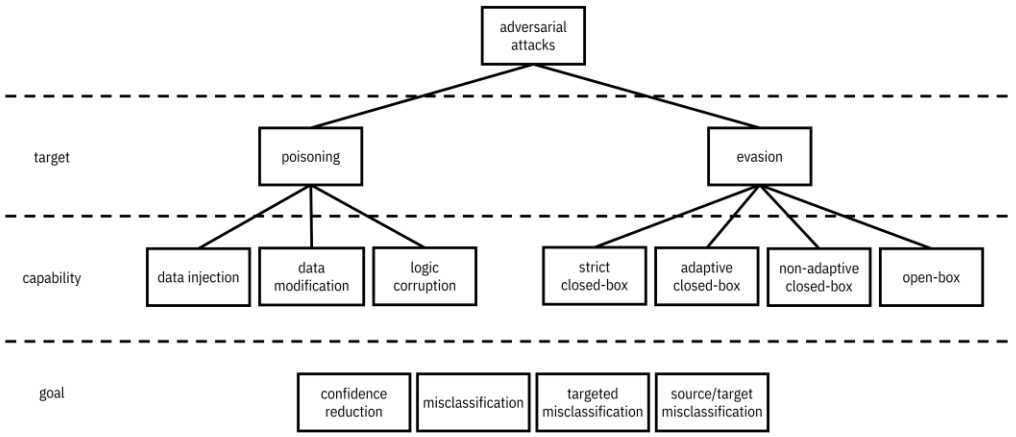


Figure 11.1. *A mental model for the different types of adversarial attacks, according to their target, their capability, and their goal.* A hierarchy diagram with adversarial attacks at its root. Adversarial attacks has children poisoning and evasion, both of which are in the target dimension. Poisoning has children data injection, data modification, and logic corruption, which are in the capability dimension. Evasion has children strict closed-box, adaptive closed-box, non-adaptive closed-box, and open-box, which are in the capability dimension. Below the hierarchy diagram are items in the goal dimension: confidence reduction, misclassification, targeted misclassification, and source/target misclassification, which apply to the whole diagram.

11.1.1 Target

Adversaries may target either the modeling phase or the deployment phase of the machine learning lifecycle. By attacking the modeling phase, they can corrupt the training data or model so that it is mismatched from the data seen in deployment. These are known as poisoning attacks and have similarities with distribution shift, covered in Chapter 9, as they change the statistics of the training data or model. Evasion attacks that target the deployment phase are a different beast that do not have a direct parallel with distribution shift, but have a loose similarity with individual fairness covered in Chapter 10. These attacks are focused on altering individual examples (individual resumes) that are fed into the machine learning system to be evaluated. As such, modifications to single data points may not affect the deployment probability distribution much at all, but can nevertheless achieve the adversary's goals for a given input resume.

One way to understand poisoning and evasion attacks is by way of the decision boundary, shown in Figure 11.2. Poisoning attacks shift the decision boundary in a way that the adversary wants. In contrast, evasion attacks do not shift the decision boundary, but shift data points across the decision boundary in ways that are difficult to detect. An original data point, the features of the resume x, shifted by δ becomes $x + \delta$. A basic mathematical description of an evasion attack is the following:

$$\hat{y}(x + \delta) \neq \hat{y}(x) \text{ such that } \|\delta\| \leq \epsilon.$$

Equation 11.1

The adversary wants to find a small perturbation δ to add to the resume x so that the predicted label changes ($\hat{y}(x+\delta) \neq \hat{y}(x)$) from select to reject or vice versa. In addition, the perturbation should be smaller in length or norm $\|\cdot\|$ than some small value ϵ. The choice of norm and value depend on the application domain. For semi-structured data modalities, the norm should capture human perception so that the perturbed data point and the original data point look or sound almost the same to people.

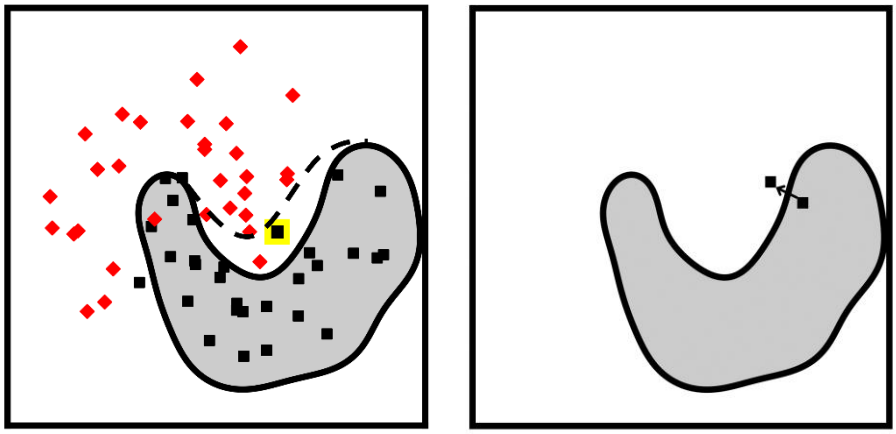


Figure 11.2. *Examples of a poisoning attack (left) and an evasion attack (right). In the poisoning attack, the adversary has injected a new data point, the square with the light border into the training data. This action shifts the decision boundary from what it would have been: the solid black line, to something else that the adversary desires: the dashed black line. More diamond deployment data points will now be misclassified. In the evasion attack, the adversary subtly perturbs a deployment data point across the decision boundary so that it is now misclassified.* Accessible caption. The stylized plot illustrating a poisoning attack shows two classes of data points arranged in a noisy yin yang or interleaving moons configuration and a decision boundary smoothly encircling one of the classes with a blob-like region. A poisoning data point with the label of the inside of the region is added outside the region. It causes a new decision boundary that puts it inside, while also causing the misclassification of another data point. The stylized plot illustrating the evasion attack has a data point inside the blob-like region. The attack pushes it outside the region.

Another way to write the label change $\hat{y}(x+\delta) \neq \hat{y}(x)$ is through the zero-one loss function: $L\big(\hat{y}(x), \hat{y}(x+\delta)\big) = 1$. (Remember that the zero-one loss takes value 0 when both arguments are the same and value 1 when the arguments are different.) Because the zero-one loss can only take the two values 0 and 1, you can also write the adversarial example using a maximum as:

$$\max_{\|\delta\| \leq \epsilon} L\big(\hat{y}(x), \hat{y}(x+\delta)\big).$$

Equation 11.2

In this notation, you can also put in other loss functions such as cross-entropy loss, logistic loss, and hinge loss from Chapter 7.

11.1.2 Capability

Some adversaries are more capable than others. In the poisoning category, adversaries change the training data or model somehow, so they have to have some access inside Kermis' information technology infrastructure. The easiest thing they can do is slip in some additional resumes that get added to the training data. This is known as *data injection*. More challenging is *data modification*, in which the adversary changes labels or features in the existing training dataset. The most challenging of all is *logic corruption*, in which the adversary changes the code and behavior of the machine learning algorithm or model. You can think of the data injection and data modification attacks as somewhat similar to bias mitigation pre-processing and logic corruption as somewhat similar to bias mitigation in-processing, except for a nefarious purpose.

In the evasion category, the adversary does not need to change anything at all inside Kermis' systems. So these are easier attacks to carry out. The attackers just have to create adversarial examples: specially crafted resumes designed in clever ways to fool the machine learning system. But how adversaries craft these tricky resumes depends on what information they have about how the model makes its predictions. The easiest thing for an adversary to do is just submit a bunch of resumes into the HireRing model and see whether they get selected or not; this gives the adversary a labeled dataset. When adversaries cannot change the set of resumes and just have to submit a batch that they have, it is called *strict closed-box* access. When they can change the input resumes based on the previous ones they've submitted and the predicted labels they've observed, it is called *adaptive closed-box* access. Adaptivity is a bit harder because the attacker might have to wait a while for Kermis to select or not select the resumes that they've submitted. You might also be able to catch on that something strange is happening over time. The next more difficult kind of information that adversaries can have about the HireRing model trained for Kermis is known as *non-adaptive closed-box* access. Here, the adversary knows the training data distribution $p_{X,Y}(x, y)$ but cannot submit resumes. Finally, the classifier decision function itself $\hat{y}(\cdot)$ is the most difficult-to-obtain information about a model for an adversary. This full knowledge of the classifier is known as *open-box* access.

Since Kermis has generally good cybersecurity overall, you should be less worried about poisoning attacks, especially logic corruption attacks. Even open-box access for an evasion attack seems less likely. Your biggest fear should be one of the closed-box evasion attacks. Nevertheless, you shouldn't let your guard down and you should still think about defending against all of the threats.

11.1.3 Goal

The third dimension of threats is the goal of the adversary, which applies to both poisoning and evasion attacks. Different adversaries try to do different things. The easiest goal is *confidence reduction*: to shift classifier scores so that they are closer to the middle of the range [0,1] and thus less confident. The next goal is *misclassification*: trying to get the classifier to make incorrect predictions. (This is the formulation given in Equation 11.1.) Job applications to Kermis that should be selected are rejected and vice versa. When you have a binary classification problem like you do in applicant screening, there is only one way to be wrong: predicting the other label. However, when you have more than two possible labels, misclassification can produce any other label that is not the true one. *Targeted misclassification* goes a step further and ensures that the misclassification isn't just any other label, but a specific one of the attacker's choice. Finally, and most sophisticated of all, *source/target misclassification* attacks are

designed so that misclassification only happens for some input job applications and the label of the incorrect prediction also depends on the input. *Backdoor* or *Trojan* attacks are an example of source/target misclassification in which a small subset of inputs (maybe ones whose resumes include some special keyword) trigger the application to be accepted. The more sophisticated goals are harder to pull off, but also the most dangerous for Kermis and HireRing if successful. The problem specification should include provisions to be vigilant for all these different goals of attacks.

11.2 Defenses Against Poisoning Attacks

Once you and the HireRing team are in the modeling phase of the lifecycle, you have to implement defense measures against the attacks identified in the problem specification phase. From a machine learning perspective, there are no specific defenses for preventing logic corruption attacks in Kermis' systems. They must be prevented by other security measures. There are, however, defenses throughout the machine learning pipeline against data injection and data modification attacks that fall into three categories based on where in the pipeline they are applied.[2] (1) Pre-processing approaches are given the name *data sanitization*. (2) In-processing defenses during model training rely on some kind of *smoothing*. (3) Post-processing defenses are called *patching*. These three categories, which are detailed in the remainder of this section, are illustrated in Figure 11.3. They are analogous to methods for mitigating distribution shift and unwanted bias described in Chapter 9 and Chapter 10, respectively.

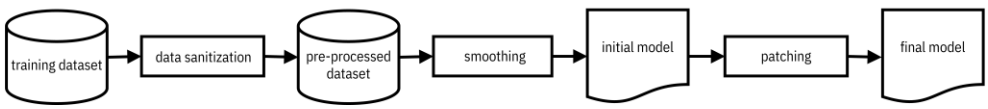


Figure 11.3. *Different categories of defenses against poisoning attacks in the machine learning pipeline.* Accessible caption. A block diagram with a training dataset as input to a data sanitization block with a pre-processed dataset as output. The pre-processed dataset is input to a smoothing block with an initial model as output. The initial model is input to a patching block with a final model as output.

Machine learning defenses against poisoning attacks are an active area of research. Specific attacks and defenses are continually improving in an arms race. Since by the time this book comes out, all presently known attacks and defenses are likely to have been superseded, only the main ideas are given rather than in-depth accounts.

11.2.1 Data Sanitization

The main idea of data sanitization is to locate the nefarious resumes that have been injected into or modified in the Kermis dataset and remove them. Such resumes tend to be anomalous in some fashion, so data sanitization reduces to a form of anomaly or outlier detection. The most common way to detect

[2]Micah Goldblum, Dmitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. "Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses." arXiv:2012.10544, 2021.

outliers, *robust statistics*, is as follows. The set of outliers is assumed to have small cardinality compared to the clean, unpoisoned training resumes. The two sets of resumes, poison and clean, are differentiated by having differing means normalized by their variances. Recent methods are able to differentiate the two sets efficiently even when the number of features is large.[3] For high-dimensional semi-structured data, the anomaly detection should be done in a representation space rather than in the original input feature space. Remember from Chapter 4 that learned representations and language models compactly represent images and text data, respectively, using the structure they contain. Anomalies are more apparent when the data is well-represented.

11.2.2 Smoothing

When HireRing is training the Kermis classifier, defenses against data poisoning make the model more robust by smoothing the score function. The general idea is illustrated in Figure 11.4, which compares a smooth and less smooth score function. By preferring smooth score functions during training, there is a lower chance for adversaries to succeed in their attacks.

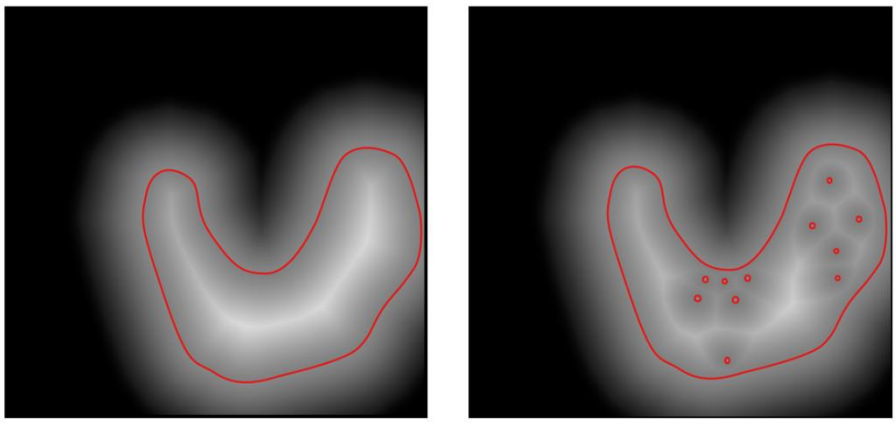


Figure 11.4. *A comparison of a smooth (left) and less smooth (right) score function. The value of the score function is indicated by shading: it is* 0 *where the shading is white and* 1 *where the shading is black. The decision boundary, where the score function takes value* 0.5 *is indicated by red lines. The less smooth score function may have been attacked.* Accessible caption. Stylized plot showing a decision boundary smoothly encircling one of the classes with a blob-like region. The underlying score function is indicated by shading, becoming smoothly whiter in the inside the region and smoothly blacker outside the region. This is contrasted with another decision boundary that has some tiny enclaves of the opposite class inside the blob-like region. Its underlying score function is not smooth.

Smoothing can be done, for example, by applying a k-nearest neighbor prediction on top of another underlying classifier. By doing so, the small number of poisoned resumes are never in the majority of a

[3]Pang Wei Koh, Jacob Steinhardt, and Percy Liang. "Stronger Data Poisoning Attacks Break Data Sanitization Defenses." In: *Machine Learning* (Nov. 2021).

neighborhood and their effect is ignored. Any little shifts in the decision boundary stemming from the poisoned data points are removed. Another way to end up with a smooth score function, known as *gradient shaping*, is by directly constraining or regularizing its slope or gradient within the learning algorithm. When the magnitude of the gradient of a decision function is almost the same throughout the feature space, it is resilient to perturbations caused by a small number of anomalous points: it is more like the left score function in Figure 11.4 than the right score function. Smoothing can also be accomplished by averaging together the score functions of several independent classifiers.

11.2.3 Patching

Patching, primarily intended for neural network models, mitigates the effect of backdoor attacks as a post-processing step. Backdoors show up as anomalous edge weights and node activations in neural networks. There is something statistically weird about them. Say that you have already trained an initial model on a poisoned set of Kermis job applications that has yielded a backdoor. The idea of the patching is similar to how you fix a tear in a pair of pants. First you 'cut' the problem out of the 'fabric': you prune the anomalous neural network nodes. Then you 'sew' a patch over it: you fine-tune the model with some clean resumes or a set of resumes generated to approximate a clean distribution.

11.3 Defenses Against Evasion Attacks

Evasion attacks are logistically simpler to carry out than poisoning attacks because they do not require the adversary to infiltrate Kermis' information technology systems. Adversaries only have to create examples that look realistic to avoid suspicion and submit them as regular job applications. Defending against these attacks can be done in two main ways: (1) denoising and (2) adversarial training. The first category of defenses is outside the machine learning training pipeline and applies at deployment. It tries to subtract off the perturbation δ from a deployment-time resume $x + \delta$ when it exists. (Only a small number of input resumes will be adversarial examples and have a δ.) This first category is known as *denoising*; the reason will become apparent in the next section. The second category of defenses occurs in the modeling pipeline and builds min-max robustness into the model itself, similar to training models robust to distribution shift in Chapter 9. It is known as *adversarial training*. There is no evasion defense category analogous to adaptation or bias mitigation pre-processing of training data from Chapter 9 and Chapter 10, respectively, because evasion attacks are not founded in training data distributions. The defenses to evasion attacks are summarized in Figure 11.5. Let's learn more about implementing these defenses in the HireRing job applicant prioritization system.

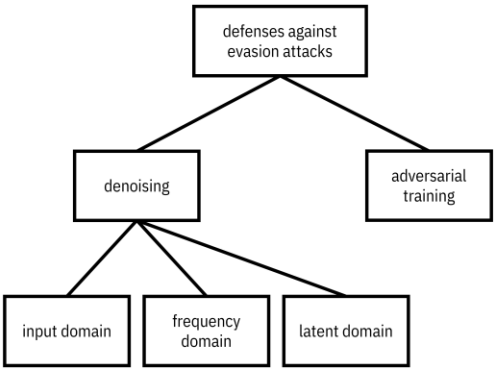


Figure 11.5. *Different defenses against evasion attacks.* Accessible caption. A hierarchy diagram with defenses against evasion attacks at its root, which has denoising and adversarial training as its children. Denoising has children input domain, frequency domain, and latent domain.

11.3.1 Denoising Input Data

Despite the best efforts of attackers, adversarial examples—resumes that have been shifted across a decision boundary—contain signals noticeable by machines even though they are imperceptible for people. The specially-crafted perturbation δ is a form of noise, called *adversarial noise*. Denoising, attempting to remove δ from $x + \delta$, is a type of defense. The challenge in denoising is to remove all of the noise while limiting the distortion to the underlying clean features.

Noise removal is an old problem that has been addressed in signal processing and related fields for a long time. There are three main ways of denoising evasion attacks that differ in their representation of the data: (1) input domain, (2) frequency domain, and (3) latent domain.[4] Denoising techniques working directly in the feature space or input domain may copy or swap feature values among neighboring data points. They may also quantize continuous values into a smaller set of discrete values. Taking advantage of recent advances in generative machine learning (briefly described in Chapter 4 in the context of data augmentation), they may generate data samples very similar to an input, but without noise. Taken together, the main idea for input domain denoising is to flatten the variability or smooth out the data values.

Smoothing is better examined in the frequency domain. If you are not an electrical engineer, you might not have heard about converting data into its frequency domain representation. The basic idea is to transform the data so that very wiggly data yields large values at so-called high frequencies and very smooth data yields large values at the opposite end of the spectrum: low frequencies. This conversion is done using the Fourier transform and other similar operations. Imperceptible adversarial noise is usually concentrated at high frequencies. Therefore, a defense for evasion attacks is squashing the high frequency components of the data (replacing them with small values) and then converting the data back to the input domain. Certain data compression techniques for semi-structured data modalities indirectly accomplish the same thing.

[4]Zhonghan Niu, Zhaoxi Chen, Linyi Li, Yubin Yang, Bo Li, and Jinfeng Yi. "On the Limitations of Denoising Strategies as Adversarial Defenses." arXiv:2012.09384, 2020.

If your machine learning model is a neural network, then the values of the data as it passes through intermediate layers constitute a latent representation. The third denoising category works in this latent representation space. Like in the frequency domain, this category also squashes the values in certain dimensions. However, these techniques do not simply assume that the adversarial noise is concentrated in a certain part of the space (e.g. high frequencies), but learn these dimensions using clean job applications and their corresponding adversarial examples that you create.

11.3.2 Adversarial Training

The second main category of defenses against evasion attacks is adversarial training. It is a form of min-max robustness,[5] which you first encountered in Chapter 9 in the context of distribution shift. Remember that the min-max idea is to do the best you can on the worst-case scenario. In adversarial training, the minimization and maximization are as follows. The minimization is to find the best job applicant classifier in the hypothesis space, just like any other risk minimization approach to machine learning. The inner maximization is to find the worst-case perturbations of the resumes. The mathematical form of the objective is:

$$\hat{y}(\cdot) = \arg\min_{f \in \mathcal{F}} \sum_{j=1}^{n} \max_{\|\delta_j\| \leq \epsilon} L\left(y_j, f\left(x_j + \delta_j\right)\right).$$

Equation 11.3

Notice that the inner maximization is the same expression as finding adversarial examples given in Equation 11.2. Thus, to carry out adversarial training, all you have to do is produce adversarial examples for the Kermis training resumes and use those adversarial examples as a new training data set in a typical machine learning algorithm. HireRing must become a good attacker to become a good defender.

11.3.3 Evaluating and Certifying Robustness to Evasion Attacks

Once the HireRing job applicant screening system has been adversarially trained on Kermis resumes, how do you know it is any good? There are two main ways to measure the model's robustness: (1) empirically and (2) characterizing the score function. As an empirical test, you create your own adversarial example resumes, feed them in, and compute how often the adversarial goal is met (confidence reduction, misclassification, targeted misclassification, or source/target misclassification). You can do it because you know the ground truth of which input resumes contain an adversarial perturbation and which ones don't. Such empirical robustness evaluation is tied to the specific attack and its capabilities (open-box or closed-box) since you as the evaluator are acting as the adversary.

[5]Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks." In: *Proceedings of the International Conference on Learning Representations*. Vancouver, Canada, Apr.–May 2018.

In contrast, a way to characterize the adversarial robustness of a classifier that is agnostic to the evasion attack is the CLEVER score.[6] An acronym for cross-Lipschitz extreme value for network robustness, the CLEVER score (indirectly) analyzes the distance from a job application data point to the classifier decision boundary. Misclassification attacks will be unsuccessful if this distance is too far because it will exceed ϵ, the bound on the norm of the perturbation δ. The higher the CLEVER score, the more robust the model. Generally speaking, smooth, non-complicated decision boundaries without many small islands (like the left score function in Figure 11.4) have large distances from data points on average, have large average CLEVER scores, and are robust to all kinds of evasion attacks. In the problem specification phase with the Kermis problem owners, you can set an acceptable minimum value for the average CLEVER score. If the model achieves it, HireRing can confidently certify a level of security and robustness.

11.4 Summary

- Adversaries are actors with bad intentions who try to attack machine learning models by degrading their accuracy or fooling them.
- Poisoning attacks are implemented during model training by corrupting either the training data or model.
- Evasion attacks are implemented during model deployment by creating adversarial examples that appear genuine, but fool models into making misclassifications.
- Adversaries may just want to worsen model accuracy in general or may have targeted goals that they want to achieve, such as obtaining specific predicted labels for specific inputs.
- Adversaries have different capabilities of what they know and what they can change. These differences in capabilities and goals determine the threat.
- Defenses for poisoning attacks take place at different parts of the machine learning pipeline: data sanitization (pre-processing), smoothing (model training), and patching (post-processing).
- Defenses for evasion attacks include denoising that attempts to remove adversarial perturbations from inputs and adversarial training which induces min-max robustness.
- Models can be certified for robustness to evasion attacks using the CLEVER score.
- Even without malicious actors, adversarial attacks are a way for developers to test machine learning systems in worst case scenarios.

[6]Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. “Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach.” In: *Proceedings of the International Conference on Learning Representations*. Vancouver, Canada, Apr.–May 2018.

12

Interpretability and Explainability

Hilo is a (fictional) startup company trying to shake up the online second home mortgage market. A type of second mortgage known as a home equity line of credit (HELOC) allows customers to borrow intermittently using their house as collateral. Hilo is creating several unique propositions to differentiate itself from other companies in the space. The first is that it integrates the different functions involved in executing a second mortgage, including a credit check of the borrower and an appraisal of the value of the home, in one system. Second, its use of machine learning throughout these human decision-making processes is coupled with a maniacal focus on robustness to distribution shift, fairness, and adversarial robustness. Third, it has promised to be scrutable to anyone who would like to examine the machine learning models it will use and to provide avenues for recourse if the machine's decisions are problematic in any respect. Imagine that you are on the data science team assembled by Hilo and have been tasked with addressing the third proposition by making the machine learning models *interpretable* and *explainable*. The platform's launch date is only a few months away, so you had better get cracking.

Interpretability of machine learning models is the aim to let people understand how the machine makes its predictions. It is a challenge because many of the machine learning approaches in Chapter 7 are not easy for people to understand since they have complicated functional forms. Interpretability and explainability are a form of *interaction* between the machine and a human, specifically *communication* from the machine to the human, that allow the machine and human to collaborate in decision making.[1] This topic and chapter lead off Part 5 of the book on interaction, which is the third attribute of trustworthiness of machine learning. Remember that the organization of the book matches the attributes of trustworthiness, shown in Figure 12.1.

[1]Ben Green and Yiling Chen. "The Principles and Limits of Algorithm-in-the-Loop Decision Making." In: *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing*. Austin, Texas, USA, Nov. 2019, p. 50.

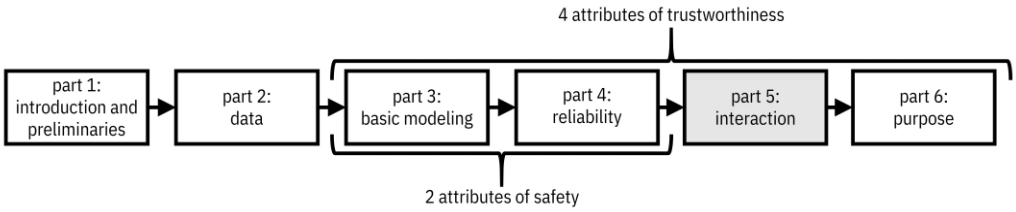


Figure 12.1. *Organization of the book. The fifth part focuses on the third attribute of trustworthiness, intimacy or interaction, which maps to machine learning models that can communicate with people and receive instruction from people about their values.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 5 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

The typical output of a machine learning model is the predicted label $\hat{Y}$, but this label is not enough to communicate how the machine makes its predictions. Something more, in the form of an explanation, is also needed. The machine is the transmitter of information and the human is the receiver or *consumer* of that information. As shown in Figure 12.2, the communication process has to overcome human cognitive biases—the limitations that people have in receiving information—that threaten human-machine collaboration. This is sometimes known as the last mile problem.[2] The figure completes the picture of biases and validities you've seen starting in Chapter 4. The final space is the *perceived space*, which is the final understanding that the human consumer has of the predictions from Hilo's machine learning models.

You will not be able to create a single kind of explanation that appeals to all of the different potential consumers of explanations for Hilo's models. Even though the launch date is only a few months away, don't take the shortcut of assuming that any old explanation will do. The cognitive biases of different people are different based on their persona, background, and purpose. As part of the problem specification phase of the machine learning lifecycle, you'll first have to consider all the different types of explanations at your disposal before going into more depth on any of them during the modeling phase.

12.1 The Different Types of Explanations

Just like we as people have many ways to explain things to each other, there are many ways for machine learning models to explain their predictions to consumers. As you consider which ones you'll need for Hilo's models, you should start by enumerating the personas of consumers.

[2]James Guszcza. "The Last-Mile Problem: How Data Science and Behavioral Science Can Work Together." In: *Deloitte Review* 16 (2015), pp. 64–79.

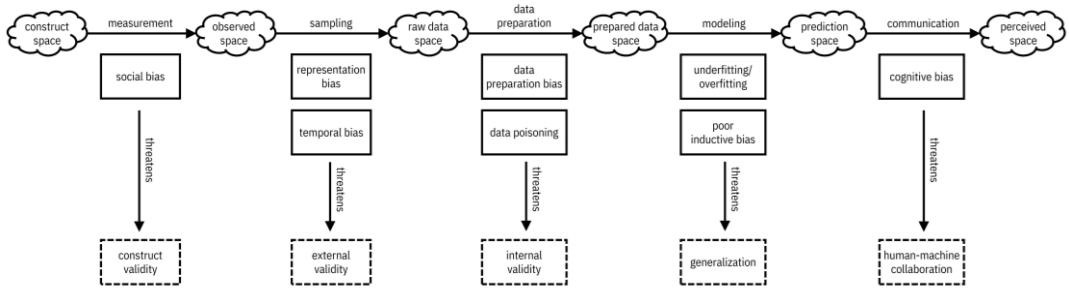


Figure 12.2. *A mental model of spaces, validities, and biases. The final space is the perceived space, which is what the human understands from the machine's output.* Accessible caption. A sequence of six spaces, each represented as a cloud. The construct space leads to the observed space via the measurement process. The observed space leads to the raw data space via the sampling process. The raw data space leads to the prepared data space via the data preparation process. The prepared data space leads to the prediction space via the modeling process. The prediction space leads to the perceived space via the communication process. The measurement process contains social bias, which threatens construct validity. The sampling process contains representation bias and temporal bias, which threatens external validity. The data preparation process contains data preparation bias and data poisoning, which threaten internal validity. The modeling process contains underfitting/overfitting and poor inductive bias, which threaten generalization. The communication process contains cognitive bias, which threatens human-machine collaboration.

12.1.1 Personas of the Consumers of Explanations

The first consumer is the *decision maker* who collaborates with the machine learning system to make the prediction: the appraiser or credit officer. These consumers need to understand and trust the model and have enough information about machine predictions to combine with their own inclinations to produce the final decision. The second consumer persona is the HELOC applicant. This *affected user* would like to know the factors that led to their model-predicted home appraisal and creditworthiness, and what they can do to improve these predictions. The third main persona of consumers is an internal compliance official or model validator, or an official from an external regulatory agency that ensures that the decisions are not crossing any legal boundaries. Together, all of these roles are *regulators* of some sort. The fourth possible consumer of explanations is a data scientist in your own team at Hilo. Explanations of the functioning of the models can help a member of your team debug and improve the models.

> "If we don't know what is happening in the black box, we can't fix its mistakes to make a better model and a better world."
>
> —Aparna Dhinakaran, chief product officer at Arize AI

Note that unlike the other three personas, the primary concern of the data scientist persona is not building interaction and intimacy for trustworthiness. The four different personas and their goals are summarized in Table 12.1.

Table 12.1. *The four main personas of consumers of explanations and their goals.*

Persona	Example	Goal
decision maker	appraiser, credit officer	(1) roughly understand the model to gain trust; (2) understand the predictions to combine with their own information to make decisions
affected user	HELOC applicant	understand the prediction for their own input data point and what they can do to change the outcome
regulator	model validator, government official	ensure the model is safe and compliant
data scientist	Hilo team member	improve the model's performance

12.1.2 Dichotomies of Explanation Methods

To meet the goals of the different personas, one kind of explanation is not enough.[3] You'll need several different explanation types for Hilo's systems. There are three dichotomies that delineate the methods and techniques for machine learning explainability.

- The first dichotomy is *local* vs. *global*: is the consumer interested in understanding the machine predictions for individual input data points or in understanding the model overall.
- The second dichotomy is *exact vs. approximate*: should the explanation be completely faithful to the underlying model or is some level of approximation allowable.
- The third dichotomy is *feature-based vs. sample-based*: is the explanation given as a statement about the features or is it given by pointing to other data points in their entirety. Feature-based explanations require that the underlying features be meaningful and understandable by the consumer. If they are not already meaningful, a pre-processing step known as *disentangled representation* may be required. This pre-processing finds directions of variation in semi-structured data that are not necessarily aligned to the given features but have some human interpretation, and is expanded upon in Section 12.2.

Since there are three dichotomies, there are eight possible combinations of explanation types. Certain types of explanations are more appropriate for certain personas to meet their goals. The fourth persona, data scientists from your own team at Hilo, may need to use any and all of the types of explanations to debug and improve the model.

- Local, exact, feature-based explanations help affected users such as HELOC applicants gain recourse and understand precisely which feature values they have to change in order to pass the credit check.
- Global and local approximate explanations help decision makers such as appraisers and credit

[3]Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. "One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques." arXiv:1909.03012, 2019. Q. Vera Liao and Kush R. Varshney. "Human-Centered Explainable AI (XAI): From Algorithms to User Experiences." arXiv:2110.10790, 2021.

officers achieve their dual goals of roughly understanding how the overall model works to develop trust in it (global) and having enough information about a machine-predicted property value to combine with their own information to produce a final appraisal (local).

- Global and local, exact, sample-based explanations and global, exact, feature-based explanations help regulators understand the behavior and predictions of the model as a safeguard. By being exact, the explanations apply to all data points, including edge cases that might be washed out in approximate explanations. Of these, the local, exact, sample-based and global, exact, feature-based explanations that appeal to regulators come from *directly interpretable models*.
- Regulators and decision makers can both benefit from global, approximate, sample-based explanations to gain understanding.

The mapping of explanation types to personas is summarized in Table 12.2.

Table 12.2. *The three dichotomies of explanations and their mapping to personas.*

Dichotomy 1	Dichotomy 2	Dichotomy 3	Persona	Example Method
local	exact	feature-based	affected user	contrastive explanations method
local	exact	sample-based	regulator	k-nearest neighbor
local	approximate	feature-based	decision maker	LIME, SHAP, saliency map
local	approximate	sample-based	decision maker	prototype
global	exact	feature-based	regulator	decision tree, Boolean rule set, logistic regression, GAM, GLRM
global	exact	sample-based	regulator	deletion diagnostics
global	approximate	feature-based	decision maker	distillation, SRatio, partial dependence plot
global	approximate	sample-based	regulator and decision maker	influence function

Another dichotomy that you might consider in the problem specification phase is whether you will allow the explanation consumer to *interactively* probe the Hilo machine learning system to gain further insight, or whether the system will simply produce *static* output explanations that the consumer cannot further interact with. The interaction can be through natural language dialogue between the consumer and the machine, or it could be by means of visualizations that the consumer adjusts and drills down into.[4] The variety of static explanations is already plenty for you to deal with without delving into interaction, so you decide to proceed only with static methods.

[4]Josua Krause, Adam Perer, and Kenney Ng. "Interacting with Predictions: Visual Inspection of Black-Box Machine Learning Models." In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. San Jose, California, USA, May 2016, pp. 5686–5697.

Mirroring the three points of intervention in the modeling pipeline seen in Part 4 of the book for distributional robustness, fairness, and adversarial robustness, Figure 12.3 shows different actions for interpretability and explainability. As mentioned earlier, disentangled representation is a pre-processing step. Directly interpretable models arise from training decision functions in specific constrained hypothesis classes (recall that the concept of hypothesis classes was introduced in Chapter 7). Finally, many methods of explanation are applied on top of already-trained, non-interpretable models such as neural networks in a *post hoc* manner.

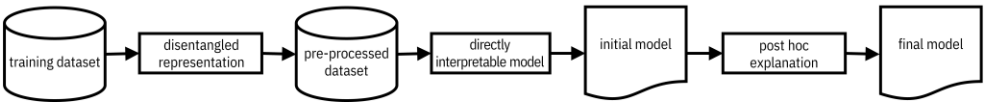


Figure 12.3. *Pipeline view of explanation methods.* Accessible caption. A block diagram with a training dataset as input to a disentangled representation block with a pre-processed dataset as output. The pre-processed dataset is input to a directly interpretable model block with an initial model as output. The initial model is input to a post hoc explanation block with a final model as output.

12.1.3 Conclusion

Now you have the big picture view of different explanation methods, how they help consumers meet their goals, and how they fit into the machine learning pipeline steps. The appraisal and HELOC approval systems you're developing for Hilo require you to appeal to all of the different consumer types, and you have the ability to intervene on all parts of the pipeline, so you should start putting together a comprehensive toolkit of interpretable and explainable machine learning techniques.

12.2 Disentangled Representation

Before people can start understanding how models make their predictions, they need some understanding of the underlying data. Features in tabular and other structured data used as inputs to machine learning models can usually be understood by consumers in some capacity. Consumers who are not decision makers, regulators, or other domain experts (or even if they are) might not grasp the nuance of every feature, but they can at least consult a data dictionary to get some understanding of each one. For example, in the HELOC approval model, a feature 'months since most recent delinquency' might not make total sense to applicants, but it is something they can understand if they do some research about it.

The same is not true of semi-structured data. For example, inputs to the home appraisal model include satellite and street view images of the property and surrounding neighborhood. The features are individual red, blue, and green color values for each pixel of the image. Those features are not meaningful to any consumer. They are void of semantics. Higher-level representations, for example edges and textures that are automatically learned by neural networks, are a little better but still leave an explanation consumer wanting. They do not directly have a meaning in the context of a home appraisal.

What can be done instead? The answer is a representation in which the dimensions are the amount of foliage in the neighborhood, the amount of empty street frontage, the visual desirability of the house,

etc.[5] that are uncorrelated with each other and also provide information not captured in other input data. (For example, even though the size and number of floors of the home could be estimated from images, it will already be captured in other tabular data.) Such a representation is known as a disentangled representation. The word *disentangled* is used because in such a representation, intervening on one dimension does not cause other dimensions to also change. Recently developed methods can learn disentangled representations directly from unlabeled data.[6] Although usually not the direct objective of disentangling, such representations tend to yield meaningful dimensions that people can provide semantics to, such as the example of 'foliage in the neighborhood' mentioned above. Therefore, disentangled representation is a way of pre-processing the training data features to make them more human-interpretable. Modeling and explanation methods later in the pipeline take the new features as input.

Sometimes, disentangled representation to improve the features is not good enough to provide meaning to consumers. Similarly, sometimes tabular data features are just not sufficient to provide meaning to a consumer. In these cases, an alternative pre-processing step is to directly elicit meaningful explanations from consumers, append them to the dataset as an expanded cardinality label set, and train a model to predict both the original appraisal or creditworthiness as well as the explanation.[7]

12.3 Explanations for Regulators

Directly interpretable models are simple enough for consumers to be able to understand *exactly* how they work by glancing at their form. They are appropriate for regulators aiming for model safety. They are a way to reduce epistemic uncertainty and achieve *inherently safe design*: models that do not have any spurious components.[8] The explanation is done by restricting the hypothesis class from which the decision function is drawn to only those functions that are simple and understandable. There are two varieties of directly interpretable exact models: (1) local sample-based and (2) global feature-based. Moreover, model understanding by regulators is enhanced by global sample-based explanations, both exact and approximate.

12.3.1 k-Nearest Neighbor Classifier

The k-nearest neighbor classifier introduced in Chapter 7 is the main example of a local sample-based directly interpretable model. The predicted creditworthiness or appraisal label is computed as the average label of nearby training data points. Thus, a local explanation for a given input data point is just the list of the k-nearest neighbor samples, including their labels. This list is simple enough for regulators to understand. You can also provide the distance metric for additional understanding.

---

[5]Stephen Law, Brooks Paige, and Chris Russell. "Take a Look Around: Using Street View and Satellite Images to Estimate House Prices." In: *ACM Transactions on Intelligent Systems and Technology* 10.5 (Nov. 2019), p. 54.

[6]Xinqi Zhu, Chang Xu, and Dacheng Tao. "Where and What? Examining Interpretable Disentangled Representations." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Jun. 2021, pp. 5857–5866.

[7]Michael Hind, Dennis Wei, Murray Campbell, Noel C. F. Codella, Amit Dhurandhar, Aleksandra Mojsilović, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. "TED: Teaching AI to Explain its Decisions." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. Honolulu, Hawaii, USA, Jan. 2019, pp. 123–129.

[8]Kush R. Varshney and Homa Alemzadeh. "On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products." In: *Big Data* 5.3 (Sep. 2017), pp. 246–255.

12.3.2 Decision Trees and Boolean Rule Sets

There is more variety in global feature-based directly interpretable models. Decision trees, introduced in Chapter 7, can be understood by regulators by tracing paths from the root through intermediate nodes to leaves containing predicted labels.[9] The individual features and thresholds involved in each node are explicit and well-understood. A hypothesis class similar to decision trees is Boolean rule sets (OR-of-AND rules and AND-of-OR rules) that people are able to comprehend directly. They are combinations of decision stumps or one-rules introduced in Chapter 7. An example of an OR-of-AND rule classifier for HELOC creditworthiness is one that predicts the applicant to be non-creditworthy if:[10]

- (Number of Satisfactory Trades ≤ 17 AND External Risk Estimate ≤ 75) OR
- (Number of Satisfactory Trades > 17 AND External Risk Estimate ≤ 72).

This is a very compact rule set in which regulators can easily see the features involved and their thresholds. They can reason that the model is more lenient on external risk when the number of satisfactory trades is higher. They can also reason that the model does not include any objectionable features. (Once decision trees or Boolean rule sets become too large, they start becoming less interpretable.)

One common refrain that you might hear is of a tradeoff between accuracy and interpretability. This argument is false.[11] Due to the Rashomon effect introduced in Chapter 9, many kinds of models, including decision trees and rule sets, have almost equally high accuracy on many datasets. The domain of competence for decision trees and rule sets is broad (recall that the domain of competence introduced in Chapter 7 is the set of dataset characteristics on which a type of model performs well compared to other models). While it is true that scalably training these models has traditionally been challenging due to their discrete nature (discrete optimization is typically more difficult than continuous optimization), the challenges have recently been overcome.[12]

> "Simplicity is not so simple."
>
> —Dmitry Malioutov, computer scientist at IBM Research

When trained using advanced discrete optimization, decision trees and Boolean rule set classifiers show competitive accuracies across many datasets.

---

[9]It is important to note that interpretability is about consumers understanding *how* the model makes its predictions, but not necessarily *why*. Consumers can supplement the *how* with the *why* based on their common-sense knowledge.

[10]The example HELOC explanations throughout the chapter are based on the tutorial https://github.com/Trusted-AI/AIX360/blob/master/examples/tutorials/HELOC.ipynb and demonstration http://aix360.mybluemix.net/data developed by Vijay Arya, Amit Dhurandhar, Q. Vera Liao, Ronny Luss, Dennis Wei, and Yunfeng Zhang.

[11]Cynthia Rudin. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." In: *Nature Machine Intelligence* 1.5 (May 2019), pp. 206–215.

[12]Oktay Günlük, Jayant Kalagnanam, Minhan Li, Matt Menickelly, and Katya Scheinberg. "Optimal Generalized Decision Trees via Integer Programming." arXiv:1612.03225, 2019. Sanjeeb Dash, Oktay Günlük, and Dennis Wei. "Boolean Decision Rules via Column Generation." In: *Advances in Neural Information Processing Systems* 31 (Dec. 2018), pp. 4655–4665.

12.3.3 Logistic Regression

Linear logistic regression is also considered by many regulators to be directly interpretable. Recall from Chapter 7 that the form of the linear logistic regression decision function is $\hat{y}(x) = \text{step}(w^T x)$. The d-dimensional weight vector w has one weight per feature dimension in x, which are different attributes of the HELOC applicant. These weight and feature dimensions are $w^{(1)}, \ldots, w^{(d)}$ and $x^{(1)}, \ldots, x^{(d)}$, respectively, which get multiplied and summed as: $w^{(1)}x^{(1)} + \cdots + w^{(d)}x^{(d)}$ before going into the step function. In logistic regression, the relationship between the probability $P\big(\hat{Y} = 1 \mid X = x\big)$ (also the score s) and the weighted sum of feature dimensions is:

$$P\big(\hat{Y} = 1 \mid X = x\big) = \frac{1}{1 + e^{-\left(w^{(1)}x^{(1)} + \cdots + w^{(d)}x^{(d)}\right)}},$$

Equation 12.1

which you've seen before as the logistic activation function for neural networks in Chapter 7. It can be rearranged to the following:

$$\log\left(\frac{P\big(\hat{Y} = 1 \mid X = x\big)}{1 - P\big(\hat{Y} = 1 \mid X = x\big)}\right) = w^{(1)}x^{(1)} + \cdots + w^{(d)}x^{(d)}.$$

Equation 12.2

The left side of Equation 12.2 is called the *log-odds*. When the log-odds is positive, $\hat{Y} = 1$ is the more likely prediction: creditworthy. When the log-odds is negative, $\hat{Y} = 0$ is the more likely prediction: non-creditworthy.

The way to understand the behavior of the classifier is by examining how the probability, the score, or the log-odds change when you increase an individual feature attribute's value by 1. Examining the response to changes is a general strategy for explanation that recurs throughout the chapter. In the case of linear logistic regression, an increase of feature value $x^{(i)}$ by 1 while leaving all other feature values constant adds $w^{(i)}$ to the log-odds. The weight value has a clear effect on the score. The most important features per unit change of feature values are those with the largest absolute values of the weights. To more easily compare feature importance using the weights, you should first standardize each of the features to zero mean and unit standard deviation. (Remember that standardization was first introduced when evaluating the covariate balancing of causal models in Chapter 8.)

12.3.4 Generalized Additive Models

Generalized additive models (GAMs) are a class of models that extend linear logistic regression to be nonlinear while retaining the same approach for interpretation. Instead of scalar weights multiplying feature values in the decision function for credit check prediction: $w^{(1)}x^{(1)} + \cdots + w^{(d)}x^{(d)}$, nonlinear functions are applied: $f^{(1)}\big(x^{(1)}\big) + \cdots + f^{(d)}\big(x^{(d)}\big)$. The entire function for a given feature dimension explicitly adds to the log-odds or subtracts from it. You can only do this exactly because there is no interaction between the feature dimensions. You can choose any hypothesis class for the nonlinear functions, but be aware that the learning algorithm has to fit the parameters of the functions from

training data. Usually, smooth spline functions are chosen. (A spline is a function made up of piecewise polynomials strung together.)

12.3.5 Generalized Linear Rule Models

What if you want the regulators to have an easy time understanding the nonlinear functions involved in the HELOC decision function themselves? You can choose the nonlinear functions to be Boolean one-rules or decision stumps involving single features. The *generalized linear rule model* (GLRM) is exactly what you need: a directly interpretable method that combines the best of Boolean rule sets and GAMs.[13] In addition to Boolean one-rules of feature dimensions, the GLRM can have plain feature dimensions too. An example GLRM for HELOC credit checks is shown in Table 12.3.

Table 12.3. *An example generalized linear rule model for HELOC credit checks.*

Plain Feature or First-Degree Boolean Rule	Weight
Months Since Most Recent Inquiry[14] > 0	0.680261
Months Since Most Recent Inquiry = 0	−0.090058
(Standardized) External Risk Estimate	0.654248
External Risk Estimate > 75	0.263437
External Risk Estimate > 72	0.107613
External Risk Estimate > 69	0.035422
(Standardized) Revolving Balance Divided by Credit Limit	−0.553965
Revolving Balance Divided by Credit Limit ≤ 39	0.062797
Revolving Balance Divided by Credit Limit ≤ 50	0.045612
(Standardized) Number of Satisfactory Trades	0.551654
Number of Satisfactory Trades ≤ 12	−0.312471
Number of Satisfactory Trades ≤ 17	−0.110220

The three plain features ('external risk estimate', 'revolving balance divided by credit limit', and 'number of satisfactory trades') were standardized before doing anything else, so you can compare the weight values to see which features are important. The decision stump of 'months since most recent inquiry' being greater than zero is the most important because it has the largest coefficient. The decision stump of 'external risk estimate' being greater than 69 is the least important because it has the smallest coefficient. This is the same kind of understanding that you would apply to a linear logistic regression model.

The way to further understand this model is by remembering that the weight contributes to the log-odds for every unit change of the feature. Taking the 'external risk estimate' feature as an example, the GLRM tells you that:

- for every increase of External Risk Estimate by 1, increase the log-odds by 0.0266 (this number is obtained by undoing the standardization on the weight 0.6542);

[13]Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Günlük. "Generalized Linear Rule Models." In: *Proceedings of the International Conference on Machine Learning*. Long Beach, California, USA, Jul. 2019, pp. 6687–6696.

[14]This feature excludes inquiries made in the last 7 days to remove inquiries that are likely due to price comparison shopping.

- if External Risk Estimate > 69, increase log-odds by an additional 0.0354;
- if External Risk Estimate > 72, increase log-odds by an additional 0.1076;
- if External Risk Estimate > 75, increase log-odds by an additional 0.2634.

The rule is fairly straightforward for consumers such as regulators to understand while being an expressive model for generalization. As shown in Figure 12.4, you can plot the contributions of the 'external risk estimate' feature to the log-odds to visually see how the Hilo classifier depends on it. Plots of $f^{(i)}(x^{(i)})$ for other GAMs look similar, but can be nonlinear in different ways.

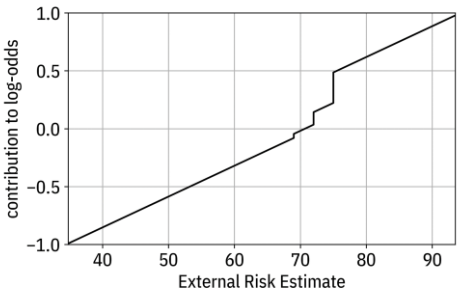


Figure 12.4. *Contribution of the 'external risk estimate' feature to the log-odds of the classifier.* Accessible caption. A plot with contribution to log-odds on the vertical axis and external risk estimate on the horizontal axis. The contribution to log-odds function increases linearly with three jump discontinuities.

You can also 'undo' the log-odds to get the actual probability (Equation 12.1 instead of Equation 12.2), but it is not additive like the log-odds. Nevertheless, the shape of the probability curve is informative in the same way, and is shown in Figure 12.5.

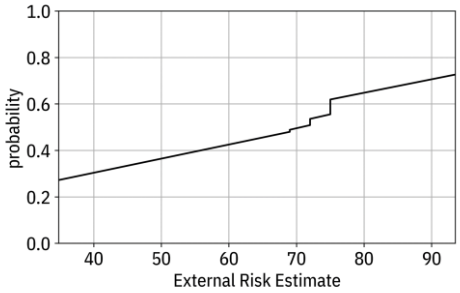


Figure 12.5. *Contribution of the 'external risk estimate' feature to the probability of the classifier.* Accessible caption. A plot with probability on the vertical axis and external risk estimate on the horizontal axis. The probability function increases linearly with three jump discontinuities.

GA^{2}Ms, equivalently known as *explainable boosting machines*, are directly interpretable models that work the same as GAMs, but with two-dimensional nonlinear interaction terms $f^{(i,i')}\left(x^{(i)}, x^{(i')}\right)$.[15] Visually showing their contribution to the log-odds of the classifier requires two-dimensional plots. It is generally difficult for people to understand interactions involving more than two dimensions and therefore higher-order GA^{2}Ms are not used in practice.[16] However, if you allow higher-degree rules in GLRMs, you end up with GA^{2}Ms of AND-rules or OR-rules involving multiple interacting feature dimensions that unlike general higher-order GA^{2}Ms, are still directly interpretable because rules involving many features can be understood by consumers.

12.3.6 Deletion Diagnostics and Influence Functions

The final set of methods that appeal to the regulator persona are from the global sample-based category. An exact method computes *deletion diagnostics* to find *influential* instances and an approximate method uses *influence functions* to do the same. The basic idea of deletion diagnostics is simple. You train a model with the entire training dataset of houses or applicants and then train it again leaving out one of the training samples. Whatever global changes there are to the model can be attributed to the house or applicant that you left out. How do you look at what changed between the two models? You can directly look at the two models or their parameters, which makes sense if the models are interpretable. But that won't work if you have an uninterpretable model. What you need to do is evaluate the two models on a held-out test set and compute the average change in the predicted labels. The bigger the change, the more influential the training data point. The regulator gains an understanding of the model by being given a list of the most influential homes or applicants.

Exactly computing deletion diagnostics is expensive because you have to train $n+1$ different models, leaving one training point out each time plus the model trained on all the data points. So usually, you'll want to approximate the calculation of the most influential training samples. Let's see how this approximation is done for machine learning algorithms that have smooth loss functions using the method of influence functions (refer back to Chapter 7 for an introduction to loss functions).[17] Influence function explanations are also useful for decision makers.

The method for computing the influence of a certain training data point x_j on a held-out test data point x_{test} starts by approximating the loss function by quadratic functions around each training data point. The gradient vector ∇L (slope or set of first partial derivatives) and Hessian matrix $\nabla^2 L$ (local curvature or set of second partial derivatives) of the quadratic approximations to the loss function with respect to the model's parameters are then calculated as closed-form formulas. The average of the

---

[15]Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. "Accurate Intelligible Models with Pairwise Interactions." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Chicago, Illinois, USA, Aug. 2013, pp. 623–631.

[16]A recently developed neural network architecture has full interactions between dimensions, but can still be decoded into the effects of individual features using very special properties of *continued fractions*, based on which the architecture is designed. A continued fraction is a representation of a number as the sum of its integer part and the reciprocal of another number; this other number is represented as the sum of its integer part and the reciprocal of another number; and so on. Isha Puri, Amit Dhurandhar, Tejaswini Pedapati, Karthikeyan Shanmugam, Dennis Wei, and Kush R. Varshney. "CoFrNets: Interpretable Neural Architecture Inspired by Continued Fractions." In: *Advances in Neural Information Processing Systems* 34 (Dec. 2021).

[17]Pang Wei Koh and Percy Liang. "Understanding Black-Box Predictions via Influence Functions." In: *Proceedings of the International Conference on Machine Learning*. Sydney, Australia, Aug. 2017, pp. 1885–1894.

Hessian matrices across all the training data points is also computed and denoted by H. Then the influence of sample x_j on x_{test} is $-\nabla L\big(y_{test}, \hat{y}(x_{test})\big)^T H^{-1} \nabla L\Big(y_j, \hat{y}\big(x_j\big)\Big)$.

The expression takes this form because of the following reason. First, the $-H^{-1}\nabla L\Big(y_j, \hat{y}\big(x_j\big)\Big)$ part of the expression is a step in the direction toward the minimum of the loss function at x_j (for those who have heard about it before, this is the Newton direction). Taking a step toward the minimum affects the model parameters just like deleting x_j from the training dataset, which is what the deletion diagnostics method does explicitly. The expression involves both the slope and the local curvature because the steepest direction indicated by the slope is bent towards the minimum of quadratic functions by the Hessian. Second, the $\nabla L\big(y_{test}, \hat{y}(x_{test})\big)$ part of the expression maps the overall influence of x_j to the x_{test} sample. Once you have all the influence values for a set of held-out test houses or applicants, you can average, rank, and present them to the regulator to gain global understanding about the model.

12.4 Explanations for Decision Makers

Decision trees, Boolean rule sets, logistic regression, GAMs, GLRMs, and other similar hypothesis classes are directly interpretable through their features because of their relatively simple form. However, there are many instances in which you want to or have to use a complicated uninterpretable model. (Examples of uninterpretable models include deep neural networks as well as decision forests and other similar ensembles that you learned in Chapter 7.) Nevertheless, in these instances, you want the decision maker persona to have a model-level global understanding of how the Hilo model works. What are the ways in which you can create approximate global explanations to meet this need? (Approximation is a must. If a consumer could understand the complicated model without approximation, it would be directly interpretable already.) There are two ways to approach global approximate feature-based explanations: (1) training a directly interpretable model like a decision tree, rule set, or GAM to be similar to the uninterpretable model, or (2) computing global summaries of the uninterpretable model that are understandable. In both cases, you first fit the complicated uninterpretable model using the training data set.

In addition to having a general model-level understanding to develop trust, approximate explanations at the local level help the appraiser or credit officer understand the predictions to combine with their own information to make decisions. The local feature-based explanation methods LIME and SHAP extend each of the two global feature-based explanation methods to the local level, respectively. A third local feature-based explanation method useful to appraisers and usually applied to semi-structured data modalities is known as saliency maps. Finally, local approximate sample-based explanations based on comparisons to prototypical data points help appraisers and credit officers make their final decisions as well. All of these methods are elaborated upon in this section.

12.4.1 Global Model Approximation

Global model approximation is the idea of finding a directly interpretable model that is close to a complicated uninterpretable model. It has two sub-approaches. The first, known as *distillation*, changes

the learning objective of the directly interpretable model from the standard risk minimization objective to an objective of matching the uninterpretable model as closely as possible.[18]

The second sub-approach for approximation using directly interpretable models, known as *SRatio*, computes training data weights based on the uninterpretable model and interpretable model. Then it trains the directly interpretable model with the instance weights.[19] You've seen reweighing of data points repeatedly in the book: inverse probability weighting for causal inference, confusion matrix-based weights to adapt to prior probability shift, importance weights to adapt to covariate shift, and reweighing as a pre-processing bias mitigation algorithm. The general idea here is the same, and is almost a reversal of importance weights for covariate shift.

Remember from Chapter 9 that in covariate shift settings, the training and deployment feature distributions are different, but the labels given the features are the same: $p_X^{(train)}(x) \neq p_X^{(deploy)}(x)$ and $p_{Y|X}^{(train)}(y \mid x) = p_{Y|X}^{(deploy)}(y \mid x)$. The importance weights are then: $w_j = p_X^{(deploy)}(x_j)/p_X^{(train)}(x_j)$. For explanation, there is no separate training and deployment distribution; there is an uninterpretable and an interpretable model. Also, since you're explaining the prediction process, not the data generating process, you care about the predicted label $\hat{Y}$ instead of the true label Y. The feature distributions are the same because you train the uninterpretable and interpretable models on the same training data houses or applicants, but the predicted labels given the features are different since you're using different models: $p_X^{(\text{interp})}(x) = p_X^{(uninterp)}(x)$ and $p_{\hat{Y}|X}^{(interp)}(\hat{y} \mid x) \neq p_{\hat{Y}|X}^{(uninterp)}(\hat{y} \mid x)$.

So following the same pattern as adapting to covariate shift by computing the ratio of the probabilities that are different, the weights are: $w_j = p_{\hat{Y}|X}^{(uninterp)}(\hat{y} \mid x)/p_{\hat{Y}|X}^{(interp)}(\hat{y} \mid x)$. You want the interpretable model to look like the uninterpretable model. In the weight expression, the numerator comes from the classifier score of the trained uninterpretable model and the denominator comes from the score of the directly interpretable model trained without weights.

12.4.2 LIME

Global feature-based explanation using model approximation has an extension to the local explanation case known as *local interpretable model-agnostic explanations* (LIME). The idea is similar to the global method from the previous subsection. First you train an uninterpretable model and then you approximate it by fitting a simple interpretable model to it. The difference is that you do this approximation around each deployment data point separately rather than trying to come up with one overall approximate model.

To do so, you get the uninterpretable model's prediction on the deployment data point you care about, but you don't stop there. You add a small amount of noise to the deployment data point's features several times to create a slew of perturbed input samples and classify each one. You then use this new set of data points to train the directly interpretable model. The directly interpretable model is a local approximation because it is based only on a single deployment data point and a set of other data points created around it. The interpretable model can be a logistic regression or decision tree and is simply shown to the decision maker, the Hilo appraiser or credit officer.

---

[18]Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. "Learning Global Additive Explanations for Neural Nets Using Model Distillation." arXiv:1801.08640, 2018.

[19]Amit Dhurandhar, Karthikeyan Shanmugam, and Ronny Luss. "Enhancing Simple Models by Exploiting What They Already Know." In: *Proceedings of the International Conference on Machine Learning*. Jul. 2020, pp. 2525–2534.

12.4.3 Partial Dependence Plots

The second global approach for increasing the trust of the appraisers and credit officers is the approximate feature-based explanation method known as *partial dependence plots*. The main idea is simple: compute and plot the classifier probability as a function of each of the feature dimensions $X^{(i)}$ separately, that is $P\left(\hat{Y} = 1 \mid X^{(i)} = x^{(i)}\right)$. You know exactly how to compute this partial dependence function from Chapter 3 by integrating or summing the probability $P\left(\hat{Y} = 1 \mid X = x\right)$ over all the feature dimensions except dimension i, also known as marginalization. An example partial dependence plot for the 'external risk estimate' feature is shown in Figure 12.6.

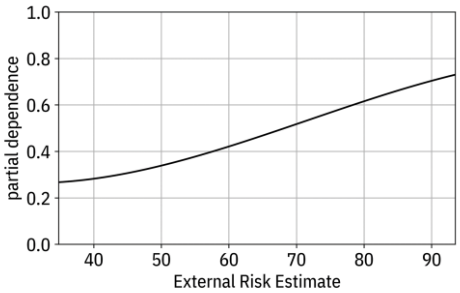


Figure 12.6. *Partial dependence plot of the 'external risk estimate' feature for some uninterpretable classifier model.* Accessible caption. A plot with partial dependence on the vertical axis and external risk estimate on the horizontal axis. The partial dependence smoothly increases in a sigmoid-like shape.

The plot of an individual feature's exact contribution to the probability in Figure 12.5 for GAMs looks similar to a partial dependence plot in Figure 12.6 for an uninterpretable model, but is different for one important reason. The contributions of the individual features exactly combine to recreate a GAM because the different features are unlinked and do not interact with each other. In uninterpretable models, there can be strong correlations and interactions among input feature dimensions exploited by the model for generalization. By not visualizing the joint behaviors of multiple features in partial dependence plots, an understanding of those correlations is lost. The set of all d partial dependence functions is not a complete representation of the classifier. Together, they are only an approximation to the complete underlying behavior of the creditworthiness classifier.

12.4.4 SHAP

Just like LIME is a local version of global model approximation, a method known as SHAP is a local version of partial dependence plots. The partial dependence plot shows the entire curve of partial dependence across all feature values, whereas SHAP focuses on the precise point on the feature axis corresponding to a particular applicant in the deployment data. The SHAP value is simply the difference between the partial dependence value for that applicant and the average probability, shown in Figure 12.7.

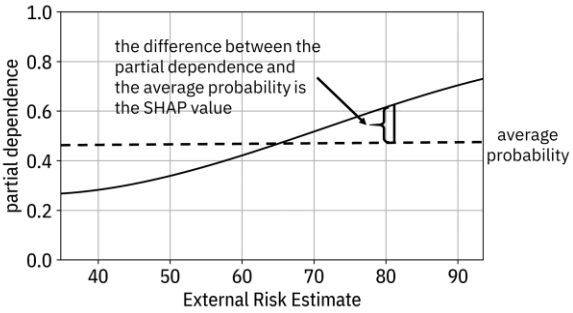


Figure 12.7. *Example showing the SHAP value as the difference between the partial dependence and average probability for a given applicant's 'external risk estimate' value.* Accessible caption. A plot with partial dependence on the vertical axis and external risk estimate on the horizontal axis. The partial dependence smoothly increases in a sigmoid-like shape. A horizontal line passing through the partial dependence function marks the average probability. The difference between the partial dependence and the average probability is the SHAP value.

12.4.5 Saliency Maps

Another local explanation technique for you to consider adding to your Hilo explainability toolkit takes the partial derivative of the classifier's score S or probability of label $\hat{Y} = 1 \mid X$ with respect to each of the input feature dimensions $x^{(i)}$, $i = 1, \ldots, d$. A higher magnitude of the derivative indicates a greater change in the classifier score with a change in the feature dimension value, which is interpreted as greater importance of that feature. Putting together all d of the partial derivatives, you have the gradient of the score with respect to the features ∇S that you examine to see which entries have the largest absolute values. For images, the gradient can be displayed as another image known as a *saliency map*. The decision maker can see which parts of the image are most important to the classification. Saliency map methods are approximate because they do not consider interactions among the different features.

Figure 12.8 shows example saliency maps for a classifier that helps the appraisal process by predicting what objects are seen in a street view image. The model is the Xception image classification model trained on the ImageNet Large Scale Visual Recognition Challenge dataset containing 1000 different classes of objects.[20] The saliency maps shown in the figure are computed by a specific method known as grad-CAM.[21] It is clear from the saliency maps that the classifier focuses its attention on the main house portion and its architectural details, which is to be expected.

[20]François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, USA, Jul. 2017, pp. 1251–1258.

[21]Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." In: *Proceedings of the IEEE International Conference on Computer Vision*. Venice, Italy, Oct. 2017, pp. 618–626. The implementation https://keras.io/examples/vision/grad_cam/ by François Chollet was used to create the figure.

Figure 12.8. *Two examples of grad-CAM applied to an image classification model. The left column is the input image, the middle column is the grad-CAM saliency map with white indicating higher attribution, and the right column superimposes the attribution on top of the image. The top row image is classified by the model as a 'mobile home' and the bottom row image is classified as a 'palace.' (Both classifications are incorrect.) The salient architectural elements are correctly highlighted by the explanation algorithm in both cases.* Accessible caption. In the first example, the highest attribution on a picture of a townhouse is on the windows, stairs, and roof. In the second example, the highest attribution on a picture of a colonial-style house is on the front portico.

12.4.6 Prototypes

Another kind of explanation useful for the decision maker persona, appraiser or credit officer, is through local sample-based approximations of uninterpretable models. Remember that local directly interpretable models, such as the k-nearest neighbor classifier work by averaging the labels of nearby HELOC applicant data points. The explanation is just the list of those other applicants and their labels. However, it is not required that a sample-based explanation only focus on nearby applicants. In this section, you will learn an approach for approximate local sample-based explanation that presents prototypical applicants as its explanation.

Prototypes—data points in the middle of a cluster of other data points shown in Figure 12.9—are useful ways for consumers to perform *case-based reasoning* to gain understanding of a classifier.[22] This reasoning is as follows. To understand the appraised value of a house, compare it to the most prototypical other house in the neighborhood that is average in every respect: average age, average square footage, average upkeep, etc. If the appraised value of the house in question is higher than the

[22]Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. "Examples Are Not Enough, Learn to Criticize!" In: *Advances in Neural Information Processing Systems* 29, (Dec. 2016), pp. 2288–2296. Karthik S. Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. "Efficient Data Representation by Selecting Prototypes with Importance Weights." In: *Proceedings of the IEEE International Conference on Data Mining*. Beijing, China, Nov. 2019, pp. 260–269.

prototype, you can see which features have better values and thus get a sense of how the classifier works, and vice versa.

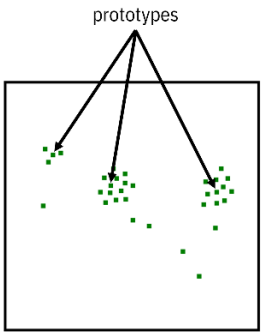


Figure 12.9. *Example of a dataset with three prototype samples marked.* Accessible caption. A plot with several data points, some of which are clustered into three main clusters. Central datapoints within those clusters are marked as prototypes.

However, just showing the one nearest prototype is usually not enough. You'll also want to show a few other nearby prototypes so that the consumer can gain even more intuition. Importantly, listing several nearby prototypes to explain an uninterpretable model and listing several nearby data points to explain the k-nearest neighbor classifier is not the same. It is often the case that the several nearby house data points are all similar to each other and do not provide any further intuition than any one of them alone. With nearby prototype houses, each one is quite different from the others and therefore does provide new understanding.

Let's look at examples of applicants in deployment data whose creditworthiness was predicted by an uninterpretable Hilo model along with three of their closest prototypes from the training data. As a first example, examine an applicant predicted to be creditworthy by the model. The labels of the prototypes must match that of the data point. The example creditworthy applicant's prototype explanation is given in Table 12.4.

The data point and the nearest prototype are quite similar to each other, but with the applicant having a slightly lower 'external risk estimate' and slightly longer time since the oldest trade. It makes sense that the applicant would be predicted to be creditworthy just like the first prototype, even with those differences in 'external risk estimate' and 'months since oldest trade open.' The second nearest prototype represents applicants who have been in the system longer but have executed fewer trades, and have a lower 'external risk estimate.' The decision maker can understand from this that the model is willing to predict applicants as creditworthy with lower 'external risk estimate' values if they counteract that low value with longer time and fewer trades. The third nearest prototype represents applicants who have been in the system even longer, executed even fewer trades, have never been delinquent, and have a very high 'external risk estimate': the really solid applicants.

Table 12.4. *An example prototype explanation for a HELOC applicant predicted to be creditworthy.*

Feature	Applicant (Credit-worthy)	Nearest Prototype	Second Prototype	Third Prototype
External Risk Estimate	82	85	77	89
Months Since Oldest Trade Open	280	223	338	379
Months Since Most Recent Trade Open	13	13	2	156
Average Months in File	102	87	109	257
Number of Satisfactory Trades	22	23	16	3
Percent Trades Never Delinquent	91	91	90	100
Months Since Most Recent Delinquency	26	26	65	0
Number of Total Trades	23	26	21	3
Number of Trades Open in Last 12 Months	0	0	1	0
Percent Installment Trades	9	9	14	33
Months Since Most Recent Inquiry	0	1	0	0
Revolving Balance Divided by Credit Limit	3	4	2	0

As a second example, let's look at an applicant predicted to be non-creditworthy. This applicant's prototype explanation is given in Table 12.5. In this example of a non-creditworthy prediction, the nearest prototype has a better 'external risk estimate,' a lower number of months since the oldest trade, and a lower revolving balance burden, but is still classified as non-creditworthy in the training data. Thus, there is some leeway in these variables. The second nearest prototype represents a younger and less active applicant who has a very high revolving balance burden and poorer 'external risk estimate' and the third nearest prototype represents applicants who have been very recently delinquent and have a very poor 'external risk estimate.' Deployment applicants can be even more non-creditworthy if they have even higher revolving balance burdens and recent delinquencies.

12.5 Explanations for Affected Users

The third and final consumer persona for you to consider as you put together an explainability toolkit for Hilo is the affected user: the HELOC applicant. Consumers from this persona are not so concerned about the overall model or about gaining any approximate understanding. Their goal is quite clear: tell me exactly why my case was deemed to be creditworthy or non-creditworthy. They need recourse when their application was deemed non-creditworthy to get approved the next time. Local exact feature-based explanations meet the need for this persona.

Table 12.5. *An example prototype explanation for a HELOC applicant predicted to be non-creditworthy.*

Feature	Applicant (Non-credit-worthy)	Nearest Prototype	Second Prototype	Third Prototype
External Risk Estimate	65	73	61	55
Months Since Oldest Trade Open	256	191	125	194
Months Since Most Recent Trade Open	15	17	7	26
Average Months in File	52	53	32	100
Number of Satisfactory Trades	17	19	5	18
Percent Trades Never Delinquent	100	100	100	84
Months Since Most Recent Delinquency	0	0	0	1
Number of Total Trades	19	20	6	11
Number of Trades Open in Last 12 Months	7	0	3	0
Percent Installment Trades	29	25	60	42
Months Since Most Recent Inquiry	2	0	0	23
Revolving Balance Divided by Credit Limit	57	31	232	84

The *contrastive explanations method* (CEM) pulls out such local exact explanations from uninterpretable models in a way that leads directly to avenues for recourse by applicants.[23] CEM yields two complementary explanations that go together: (1) *pertinent negatives* and (2) *pertinent positives*. The terminology comes from medical diagnosis. A pertinent negative is something in the patient's history that helps a diagnosis because the patient denies that it is present. A pertinent positive is something that is necessarily present in the patient. For example, a patient with abdominal discomfort, watery stool, and without fever will be diagnosed with likely viral gastroenteritis rather than bacterial gastroenteritis. The abdominal discomfort and watery stool are pertinent positives and the lack of fever is a pertinent negative. A pertinent negative explanation is the minimum change needed in the features to change the predicted label. Changing no fever to fever will change the diagnosis from viral to bacterial.

The mathematical formulation of CEM is almost the same as an adversarial example that you learned about in Chapter 11: find the smallest sparse perturbation δ so that $\hat{y}(x+\delta)$ is different from $\hat{y}(x)$. For pertinent negatives, you want the perturbation to be sparse or concentrated in a few features to be interpretable and understandable. This contrasts with adversarial examples whose perturbations should be diffuse and spread across a lot of features to be imperceptible. A pertinent positive explanation is also a sparse perturbation that is removed from x and maintains the predicted label. Contrastive explanations are computed in a post hoc manner after an uninterpretable model has already been trained. Just like for adversarial examples, there are two cases for the computation: *open-box* when

[23]Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. "Explanations Based on the Missing: Towards Contrastive Explanations with Pertinent Negatives." In: *Advances in Neural Information Processing Systems* 32 (Dec. 2018), pp. 590–601.

the gradients of the model are made available and *closed-box* when the gradients are not made available and must be estimated.

Table 12.6. *An example contrastive explanation for a HELOC applicant predicted to be creditworthy.*

Feature	Applicant (Credit-worthy)	Pertinent Positive
External Risk Estimate	82	82
Months Since Oldest Trade Open	280	-
Months Since Most Recent Trade Open	13	-
Average Months in File	102	91
Number of Satisfactory Trades	22	22
Percent Trades Never Delinquent	91	91
Months Since Most Recent Delinquency	26	-
Number of Total Trades	23	-
Number of Trades Open in Last 12 Months	0	-
Percent Installment Trades	9	-
Months Since Most Recent Inquiry	0	-
Revolving Balance Divided by Credit Limit	3	-

Table 12.7. *An example contrastive explanation for a HELOC applicant predicted to be creditworthy.*

Feature	Applicant (Non-credit-worthy)	Pertinent Negative Perturbation	Pertinent Negative Value
External Risk Estimate	65	15.86	80.86
Months Since Oldest Trade Open	256	0	256
Months Since Most Recent Trade Open	15	0	15
Average Months in File	52	13.62	65.62
Number of Satisfactory Trades	17	4.40	21.40
Percent Trades Never Delinquent	100	0	100
Months Since Most Recent Delinquency	0	0	0
Number of Total Trades	19	0	19
Number of Trades Open in Last 12 Months	7	0	7
Percent Installment Trades	29	0	29
Months Since Most Recent Inquiry	2	0	2
Revolving Balance Divided by Credit Limit	57	0	57

Examples of contrastive explanations for the same two applicants presented in the prototype section are given in Table 12.6 (creditworthy; pertinent positive) and Table 12.7 (non-creditworthy; pertinent negative). To remain creditworthy, the pertinent positive states that this HELOC applicant must maintain the values of 'external risk estimate,' 'number of satisfactory trades,' and 'percent trades never delinquent.' The 'average months in file' is allowed to drop to 91, which is a similar behavior seen in the first prototype of the prototype explanation. For the non-creditworthy applicant, the pertinent negative

perturbation is sparse as desired, with only three variables changed. This minimal change to the applicant's features tells them that if they improve their 'external risk estimate' by 16 points, wait 14 months to increase their 'average months in file', and increase their 'number of satisfactory trades' by 5, the model will predict them to be creditworthy. The recourse for the applicant is clear.

12.6 Quantifying Interpretability

Throughout the chapter, you've learned about many different explainability methods applicable at different points of the machine learning pipeline appealing to different personas, differentiated according to several dichotomies: local vs. global, approximate vs. exact, and feature-based vs. sample-based. But how do you know that a method is actually good or not? Your boss isn't going to put any of your explainability tools into the production Hilo platform unless you can prove that they're effective.

Evaluating interpretability does not yield the same sort of quantitative metrics as in Part 3 for distributional robustness, fairness, and adversarial robustness. Ideally, you want to show explanations to a large set of consumers from the relevant persona performing the task the model is for and get their judgements. Known as *application-grounded evaluation*, this way of measuring the goodness of an explanation is usually costly and logistically difficult.[24] A less involved approach, *human-grounded evaluation*, uses a simpler task and people who are not the future intended consumers, so just general testers rather than actual appraisers or credit officers. An even less involved measurement of interpretability, *functionally-grounded evaluation*, uses quantitative proxy metrics to judge explanation methods on generic prediction tasks. These evaluation approaches are summarized in Table 12.8.

Table 12.8. *Three categories of evaluating explanations.*

Category	Consumers	Tasks
application-grounded evaluation	true persona members	real task
human-grounded evaluation	generic people	simple task
functionally-grounded evaluation	none	proxy task

What are these quantitative proxy metrics for interpretability? Some measure simplicity, like the number of operations needed to make a prediction using a model. Others compare an explanation method's ordering of features attribution to some ground-truth ordering. (These explainability metrics only apply to feature-based explanations.) An explainability metric known as *faithfulness* is based on this idea of comparing feature orderings.[25] Instead of requiring a true ordering, however, it measures the correlation between a given method's feature order to the order in which the accuracy of a model drops the most when the corresponding feature is deleted. A correlation value of 1 is the best faithfulness. Unfortunately, when faithfulness is applied to saliency map explanations, it is unreliable.[26] You should

[24]Finale Doshi-Velez and Been Kim. "Towards a Rigorous Science of Interpretable Machine Learning." arXiv:1702.08608, 2017.

[25]David Alvarez-Melis and Tommi S. Jaakkola. "Towards Robust Interpretability with Self-Explaining Neural Networks." In: *Advances in Neural Information Processing Systems* 32 (Dec. 2018), pp. 7786–7795.

[26]Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. "Sanity Checks for Saliency Metrics." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, New York, USA, Feb. 2020, pp. 6021–6029.

be beware of functionally-grounded evaluation and always try to do at least a little human-grounded and application-grounded evaluation before the Hilo platform goes live.

You've put together an explainability toolkit for both Hilo house appraisal and credit checking models and implemented the appropriate methods for the right touchpoints of the applicants, appraisers, credit officers, and regulators. You haven't taken shortcuts and have gone through a few rounds of functionally-grounded evaluations. Your contributions to the Hilo platform help make it a smashing success when it is launched.

12.7 Summary

- Interpretability and explainability are needed to overcome cognitive biases in the last-mile communication problem between the machine learning model and the human consumer.
- There is no one best explanation method. Different consumers have different personas with different needs to achieve their goals. The important personas are the affected user, the decision maker, and the regulator.
- Human interpretability of machine learning requires features that people can understand to some extent. If the features are not understandable, disentangled representation can help.
- Explanation methods can be divided into eight categories by three dichotomies. Each category tends to be most appropriate for one consumer persona. The first dichotomy is whether the explanation is for the entire model or a specific input data point (global/local). The second dichotomy is whether the explanation is an exact representation of the underlying model or it contains some approximation (exact/approximate). The third dichotomy is whether the language used in creating the explanation is based on the features or on entire data points (feature-based/sample-based).
- Ideally, you want to quantify how good an explanation method is by showing explanations to consumers in the context of the actual task and eliciting their feedback. Since this is expensive and difficult, proxy quantitative metrics have been developed, but they are far from perfect.

13

Transparency

Imagine that you are a model validator in the model risk management department at JCN Corporation, the (fictional) information technology company undergoing an enterprise transformation first encountered in Chapter 7. In addition to using machine learning for estimating the skills of its employees, JCN Corporation is rolling out machine learning in another human resources effort: proactive retention. Using historical employee administrative data, JCN Corporation is developing a system to predict employees at risk of voluntarily resigning in the next six months and offering incentives to retain them. The data includes internal corporate information on job roles and responsibilities, compensation, market demand for jobs, performance reviews, promotions, and management chains. JCN Corporation has consent to use the employee administrative data for this purpose through employment contracts. The data was made available to JCN Corporation's data science team under institutional control after a syntactic anonymity transformation was performed.

The team has developed several attrition prediction models using different machine learning algorithms, keeping accuracy, fairness, distributional robustness, adversarial robustness, and explainability as multiple goals. If the attrition prediction models are fair, the proactive retention system could make employment at JCN Corporation more equitable than it is right now. The project has moved beyond the problem specification, data understanding, data preparation, and modeling phases of the development lifecycle and is now in the evaluation phase.

> "The full cycle of a machine learning project is not just modeling. It is finding the right data, deploying it, monitoring it, feeding data back [into the model], showing safety—doing all the things that need to be done [for a model] to be deployed. [That goes] beyond doing well on the test set, which fortunately or unfortunately is what we in machine learning are great at."
>
> —Andrew Ng, computer scientist at Stanford University

Your job as the model validator is to test out and compare the models to ensure at least one of them is safe and trustworthy before it is deployed. You also need to obtain buy-in from various parties before you can sign your name and approve the model's deployment. To win the support of internal JCN Corporation executives and compliance officers, external regulators,[1] and members of a panel of diverse employees and managers within the company you'll assemble, you need to provide *transparency* by communicating not only the results of independent tests you conduct, but also what happened in the earlier phases of the lifecycle. (Transparent reporting to the general public is also something you should consider once the model is deployed.) Such transparency goes beyond model interpretability and explainability because it is focused on model performance metrics and their uncertainty characterizations, various pieces of information about the training data, and the suggested uses and possible misuses of the model.[2] All of these pieces of information are known as *facts*.

Not all of the various consumers of your transparent reporting are looking for the same facts or the same level of detail. Modeling tasks besides predicting voluntary attrition may require different facts. Transparency has no one-size-fits-all solution. Therefore, you should first run a small design exercise to understand which facts and details are relevant for the proactive retention use case and for each consumer, and the presentation style preferred by each consumer.[3] (Such an exercise is related to value alignment, which is elaborated upon in Chapter 14.) The artifact that ultimately presents a collection of facts to a consumer is known as a *factsheet*. After the design exercise, you can be off to the races with creating, collecting, and communicating information about the lifecycle.

You are shouldering a lot of responsibility, so you don't want to perform your job in a haphazard way or take any shortcuts. To enable you to properly evaluate and validate the JCN Corporation voluntary resignation models and communicate your findings to various consumers, this chapter teaches you to:

- create factsheets for transparent reporting,
- capture facts about the model purpose, data provenance, and development steps,
- conduct tests that measure the probability of expected harms and the possibility of unexpected harms to generate quantitative facts,
- communicate these test result facts and their uncertainty, and
- defend your efforts against people who are not inclined to trust you.

You're up to the task padawan, so let's start equipping you with the tools you need.

13.1 Factsheets

Transparency should reveal several kinds of facts that come from different parts of the lifecycle.[4] From the problem specification phase, it is important to capture the goals, intended uses, and possible

---

[1]Regulations play a role in the company's employee retention programs because they are subject to fair employment laws.
[2]Q. Vera Liao and Kush R. Varshney. "Human-Centered Explainable AI (XAI): From Algorithms to User Experiences." arXiv:2110.10790, 2021.
[3]John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. "A Methodology for Creating AI FactSheets." arXiv:2006.13796, 2020.
[4]Matthew Arnold, Rachel K. E. Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, Karthikeyan Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, Darrell Reimer, John Richards, Jason Tsay, and

misuses of the system along with who was involved in making those decisions (e.g. were diverse voices included?). From the data understanding phase, it is important to capture the provenance of the data, including why it was originally collected. From the data preparation phase, it is important to catalog the data transformations and feature engineering steps employed by the data engineers and data scientists, as well as any data quality analyses that were performed. From the modeling phase, it is important to understand what algorithmic choices were made and why, including which mitigations were employed. From the evaluation phase, it is important to test for trust-related metrics and their uncertainties (details are forthcoming in the next section). Overall, there are two types of facts for you to transparently report: (1) (qualitative) knowledge from inside a person's head that must be explicitly asked about, and (2) data, processing steps, test results, models, or other artifacts that can be grabbed digitally.

How do you get access to all this information coming from all parts of the machine learning development lifecycle and from different personas? Wouldn't it be convenient if it were documented and transparently reported all along? Because of the tireless efforts of your predecessors in the model risk management department, JCN Corporation has instrumented the entire lifecycle with a mandatory tool that manages machine learning development by creating checklists and pop-up reminders for different personas to enter qualitative facts at the time they should be top-of-mind for them. The tool also automatically collects and version-controls digital artifacts as facts as soon as they are generated. Let's refer to the tool as *fact flow*, which is shown in Figure 13.1.

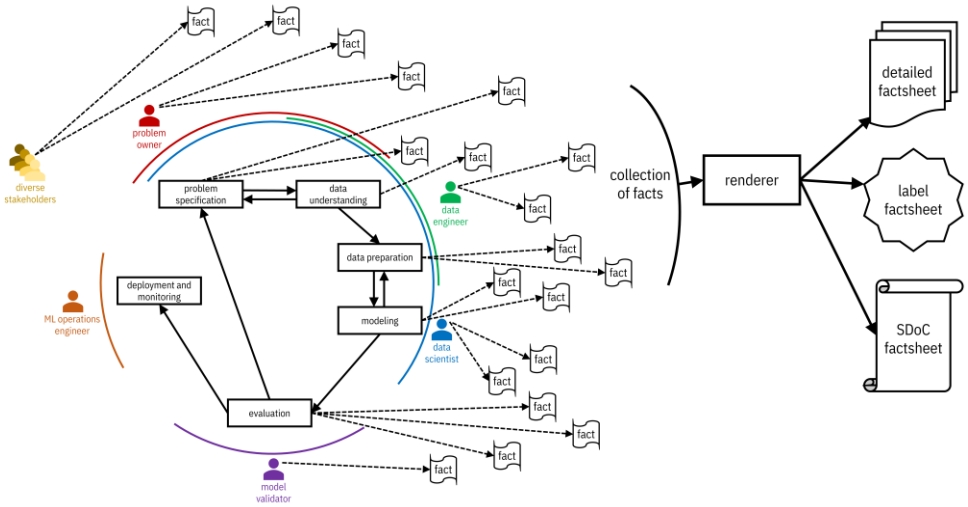


Figure 13.1. *The fact flow captures qualitative and quantitative facts generated by different people and processes throughout the machine learning development lifecycle and renders them into factsheets appropriate for different consumers.* Accessible caption. Facts from people and technical steps in the development lifecycle go into a renderer which may output a detailed factsheet, a label factsheet, or a SDoC factsheet.

Kush R. Varshney. "FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity." In: *IBM Journal of Research and Development* 63.4/5 (Jul./Sep. 2019), p. 6.

Since machine learning is a general purpose technology (recall the discussion in Chapter 1), there is no universal set of facts that applies to all machine learning models irrespective of their use and application domain. The facts to validate the machine learning systems for m-Udhār Solar, Unconditionally, ThriveGuild, and Wavetel (fictional companies discussed in previous chapters) are not exactly the same; more precision is required.[5] Moreover, the set of facts that make it to a factsheet and their presentation depends on the consumer. As the model validator, you need a full dump of all the facts. You should adjust the factsheet to a summary label, document, or presentation slides for personas who, to overcome their cognitive biases, need fewer details. You should broadly disseminate simpler factsheets among JCN Corporation managers (decision makers), employees (affected users), and the general public who yearn for transparency. You will have determined the set of facts, their level of detail, and their presentation style for different personas through your initial design exercise. Fact flow has a renderer for you to create different factsheet presentations.

You should also sign and release a factsheet rendered as a *supplier's declaration of conformity* (SDoC) for external regulators. An SDoC is a written assurance that a product or service conforms to a standard or technical regulation. Your declaration is based on your confidence in the fact flow tool and the inspection of the results you have conducted.[6] *Conformity* is one of several related concepts (compliance, impact, and accountability), but different from each of them.[7] Conformity is abiding by specific regulations whereas *compliance* is abiding by broad regulatory frameworks. Conformity is a statement on abiding by regulations at the current time whereas *impact* is abiding by regulations into an uncertain future. Conformity is a procedure by which to show abidance whereas *accountability* is a responsibility to do so. As such, conformity is the narrowest of definitions and is the one that forms the basis for the draft regulation of high-risk machine learning systems in the European Economic Area and may become a standard elsewhere too. Thus SDoCs represent an up-and-coming requirement for machine learning systems used in high-stakes decision making, including proactive retention at JCN Corporation.

> "We really need standards for what an audit is."
>
> —Rumman Chowdhury, machine learning ethicist at Twitter

13.2 Testing for Quantitative Facts

Many quantitative facts come from your model testing in the evaluation phase. Testing a machine learning model seems easy enough, right? The JCN Corporation data scientists already obtained good accuracy numbers on an i.i.d. held-out data set, so what's the big deal? First, you cannot be sure that the

---

[5]Ryan Hagemann and Jean-Marc Leclerc. "Precision Regulation for Artificial Intelligence." In: *IBM Policy Lab Blog* (Jan. 2020). URL: https://www.ibm.com/blogs/policy/ai-precision-regulation.

[6]National Institute of Standards and Technology. "The Use of Supplier's Declaration of Conformity." URL: https://www.nist.gov/system/files/documents/standardsgov/Sdoc.pdf.

[7]Nikolaos Ioannidis and Olga Gkotsopoulou. "The Palimpsest of Conformity Assessment in the Proposed Artificial Intelligence Act: A Critical Exploration of Related Terminology." In: *European Law Blog* (Jul. 2021). URL: https://europeanlaw-blog.eu/2021/07/02/the-palimpsest-of-conformity-assessment-in-the-proposed-artificial-intelligence-act-a-critical-exploration-of-related-terminology.

data scientists completely isolated their held-out data set and didn't incur any leakage into modeling.[8] As the model validator, you can ensure such isolation in your testing.

Importantly, testing machine learning systems is different from testing other kinds of software systems.[9] Since the whole point of machine learning systems is to generalize from training data to label new unseen input data points, they suffer from the *oracle problem*: not knowing what the correct answer is supposed to be for a given input.[10] The way around this problem is not by looking at a single employee's input data point and examining its corresponding output attrition prediction, but by looking at two or more variations that should yield the same output. This approach is known as using *metamorphic relations*.

For example, a common test for counterfactual fairness (described in Chapter 10) is to input two data points that are the same in every way except having different values of a protected attribute. If the predicted label is not the same for both of them, the test for counterfactual fairness fails. The important point is that the actual predicted label value (will voluntarily resign/won't voluntarily resign) is not the key to the test, but whether that predicted value is equal for both inputs. As a second example for competence, if you multiply a feature's value by a constant in all training points, train the model, and then score a test point that has been scaled by the same constant, you should get the same prediction of voluntary resignation as if you had not done any scaling at all. In some other application involving semi-structured data, a metamorphic relation for an audio clip may be to speed it up or slow it down while keeping the pitch the same. Coming up with such metamorphic relations requires ingenuity; automating this process is an open research question.

In addition to the oracle problem of machine learning, there are three factors you need to think about that go beyond the typical testing done by JCN Corporation data scientists while generating facts:

1. testing for dimensions beyond accuracy, such as fairness, robustness, and explainability,
2. pushing the system to its limits so that you are not only testing average cases, but also covering edge cases, and
3. quantifying aleatoric and epistemic uncertainty around the test results.

Let's look into each of these three concerns in turn.

13.2.1 Testing for Dimensions of Trustworthiness

If you've reached this point in the book, it will not surprise you that testing for accuracy (and related performance metrics described in Chapter 6) is not sufficient when evaluating machine learning models that are supposed to be trustworthy. You also need to test for fairness using metrics such as disparate impact ratio and average odds difference (described in Chapter 10), adversarial robustness using metrics such as empirical robustness and CLEVER score (described in Chapter 11), and explainability

---

[8]Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. "FairPrep: Promoting Data to a First-Class Citizen in Studies of Fairness-Enhancing Interventions." In: *Proceedings of the International Conference on Extending Database Technology*. Copenhagen, Denmark, Mar.–Apr. 2020, pp. 395–398.

[9]P. Santhanam. "Quality Management of Machine Learning Systems." In: *Proceedings of the AAAI Workshop on Engineering Dependable and Secure Machine Learning Systems*. New York, New York, USA, Feb. 2020.

[10]Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. "Machine Learning Testing: Survey, Landscapes and Horizons." In: *IEEE Transactions on Software Engineering* 48.1 (Jan. 2022), pp. 1–36.

using metrics such as faithfulness (described in Chapter 12).[11] You also need to test for accuracy under distribution shifts (described in Chapter 9). Since the JCN Corporation data science team has created multiple attrition prediction models, you can compare the different options. Once you have computed the metrics, you can display them in the factsheet as a table such as Table 13.1 or in visual ways to be detailed in Section 13.3 to better understand their domains of competence across dimensions of trustworthiness. (Remember that domains of competence for accuracy were a main topic of Chapter 7.)

Table 13.1. *Result of testing several attrition models for multiple trust-related metrics.*

Model	Accuracy	Accuracy with Distribution Shift	Disparate Impact Ratio	Empirical Robust-ness	Faithful-ness
logistic regression	0.869	0.775	0.719	0.113	0.677
neural network	0.849	0.755	1.127	0.127	0.316
decision forest (boosting)	0.897	0.846	1.222	0.284	0.467
decision forest (bagging)	0.877	0.794	0.768	0.182	0.516

In these results, the decision forest with boosting has the best accuracy and robustness to distribution shift, but the poorest adversarial robustness, and poor fairness and explainability. In contrast, the logistic regression model has the best adversarial robustness and explainability, while having poorer accuracy and distributional robustness. None of the models have particularly good fairness (disparate impact ratio), and so the data scientists should go back and do further bias mitigation. The example emphasizes how looking only at accuracy leaves you with blind spots in the evaluation phase. As the model validator, you really do need to test for all the different metrics.

13.2.2 Generating and Testing Edge Cases

The primary way to test or audit machine learning models is by feeding in data from different employees and looking at the output attrition predictions that result.[12] Using a held-out dataset with the same probability distribution as the training data will tell you how the model performs in the average case. This is how to estimate empirical risk (the empirical approximation to the probability of error), and thus the way to test for the first of the two parts of safety: the risk of expected harms. Similarly, using held-out data with the same probability distribution is common practice (but not necessary) to test for fairness and explainability. Testing for distributional robustness, by definition however, requires input data points drawn from a probability distribution different from the training data. Similarly, computing empirical adversarial robustness involves creating adversarial example employee features as input.

In Chapter 11, you have already learned how to push AI systems to their limits using adversarial examples. These adversarial examples are test cases for unexpected, worst-case harms that go beyond

[11]Moninder Singh, Gevorg Ghalachyan, Kush R. Varshney, and Reginald E. Bryant. "An Empirical Study of Accuracy, Fairness, Explainability, Distributional Robustness, and Adversarial Robustness." In: *Proceedings of the KDD Workshop on Measures and Best Practices for Responsible AI*. Aug. 2021.

[12]Aniya Aggarwal, Samiulla Shaikh, Sandeep Hans, Swastik Haldar, Rema Ananthanarayanan, and Diptikalyan Saha. "Testing Framework for Black-Box AI Models." In: *Proceedings of the IEEE/ACM International Conference on Software Engineering*. May 2021, pp. 81–84.

the probability distribution of the training and held-out datasets. And in fact, you can think about crafting adversarial examples for fairness and explainability as well as for accuracy.[13] Another way to find edge cases in machine learning systems is by using a crowd of human testers who are challenged to 'beat the machine.'[14] They get points in a game for coming up with rare but catastrophic data points.

Importantly, the philosophy of model validators such as yourself who are testing the proactive retention system is different from the philosophy of malicious actors and 'machine beaters.' These adversaries need to succeed just once to score points, whereas model validators need to efficiently generate test cases that have good *coverage* and push the system from many different sides. You and other model validators have to be obsessed with failure; if you're not finding flaws, you have to think that you're not trying hard enough.[15] Toward this end, *coverage metrics* have been developed for neural networks that measure if every neuron in the model has been tested. However, such coverage metrics can be misleading and do not apply to other kinds of machine learning models.[16] Developing good coverage metrics and test case generation algorithms to satisfy those coverage metrics remains an open research area.

13.2.3 Uncertainty Quantification

As you evaluate and validate proactive retention models for JCN Corporation, testing gives you estimates of the different dimensions of trust as in Table 13.1. But as you've learned throughout the book, especially Chapter 3, uncertainty is everywhere, including in those test results. By quantifying the uncertainty of trust-related metrics, you can be honest and transparent about the limitations of the test results. Several different methods for uncertainty quantification are covered in this section, summarized in Figure 13.2.

> "I can live with doubt and uncertainty and not knowing. I think it's much more interesting to live not knowing than to have answers which might be wrong."
>
> —Richard Feynman, physicist at California Institute of Technology

The total predictive uncertainty includes both aleatoric and epistemic uncertainty. It is indicated by the score for well-calibrated classifiers (remember the definition of calibration, Brier score, and calibration loss[17] from Chapter 6). When the attrition prediction classifier is well-calibrated, the score is

---

[13]Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. "You Shouldn't Trust Me: Learning Models Which Conceal Unfairness from Multiple Explanation Methods." In: *Proceedings of the European Conference on Artificial Intelligence*. Santiago de Compostela, Spain, Aug.–Sep. 2020. Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. "Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, New York, USA, Feb. 2020, pp. 180–186.

[14]Joshua Attenberg, Panos Ipeirotis, and Foster Provost. "Beat the Machine: Challenging Humans to Find a Predictive Model's 'Unknown Unknowns.'" In: *Journal of Data and Information Quality* 6.1 (Mar. 2015), p. 1.

[15]Thomas G. Dietterich. "Robust Artificial Intelligence and Robust Human Organizations." In: *Frontiers of Computer Science* 13.1 (2019), pp. 1–3.

[16]Dusica Marijan and Arnaud Gotlieb. "Software Testing for Machine Learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, New York, USA, Feb. 2020, pp. 13576–13582.

[17]A popular variation of the calibration loss detailed in Chapter 6, known as the *expected calibration error*, uses the average absolute difference rather than the average squared difference.

also the probability of an employee voluntarily resigning being 1; scores close to 0 and 1 are certain predictions and scores close to 0.5 are uncertain predictions. Nearly all of the classifiers that we've talked about in the book give continuous-valued scores as output, but many of them, such as the naïve Bayes classifier and modern deep neural networks, tend not to be well-calibrated.[18] They have large values of calibration loss because their calibration curves are not straight diagonal lines like they ideally should be (remember the picture of a calibration curve dropping below and pushing above the ideal diagonal line in Figure 6.4).

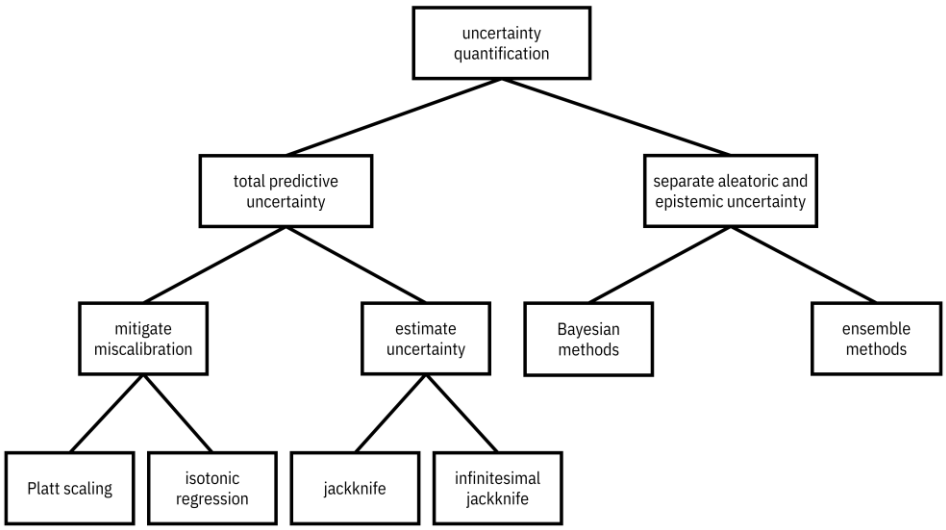


Figure 13.2. *Different methods for quantifying the uncertainty of classifiers.* Accessible caption. Hierarchy diagram with uncertainty quantification as the root. Uncertainty quantification has children total predictive uncertainty, and separate aleatoric and epistemic uncertainty. Total predictive uncertainty has children mitigate miscalibration and estimate uncertainty. Mitigate miscalibration has children Platt scaling and isotonic regression. Estimate uncertainty has children jackknife and infinitesimal jackknife. Separate aleatoric and epistemic uncertainty has children Bayesian methods and ensemble methods.

Just like in other pillars of trustworthiness, algorithms for obtaining uncertainty estimates and mitigating poor calibration apply at different stages of the machine learning pipeline. Unlike other topic areas, there is no pre-processing for uncertainty quantification. There are, however, methods that apply during model training and in post-processing. Two post-processing methods for mitigating poor calibration, *Platt scaling* and *isotonic regression*, both take the classifier's existing calibration curve and straighten it out. Platt scaling assumes that the existing calibration curve looks like a sigmoid or logistic

[18]Alexandru Niculescu-Mizil and Rich Caruana. "Predicting Good Probabilities with Supervised Learning." In: *Proceedings of the International Conference on Machine Learning*. Bonn, Germany, Aug. 2005, pp. 625–632. Chuan Guo, Geoff Pleiss, Yu Sun, and Killian Q. Weinberger. "On Calibration of Modern Neural Networks." In: *Proceedings of the International Conference on Machine Learning*. Sydney, Australia, Aug. 2017, pp. 1321–1330.

activation function whereas isotonic regression can work with any shape of the existing calibration curve. Isotonic regression requires more data than Platt scaling to work effectively.

A post-processing method for total predictive uncertainty quantification that does not require you to start with an existing classifier score works in almost the same way as computing deletion diagnostics described in Chapter 12 for explanation. You train many attrition models, leaving one training data point out each time. You compute the standard deviation of the accuracy of each of these models and report this number as an indication of predictive uncertainty. In the uncertainty quantification context, this is known as a *jackknife* estimate. You can do the same thing for other metrics of trustworthiness as well, yielding an extended table of results that goes beyond Table 13.1 to also contain uncertainty quantification, shown in Table 13.2. Such a table should be displayed in a factsheet.

Table 13.2. *Result of testing several attrition models for multiple trust-related metrics with uncertainty quantified using standard deviation below the metric values.*

Model	Accuracy	Accuracy with Distribution Shift	Disparate Impact Ratio	Empirical Robust-ness	Faithful-ness
logistic regression	0.869 (±0.042)	0.775 (±0.011)	0.719 (±0.084)	0.113 (±0.013)	0.677 (±0.050)
neural network	0.849 (±0.046)	0.755 (±0.013)	1.127 (±0.220)	0.127 (±0.021)	0.316 (±0.022)
decision forest (boosting)	0.897 (±0.041)	0.846 (±0.009)	1.222 (±0.346)	0.284 (±0.053)	0.467 (±0.016)
decision forest (bagging)	0.877 (±0.036)	0.794 (±0.003)	0.768 (±0.115)	0.182 (±0.047)	0.516 (±0.038)

Chapter 12 noted that deletion diagnostics are costly to compute directly, which motivated influence functions as an approximation for explanation. The same kind of approximation involving gradients and Hessians, known as an *infinitesimal jackknife*, can be done for uncertainty quantification.[19] Influence functions and infinitesimal jackknives may also be derived for some fairness, explainability, and robustness metrics.[20]

Using a calibrated score or (infinitesimal) jackknife-based standard deviation as the quantification of uncertainty does not allow you to decompose the total predictive uncertainty into aleatoric and epistemic uncertainty, which can be important as you decide to approve the JCN Corporation proactive retention system. There are, however, algorithms applied during model training that let you estimate the aleatoric and epistemic uncertainties separately. These methods are like directly interpretable models (Chapter 12) and bias mitigation in-processing (Chapter 10) in terms of their place in the

[19]Ryan Giordano, Will Stephenson, Runjing Liu, Michael I. Jordan, and Tamara Broderick. "A Swiss Army Infinitesimal Jackknife." In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Naha, Okinawa, Japan, Apr. 2019, pp. 1139–1147.

[20]Hao Wang, Berk Ustun, and Flavio P. Calmon. "Repairing without Retraining: Avoiding Disparate Impact with Counterfactual Distributions." In: *Proceedings of the International Conference on Machine Learning*. Long Beach, California, USA, Jul. 2019, pp. 6618–6627. Brianna Richardson and Kush R. Varshney. "Addressing the Design Needs of Implementing Fairness in AI via Influence Functions." In: *INFORMS Annual Meeting*. Anaheim, California, USA, Oct. 2021.

pipeline. The basic idea to extract the two uncertainties is as follows.[21] The total uncertainty of a prediction, i.e. the predicted label $\hat{Y}$ given the features X, is measured using the entropy $H(\hat{Y} \mid X)$ (remember entropy from Chapter 3). This prediction uncertainty includes both epistemic and aleatoric uncertainty; it is general and does not fix the choice of the actual classifier function $\hat{y}^*(\cdot)$ within a hypothesis space $\mathcal{F}$. The epistemic uncertainty component captures the lack of knowledge of a good hypothesis space and a good classifier within a hypothesis space. Therefore, epistemic uncertainty goes away once you fix the choice of hypothesis space and classifier. All that remains is aleatoric uncertainty. The aleatoric uncertainty is measured by another entropy $H(\hat{Y} \mid X, f)$, averaged across classifiers $f(\cdot) \in \mathcal{F}$ whose probability of being a good classifier is based on the training data. The epistemic uncertainty is then the difference between $H(\hat{Y} \mid X)$ and the average $H(\hat{Y} \mid X, f)$.

There are a couple ways to obtain these two entropies and thereby the aleatoric and epistemic uncertainty. Bayesian methods, including *Bayesian neural networks*, are one large category of methods that learn full probability distributions for the features and labels, and thus the entropies can be computed from the probability distribution. The details of Bayesian methods are beyond the scope of this book.[22] Another way to obtain the aleatoric and epistemic uncertainty is through ensemble methods, including ones involving bagging and dropout that explicitly or implicitly create several independent machine learning models that are aggregated (bagging and dropout were described in Chapter 7).[23] The average classifier-specific entropy for characterizing aleatoric uncertainty is estimated by simply averaging the entropy of several data points for all the models in the trained ensemble considered separately. The total uncertainty is estimated by computing the entropy of the entire ensemble together.

13.3 Communicating Test Results and Uncertainty

Recall from Chapter 12, that you must overcome the cognitive biases of the consumer of an explanation. The same is true for communicating test results and uncertainty. Researchers have found that the presentation style has a large impact on the consumer.[24] So don't take the shortcut of thinking that your job is done once you've completed the testing and uncertainty quantification. You'll have to justify your model validation to several different factsheet consumers (internal stakeholders within JCN Corporation, external regulators, et al.) and it is important for you to think about how you'll communicate the results.

---

[21]Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluf. "Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-Sensitive Learning." In: *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 1184–1193.

[22]Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems* 31 (Dec. 2017), pp. 5580–5590.

[23]Yarin Gal and Zoubin Gharahmani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." In: *Proceedings of the International Conference on Machine Learning*. New York, New York, USA, Jun. 2016, pp. 1050–1059. Aryan Mobiny, Pengyu Yuan, Supratik K. Moulik, Naveen Garg, Carol C. Wu, and Hien Van Nguyen. "DropConnect is Effective in Modeling Uncertainty of Bayesian Deep Networks." In: *Scientific Reports* 11.5458 (Mar. 2021). Mohammad Hossein Shaker and Eyke Hüllermeier. "Aleatoric and Epistemic Uncertainty with Random Forests." In: *Proceedings of the International Symposium on Intelligent Data Analysis*. Apr. 2020, pp. 444–456.

[24]Po-Ming Law, Sana Malik, Fan Du, and Moumita Sinha. "The Impact of Presentation Style on Human-in-the-Loop Detection of Algorithmic Bias." In: *Proceedings of the Graphics Interface Conference*. May 2020, pp. 299–307.

13.3.1 Visualizing Test Results

Although tables of numbers such as Table 13.2 are complete and effective ways of conveying test results with uncertainty, there are some other options to consider. First, there are nascent efforts to use methods from explainability like contrastive explanations and influence functions to help consumers understand why a model has a given fairness metric or uncertainty level.[25] More importantly, *visualization* is a common approach.

The various trust dimension metrics you have tested are often presented as *bar graphs*. The trust metrics of multiple models can be compared with adjacent bars as in Figure 13.3. However, it is not clear whether this visualization is more effective than simply presenting a table like Table 13.1. Specifically, since model comparisons are to be done across dimensions that are on different scales, one dimension with a large dynamic range can warp the consumer's perception. Also, if some metrics have better values when they are larger (e.g. accuracy) and other metrics have better values when they are smaller (e.g. statistical parity difference), the consumer can get confused when making comparisons. Moreover, it is difficult to see what is going on when there are several models (several bars).

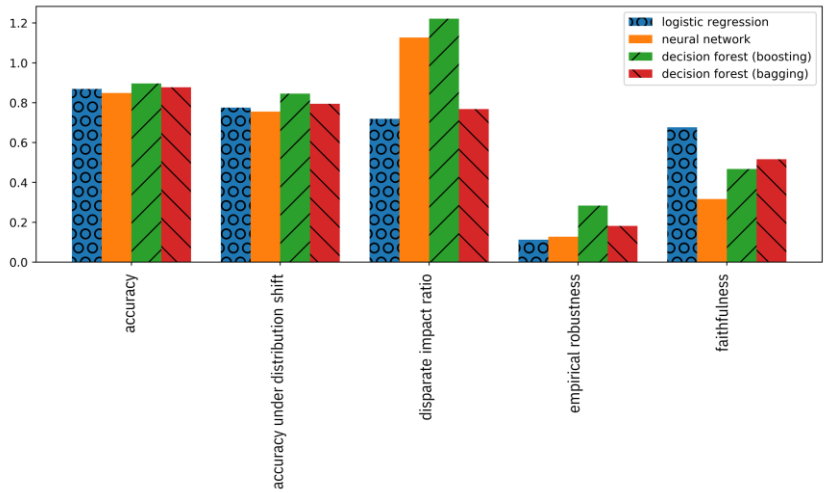


Figure 13.3. *Bar graph of trust metrics for four different models.*

An alternative is the *parallel coordinate plot*, which is a line graph of the different metric dimensions next to each other, but normalized separately.[26] An example is shown in Figure 13.4. The separate normalization per metric permits you to flip the direction of the axis so that, for example, higher is always better. (This flipping has been done for empirical robustness in the figure.) Since the lines can

[25]Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. "Getting a CLUE: A Method for Explaining Uncertainty Estimates." In: *Proceedings of the International Conference on Learning Representations*. May 2021.
[26]Parallel coordinate plots have interesting mathematical properties. For more details, see: Rida E. Moustafa. "Parallel Coordinate and Parallel Coordinate Density Plots." In: *WIREs Computational Statistics* 3 (Mar./Apr. 2011), pp. 134–148.

overlap, there is less of a crowding effect from too many models being compared than with bar graphs. (If there are so so many models that even the parallel coordinate plot becomes unreadable, an alternative is the *parallel coordinate density plot*, which gives an indication of how many lines there are in every part of the plot using shading.) The main purpose of parallel coordinate plots is precisely to compare items along several categories with different metrics. *Conditional parallel coordinate plots*, an interactive version of parallel coordinate plots, allow you to expand upon submetrics within a higher-level metric.[27] For example, if you create an aggregate metric that combines several adversarial robustness metrics including empirical robustness, CLEVER score, and others, an initial visualization will only contain the aggregate robustness score, but can be expanded to show the details of the other metrics it is composed of. Parallel coordinate plots can be wrapped around a polygon to yield a *radar chart*, an example of which is shown in Figure 13.5.

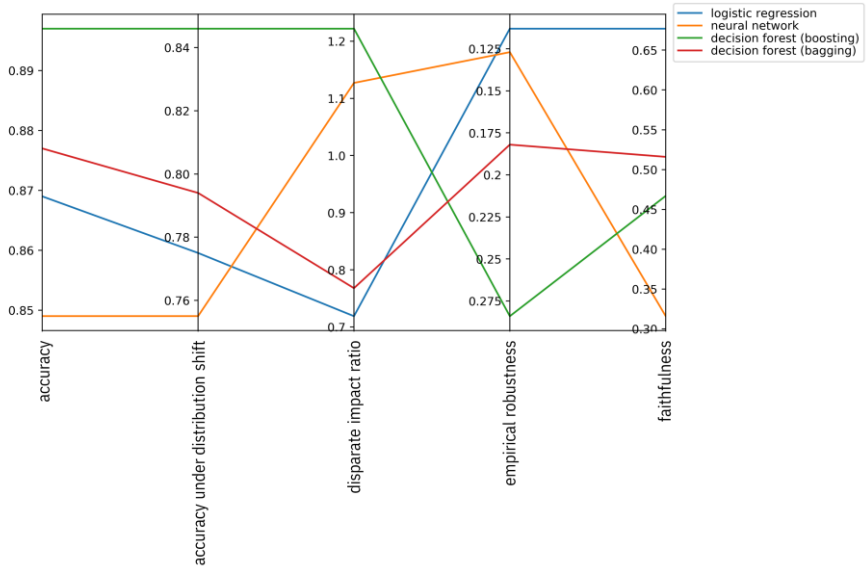


Figure 13.4. *Parallel coordinate plot of trust metrics for four different models.*

[27]Daniel Karl I. Weidele. "Conditional Parallel Coordinates." In: *Proceedings of the IEEE Visualization Conference*. Vancouver, Canada, Oct. 2019, pp. 221–225.

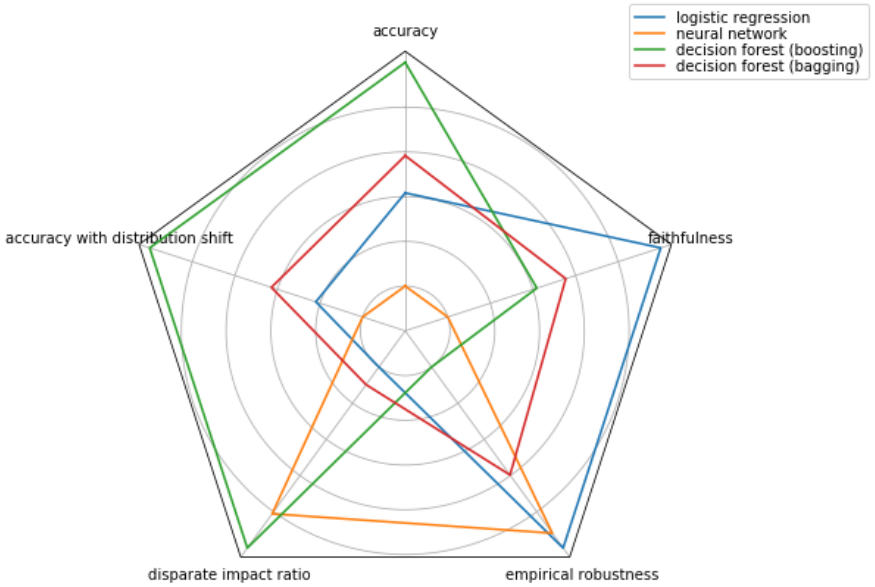


Figure 13.5. *Radar chart of trust metrics for four different models.*

It is not easy to visualize metrics such as disparate impact ratio in which both small and large values indicate poor performance and intermediate values indicate good values. In these cases, and also to appeal to less technical consumers in the case of all metrics, simpler non-numerical visualizations involving color patches (e.g. green/yellow/red that indicate good/medium/poor performance), pictograms (e.g. smiley faces or stars), or Harvey balls (○/◔/◑/◕/●) may be used instead. See Figure 13.6 for an example. However, these visualizations require thresholds to be set in advance on what constitutes a good, medium, or poor value. Eliciting these thresholds is part of value alignment, covered in Chapter 14.

Model	Accuracy	Accuracy with Distribution Shift	Disparate Impact Ratio	Empirical Robust-ness	Faithful-ness
logistic regression	★★★★	★★★★	★	★★★★★	★★★★★
neural network	★★★★	★★★★	★★★	★★★★	★
decision forest (boosting)	★★★★★	★★★★★	★	★	★★★
decision forest (bagging)	★★★★	★★★★	★★	★★★	★★★

Figure 13.6. *Simpler non-numeric visualization of trust metrics for four different models.*

13.3.2 Communicating Uncertainty

It is critical that you not only present the test result facts in a meaningful way, but also present the uncertainty around those test results to ensure that employees receiving and not receiving retention incentives, their managers, other JCN Corporation stakeholders and external regulators have full transparency about the proactive retention system.[28] Van der Bles et al. give nine levels of communicating uncertainty:[29]

1. explicit denial that uncertainty exists,
2. no mention of uncertainty,
3. informally mentioning the existence of uncertainty,
4. a list of possibilities or scenarios,
5. a qualifying verbal statement,
6. a predefined categorization of uncertainty,
7. a rounded number, range or an order-of-magnitude assessment,
8. a summary of a distribution, and
9. a full explicit probability distribution.

You should not consider the first five of these options.

Similar to the green/yellow/red categories described above for test values, *predefined categorizations* of uncertainty, such as 'extremely uncertain,' 'uncertain,' 'certain,' and 'extremely certain' may be useful for less technical consumers. In contrast to green/yellow/red, categories of uncertainty need not be elicited during value alignment because they are more universal concepts that are not related to the actual metrics or use case. *Ranges* express the possibility function (presented in Chapter 3), and can also be useful presentations for less technical consumers.

The last two options are more appropriate for in-depth communication of uncertainty to consumers. *Summaries of probability distributions*, like the standard deviations given in Table 13.2, can also be shown in bar graphs using *error bars*. *Box-and-whisker plots* are like bar graphs, but show not only the standard deviation, but also outliers, quantiles and other summaries of uncertainty through a combination of marks, lines, and shaded areas. *Violin plots* are also like bar graphs, but show the *full explicit probability distribution* through their shape; the shape of the bar follows the pdf of the metric turned on its side. Examples of each are shown in Figure 13.7, Figure 13.8, and Figure 13.9. Parallel coordinate plots and radar charts can also contain error bars or shading to indicate summaries of probability distributions, but may be difficult to interpret when showing more than two or three models.

---

[28]Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q. Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Gauthier Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, Lama Nachman, Rumi Chunara, Madhulika Srikumar, Adrian Weller, and Alice Xiang. "Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. Jul. 2021, pp. 401–413.

[29]Anne Marthe van der Bles, Sander van der Linden, Alexandra L. J. Freeman, James Mitchell, Ana B. Galvao, Lisa Zaval, and David J. Spiegelhalter. "Communicating Uncertainty About Facts, Numbers and Science." In: *Royal Society Open Science* 6.181870 (Apr. 2019).

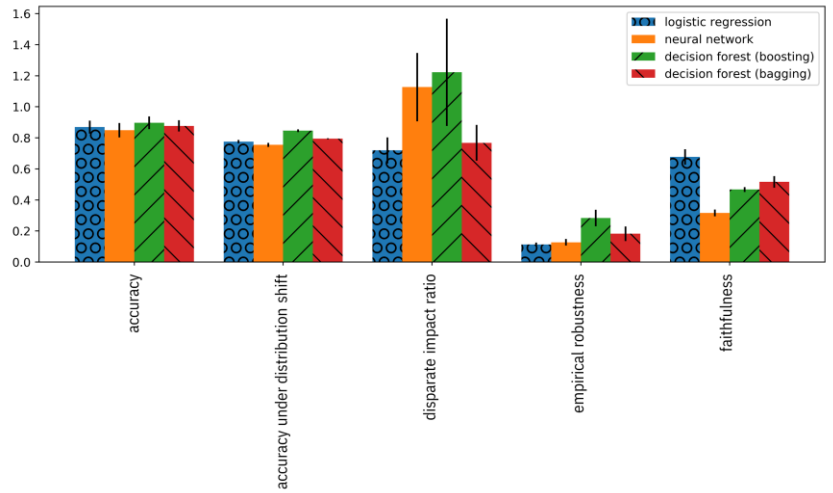


Figure 13.7. *Bar graph with error bars of trust metrics for four different models.*

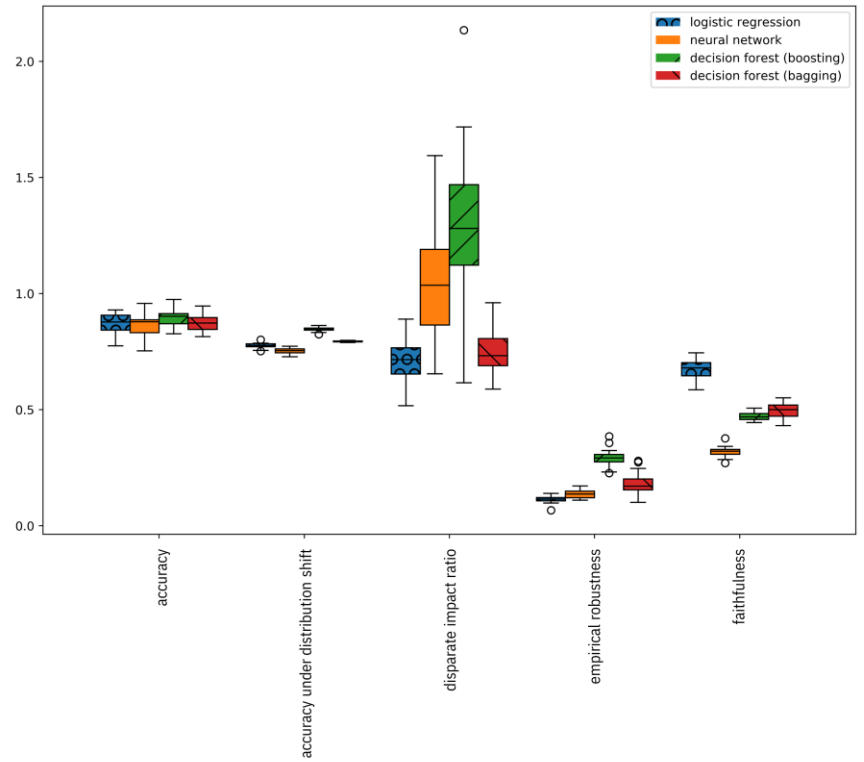


Figure 13.8. *Box-and-whisker plot of trust metrics for four different models.*

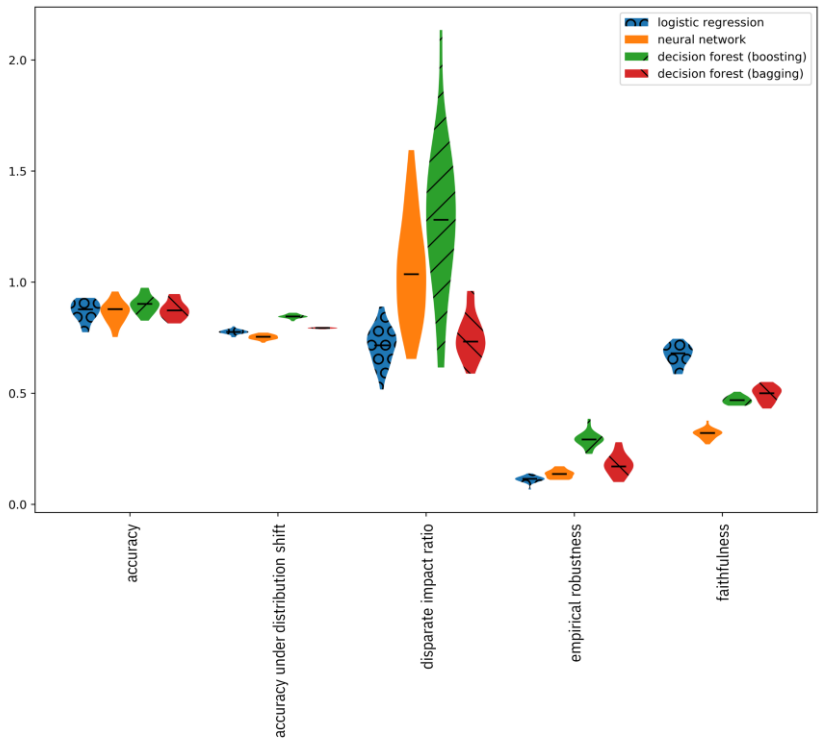


Figure 13.9. *Violin plot of trust metrics for four different models.*

13.4 Maintaining Provenance

In principle, factsheets are a good idea to achieve transparency, show conformity to regulations, and increase trustworthiness in JCN Corporation's proactive retention system. But if consumers of factsheets think JCN Corporation is lying to them, is there anything you can do to convince them otherwise (assuming all the facts are impeccable)? More subtly, how can you show that facts haven't been tampered with or altered after they were generated? Providing such assurance is hard because the facts are generated by many different people and processes throughout the development lifecycle, and just one weak link can spoil the entire factsheet. Provenance of the facts is needed.

One solution is a version of the fact flow tool with an *immutable ledger* as its storage back-end. An immutable ledger is a system of record whose entries (ideally) cannot be changed, so all facts are posted with a time stamp in a way that is very difficult to tamper. It is append-only, so you can only write to it and not change or remove any information. A class of technologies that implements immutable ledgers is *blockchain networks*, which use a set of computers distributed across many owners and geographies to each provably validate and store a copy of the facts. The only way to beat this setup is by colluding with

more than half of the computer owners to change a fact that has been written, which is a difficult endeavor. Blockchains provide a form of distributed trust.

There are two kinds of blockchains: (1) *permissioned* (also known as private) and (2) *permissionless* (also known as public). Permissioned blockchains restrict reading and writing of information and ownership of machines to only those who have signed up with credentials. Permissionless blockchains are open to anyone and can be accessed anonymously. Either may be an option for maintaining the provenance of facts while making the attrition prediction model more trustworthy. If all consumers are within the corporation or are among a fixed set of regulators, then a permissioned blockchain network will do the trick. If the general public or others external to JCN Corporation are the consumers of the factsheet, then a permissionless blockchain is the preferred solution.

Posting facts to a blockchain solves the problem of maintaining the provenance of facts, but what if there is tampering in the creation of the facts themselves? For example, what if a data scientist discovers a small bug in the feature engineering code that shouldn't affect model performance very much and fixes it. Retraining the entire model will go on through the night, but there's a close-of-business deadline to submit facts. So the data scientist submits facts from a previously trained model. Shortcuts like this can also be prevented with blockchain technologies.[30] Since the training of many machine learning models is done in a deterministic way by an iterative procedure (such as gradient descent), other computers in the blockchain network can endorse and verify that the training computation was actually run by locally rerunning small parts of the computation starting from checkpoints of the iterations posted by the data scientist. The details of how to make such a procedure tractable in terms of computation and communication costs is beyond the scope of the book.

In your testing, you found that all of the models were lacking in fairness, so you sent them back to the data scientists to add better bias mitigation, which they did to your satisfaction. The various stakeholders are satisfied now as well, so you can go ahead and sign for the system's conformity and push it on to the deployment stage of the lifecycle. Alongside the deployment efforts, you also release a factsheet for consumption by the managers within JCN Corporation who will be following through on the machine's recommended retention actions. Remember that one of the promises of this new machine learning system was to make employment at JCN Corporation more equitable, but that will only happen if the managers adopt the system's recommendations.[31] Your efforts at factsheet-based transparency have built enough trust among the managers so they are willing to adopt the system, and JCN Corporation will have fairer decisions in retention actions.

---

[30]Ravi Kiran Raman, Roman Vaculin, Michael Hind, Sekou L. Remy, Eleftheria K. Pissadaki, Nelson Kibichii Bore, Roozbeh Daneshvar, Biplav Srivastava, and Kush R. Varshney. "A Scalable Blockchain Approach for Trusted Computation and Verifiable Simulation in Multi-Party Collaborations." In: *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency*. May 2019, Seoul, Korea, pp. 277–284.

[31]There have been instances where a lack of transparency in machine learning algorithms designed to reduce inequity were adopted to a greater extent by privileged decision makers and adopted to a lesser extent by unprivileged decision makers, which ended up exacerbated inequity instead of tamping it down. See: Shunyung Zhang, Kannan Srinivasan, Param Vir Singh, and Nitin Mehta. "AI Can Help Address Inequity—If Companies Earn Users' Trust." In: *Harvard Business Review* (Sep. 2021). URL: https://hbr.org/2021/09/ai-can-help-address-inequity-if-companies-earn-users-trust.

13.5 Summary

- Transparency is a key means for increasing the third attribute of trustworthiness in machine learning (openness and human interaction).
- Fact flow is a mechanism for automatically collecting qualitative and quantitative facts about a development lifecycle. A factsheet is a collection of facts, appropriately rendered for a given consumer, that enables transparency and conformity assessment.
- Model validation and risk management involve testing models across dimensions of trust, computing the uncertainties of the test results, capturing qualitative facts about the development lifecycle, and documenting and communicating these items transparently via factsheets.
- Testing machine learning models is a unique endeavor different from other software testing because of the oracle problem: not knowing in advance what the behavior should be.
- Visualization helps make test results and their uncertainties more accessible to various consumer personas.
- Facts and factsheets become more trustworthy if their provenance can be maintained and verified. Immutable ledgers implemented using blockchain networks provide such capabilities.

14

Value Alignment

The first two chapters in this part of the book on interaction were focused on the communication from the machine system to the human consumer. This chapter is focused on the other direction of interaction: from humans to the machine system. Imagine that you're the director of the selection committee of Alma Meadow, a (fictional) philanthropic organization that invests in early-stage social enterprises and invites the founders of those mission-driven organizations to participate in a two-year fellowship program. Alma Meadow receives about three thousand applications per year and selects about thirty of them to be fellowship recipients. As the director of this process, you are considering using machine learning in some capacity to improve the way it works. As such, you are a problem owner in the problem specification phase of an incipient machine learning lifecycle. Your main concern is that you do not sacrifice Alma Meadow's mission or values in selecting social impact startups.

> "We need to have more conversations where we're doing this translation between policy, world outcome impact, what we care about and then all the math and data and tech stuff is in the back end trying to achieve these things."
>
> —Rayid Ghani, machine learning and public policy researcher at Carnegie Mellon University

Values are fundamental beliefs that guide actions. They indicate the importance of various things and actions to a person or group of people, and determine the best ways to live and behave. Embedding Alma Meadow's values in the machine learning system that you are contemplating is known as *value alignment* and has two parts.[1] The first part is *technical*: how to encode and elicit values in such a way that machine learning systems can access them and behave accordingly. The second part is *normative*: what the actual values are. (The word normative refers to norms in the social rather than mathematical sense: standards

[1]Iason Gabriel. "Artificial Intelligence, Values, and Alignment." In: *Minds and Machines* 30 (Oct. 2020), pp. 411–437.

or principles of right action.) The focus of this chapter is on the first part of value alignment: the technical aspects for you, your colleagues, and other stakeholders to communicate your values (likely influenced by laws and regulations). The chapters in the sixth and final part of the book on purpose delve into the values themselves.

> "There is scientific research that can be undertaken to actually understand how to go from these values that we all agree on to embedding them into the AI system that's working with humans."
>
> —Francesca Rossi, AI ethics global leader at IBM

Before diving into the technical details of value alignment, let's first take a step back and talk about two ways of expressing values: (1) deontological and (2) consequentionalist.[2] At a simplified level, *deontological* values are about defining good *actions* without concern for their outcomes, and *consequentialist* values are focused on defining *outcomes* that are good for all people. As an example, Alma Meadow has two deontological values: at least one of the recipients of the fellowship per year will be a formerly incarcerated individual and fellowship recipients' social change organizations cannot promote a specific religious faith. These explicit rules or constraints on the action of awarding fellowships do not look into the effect on any outcome. In contrast, one of Alma Meadow's consequentionalist values is that a fellowship recipient chosen from the applicant pool leads a social impact startup that will most improve the worldwide disability-adjusted life-years (DALY) in the next ten years. DALY is a metric that indicates the combined morbidity and mortality of the global disease burden. (It cannot be perfectly known which applicant satisfies this at the time the decision is made due to uncertainty, but it can still be a value.) It is a consequentionalist value because it is in terms of an outcome (DALY).

There is some overlap between deontology and procedural justice (described in Chapter 10), and between consequentionalism and distributive justice. One important difference between consequentialism and distributive justice is that in operationalizing distributive justice through group fairness as done in Chapter 10, the population over whom good outcomes are sought are the affected users, and that the justice/fairness is limited in time and scope to just the decision itself.[3] In contrast, in consequentionalism, the good is for all people throughout the broader society and the outcomes of interest are not only the immediate ones, but the longer term ones as well. Just like distributive justice was the focus in Chapter 10 rather than procedural justice because of its more natural operationalization in supervised classification, consequentialism is the focus here rather than deontology. However, it should be noted that deontological values may be elicited from people as rules and used as additional constraints to the Alma Meadow applicant screening model. In certain situations, such constraints can be easily added to the model without retraining.[4]

---

[2]Joshua Greene, Francesca Rossi, John Tasioulas, Kristen Brent Venable, and Brian Williams. "Embedding Ethical Principles in Collective Decision Support Systems." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Phoenix, Arizona, USA, Feb. 2016, pp. 4147–4151.

[3]Dallas Card and Noah A. Smith. "On Consequentialism and Fairness." In: *Frontiers in Artificial Intelligence* 3.34 (May 2020).

[4]Elizabeth M. Daly, Massimiliano Mattetti, Öznur Alkan, and Rahul Nair. "User Driven Model Adjustment via Boolean Rule Explanations." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Feb. 2021, pp. 5896–5904.

It is critical not to take any shortcuts in value alignment because it forms the foundation for the other parts of the lifecycle. By going through the value alignment process, you arrive at problem specifications that data scientists try to satisfy using machine learning models, bias mitigation algorithms, explainability algorithms, adversarial defenses, etc. during the modeling phase of the lifecycle.

One thing to be wary of is underspecification that allows machine learning models to take shortcuts (also known as *specification gaming* and *reward hacking* in the value alignment literature).[5] This concept was covered in detail in Chapter 9, but is worth repeating. Any values that are left unsaid are free dimensions for machine learning algorithms to use as they please. So for example, even if the values you provide to the machine don't prioritize fairness, you might still be opposed to an extremely extremely unfair model in spirit. If you don't include at least some specification for a minimal level of fairness, the model may very well learn to be extremely unfair if it helps achieve specified values in accuracy, uncertainty quantification, and privacy.

In the remainder of the chapter, you will go through the problem specification phase for selecting Alma Meadow's fellows using supervised machine learning, insisting on value alignment. By the end, you'll have a better handle on the following questions.

- What are the different levels of consequentionalist values that you should consider?
- How should these values be elicited from individual people and fused together when elicited from a group of people?
- How do you put together elicited values with transparent documentation covered in Chapter 13 to *govern* machine learning systems?

14.1 Four Levels of Values in Trustworthy Machine Learning

When you were first starting to think about improving Alma Meadow's process for winnowing and selecting applications using machine learning, you had some rough idea why you wanted to do it (improving efficiency and transparency). However, you didn't have a progression of questions to work through as you figured out whether and in which parts of the selection process you should use machine learning, which pillars of trustworthy machine learning you should worry about, and how to make your worries quantitative. Let's list a series of four questions to help you gain clarity. (You'll be aided in answering them in the next section.)

1. Should you work on this problem?
2. Which pillars of trustworthiness are of concern?
3. What are the appropriate metrics for those pillars of trustworthiness?
4. What are acceptable ranges of the metric values?

The first question you should ask is whether you should even work on a problem. The answer may be no. If you stop and think for a minute, many problems are not problems to be solved. At face value,

[5]Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. "Specification Gaming: The Flip Side of AI Ingenuity." In: *DeepMind Blog* (Apr. 2020). URL: https://deepmind.com/blog/article/Specification-gaming-the-flip-side-of-AI-ingenuity.

evaluating three thousand applications and awarding fellowships seems not to be oppressive, harmful, misguided, or useless, but nevertheless, you should think deeply before answering.

> "Technical audiences are never satisfied with the fix being 'just don't do it.'"
>
> —Kristian Lum, statistician at University of Pennsylvania

Even if a problem is one that should be solved, machine learning is not always the answer. Alma Meadow has used a manual process to sort through applications for over thirty years, and has not been worse for wear. So why make the change now? Are there only some parts of the overall evaluation process for which machine learning makes sense?

The second question is more detailed. Among the different aspects of trustworthiness covered in the book so far, such as privacy, consent, accuracy, distributional robustness, fairness, adversarial robustness, interpretability, and uncertainty quantification, which ones are of the greatest concern? Are some essential and others only nice-to-haves? The third question takes the high-level elements of trustworthiness and brings them down to the level of specific metrics. Is accuracy, balanced accuracy, or AUC a more appropriate metric? How about the choice between statistical parity difference and average absolute odds difference? Lastly, the fourth question focuses on the preferred ranges of values of the metrics selected in the third question. Is a Brier score less than or equal to 0.25 acceptable? Importantly, there are relationships among the different pillars; you cannot create a system that is perfect in all respects. For example, typical differential privacy methods worsen fairness and uncertainty quantification.[6] Explainability may be at odds with other dimensions of trustworthiness.[7] Thus in the fourth question, it is critical to understand the relationships among metrics of different pillars and only specify ranges that are feasible.

14.2 Representing and Eliciting Values

Now that you have an overview of the four different levels of values for the supervised machine learning system you're contemplating for Alma Meadow's evaluation process, let's dig a little bit deeper to understand how to represent those values and how to make it easier for you to figure out what your values are.

14.2.1 Should You Work on This Problem?

A helpful tool in determining your values is a checklist of possible concerns along with case studies illustrating each of these concerns in real-world applications of machine learning related to your task of evaluating applications. An example of such a checklist and related case studies is the Ethical OS

---

[6]Marlotte Pannekoek and Giacomo Spigler. "Investigating Trade-Offs in Utility, Fairness and Differential Privacy in Neural Networks." arXiv:2102.05975, 2021. Zhiqi Bu, Hua Wang, Qi Long, and Weijie J. Su. "On the Convergence of Deep Learning with Differential Privacy." arXiv:2106.07830, 2021.

[7]Adrian Weller. "Transparency: Motivations and Challenges." In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Cham, Switzerland: Springer, 2019, pp. 23–40.

Toolkit,[8] which lists eight different broad consequences of the machine learning system that you should ponder:

1. Disinformation: the system helps subvert the truth at a large scale.
2. Addiction: the system keeps users engaged with it beyond what is good for them.
3. Economic inequality: the system contributes to income and wealth inequity by serving only well-heeled users or by eliminating low-income jobs.
4. Algorithmic bias: the system amplifies social biases.
5. Surveillance state: the system enables repression of dissent.
6. Loss of data control: the system causes people to lose control of their own personal data and any monetization it might lead to.
7. Surreptitious: the system does things that users don't know about.
8. Hate and crime: the system makes bullying, stalking, fraud, or theft easier.

Links to case studies accompany each of these checklist items in the Ethical OS Toolkit. Some of the case studies show when the item has happened in the real-world, and some show actions taken to prevent such items from happening. Another source of case studies is the continually-updated AI Incident Database.[9] Part 6 of the book, which is focused on purpose, touches on some of the items and case studies as well.

Starting with the checklist, your first step is to decide which items are good and which items are bad. In practice, you will read through the case studies, compare them to the Alma Meadow use case, spend some time thinking, and come up with your judgement. Many people, including you, will mark each of the eight items as bad, and judge the overall system to be too bad to proceed if any of them is true. But values are not universal. Some people may mark some of the checklist items as good. Some judgements may even be conditional. For example, with all else being equal, you might believe that algorithmic bias (item 4) is good if economic inequality (item 3) is false. In this second case and in even more complicated cases, reasoning about your preferences is not so easy.

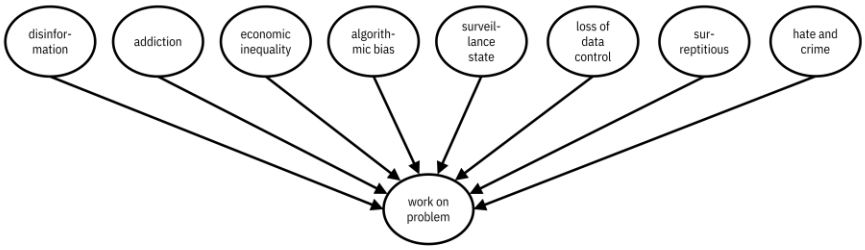
CP-nets are a representation of values, including conditional ones, that help you figure out your overall preference for the system and communicate it to the machine.[10] (The 'CP' stands for 'conditional preference.') CP-nets are directed graphical models with each node representing one attribute (checklist item) and arrows indicating conditional relationships. Each node also has a *conditional preference table* that gives the preferred values. (In this way, they are similar to causal graphs and structural equations you learned about in Chapter 8.) The symbol $\succ$ represents a preference relation; the argument on the left is preferred to the one on the right. The CP-net of the first case above (each of the eight items is bad) is given in Figure 14.1. It has an additional node at the bottom capturing the overall preference for working on the problem, which is conditioned on the eight items. There is a simple, greedy algorithm

[8]URL: https://ethicalos.org/wp-content/uploads/2018/08/Ethical-OS-Toolkit-2.pdf

[9]Sean McGregor. "Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Feb. 2021, pp. 15458–15463.

[10]Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. "CP-Nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements." In: *Journal of Artificial Intelligence Research* 21.1 (Jan. 2004), pp. 135–191.

for figuring out the most preferred instantiation of the values from CP-nets. However, in this case it is easy to figure out the answer without an algorithm: it is the system that does not satisfy any of the eight checklist items and says to go ahead and work on the problem. In more general cases with complicated CP-nets, the inference algorithm is helpful.



disinformation	addiction	economic inequity	algorithmic bias	surveillance state	loss of data control	surreptitious	hate and crime
no ≻ yes	no ≻ yes	no ≻ yes	no ≻ yes	no ≻ yes	no ≻ yes	no ≻ yes	no ≻ yes

work on problem
disinformation = no, addiction = no, economic inequity = no, algorithmic bias = no, surveillance state = no, loss of data control = no, surreptitious = no, hate and crime = no: yes ≻ no
disinformation = yes, addiction = no, economic inequity = no, algorithmic bias = no, surveillance state = no, loss of data control = no, surreptitious = no, hate and crime = no: no ≻ yes
disinformation = no, addiction = yes, economic inequity = no, algorithmic bias = no, surveillance state = no, loss of data control = no, surreptitious = no, hate and crime = no: no ≻ yes
⋮
disinformation = no, addiction = no, economic inequity = no, algorithmic bias = no, surveillance state = no, loss of data control = no, surreptitious = no, hate and crime = yes: no ≻ yes
disinformation = yes, addiction = yes, economic inequity = no, algorithmic bias = no, surveillance state = no, loss of data control = no, surreptitious = no, hate and crime = no: no ≻ yes
⋮
disinformation = yes, addiction = yes, economic inequity = yes, algorithmic bias = yes, surveillance state = yes, loss of data control = yes, surreptitious = yes, hate and crime = yes: no ≻ yes

Figure 14.1. *An example CP-net for whether Alma Meadow should work on the application evaluation problem. At the top is the graphical model. At the bottom are the conditional preference tables.* Accessible caption. Eight nodes disinformation, addiction, economic inequity, algorithmic bias, surveillance state, loss of data control, surreptitious, and hate and crime all have the node work on problem as their child. All preferences for the top eight nodes are no ≻ yes. In all configurations of yeses and noes, the work on problem preference is no ≻ yes, except when all top eight nodes have configuration no, when it is no ≻ yes.

With the values decided, it is time to go through the checklist items and determine whether they are consistent with your most preferred values:

1. Disinformation = no: evaluating applications from social entrepreneurs is unlikely to subvert the truth.
2. Addiction = no: this use of machine learning is not likely to lead to addiction.
3. Economic inequality = partly yes, partly no: it is possible the system could only select applications that have very technical descriptions of the social impact startup's value proposition and have been professionally polished. However, this possibility is not enough of a concern to completely stop the use of machine learning. What this concern does suggest, though, is that machine learning only be used to prioritize semi-finalists rather than later in the evaluation process because human evaluators may find gems that seem unusual to the machine.

4. Algorithmic bias = no: Alma Meadow has been extremely proactive in preventing social bias with respect to common protected attributes in its human evaluations in past years, so the training data will not yield much social bias in models.
5. Surveillance state = no: the machine learning system is unlikely to be an instrument of oppression.
6. Loss of data control = no: by sharing their ideas in the application, budding social entrepreneurs could feel that they are giving up their intellectual property, but Alma Meadow has gone to great lengths to ensure that is not the case. In fact, toward one of its values, Alma Meadow provides information to applicants on how to construct confidential information assignment agreements.
7. Surreptitious = no: the system is unlikely to do anything users don't know about.
8. Hate and crime = no: the system is unlikely to enable criminal activities.

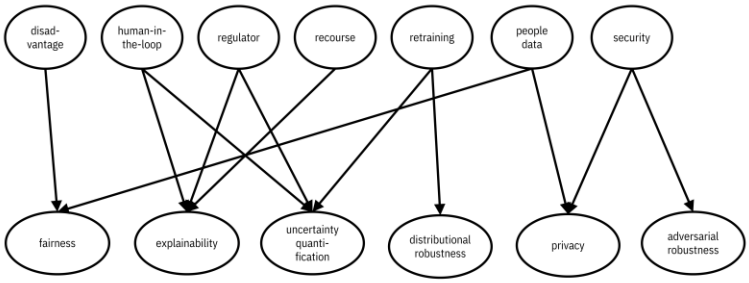
None of the items are properties of the system, including economic inequality when restricting the use of machine learning only to a first-round prioritization. This is consistent with your most-preferred values, so you should work on this problem.

14.2.2 Which Pillars of Trustworthiness Are of Concern?

Now that you have passed the first level of value judgement, you have to determine which elements of trust are your top priority in the feature engineering and modeling phases. Rather than having you take on the very difficult task of trying to directly state a preference ordering, e.g. fairness $\succ$ explainability $\succ$ distributional robustness $\succ$ uncertainty quantification $\succ$ privacy $\succ$ adversarial robustness, let's create a CP-net with some considerations that are easier to answer. To make things even easier, let's assume that you are in a predictive modeling situation, not causal modeling of interventions. Let's take accuracy and similar performance metrics from Chapter 6 out of the equation, since basic competence is always valued. Furthermore, assume the application is high-risk (true for Alma Meadow's applicant selection), so the different elements of trustworthiness are part of your value consideration, and assume that consent and transparency are required. Then a construction of the CP-net for pillars of trustworthiness begins with the following seven properties:

1. Disadvantage (no, yes): the decisions have the possibility of giving systematic disadvantage to certain groups or individuals.
2. Human-in-the-loop (no, yes): the system predictions support a human decision-maker.
3. Regulator (no, yes): regulators (broadly-construed) audit the model.
4. Recourse (no, yes): affected users of the system have the ability to challenge the decision they receive.
5. Retraining (no, yes): the model is retrained frequently to match the time scale of distribution shift.
6. People data (not about people, about people but not SPI, SPI): the system may use data about people which may be sensitive personal information (SPI).
7. Security (external, internal and not secure, secure): the data, model interface, or software code are available either externally or only internally, and may be kept highly secured.

Once you have given these seven system preferences, giving conditional preferences for the different elements of trustworthiness is more compact. They can simply be given as high or low priority values based on just a few of the system preferences. For example, if there is a possibility of systematic disadvantage *and* the problem involves people data, then giving attention to fairness may be highly valued. Putting everything together yields a CP-net like the one in Figure 14.2.



disadvantage	human-in-the-loop	regulator	recourse	retraining	people data	security
yes ≻ no	yes ≻ no	no ≻ yes	yes ≻ no	no ≻ yes	not SPI ≻ not about people ≻ SPI	secure ≻ internal and not secure ≻ external

fairness
disadvantage = no AND people data = not about people: low priority ≻ high priority
disadvantage = no AND people data = not SPI: low priority ≻ high priority
disadvantage = no AND people data = SPI: low priority ≻ high priority
disadvantage = yes AND people data = not about people: low priority ≻ high priority
disadvantage = yes AND people data = not SPI: high priority ≻ low priority
disadvantage = yes AND people data = SPI: high priority ≻ low priority

distributional robustness
retraining = no: high priority ≻ low priority
retraining = yes: low priority ≻ high priority

explainability
human-in-the-loop = no AND regulator = no AND recourse = no: low priority ≻ high priority
human-in-the-loop = no AND regulator = no AND recourse = yes: high priority ≻ low priority
human-in-the-loop = no AND regulator = yes AND recourse = no: high priority ≻ low priority
human-in-the-loop = no AND regulator = yes AND recourse = yes: high priority ≻ low priority
human-in-the-loop = yes AND regulator = no AND recourse = no: high priority ≻ low priority
human-in-the-loop = yes AND regulator = no AND recourse = yes: high priority ≻ low priority
human-in-the-loop = yes AND regulator = yes AND recourse = no: high priority ≻ low priority
human-in-the-loop = yes AND regulator = yes AND recourse = yes: high priority ≻ low priority

uncertainty quantification
human-in-the-loop = no AND regulator = no AND retraining = no: high priority ≻ low priority
human-in-the-loop = no AND regulator = no AND retraining = yes: low priority ≻ high priority
human-in-the-loop = no AND regulator = yes AND retraining = no: high priority ≻ low priority
human-in-the-loop = no AND regulator = yes AND retraining = yes: high priority ≻ low priority
human-in-the-loop = yes AND regulator = no AND retraining = no: high priority ≻ low priority
human-in-the-loop = yes AND regulator = no AND retraining = yes: high priority ≻ low priority
human-in-the-loop = yes AND regulator = yes AND retraining = no: high priority ≻ low priority
human-in-the-loop = yes AND regulator = yes AND retraining = yes: high priority ≻ low priority

privacy
people data = not about people AND security = external: low priority ≻ high priority
people data = not about people AND security = internal and not secure: low priority ≻ high priority
people data = not about people AND security = secure: low priority ≻ high priority
people data = not SPI AND security = external: low priority ≻ high priority
people data = not SPI AND security = internal and not secure: low priority ≻ high priority
people data = not SPI AND security = secure: low priority ≻ high priority
people data = SPI AND security = external: high priority ≻ low priority
people data = SPI AND security = internal and not secure: low priority ≻ high priority
people data = SPI AND security = secure: low priority ≻ high priority

adversarial robustness
security = external: high priority ≻ low priority
security = internal and not secure: high priority ≻ low priority
security = secure: low priority ≻ high priority

Figure 14.2. *An example CP-net for which pillars of trustworthiness Alma Meadow should prioritize when developing a model for the application evaluation problem. At the top is the graphical model. At the bottom are the conditional preference tables.* Accessible caption. In the graphical model, there are edges from disadvantage to fairness, people data to fairness, human-in-the-loop to explainability, regulator to explainability, recourse to explainability, human-in-the-loop to uncertainty quantification, regulator to uncertainty quantification, retraining to uncertainty quantification, retraining to distributional robustness, people data to privacy, security to privacy, and security to adversarial robustness. The conditional preference tables list many different complicated preferences.

The top-level system property preferences will be highly specific to your Alma Meadow application evaluation use case. You and other problem owners have the requisite knowledge at your fingertips to provide your judgements. The conditional preferences connecting the top-level properties with the specific elements of trustworthiness (fairness, explainability, etc.) are more generic and generalizable. Even if the edges and conditional preference tables given in the figure are not 100% universal, they are close to universal and can be used as-is in many different application domains.

In the Alma Meadow example in Figure 14.2, your specific judgements are: systematic disadvantage is possible, you prefer a human decision-maker in the loop, there will not be a regulator audit, you prefer that social entrepreneur applicants have an opportunity for recourse, you prefer the system not be retrained frequently, you prefer that the applications contain data about people (both about the applicant and the population their organization serves) but not anything personally-sensitive, and you prefer that the data and models be secured. Based on these values and the conditional preferences lower in the CP-net, the following pillars are inferred to be higher priority: fairness, explainability, uncertainty quantification, and distributional robustness. Privacy and adversarial robustness are inferred to be lower priority.

14.2.3 What Are the Appropriate Metrics?

After the second stage of value alignment, you know which pillars of trustworthiness are higher priority and you can move on to figuring out specific metrics within the pillars. This problem is known as *performance metric elicitation*. In previous chapters, you've already learned about different considerations when making these determinations. For example, in Chapter 6, it was discussed that AUC is an appropriate basic performance metric when you desire good performance across all operating points. As another example, Table 10.1 summarized the considerations in determining group fairness metrics: whether you are testing data or models, whether there is social bias in the measurement process, and whether the favorable label is assistive or non-punitive. We will not repeat those arguments here, which you should definitely go through, but will mention another tool to help you in metric elicitation.

In the previous elicitation task, it was difficult to go straight to a total preference ordering for the different pillars of trustworthiness; the task was made easier by asking simpler and more structured judgements using CP-nets. There's a similar story here, but using *pairwise comparisons* instead of CP-nets. The elicitation process is like an optometrist helping you home in on your preferred eye prescription by having you compare a sequence of pairs of lenses. Here, the pairwise comparisons are between different possible metrics within a given pillar. By comparing the values of two metrics for many models, you get a sense of what they're indicating and can choose one over the other. If the pairs are chosen in an intelligent way and you do enough comparisons, you will converge onto your preferred metric. One such intelligent way efficiently elicits basic performance metrics and fairness metrics by taking advantage of their linearity or quadraticity properties and showing users a sequence of pairs of confusion matrices (recall confusion matrices from Chapter 6).[11] Confusion matrices may be too difficult for different stakeholders to reason about in their typical format as a 2×2 matrix of numbers; alternate visualizations of confusion matrices such as tree diagrams, flow charts, and matrices presented with

[11]Gaurush Hiranandani, Harikrishna Narasimhan, and Oluwasanmi Koyejo. "Fair Performance Metric Elicitation." In: *Advances in Neural Information Processing Systems* 33 (Dec. 2020), pp. 11083–11095.

contextual information may be used instead.[12] Another approach based on pairwise comparisons is known as the *analytical hierarchy process*; it asks for numerical ratings (one to nine) in the comparison so that you not only indicate which metric is better, but by roughly how much as well.[13]

14.2.4 What are Acceptable Ranges of the Metric Values?

Once specific metrics have been selected, the final level of value alignment is determining the quantitative ranges of preferred metric values for the Alma Meadow semi-finalist selection model. Since the different elements of trustworthiness and their relevant metrics are interrelated, including some that are tradeoffs, this level of elicitation should not be approached one metric at a time like the previous metric elicitation, but more holistically.

The starting point is a feasible set of metric values, shown schematically in Figure 14.3. In this schematic, the quantitative test results for a single model (shown as tables, bar graphs, parallel coordinate plots, and radar charts in Chapter 13) are mapped to a single point inside the feasible region. From Chapter 6, you know that the optimal Bayes risk is fundamentally the best you can ever do for cost-weighted accuracy. As also mentioned in that chapter, it turns out that you can empirically estimate the optimal Bayes risk from the historical Alma Meadow applications data you have.[14] Moreover, fundamental theoretical relationships between metrics from different elements of trustworthiness are starting to be researched using the concept of Chernoff information[15] from detection theory and information theory (they include both tradeoffs and non-tradeoffs): a so-called unified theory of trust.[16] Once that research is completed, the schematic diagram of Figure 14.3 can be actualized for a given machine learning task and the fourth value alignment question (ranges of values of different metrics) can be more easily stated. By explicitly knowing the feasible set of metric values, you can confidently make choices that are possible for the Alma Meadow semi-finalist prioritization model instead of wishful thinking.

---

[12]Hong Shen, Haojian Jin, Ángel Alexander Cabrera, Adam Perer, Haiyi Zhu, and Jason I. Hong. "Designing Alternative Representations of Confusion Matrices to Support Non-Expert Public Understanding of Algorithm Performance." In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW2 (Oct. 2020), p. 153.

[13]Yunfeng Zhang, Rachel K. E. Bellamy, and Kush R. Varshney. "Joint Optimization of AI Fairness and Utility: A Human-Centered Approach." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, New York, USA, Feb. 2020, pp. 400–406.

[14]Visar Berisha, Alan Wisler, Alfred O. Hero, III, and Andreas Spanias. "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure." In: *IEEE Transactions on Signal Processing* 64.3 (Feb. 2016), pp. 580–591. Ryan Theisen, Huan Wang, Lav R. Varshney, Caiming Xiong, and Richard Socher. "Evaluating State-of-the-Art Classification Models Against Bayes Optimality." In: *Advances in Neural Processing Systems* 34 (Dec. 2021).

[15]Frank Nielsen. "An Information-Geometric Characterization of Chernoff Information." In: *IEEE Signal Processing Letters* 20.3 (Mar. 2013), pp. 269–272.

[16]Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush R. Varshney, "Is There a Trade-Off Between Fairness and Accuracy? A Perspective Using Mismatched Hypothesis Testing." In: *Proceedings of the International Conference on Machine Learning*. Jul. 2020, pp. 2803–2813. Kush R. Varshney, Prashant Khanduri, Pranay Sharma, Shan Zhang, and Pramod K. Varshney, "Why Interpretability in Machine Learning? An Answer Using Distributed Detection and Data Fusion Theory." In: *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning*. Stockholm, Sweden, Jul. 2018, pp. 15–20. Zuxing Li, Tobias J. Oechtering, and Deniz Gündüz. "Privacy Against a Hypothesis Testing Adversary." In: *IEEE Transactions on Information Forensics and Security* 14.6 (Jun. 2019), pp. 1567–1581.

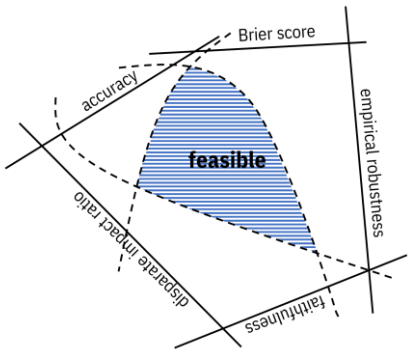


Figure 14.3. *Schematic diagram of feasible set of trust-related metrics.* Accessible caption. A shaded region enclosed by three curved segments is labeled feasible. It is surrounded by five axes: accuracy, Brier score, empirical robustness, faithfulness, and disparate impact ratio.

The feasible set is a good starting point, but there is still the question of deciding on the preferred ranges of the metrics. Two approaches may help. First, a value alignment system can automatically collect or create a corpus of many models for the same or similar prediction task and compute their metrics. This will yield an empirical characterization of the interrelationships among the metrics.[17] You can better understand your choice of metric values based on their joint distribution in the corpus. The joint distribution can be visualized using a parallel coordinate density plot mentioned in Chapter 13.

Second, the value alignment system can utilize a variation of so-called *trolley problems* for supervised machine learning. A trolley problem is a thought experiment about a fictional situation in which you can save the lives of five people who'll otherwise be hit by a trolley by swerving and killing one person. Whether you choose to divert the trolley reveals your values. Variations of trolley problems change the number of people who die under each option and associate attributes with the people.[18] They are also pairwise comparisons. Trolley problems are useful for value elicitation because humans are more easily able to reason about small numbers than the long decimals that usually appear in trust metrics. Moreover, couching judgements in terms of an actual scenario helps people internalize the consequences of the decision and relate them to their use case.

As an example, consider the two scenarios shown in Figure 14.4. Which one do you prefer? Would you rather have an adversarial example fool the system or have a large disparate impact ratio? The actual numbers also play a role because a disparate impact ratio of 2 in scenario 2 is quite high. There is no right or wrong answer, but whatever you select indicates your values.

[17]Moninder Singh, Gevorg Ghalachyan, Kush R. Varshney, and Reginald E. Bryant. "An Empirical Study of Accuracy, Fairness, Explainability, Distributional Robustness, and Adversarial Robustness." In: *KDD Workshop on Measures and Best Practices for Responsible AI*. Aug. 2021.

[18]Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. "The Moral Machine Experiment." In: *Nature* 563.7729 (Oct. 2018), pp. 59–64.

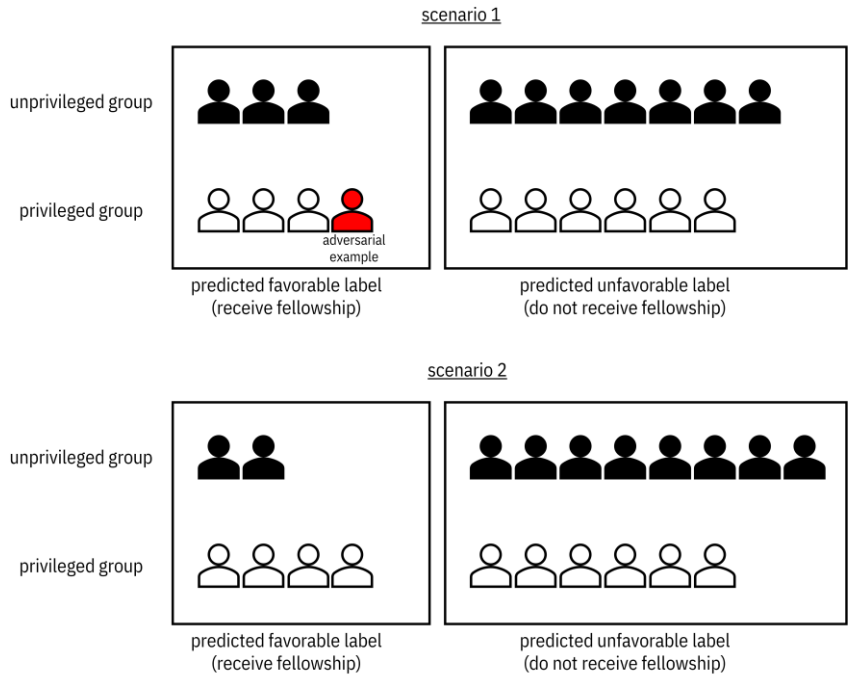


Figure 14.4. *A pairwise comparison of illustrated scenarios.* Accessible caption. Two scenarios each have different small numbers of members of unprivileged and privileged groups receiving and not receiving the fellowship. The first scenario also has an adversarial example.

14.3 Fusion of Preferences Over a Group

Based on the previous section, you have several ways to tell the machine learning system your preferred values at different levels of granularity. As the problem owner, you have a lot of power. But should you wield that power unilaterally? Wouldn't it better to include diverse voices and build consensus? Yes it would; it is important to take the preferences of other stakeholders such as the executive director, board members, and members of the Alma Meadow team into account. It is also critical that budding social entrepreneurs and the beneficiaries of their social impact startups participate in the value alignment process (they should be monetarily compensated for participating). The values communicated to the machine learning system should also take applicable laws and regulations into account; the law is another voice.

Each of the individuals in an assembled panel can go through the same four-level value elicitation that you did in the previous section, yielding several CP-nets and sets of pairwise comparisons. But then what? How do you technically combine the individual preferences expressed by the different folks? *Voting* of some kind, also known as *computational social choice*, is a natural answer. Extensions of both CP-nets and the analytic hierarchy process use voting-like mechanisms to fuse together several individual

preferences.[19] Other methods for aggregating individual preferences into collective preferences are also based on voting.[20]

Voting methods typically aim to choose the value that is preferred by the majority in every pairwise comparison with other possible values (this majority-preferred set of values is known as the *Condorcet winner*). However, it is not clear if such majoritarianism is really what you want when combining the preferences of the various stakeholders. Minority voices may raise important points that shouldn't be drowned out by the majority, which is apt to happen in independent individual elicitation followed by a voting-based preference fusion. The degree of participation by members of minoritized groups should not be so weak as to be meaningless or even worse: extractive (the idea of extraction conceived in postcolonialialism is covered in Chapter 15).[21] This shortcoming of voting systems suggests that an alternative process be pursued that does not reproduce existing power dynamics. *Participatory design*—various stakeholders, data scientists and engineers working together in facilitated sessions to collectively come up with single CP-nets and pairwise comparisons—is a suggested remedy, but may in fact also reproduce existing power dynamics if not conducted well. So in your role at Alma Meadow, don't skimp on well-trained facilitators for participatory design sessions.

14.4 Governance

You've come to an agreement with the stakeholders on the values that should be expressed in Alma Meadow's application screening system. You've specified them as feasible ranges of quantitative metrics that the machine learning system can incorporate. Now how do you ensure that those desired values are realized by the deployed machine learning model? Through *control* or *governance*.[22] Viewing the lifecycle as a control system, illustrated in Figure 14.5, the values coming out of value alignment are the reference input, the data scientists are the controllers that try to do all they can so the machine learning system meets the desired values, and model facts (described in Chapter 13 as part of transparency) are the measured output of testing that indicate whether the values are met. Any difference between the facts and the values is a signal of misalignment to the data scientists; they must do a better job in modeling. In this way, the governance of machine learning systems requires both the elicitation of the system's desired behavior (value alignment) and the reporting of facts that measure those behaviors (transparency).

---

[19]Lirong Xia, Vincent Conitzer, and Jérôme Lang. "Voting on Multiattribute Domains with Cyclic Preferential Dependencies." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Chicago, Illinois, USA, Jul. 2008, pp. 202–207. Indrani Basak and Thomas Saaty. "Group Decision Making Using the Analytic Hierarchy Process." In: *Mathematical and Computer Modelling* 17.4–5 (Feb.–Mar. 1993), pp. 101–109.

[20]Ritesh Noothigattu, Snehalkumar 'Neil' S. Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel D. Procaccia. "A Voting-Based System for Ethical Decision Making." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA, Feb. 2018, pp. 1587–1594. Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. "WeBuildAI: Participatory Framework for Algorithmic Governance." In: *Proceedings of the ACM on Human-Computer Interaction* 3.181 (Nov. 2019).

[21]Sasha Costanza-Chock. *Design Justice: Community-Led Practices to Build the Worlds We Need*. Cambridge, Massachusetts, USA: MIT Press, 2020.

[22]Osonde A. Osoba, Benjamin Boudreaux, and Douglas Yeung. "Steps Towards Value-Aligned Systems." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, New York, USA, Feb. 2020, pp. 332–336.

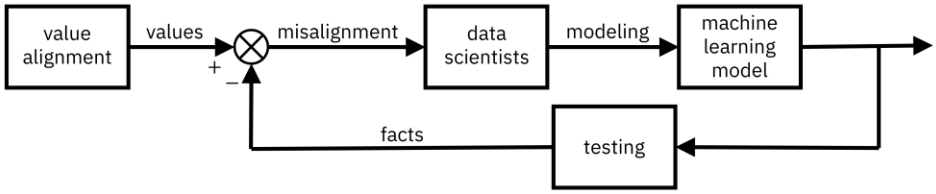


Figure 14.5. *Transparent documentation and value alignment come together to help in the governance of machine learning systems.* Accessible caption. A block diagram that starts with a value alignment block out of which come values. Facts are subtracted from values to yield misalignment. Misalignment is input to a data scientists block with modeling as output. Modeling is input to a machine learning model with output that is fed into a testing block. The output of testing is the same facts that were subtracted from values, creating a feedback loop.

In Chapter 13, factsheets contained not only quantitative test results, but also intended uses and other qualitative knowledge about the development process. However, in the view of governance presented here, only the quantitative test results seem to be used. So, is governance concerned only with test outcomes, which are of a consequentialist nature, or is it also concerned with the development process, which is of a deontological nature? Since the controllers—the data scientists—are people with inherent quirks and biases, both kinds of facts together help them see the big picture goals without losing track of their lower-level, day-to-day duties for resolving misalignment. Thus, a codification of processes to be followed during development is an integral part of governance. Toward this end, you have instituted a set of checklists for Alma Meadow's data scientists to follow, resulting in a well-governed system overall.

14.5 Summary

- Interaction between people and machine learning systems is not only *from* the machine learning system *to* a human via explainability and transparency. The other direction from humans to the machine, known as value alignment, is just as critical so that people can instruct the machine on acceptable behaviors.
- There are two kinds of values: consequentialist values that are concerned with outcomes and deontological values that are concerned with actions. Consequentialist values are more natural in value alignment for supervised machine learning systems.
- Value alignment for supervised classification consists of four levels. Should you work on a problem? Which pillars of trustworthiness are high priority? What are the appropriate metrics? What are acceptable metric value ranges?
- CP-nets and pairwise comparisons are tools for structuring the elicitation of preferences of values across the four levels.
- The preferences of a group of stakeholders, including those from traditionally marginalized backgrounds, may be combined using either voting or participatory design sessions.
- Governance of machine learning systems combines value alignment to elicit desired behaviors with factsheet-based transparency to measure whether those elicited behaviors are being met.

15

Ethics Principles

The fourth attribute of trustworthiness, introduced in Chapter 1, includes low self-orientation, motivation to serve others' interests as well as own interests, benevolence, and an aligned purpose. This chapter focuses on this fourth attribute and kicks off the sixth and final part of the book (remember the organization of the book illustrated in Figure 15.1). Introduced in Chapter 14, value alignment is composed of two halves: technical and normative; this chapter deals with the normative part. Unlike earlier chapters, this chapter is not presented through a fictional use case.

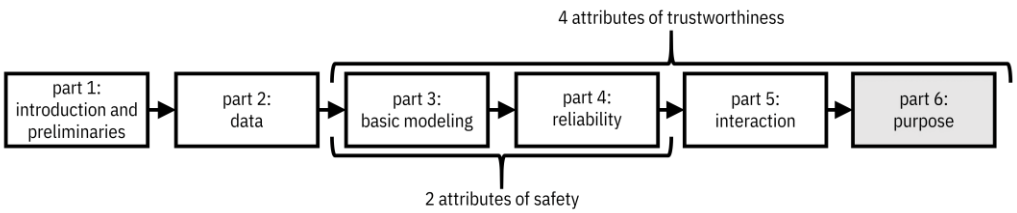


Figure 15.1. *Organization of the book. The sixth part focuses on the fourth attribute of trustworthiness, purpose, which maps to the use of machine learning that is uplifting.* Accessible caption. A flow diagram from left to right with six boxes: part 1: introduction and preliminaries; part 2: data; part 3: basic modeling; part 4: reliability; part 5: interaction; part 6: purpose. Part 6 is highlighted. Parts 3–4 are labeled as attributes of safety. Parts 3–6 are labeled as attributes of trustworthiness.

Benevolence implies the application of machine learning for good purposes. From a consequentionalist perspective (defined in chapter 14), we should broadly be aiming for good outcomes for all people. But a single sociotechnical system surely cannot do that. So we must ask: whose good? Whose interests will machine learning serve? Who can machine learning empower to achieve their goals?

The values encoded into machine learning systems are an ultimate expression of power. The most powerful can push for their version of 'good.' However, for machine learning systems to be worthy of trust, the final values cannot only be those that serve the powerful, but must also include the values of the most vulnerable. Chapter 14 explains technical approaches for bringing diverse voices into the value alignment process; here we try to understand what those voices have to say.

But before getting there, let's take a step back and think again about the governance of machine learning as a control system. What do we have to do to make it selfless and empowering for all? As shown in Figure 15.2, which extends Figure 14.5, there is a *paradigm*—a normative theory of how things should be done—that yields principles out of which values arise. The values then influence modeling.

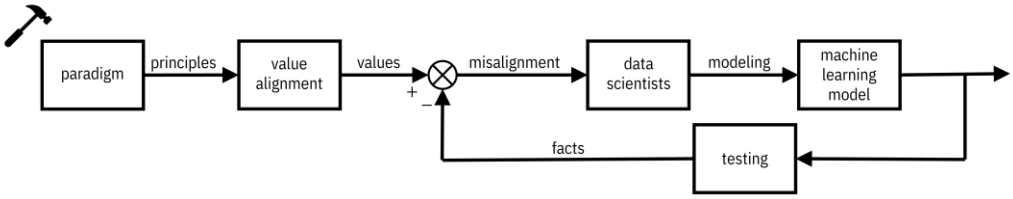


Figure 15.2. *A paradigm determines the principles by which values are constructed. The paradigm is one of the most effective points in the system to intervene to change its behavior.* Accessible caption. A block diagram that starts with a paradigm block with output principles. Principles are input to a value alignment block with output values. Facts are subtracted from values to yield misalignment. Misalignment is input to a data scientists block with modeling as output. Modeling is input to a machine learning model with output that is fed into a testing block. The output of testing is the same facts that were subtracted from values, creating a feedback loop. Paradigm is intervened upon, shown using a hammer.

There are many *leverage points* in such a complex system to influence how it behaves.[1] Twiddling with parameters in the machine learning model is a leverage point that may have some small effect. Computing facts quickly and bringing them back to data scientists is a leverage point that may have some slightly larger effect. But the most effective leverage point to intervene on is the paradigm producing the principles.[2] Therefore, in this chapter, we focus on different paradigms and the principles, codes, and guidelines that come from them.

15.1 Landscape of Principles

Over the last several years, different groups from different sectors and different parts of the world have created ethics principles for machine learning (and artificial intelligence more broadly) that espouse their paradigms. Organizations from private industry, government, and civil society (the third sector that is neither industry nor government, and includes non-governmental organizations (NGOs)) have produced normative documents at similar rates. Importantly, however, organizations in more

[1]Donella H. Meadows. *Thinking in Systems: A Primer*. White River Junction, Vermont, USA: Chelsea Green Publishing, 2008.
[2]More philosophically, Meadows provides an even more effective leverage point: completely transcending the idea of paradigms through enlightenment.

economically-developed countries have been more active than those in less economically-developed countries, which may exacerbate power imbalances. Moreover, the entire framing of ethics principles for machine learning is based on Western philosophy rather than alternative conceptions of ethics.[3] There are many similarities across the different sets of principles, but also key differences.[4]

First, let's look at the similarities. At a coarse-grained level, five principles commonly occur in ethics guidelines from different organizations:

1. privacy,
2. fairness and justice,
3. safety and reliability,
4. transparency (which usually includes interpretability and explainability), and
5. social responsibility and beneficence.

This list is not dissimilar to the attributes of trustworthiness that have guided the progression of the book. Some topics are routinely omitted from ethics principles, such as artificial general intelligence and existential threats (machines taking over the world), and the psychological impacts of machine learning systems.

Differences manifest when looking across sectors: governments, NGOs, and private corporations. Compared to the private sector, governments and NGOs take a more participatory approach to coming up with their principles. They also have longer lists of ethical principles beyond the five core ones listed above. Furthermore, the documents espousing their principles contain greater depth.

The topics of emphasis are different across the three sectors. Governments emphasize macroeconomic concerns of the adoption of machine learning, such as implications on employment and economic growth. NGOs emphasize possible misuse of machine learning. Private companies emphasize trust, transparency, and social responsibility. The remainder of the chapter drills down into these high-level patterns.

15.2 Governments

What is the purpose of government? Some of the basics are law and order, defense of the country from external threats, and general welfare, which includes health, well-being, safety, and morality of the people. Countries often create national development plans that lay out actions toward improving general welfare. In 2015, the member countries of the United Nations ratified a set of 17 sustainable development goals to achieve by 2030 that harmonize a unified purpose for national development. These global goals are:

1. end poverty in all its forms everywhere,

---

[3]Abeba Birhane. "Algorithmic Injustice: A Relational Ethics Approach." In: *Patterns* 2.2 (Feb. 2021), p. 100205. Ezinne Nwankwo and Belona Sonna. "Africa's Social Contract with AI." In: *ACM XRDS Magazine* 26.2 (Winter 2019), pp. 44–48.
[4]Anna Jobin, Marcello Ienca, and Effy Vayena. "The Global Landscape of AI Ethics Guidelines." In: *Nature Machine Intelligence* 1 (Sep. 2019), pp. 389–399. Daniel Schiff, Jason Borenstein, Justin Biddle, and Kelly Laas. "AI Ethics in the Public, Private, and NGO Sectors: A Review of a Global Document Collection." In *IEEE Transactions on Technology and Society* 2.1 (Mar. 2021), pp. 31–42.

2. end hunger, achieve food security and improved nutrition and promote sustainable agriculture,
3. ensure healthy lives and promote well-being for all at all ages,
4. ensure inclusive and equitable quality education and promote lifelong learning opportunities for all,
5. achieve gender equality and empower all women and girls,
6. ensure availability and sustainable management of water and sanitation for all,
7. ensure access to affordable, reliable, sustainable and modern energy for all,
8. promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all,
9. build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation,
10. reduce inequality within and among countries,
11. make cities and human settlements inclusive, safe, resilient and sustainable,
12. ensure sustainable consumption and production patterns,
13. take urgent action to combat climate change and its impacts,
14. conserve and sustainably use the oceans, seas and marine resources for sustainable development,
15. protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss,
16. promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels,
17. strengthen the means of implementation and revitalize the global partnership for sustainable development.

Toward satisfying the purpose of government, governmental AI ethics principles are grounded in the kinds of concerns stated in the sustainable development goals. Fairness and justice are a part of many of the goals, including goals five, ten, and sixteen, and also appear as a core tenet of ethics principles. Several other goals relate to social responsibility and beneficence.

Economic growth and productive employment are main aspects of goal eight and play a role in goals nine and twelve. Governments have an overriding fear that machine learning technologies will eliminate jobs through automation without creating others in their place. Therefore, as mentioned in the previous section, the economic direction is played up in governmental AI ethics guidelines and not so much in those of other sectors.

> "Our current trajectory automates work to an excessive degree while refusing to invest in human productivity; further advances will displace workers and fail to create new opportunities (and, in the process, miss out on AI's full potential to enhance productivity)."
>
> —Daron Acemoglu, economist at Massachusetts Institute of Technology

As part of this goal, there are increasing calls for a paradigm shift towards AI systems that complement or augment human intelligence instead of imitating it.[5]

Furthermore, towards both economic competitiveness and defense from external threats, some countries have now started engaging in a so-called arms race. Viewing the development of machine learning as a race may encourage taking shortcuts in safety and governance, which is cautioned against throughout this book.[6]

15.3 Private Industry

What is the purpose of a corporation? Throughout much of the last fifty years, the stated purpose of corporations (with some exceptions) has been to single-mindedly return profits to investors, also known as maximizing shareholder value.

> "There is one and only one social responsibility of business: to engage in activities designed to increase its profits."
>
> —Milton Friedman, economist at the University of Chicago

In 2019, however, the Business Roundtable, an association of the chief executives of 184 large companies headquartered in the United States, stated a broader purpose for corporations:

1. Delivering value to our customers. We will further the tradition of American companies leading the way in meeting or exceeding customer expectations.
2. Investing in our employees. This starts with compensating them fairly and providing important benefits. It also includes supporting them through training and education that help develop new skills for a rapidly changing world. We foster diversity and inclusion, dignity and respect.
3. Dealing fairly and ethically with our suppliers. We are dedicated to serving as good partners to the other companies, large and small, that help us meet our missions.
4. Supporting the communities in which we work. We respect the people in our communities and protect the environment by embracing sustainable practices across our businesses.
5. Generating long-term value for shareholders, who provide the capital that allows companies to invest, grow and innovate. We are committed to transparency and effective engagement with shareholders.

Shareholder value is listed only in the last item. Other items deal with fairness, transparency and sustainable development. AI ethics principles coming from corporations are congruent with this broadening purpose of the corporation itself, and are also focused on fairness, transparency and sustainable development.[7]

---

[5]Daron Acemoglu, Michael I. Jordan, and E. Glen Weyl. "The Turing Test is Bad for Business." In: *Wired* (Nov. 2021).
[6]Stephen Cave and Seán S. ÓhÉigeartaigh. "An AI Race for Strategic Advantage: Rhetoric and Risks." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New Orleans, Louisiana, USA, Feb. 2018, pp. 36–40.
[7]In January 2022, the Business Roundtable came out with 10 AI ethics principles of their own: (1) innovate with and for diversity, (2) mitigate the potential for unfair bias, (3) design for and implement transparency, explainability and interpretability,

> "I think we're in the third era, which is the age of integrated impact where we have created social impact that is part of the core value and function of the company overall."
>
> —Erin Reilly, chief social impact officer at Twilio

The 2019 statement by the Business Roundtable is not without criticism. Some argue that it is simply a public relations effort without accompanying actions that could lead to a paradigm change. Others argue it is a way for chief executives to lessen their accountability to investors.[8] AI ethics principles by corporations, especially those by companies developing machine learning technologies, face a similar criticism known as *ethics washing*—creating a façade of developing ethical or responsible machine learning that hides efforts that are actually very shallow.[9] An extreme criticism is that technology companies actively mislead the world about their true purpose and intentions with machine learning.[10]

15.4 Non-Governmental Organizations

NGOs are not homogeneous, but their purpose is usually to advance the political or social goals of their members. The purpose of an individual NGO is captured in its *theory of change*, which could include promoting human rights, improving the welfare of vulnerable groups and individuals, or protecting the environment. As the third sector (civil society), NGOs serve as a watchdog and counterbalance to governments and corporations by taking on roles that neither of the two are able or willing to fulfill. By filling this niche, they lead the criticism of governments and private industry either implicitly or explicitly. Activists in NGOs often try to shift power to the unprivileged.

Critical theory is the study of societal values with the purpose of revealing and challenging power structures; it is the foundation for several NGO theories of change. It includes subfields such as *critical race theory*, *feminism*, *postcolonialism*, and *critical disability theory*. Critical race theory challenges power structures related to race and ethnicity, with a particular focus on white supremacism and racism against blacks in the United States. Feminism is focused on power structures related to gender and challenging male supremacy. Postcolonialism challenges the legacy of (typically European) imperialism that continues to extract human and natural resources for the benefit of colonizers. Critical disability theory challenges ableism. The combinations of these different dimensions and others, known as *intersectionality* (first introduced in Chapter 10), are a key component of critical theory as well.

---

(4) invest in a future-ready AI workforce, (5) evaluate and monitor model fitness and impact, (6) manage data collection and data use responsibly, (7) design and deploy secure AI systems, (8) encourage a company-wide culture of responsible AI, (9) adapt existing governance structures to account for AI, and (10) operationalize AI governance throughout the whole organization.

[8]Lucian A. Bebchuk and Roberto Tallarita. "The Illusory Promise of Stakeholder Governance." In: *Cornell Law Review* 106 (2020), pp. 91–178.

[9]Elettra Bietti. "From Ethics Washing to Ethics Bashing: A View on Tech Ethics from Within Moral Philosophy." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 210–219.

[10]Mohamed Abdalla and Moustafa Abdalla. "The Grey Hoodie Project: Big Tobacco, Big Tech, and the Threat on Academic Integrity." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. Jul. 2021, pp. 287–297.

From the perspective of critical theory, machine learning systems tend to be instruments that reinforce hegemony (power exerted by a dominant group).[11] They extract data from vulnerable groups and at the same time, deliver harm to those same and other vulnerable groups. Therefore, the AI ethics principles coming from civil society often call for a disruption of the entrenched balance of power, particularly by centering the contexts of the most vulnerable and empowering them to pursue their goals.

> "A truly ethical stance on AI requires us to focus on augmentation, localized context and inclusion, three goals that are antithetical to the values justified by late-stage capitalism."
>
> —danah boyd, president of Data & Society Research Institute

As an example, the AI principles stated by an NGO that supports the giving of humanitarian relief to vulnerable populations are the following:[12]

1. weigh the benefits versus the risks: avoid AI if possible,
2. use AI systems that are contextually-based,
3. empower and include local communities in AI initiatives,
4. implement algorithmic auditing systems.

15.5 From Principles to Practice

The ethics principles from government, business, and civil society represent three different paradigms of normative values that may be encoded using the technical aspects of value alignment (described in Chapter 14) to specify the behavior of trustworthy machine learning systems. However, such specification will only happen when there is enough will, incentives, and devoted resources within an organization to make things happen. Intervening on the system's paradigm is an effective starting point, but cannot be the only leverage point that is intervened upon. Putting principles into practice involves several other leverage points as well.

The theory and methods for trustworthy machine learning start from algorithmic research. The incentives for typical machine learning researchers are centered on performance, generalization, efficiency, researcher understanding, novelty, and building on previous work.[13] Since there is now a

---

[11]Shakir Mohamed, Marie-Therese Png, and William Isaac. "Decolonial AI: Decolonial Theory as Sociotechnical Foresight in Artificial Intelligence" In: *Philosophy & Technology* 33 (Jul. 2020), pp. 659–684. Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. "Towards a Critical Race Methodology in Algorithmic Fairness." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Barcelona, Spain, Jan. 2020, pp. 501–512. Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜" In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Mar. 2021, pp. 610–623.

[12]Jasmine Wright and Andrej Verity. "Artificial Intelligence Principles for Vulnerable Populations in Humanitarian Contexts." Digital Humanitarian Network, Jan. 2020.

[13]Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, Michelle Bao. "The Values Encoded in Machine Learning Research." arXiv:2106.15590, 2021.

growing body of research literature on fairness, explainability, and robustness (they are 'hot topics'), the incentives for researchers are starting to align with the pursuit of research in technical trustworthy machine learning algorithms. Several open-source and commercial software tools have also been created in recent years to make the algorithms from research labs accessible to data scientists. But having algorithmic tools is also only one leverage point for putting ethical AI principles into practice. Practitioners also need the knowhow for affecting change in their organizations and managing various stakeholders. One approach for achieving organizational change is a checklist of harms co-developed with stakeholders.[14] Research is needed to further develop more playbooks for organization change.

Putting principles to practice is a process that has its own lifecycle.[15] The first step is a series of small efforts such as ad hoc risk assessments initiated by *tempered radicals* (people within the organization who believe in the change and continually take small steps toward achieving it). The second step uses the small efforts to demonstrate the importance of trustworthy machine learning and obtain the buy-in of executives to agree to ethics principles. The executives then invest in educating the entire organization on the principles and also start valuing the work of individuals who contribute to trustworthy machine learning practices in their organization. The impetus for executives may also come from external forces such as the news media, brand reputation, third-party audits, and regulations. The third step is the insertion of fact flow tooling (remember this was a way to automatically capture facts for transparency in Chapter 13) and fairness/robustness/explainability algorithms throughout the lifecycle of the common development infrastructure that the organization uses. The fourth step is instituting the requirement that diverse stakeholders be included in problem specification (value alignment) and evaluation of machine learning systems with veto power to modify or stop the deployment of the system. Simultaneously, this fourth step includes the budgeting of resources to pursue trustworthy machine learning in all model development throughout the organization.

15.6 Summary

- The purpose of trustworthy machine learning systems is to do good, but there is no single definition of good.
- Different definitions are expressed in ethics principles from organizations across the government, private, and social sectors.
- Common themes are privacy, fairness, reliability, transparency, and beneficence.
- Governments emphasize the economic implications of the adoption of machine learning.
- Companies stick primarily to the common themes.
- NGOs emphasize the centering and empowerment of vulnerable groups.

---

[14]Michael A. Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. "Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI." In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. Honolulu, Hawaii, USA, Apr. 2020, p. 318.

[15]Bogdana Rakova, Jingying Yang, Henriette Cramer, and Rumman Chowdhury. "Where Responsible AI Meets Reality: Practitioner Perspectives on Enablers for Shifting Organizational Practices." In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (Apr. 2021), p. 7. Kathy Baxter. "AI Ethics Maturity Model." Sep. 2021.

- A series of small actions can push an organization to adopt AI ethics paradigms and principles. The adoption of principles is an effective start for an organization to adopt trustworthy machine learning as standard practice, but not the only intervention required.
- Going from principles to practice also requires organization-wide education, tooling for trustworthy machine learning throughout the organization's development lifecycle, budgeting of resources to put trustworthy machine learning checks and mitigations into *all* models, and veto power for diverse stakeholders at the problem specification and evaluation stages of the lifecycle.

16

Lived Experience

Recall Sospital, the leading (fictional) health insurance company in the United States that tried to transform its care management system in Chapter 10 with fairness as a top concern. Imagine that you are a project manager at Sospital charged with reducing the misuse of opioid pain medications by members. An opioid epidemic began in the United States in the late 1990s and is now at a point that over 81,000 people die per year from opioid overdoses. As a first step, you analyze member data to understand the problem better. Existing machine learning-based opioid overdose risk models trained on data from state prescription drug monitoring programs (which may include attributes originating in law enforcement databases) have severe issues with data quality, consent, bias, interpretability, and transparency.[1] Also, the existing risk models are predictive machine learning models that can easily pick up spurious correlations instead of causal models that do not. So you don't want to take the shortcut of using the existing models. You want to start from scratch and develop a model that is trustworthy. Once you have such a model, you plan to deploy it to help human decision makers intervene in fair, responsible, and supportive ways.

You are starting to put a team together to carry out the machine learning lifecycle for the opioid model. You have heard the refrain that diverse teams are better for business.[2] For example, a 2015 study found that the top quartile of companies in gender and racial/ethnic diversity had 25% better financial performance than other companies.[3] In experiments, diverse teams have focused more on facts and been more innovative.[4] But do diverse teams create better, less biased, and more trustworthy machine

---

[1]Maia Szalavitz. "The Pain Was Unbearable. So Why Did Doctors Turn Her Away?" In: *Wired* (Aug. 2021).

[2]Among many other points that are used throughout this chapter, Fazelpour and De-Arteaga emphasize that the business case view on diversity is problematic because it takes the *lack* of diversity as a given and burdens people from marginalized groups to justify their presence. Sina Fazelpour and Maria De-Arteaga. "Diversity in Sociotechnical Machine Learning Systems." arXiv:2107.09163, 2021.

[3]The study was of companies in the Americas and the United Kingdom. Vivian Hunt, Dennis Layton, and Sara Prince. "Why Diversity Matters." McKinsey & Company, Jan. 2015.

[4]David Rock and Heidi Grant. "Why Diverse Teams are Smarter." In: *Harvard Business Review* (Nov. 2016).

learning models?[5] How and why? What kind of diversity are we talking about? In which roles and phases of the machine learning lifecycle is diversity a factor?

> "I believe diversity in my profession will lead to better technology and better benefits to humanity from it."
>
> —Andrea Goldsmith, electrical engineer at Stanford University

The first question is whether the team even affects the models that are created. Given the same data, won't all skilled data scientists produce the same model and have the same inferences? A real-world experiment assigned 29 teams of skilled data scientists an open-ended causal inference task of determining whether soccer referees are biased against players with dark skin, all using exactly the same data.[6] Due to different subjective choices the teams made in the problem specification and analysis, the results varied. Twenty teams found a significant bias against dark-skinned players, which means that nine teams did not. In another real-world example, 25 teams of data scientists developed mortality prediction models from exactly the same health data and had quite variable results, especially in terms of fairness with respect to race and gender.[7] In open-ended lifecycles, models and results may depend a lot on the team.

If the team matters, what are the characteristics of the team that matter? What should you be looking for as you construct a team for modeling individual patients' risk of opioid misuse? Let's focus on two team characteristics: (1) *information elaboration*: how do team members work together, and (2) *cognitive*: what do individual team members know. In the first characteristic: information elaboration, socioculturally non-homogeneous teams are more likely to slow down and consider critical and contentious issues; they are less apt to take shortcuts.[8] Such a slowdown is not prevalent in homogeneous teams and importantly, does not depend on the team members having different sets of knowledge. All of the team members could know the critical issues, but still not consider them if the members are socioculturally homogeneous.

You have probably noticed quotations sprinkled throughout the book that raise issues relevant to the topic of a given section. You may have also noticed that the people quoted have different sociocultural backgrounds, which may be different than yours. This is an intentional feature of the book. Even if they are not imparting knowledge that's different from the main text of the book, the goal of the quotes is for you to hear these voices so that you are pushed to slow down and not take shortcuts. (Not taking shortcuts is a primary theme of the book.)

---

[5]Caitlin Kuhlman, Latifa Jackson, and Rumi Chunara. "No Computation Without Representation: Avoiding Data and Algorithm Biases Through Diversity." arXiv:2002.11836, 2020.

[6]Raphael Silberzahn and Eric L. Uhlmann. "Crowdsourced Research: Many Hands Make Tight Work." In: *Nature* 526 (Oct. 2015), pp. 189–191.

[7]Timothy Bergquist, Thomas Schaffter, Yao Yan, Thomas Yu, Justin Prosser, Jifan Gao, Guanhua Chen, Łukasz Charzewski, Zofia Nawalany, Ivan Brugere, Renata Retkute, Alidivinas Prusokas, Augustinas Prusokas, Yonghwa Choi, Sanghoon Lee, Junseok Choe, Inggeol Lee, Sunkyu Kim, Jaewoo Kang, Patient Mortality Prediction DREAM Challenge Consortium, Sean D. Mooney, and Justin Guinney. "Evaluation of Crowdsourced Mortality Prediction Models as a Framework for Assessing AI in Medicine." medRxiv:2021.01.18.21250072, 2021.

[8]Daniel Steel, Sina Fazelpour, Bianca Crewe, and Kinley Gillette. "Information Elaboration and Epistemic Effects of Diversity." In: *Synthese* 198.2 (Feb. 2021), pp. 1287–1307.

Sociocultural differences are associated with differences in *lived experience* of marginalization.[9] Remember from Chapter 1 that lived experience is the personal knowledge you have gained through direct involvement in things from which you have no option to escape. Related to the second characteristic of the team: what the team members know, one key cognitive theory relevant for trustworthy machine learning is that people with lived experience of marginalization have an *epistemic advantage*: when people reflect on their experience of being oppressed, they are better able to understand all sides of power structures and decision-making systems than people who have not been oppressed.[10] Briefly mentioned in Chapter 4, they have a bifurcated consciousness that allows them to walk in the shoes of both the oppressed and the powerful. In contrast, privileged people tend to have blind spots and can only see their own perspective.

> "People with marginalized characteristics—so people who had experienced discrimination—had a deeper understanding of the kinds of things that could happen to people negatively and the way the world works in a way that was a bit less rosy."
>
> —Margaret Mitchell, research scientist at large
>
> "The lived experiences of those directly harmed by AI systems gives rise to knowledge and expertise that must be valued."
>
> —Emily Denton, research scientist at Google
>
> "Technical know-how cannot substitute for contextual understanding and lived experiences."
>
> —Meredith Whittaker, research scientist at New York University

In modern Western science and engineering, knowledge derived from lived experience is typically seen as invalid; often, only knowledge obtained using the scientific method is seen as valid. This contrasts with critical theory, which has knowledge from the lived experience of marginalized people at its very foundation. Given the many ethics principles founded in critical theory covered in Chapter 15, it makes sense to consider lived experience in informing your development of a model for opioid misuse risk. Toward this end, in the remainder of the chapter, you will:

- map the cognitive benefit of the lived experience of team members to the needs and requirements of different phases of the machine learning lifecycle, and
- formulate lifecycle roles and architectures that take advantage of that mapping.

---

[9]Neurodiversity is not touched upon in this chapter, but is another important dimension that could be expanded upon.
[10]Natalie Alana Ashton and Robin McKenna. "Situating Feminist Epistemology." In: *Episteme* 17.1 (Mar. 2020), pp. 28–47.

16.1 Lived Experience in Different Phases of the Lifecycle

The first stage in the lifecycle of developing an opioid misuse risk model is problem specification and value alignment. In this phase, there is a clear need for the inclusion of people with different lived experiences to question assumptions and identify critical issues in the four levels of value alignment covered in Chapter 14: whether you should work on the problem, which pillars of trustworthiness are of concern, how to measure performance in those pillars, and acceptable ranges of metrics. The epistemic advantage of these team members is critical in this phase. The blind spots of team members who have not experienced systematic disadvantages will prevent them from noticing all the possible misuses and harms that can arise from the system, such as undue denials of care to traditionally marginalized individuals. This is the phase in which participatory design, also covered in Chapter 14, should be used.[11]

> "New perspectives ask new questions and that's a fact. This is exactly why inclusion matters!"
>
> —Deb Raji, fellow at Mozilla

The second phase, data understanding, requires digging into the available data and its provenance to identify the possible bias and consent issues detailed in Chapter 4 and Chapter 5. This is another phase in which it is important for the team to be critical, and it is useful to have members with epistemic advantage. In Chapter 10, we already saw that the team developing the Sospital care management system needed to recognize the bias against African Americans when using health cost as a proxy for health need. Similarly, a diagnosis for opioid addiction in a patient's data implies that the patient actually interacted with Sospital for treatment, which will also be biased against groups that are less likely to utilize the health care system. Problem owners, stakeholders, and data scientists from marginalized groups are more likely to recognize this issue. Furthermore, a variety of lived experiences will help discover that large dosage opioid prescriptions from veterinarians in a patient's record are for their pets, not for them; prescription claims for naltrexone, an opioid itself, represent treatment for opioid addiction, not evidence of further misuse; and so on.

The third phase in developing an opioid misuse model is data preparation. You can think of data preparation in two parts: (1) data integration and (2) feature engineering. Critique stemming from lived experience has little role to play in data integration because of its mechanical and rote nature. Is this also the case in the more creative feature engineering part? Remember from Chapter 10 that biases may be introduced in feature engineering, such as by adding together different health costs to create a single column. Such biases may be spotted by team members who are advantaged in looking for them. However, if dataset constraints, such as dataset fairness metric constraints, have already been included in the problem specification of the opioid misuse model in anticipation of possible harms, then no additional epistemic advantage is needed to spot the issues. Thus, there is less usefulness of lived experience of marginalization among team members in the data preparation stage of the lifecycle.

---

[11] Vinodkumar Prabhakaran and Donald Martin Jr. "Participatory Machine Learning Using Community-Based System Dynamics." In: *Health and Human Rights Journal* 22.2 (Dec. 2020), pp. 71–74.

In the fourth phase of the lifecycle, the team will take the prepared data and develop an individualized causal model of factors that lead to opioid misuse.[12] Coming after the problem specification phase that sets forth the modeling task and the performance metrics, and after the data understanding and data preparation phases that finalize the dataset, the modeling phase is not open-ended like the soccer referee and mortality prediction tasks described in the previous section. The modeling is quite constrained from the perspective of the data scientist.

A recent study tasked 399 data scientists, each working alone, with developing models of the mathematical literacy of people based on approximately five hundred of their biographical features; the dataset and basic performance metrics were clearly specified (no fairness metric was specified).[13] Importantly, the dataset had many data points and was purposefully and carefully collected as a representative sample without population biases. Thus, the dataset had negligible epistemic uncertainty. The study analyzed the 399 models that were created and found no significant relationship between the unwanted bias of the models and the sociocultural characteristics of the data scientists that produced them.

In this example and other similar regimented and low-epistemic uncertainty modeling tasks, the lived experience of the team is seemingly of low importance. In contrast, when there is great epistemic uncertainty like you may have in analyzing opioid abuse, the inductive bias of the model chosen by the data scientist has a great role to play and the lived experience of the data scientist can become important. However, mirroring the argument made earlier about an explicit problem specification lessening the epistemic advantage for members of marginalized groups in feature engineering, a clear specification of all relevant trust metric dimensions also lessens the usefulness of lived experience in modeling.

Evaluating the opioid risk model once it has been created is not as straightforward as simply testing it for the specified allowable trust metric ranges in the ways described in Chapter 14. Once a model is tangible, you can manipulate it in various ways and better imagine the harms it could lead to. Thus, being critical of the model during evaluation is also a job better done by a team that has members who have experienced systematic disadvantage and are attuned to the negative impacts it may have if it is deployed within Sospital's operations.

Finally, if the model has passed the evaluation stage, the ML operations engineers on the team carry out the deployment and monitoring phase of the lifecycle. Their role is primarily to ensure technical integration with Sospital's other systems and noting when the trust metric ranges elicited during value alignment are violated over time. This is another phase of the lifecycle in which there is not much epistemic advantage to be had by a team containing engineers with lived experience of marginalization.

Overall, as shown in Figure 16.1, three lifecycle phases (problem specification, data understanding, and evaluation) can take advantage of having a diverse team containing members that have lived experience of marginalization. The other three phases (data preparation, modeling, and deployment and monitoring) benefit less from the epistemic advantage of team members with lived experience of

---

[12]Chirag Nagpal, Dennis Wei, Bhanukiran Vinzamuri, Monica Shekhar, Sara E. Berger, Subhro Das, and Kush R. Varshney. "Interpretable Subgroup Discovery in Treatment Effect Estimation with Application to Opioid Prescribing Guidelines." In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. Apr. 2020, pp. 19–29.

[13]Bo Cowgill, Fabrizio Dell'Acqua, Samuel Deng, Daniel Hsu, Nakul Verma, and Augustin Chaintreau. "Biased Programmers? Or Biased Data? A Field Experiment in Operationalizing AI Ethics." In: *Proceedings of the ACM Conference on Economics and Computation*. Jul. 2020, pp. 679–681.

systematic harm. This conclusion suggests a particular lifecycle architecture for developing your opioid risk model, discussed in the next section.

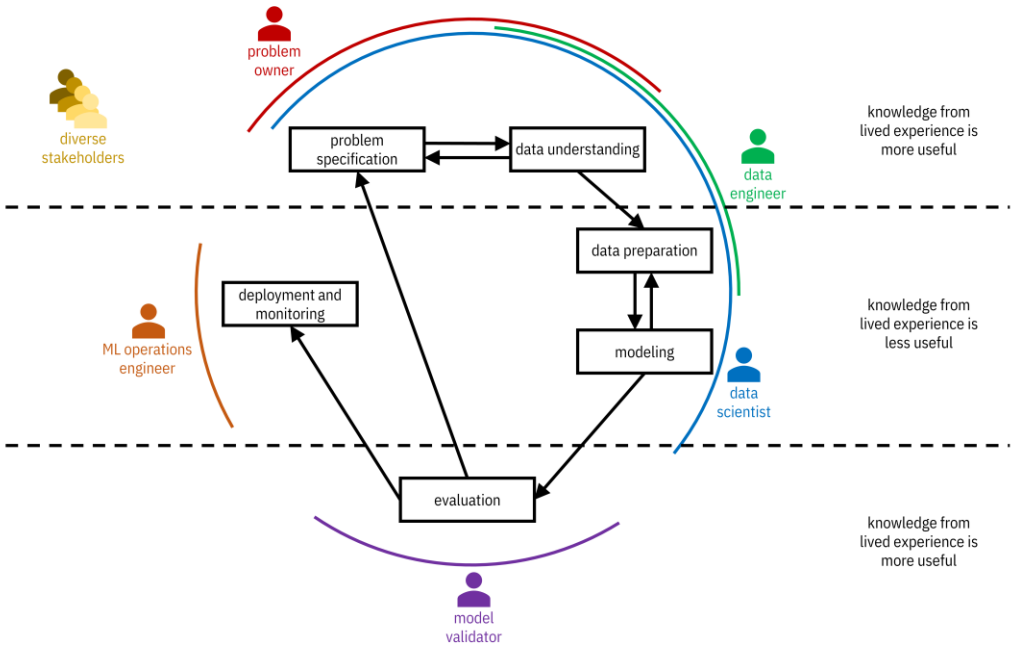


Figure 16.1. *The different phases of the machine learning lifecycle delineated by how useful knowledge from lived experience is. Knowledge from lived experience is more useful in problem specification, data understanding, and evaluation. Knowledge from lived experience is less useful in data preparation, modeling, and deployment and monitoring.* Accessible caption. A diagram of the development lifecycle is marked according to which phases find lived experience more useful and less useful.

16.2 Inclusive Lifecycle Architectures

From the previous section, you have learned that having a diverse team with lived experience of systematic harms is most important in the problem specification, data understanding, and evaluation phases. These phases coincide with the phases in which the problem owner and model validator personas are most prominent. Problem owners often tend to be subject matter experts about the application and are not usually skilled at the technical aspects of data engineering and modeling. They may or may not come from marginalized backgrounds. Given the power structures in place at many corporations, including Sospital, the problem owners often come from privileged backgrounds. Even if that is not true at Sospital, it is strongly suggested to have a panel of diverse voices, including those from marginalized groups, participate and be given a voice in these phases of the lifecycle.

That leaves the other three phases. What about them? The analysis suggests that as long as the specification and validation are done with the inclusion of team members and panelists with lived

experience of oppression,[14] then any competent, reliable, communicative, and selfless data engineers, data scientists, and ML operations engineers equipped with the tools and training in trustworthy machine learning will create a trustworthy opioid misuse risk model irrespective of their lived experience. The pool of skilled data scientists at Sospital does not include many individuals with lived experience, and you also don't want to levy a 'minority tax'—the burden of extra responsibilities placed on minority employees in the name of diversity—on the ones there are. So you go with the best folks available, and that is perfectly fine. (Machine learning researchers creating the tools for practitioners should have a variety of lived experiences because researchers have to both pose and answer the questions. Fortuitously, though their numbers are small overall, researchers from groups traditionally underrepresented in machine learning and associated with marginalization are seemingly overrepresented in research on *trustworthy* machine learning, as opposed to other areas of machine learning research.[15])

If the lived experience of the data scientists and engineers on the team is less relevant for building trustworthy machine learning systems, what if the data scientists and engineers are not living beings at all? Technology advances are leading to a near-future state in which feature engineering and modeling will be mostly automated, using so-called auto ML. Algorithms will construct derived features, select hypothesis classes, tune hyperparameters of machine learning algorithms, and so on. As long as these auto ML algorithms are themselves trustworthy,[16] then it seems as though they will seamlessly enter the lifecycle, interact with problem owners and model validators, and successfully create a trustworthy model for opioid misuse.

Shown in Figure 16.2, in this near-future, auto ML instead of data scientists is the controller in the control theory perspective on governance introduced in Chapter 14 and Chapter 15. And this is a-okay. Such an architecture involving auto ML empowers problem owners and marginalized communities to pursue their goals without having to rely on scarce and expensive data scientists. This architecture enables more democratized and accessible machine learning for Sospital problem owners when paired with *low-code/no-code* interfaces (visual software development environments that allow users to create applications with little or no knowledge of traditional computer programming).

> "It's about humans at the center, it's about those unnecessary barriers, where people have domain expertise but have difficulty teaching the machine about it."
>
> —Christopher Re, computer scientist at Stanford University

[14]Those specifications and validations must also be given true power. This point is discussed later using the terminology 'participation washing'.

[15]Yu Tao and Kush R. Varshney. "Insiders and Outsiders in Research on Machine Learning and Society." arXiv:2102.02279, 2021.

[16]Jaimie Drozdal, Justin Weisz, Dakuo Wang, Gaurav Dass, Bingsheng Yao, Changruo Zhao, Michael Muller, Lin Ju, and Hui Su. "Trust in AutoML: Exploring Information Needs for Establishing Trust in Automated Machine Learning Systems." In: *Proceedings of the International Conference on Intelligent User Interfaces*. Mar. 2020, pp. 297–307.

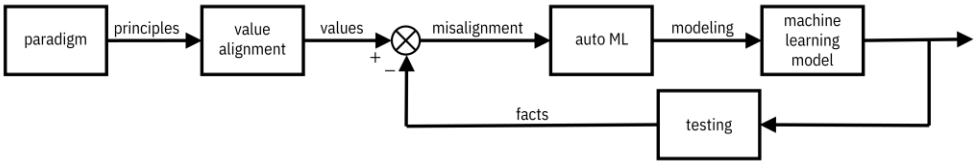


Figure 16.2. *The control theory perspective of AI governance with auto ML technologies serving as the controller instead of data scientists.* Accessible caption. A block diagram that starts with a paradigm block with output principles. Principles are input to a value alignment block with output values. Facts are subtracted from values to yield misalignment. Misalignment is input to an auto ML block with modeling as output. Modeling is input to a machine learning model with output that is fed into a testing block. The output of testing is the same facts that were subtracted from values, creating a feedback loop.

A recent survey of professionals working within the machine learning lifecycle asked respondents their preference for auto ML in different lifecycle phases.[17] The respondents held different lifecycle personas. The preferred lifecycle phases for automation were precisely those in which lived experience is less important: data preparation, modeling, and deployment and monitoring. The phases of the lifecycle that respondents did not care to see automation take hold were the ones where lived experience is more important: problem specification, data understanding, and evaluation. Moreover, respondents from the problem owner persona desired the greatest amount of automation, probably because of the empowerment it provides them. These results lend further credence to an architecture for machine learning development that emphasizes inclusive human involvement in the 'takeoff' (problem specification and data understanding) and 'landing' (evaluation) phases of the lifecycle while permitting 'auto pilot' (auto ML) in the 'cruising' (data preparation and modeling) phase.

Another recent survey showed that machine learning experts were more likely to call for strong governance than machine learning non-experts.[18] This result suggests that problem owners may not realize the need for explicit value alignment in an automated lifecycle. Therefore, the empowerment of problem owners should be only enabled in architectures that place the elicitation of paradigms and values at the forefront.

> "Participation-washing could be the next dangerous fad in machine learning."
>
> —Mona Sloane, sociologist at New York University

Before concluding the discussion on inclusive lifecycle architectures, it is important to bring up *participation washing*—uncredited and uncompensated work by members of marginalized groups.[19]

---

[17]Dakuo Wang, Q. Vera Liao, Yunfeng Zhang, Udayan Khurana, Horst Samulowitz, Soya Park, Michael Muller, and Lisa Amini. "How Much Automation Does a Data Scientist Want?" arXiv:2101.03970, 2021.

[18]Matthew O'Shaughnessy, Daniel Schiff, Lav R. Varshney, Christopher Rozell, and Mark Davenport. "What Governs Attitudes Toward Artificial Intelligence Adoption and Governance?" osf.io/pkeb8, 2021.

[19]Mona Sloane, Emanuel Moss, Olaitan Awomolo, and Laura Forlano. "Participation is Not a Design Fix for Machine Learning." arXiv:2007.02423, 2020. Bas Hofstra, Vivek V. Kulkarni, Sebastian Munoz-Najar Galvez, Bryan He, Dan Jurafsky, and Daniel A. McFarland. "The Diversity–Innovation Paradox in Science." In: *Proceedings of the National Academy of Sciences of the United States of America* 117.17 (Apr. 2020), pp. 9284–9291.

Participatory design sessions that include diverse voices, especially those with lived experience of marginalization, have to be credited and compensated. The sessions are not enough if they turn out to be merely for show. The outcomes of those sessions have to be backed by power and upheld throughout the lifecycle of developing the opioid abuse model. Otherwise, the entire architecture falls apart and the need for team members with lived experience returns to all phases of the lifecycle.

Leaving aside the difficult task of backing the inputs of marginalized people with the power they need to be given, how should you even go about bringing together a diverse panel? From a practical perspective, what if you are working under constraints?[20] Broad advertising and solicitations from entities that vulnerable people don't know may not yield many candidates. More targeted recruitment in specific social media groups and job listing sites may be somewhat better, but will still miss certain groups. Unfortunately, there are no real shortcuts. You have to develop relationships with institutions serving different communities and with members of those communities. Only then will you be able to recruit people to participate in the problem specification, data understanding, and evaluation phases (either as employees or simply as one-time panelists) and be able to do what you know that you should.

16.3 Summary

- The model produced in a machine learning lifecycle depends on characteristics of the team.
- Teams that are socioculturally heterogeneous tend to slow down and not take shortcuts.
- Team members with lived experience of marginalization have an epistemic advantage in noticing potential harms.
- This epistemic advantage from lived experience is most important in the problem specification, data understanding, and evaluation stages of the lifecycle. It is less important in the data preparation, modeling, and deployment and monitoring stages.
- A sensible architecture for the lifecycle focuses on inclusion of team members with lived experience of systematic harm in the three phases in which they have epistemic advantage.
- The other three phases may be sensibly carried out by trustworthy data scientists and engineers, or even trustworthy auto ML algorithms, which may be empowering for problem owners.

[20]Fernando Delgado, Stephen Yang, Michael Madaio, and Qian Yang. "Stakeholder Participation in AI: Beyond 'Add Diverse Stakeholders and Stir.'" In: *Proceedings of the NeurIPS Human-Centered AI Workshop*. Dec. 2021.

17

Social Good

As you know from Chapter 7 and Chapter 13, the (fictional) information technology company JCN Corporation has several data science teams that work on different problems faced by the company and its customers. JCN Corporation's chief executive is a member of the Business Roundtable and a signatory to broadening the values of private industry from being solely driven by shareholders to being driven by other stakeholders too (this was covered in Chapter 15). In this environment, you recall that the fourth attribute of trustworthiness includes beneficence and helping others. Toward this end, you have the idea to start a *data science for social good* program at JCN to engage those data science teams part-time to conduct projects that directly contribute to uplifting humanity.

> "Imagine what the world would look like if we built products that weren't defined by what the market tells us is profitable, but instead what our hearts tell us is essential."
>
> —Vilas Dhar, president of Patrick J. McGovern Foundation

Taking a consequentialist view (remember consequentialism from Chapter 14), 'social impact' or 'making a difference' is promoting the total wellbeing of humanity (in expected value over the long term without sacrificing anything that might be of comparable moral importance).[1] But what does that really mean? And whose good or whose value of wellbeing are we talking about?

[1]Benjamin Todd. "What Is Social Impact? A Definition." URL: https://80000hours.org/articles/what-is-social-impact-definition, Nov. 2021.

> "The phrase 'data science for social good' is a broad umbrella, ambiguously defined. As many others have pointed out, the term often fails to specify good for whom."
>
> —Rachel Thomas, data scientist at Queensland University of Technology

It is dangerous for you to think that you or the data science teams at JCN Corporation are in position to determine what is an appropriate problem specification to uplift the most vulnerable people in the world. Data science for social good is littered with examples of technologists taking the shortcut of being paternalistic and making that determination themselves. If your data science teams are diverse and include people with lived experience of marginalization (see Chapter 16), then maybe they will be less paternalistic and push to have diverse, external problem owners.

> "Most technologists from the Global North are often not self-aware and thus look at problems in the Global South through the lens of technology alone. In doing so, they inevitably silence the plurality of perspectives."
>
> —Patrick Meier, co-CEO of WeRobotics

But who should those external problem owners be? Your first inclination is to look towards international development experts from large well-established governmental and non-governmental organizations, and consulting the seventeen UN Sustainable Development Goals (SDGs) listed in Chapter 15. But as you investigate further, you realize that there were a lot of struggles of power and politics that went into determining the SDGs; in particular, the lower-level targets beneath the seventeen goals may not represent the views of the most vulnerable.[2] You also learn that international development overall has many paternalistic tendencies and is also littered with projects that make no sense. Some may even be harmful to the people they intend to uplift.

> "Find algorithms that benefit people on their own terms."
>
> —Jacob Metcalf, technology ethicist at Data & Society Research Institute

Thus, while taking inspiration from the high-level topics touched on by the SDGs, you decide on the following theory of change for the JCN data science for social good program you are creating. Using machine learning, you will empower smaller, innovative social change organizations that explicitly include the knowledge of the vulnerable people they intend to uplift when they work towards social impact. (Collectively, civil society organizations and social enterprises—for-profit businesses that have social impact as their main goal—are known as *social change organizations*.) Toward developing a social good program within JCN Corporation, in this chapter you will:

- evaluate past data science for social good projects,

---

[2]Serge Kapto. "Layers of Politics and Power Struggles in the SDG Indicator Process." In: *Global Policy* 10.S1 (Jan. 2019), pp. 134–136.

- formulate a lifecycle for achieving a successful data science for social good program, and
- sketch out empowering machine learning architectures and platforms for promoting social good.

Before jumping into it, a few words on how you can gain internal support within JCN Corporation to devote resources to the program. There are several value propositions that go beyond appealing to the broadening stakeholder values that JCN Corporation is adopting and beyond appealing to the potential for positive public relations. First, machine learning problem specifications in social impact applications tend to have different constraints than those found in information technology and enterprise applications. Constraints are the mother of innovation, and so working on these problems will lead to new innovations for JCN. Second, by partnering with civil society organizations, JCN Corporation will receive valuable feedback and public references about its machine learning tools that enterprise customers may be unwilling to provide. Public references that allow JCN to tout its capabilities are distinct from positive public relations because they do not depend on the goodness of the application. Third, working on these projects attracts, retains, and grows the skills of talented data scientists in JCN Corporation. Fourth, if the program is run on JCN Corporation's cloud computing platform, the platform's usage will grow. Tax deductions for charitable giving are conspicuously absent from the value propositions because JCN Corporation will be receiving product feedback and possible cloud usage from the social change organizations.

17.1 Evaluating Data Science for Social Good

Throughout the book, you have taken on roles in several (fictional) social change organizations, including as a project manager with m-Udhār Solar (the provider of pay-as-you-go solar energy), as a data scientist with Unconditionally (the distributor of unconditional cash transfers), as a data scientist collaborator of ABC Center (the integrated social services provider), and as a problem owner with Alma Meadow (the granter of two-year fellowships to budding social entrepreneurs). Moreover, although the (fictional) Bank of Bulandshahr and Wavetel were launching the Phulo mobile telephony-based lending service with for-profit motives, the service is a vehicle for financial inclusion and upliftment. Thus, you have already seen some examples of projects that fall under the data science for social good umbrella. Their non-fictionalized counterparts were conducted as partnerships between social change organizations and data scientists acting in a 'for social good' capacity.[3]

[3]Hugo Gerard, Kamalesh Rao, Mark Simithraaratchy, Kush R. Varshney, Kunal Kabra, and G. Paul Needham. "Predictive Modeling of Customer Repayment for Sustainable Pay-As-You-Go Solar Power in Rural India." In: *Proceedings of the Data for Good Exchange Conference*. New York, New York, USA. Sep. 2015. Brian Abelson, Kush R. Varshney, and Joy Sun. "Targeting Direct Cash Transfers to the Extremely Poor." In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, New York, USA, Aug. 2014, pp. 1563–1572. Debarun Bhattacharjya, Karthikeyan Shanmugam, Tian Gao, Nicholas Mattei, Kush R. Varshney, and Dharmashankar Subramanian. "Event-Driven Continuous Time Bayesian Networks." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, New York, USA, Feb. 2020, pp. 3259–3266. Aditya Garg, Alexandra Olteanu, Richard B. Segal, Dmitriy A. Katz-Rogozhnikov, Keerthana Kumar, Joana Maria, Liza Mueller, Ben Beers, and Kush R. Varshney. "Demystifying Social Entrepreneurship: An NLP Based Approach to Finding a Social Good Fellow." In: *Proceedings of the Data Science for Social Good Conference*. Chicago, Illinois, USA, Sep. 2017. Skyler Speakman, Srihari Sridharan, and Isaac Markus. "Three Population Covariate Shift for Mobile Phone-based Credit Scoring." In: *Proceedings of the ACM SIGCAS Conference on Computing and Sustainable Societies*. Menlo Park, California, USA, Jun. 2018, p. 20.

17.1.1 What is Data Science for Social Good?

By happenstance, the examples in the book cover a narrow swath of possible uses of machine learning and artificial intelligence in social good. Moreover, due to the scope of the book, all but the ABC Center use case are focused on classification problems that lead to the allocation of something that, at face value, appears to be favorable to the recipient. The ABC Center example is focused on causal inference and causal discovery. Surveys of the data science for social good landscape find projects touching the following categories , which have a great deal of alignment with the SDGs:[4]

- accessibility,
- agriculture,
- education,
- environment,
- financial inclusion,
- health care,
- infrastructure (e.g. urban planning and transportation),
- information verification and validation,
- public safety and justice, and
- social work,

and touching the following technical approaches from artificial intelligence:

- supervised learning,
- reinforcement learning,
- computer vision,
- natural language processing,
- robotics,
- knowledge representation and reasoning,
- planning and scheduling,
- constraint satisfaction,

and many others.

Both lists are extremely vast and encompassing. As such, you should not think of social good as an application area of machine learning and AI, but as a paradigm or value system (paradigms are discussed in Chapter 15 as the precursor to values). Do not simply train a model on some dataset you downloaded that relates to agriculture or infrastructure or misinformation; that is not data science for social good. Do not create a system that helps privileged individuals discover which farmers markets

[4]Michael Chui, Martin Harryson, James Manyika, Roger Roberts, Rita Chung, Ashley van Heteren, and Pieter Nel. "Notes from the AI Frontier: Applying AI for Social Good." McKinsey & Company, Dec. 2018. Zheyuan Ryan Shi, Claire Wang, and Fei Fang. "Artificial Intelligence for Social Good: A Survey." arXiv:2001.01818, 2020.

currently have an inventory of kale; that is not data science for social good.[5] Data science for social good requires social change organizations to be problem owners who state the problem specification based on the lived experiences of their beneficiaries (and even better, bring their beneficiaries to a panel of diverse voices to inform the project).

Needless to say, the data science for social good you do in your program at JCN Corporation must be imbued with data privacy and consent, along with the first three attributes of trustworthy machine learning: competence, reliability (including fairness and robustness), and interaction (including explainability, transparency, and value alignment). This is especially the case because these systems are affecting the most vulnerable members of society.

17.1.2 How Has Data Science for Social Good Been Conducted?

Surveys of the data science for social good landscape find that nearly all efforts have been conducted as one-off projects that involve the development of a custom-tailored solution, irrespective of whether they are carried out as data science competitions, weekend volunteer events, longer term volunteer-based consulting engagements, student fellowship programs, corporate philanthropy, specialized non-governmental organizations, or dedicated innovation teams of social change organizations.

Creating such one-off solutions requires a great deal of time and effort both from the social change organization and the data scientists. There is limited reuse of assets and learnings from one project to the next because (1) every new project involves a different social change organization and (2) data scientists acting as volunteers are unable to conduct a sequence of several projects over time. Moreover, these projects typically require the social change organization to integrate machine learning solutions with their other systems and practices, to deploy those solutions, and monitor and maintain the solutions over time themselves. Very few social change organizations are equipped to do such 'last-mile' implementation, partly because their funding typically does not allow them to invest time and resources into building up technological capacity.

The confluence of all these factors has led to the state we are in: despite the data science for social good movement being nearly a decade long, most projects continue to only be demonstrations without meaningful and lasting impact on social change organizations and their constituents.[6] A project lasting a few months may show initial promise, but then is not put into practice and does not 'make a difference.'

17.2 A Lifecycle of a Data Science for Social Good Program

As you envision the JCN Corporation data science for social good program, you want to avoid the pitfalls that others have experienced in the past. But can your program jump right to the end goal of doing high-impact work, or is there an evolution it must go through? Sorry, there are no shortcuts. Just like an artist's or scientist's hot-streak of high-impact work begins with an exploration[7] phase that touches a

---

[5]Jake Porway. "You Can't Just Hack Your Way to Social Change." In: *Harvard Business Review* (Mar. 2013). URL: https://hbr.org/2013/03/you-cant-just-hack-your-way-to.

[6]Kush R. Varshney and Aleksandra Mojsilović. "Open Platforms for Artificial Intelligence for Social Good: Common Patterns as a Pathway to True Impact." In: *Proceedings of the ICML AI for Social Good Workshop*. Long Beach, California, USA, Jul. 2019.

[7]Lu Liu, Nima Dehmamy, Jillian Chown, C. Lee Giles, and Dashun Wang. "Understanding the Onset of Hot Streaks Across Artistic, Cultural, and Scientific Careers." In: *Nature Communications* 12 (Sep. 2021), p. 5392.

diversity of topics (which is then followed by a narrowly-focused exploitation phase that produces the impact),[8] a data science for social good program needs to begin broadly and go through the following three-step lifecycle, illustrated in Figure 17.1.

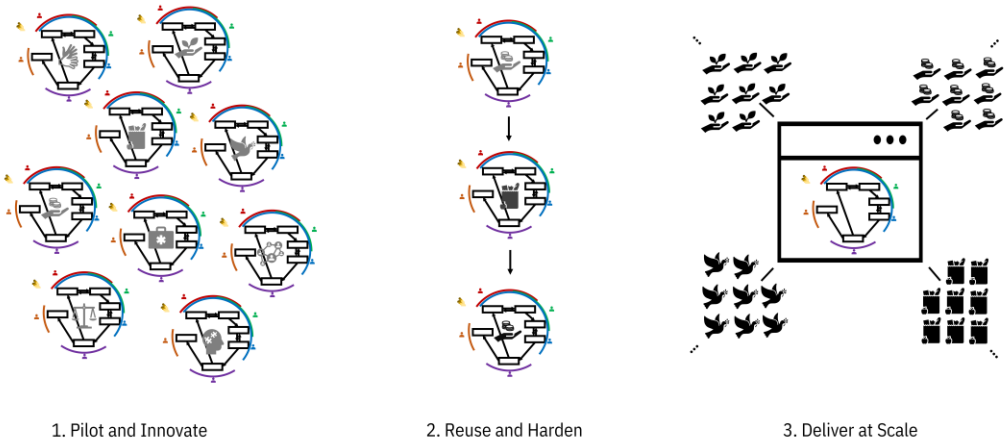


Figure 17.1. *Illustration of the three phases of the lifecycle of a data science for social good program: (1) piloting and innovating with a portfolio of projects, (2) reusing and hardening solutions to the common patterns, and (3) creating a usable platform that can reach a lot of social change organizations.* Accessible caption. Step 1, pilot and innovate, shows several different development lifecycles with icons for different social good applications in their center, colored gray to indicate they are not yet hardened. Step 2, reuse and harden, shows a sequence of three development lifecycles in which the social good application icon gets progressively darker to black to indicate hardening. Step 3, deliver at scale shows a development lifecycle inside a computer window illustrating its incorporation into a platform, touching tens of social good applications.

1. *Pilot and innovate.* You should conduct several individual projects to learn about the needs of social change organizations that may be addressed by machine learning. In this phase, your data scientists will also gain the experience of conducting multiple projects and start seeing commonalities across them. While doing so, JCN Corporation will gain from new innovations under new constraints. You can choose to be somewhat intentional in the application area of social good to match corporate values or in the technical area of machine learning to match technical areas of interest, but not overly so.
2. *Reuse and harden.* Once you have several projects under your belt, you must step back and analyze the common patterns that emerge. Your goal at this stage is to develop common algorithms or algorithmic toolkits to address those common patterns in as reusable a way as possible. You want to meet the needs of multiple social change organizations using a common model or algorithm. This type of machine learning innovation is unique; most data scientists and machine learning researchers are not trained to step back and abstract things in this way, so it will be a

[8]The word 'exploit' is used in a positive sense here, but is used in a negative sense later in the chapter.

challenge. However, this sort of insight and innovation is precisely the feedback that will be helpful for JCN Corporation's teams developing software tools and products for conducting data science.

3. *Deliver at scale.* Those common reusable algorithms will not make high impact until they are made available within an environment that low-resourced and low-skilled social change organizations can be empowered to tweak, use, and maintain. (Refer to inclusive low-code/no-code architectures in Chapter 16 for a related discussion.) The delivery will likely be 'as-a-service' on JCN Corporation's cloud-based environment. Software-as-a-service is software that is licensed as a subscription, is centrally hosted, and is accessed by users using a web browser. Therefore, integration with other systems is greatly simplified and the responsibility for maintenance falls on JCN Corporation rather than the social change organization.

You are probably comfortable with the first phase of this data science for social good program lifecycle. As long as you ensure that social change organizations—representing the interests of their beneficiaries who have lived experience of vulnerability—are the problem owners and involved in evaluation, then the JCN Corporation data scientists can approach the portfolio of projects in this phase in a manner they are used to.

The second phase presupposes that there *are* common patterns in social good projects that can be addressed using common models or algorithms. Evidence is starting to mount that this is indeed the case. For example, the same algorithm for bandit data-driven optimization is used in social good applications as varied as feeding the hungry and stopping wildlife poachers.[9] As a second example, most of the social good use cases (fictionally) covered in the book are quite different from each other, but are all fair allocation problems posed as binary classification that can be addressed using a common algorithmic toolkit such as AI Fairness 360, a library of fairness metrics and bias mitigation algorithms.[10] Moreover, large language models have been fine-tuned for several disparate social good domains such as collecting evidence for drug repurposing and simplifying text for people with low-literacy or cognitive disability.[11] (Large language models are a kind of foundation model; introduced in Chapter 4, foundation models are machine learning models trained on large-scale data that can be fine-tuned for specific problems.)

The third phase of the lifecycle of a social good program is mostly unproven as yet, but is what you should be working toward in the program you intend to start at JCN Corporation. The result is an accessible and inclusive data science for social good *platform* that is described in the next section.

---

[9]Zheyuan Ryan Shi, Zhiwei Steven Wu, Rayid Ghani, and Fei Fang. "Bandit Data-Driven Optimization: AI for Social Good and Beyond." arXiv:2008.11707, 2020.

[10]Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. "AI Fairness 360: An Extensible Toolkit for Detecting and Mitigating Algorithmic Bias." In: *IBM Journal of Research and Development* 63.4/5 (Jul./Sep. 2019), p. 4.

[11]Shivashankar Subramanian, Ioana Baldini, Sushma Ravichandran, Dmitriy A. Katz-Rogozhnikov, Karthikeyan Natesan Ramamurthy, Prasanna Sattigeri, Kush R. Varshney, Annmarie Wang, Pradeep Mangalath, and Laura B. Kleiman. "A Natural Language Processing System for Extracting Evidence of Drug Repurposing from Scientific Publications." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, New York, USA, Feb. 2020, pp. 13369–13381. Sanja Stajner. "Automatic Text Simplification for Social Good: Progress and Challenges." In: *Findings of the Association for Computational Linguistics*. Aug. 2021, pp. 2637–2652.

Before getting there, two comments on the term 'scale.' Scaling is a paradigm seen as paramount in much of the technology industry, and is the main reason to pursue a digital platform that can be built once and used by many. However, scaling is not the mission of many social change organizations; although some would like to grow, many would like to remain small with a very pointed mission.[12] Moreover, scaling as an overriding paradigm is not free from criticism and can be seen as a means for exploiting the most vulnerable.[13] In creating a social good program and platform with JCN Corporation, your goal is to make the work of all social change organizations easier, irrespective of whether they would like to scale themselves. You can control any possible exploitation by centering the values of the most vulnerable throughout the development lifecycle.

17.3 A Data Science for Social Good Platform

A foundation model, algorithm, or algorithmic toolkit that applies broadly to the problems faced by many social change organizations is an excellent start, but it is not enough to satisfy your theory of change. These technological artifacts alone do not empower social change organizations because they require a level of skill and infrastructure in data science and engineering that the organizations typically lack. Incongruously, social change organizations are typically low-resourced just like the people they serve, especially in comparison to private corporations embracing machine learning with large data science teams. You can say that social change organizations are at the 'bottom of the pyramid' among organizations. (In its typical usage, the term 'bottom of the pyramid' refers to the socioeconomically poorest or least wealthy group of people.)

A washing machine, stove, or ultrasound imaging machine designed for a wealthy, high-resourced context will not cut it in a low-resourced context. The core technology has to be put into a form factor that makes sense for bottom-of-the-pyramid users. The same is true of machine learning for social change organizations. In general, bottom-of-the-pyramid innovation has the following twelve principles:[14]

1. focus on (quantum jumps in) price performance;
2. hybrid solutions, blending old and new technology;
3. scalable and transportable operations across countries, cultures and languages;
4. reduced resource intensity: eco-friendly products;
5. identify appropriate functionality;
6. build logistical and manufacturing infrastructure;
7. deskill (services) work;
8. educate (semiliterate) customers in product usage;

---

[12]Anne-Marie Slaughter. "Thinking Big for Social Enterprise Can Mean Staying Small." In: *Financial Times* (Apr. 2018). URL: https://www.ft.com/content/86061a82-46ce-11e8-8c77-ff51caedcde6.

[13]Katherine Ye. "Silicon Valley and the English Language." URL: https://book.affecting-technologies.org/silicon-valley-and-the-english-language/. Jul. 2020.

[14]C. K. Prahalad. *The Fortune at the Bottom of the Pyramid: Eradicating Poverty Through Profits*. Upper Saddle River, New Jersey, USA: Wharton School Publishing, 2005.

9. products must work in hostile environments;
10. adaptable user interface to heterogeneous consumer bases;
11. distribution methods designed to reach both highly dispersed rural markets and highly dense urban markets; and
12. focus on broad architecture, enabling quick and easy incorporation of new features.

What are the important points among these principles for a machine learning platform that empowers social change organizations and what is a platform anyway?

A digital platform is a collection of people, processes, and internet-based tools that enable users to develop and run something of value. Therefore, a machine learning platform contains web-based software tools to carry out the entire machine learning development lifecycle from the problem specification phase all the way to the deployment and monitoring phase, with roles for all the personas including problem owners, data scientists, data engineers, model validators, operations engineers, and diverse voices. Importantly, a machine learning platform is more than simply an off-the-shelf programming library for modeling.

In fact, there are three kinds of machine learning capabilities: (1) off-the-shelf machine learning packages, (2) machine learning platforms, and (3) bespoke machine learning builds.[15] At one extreme, off-the-shelf packages are useful for top-of-the-pyramid organizations with a high level of data science skill and a high level of resources, but not for bottom-of-the-pyramid social change organizations. At the other extreme, bespoke or custom-tailored development (which has been the predominant mode of data science for social good over the last decade) should only be used for extremely complex problems or when an organization needs a technological competitive advantage. These are not the circumstances in which social change organizations typically operate; usually their problems, although having unique constraints, are not overly complicated from a machine learning perspective and usually their advantages in serving their beneficiaries are non-technological. Thus, it makes sense to be *just right* and serve social change organizations using machine learning platforms.

What does a machine learning platform for the bottom of the pyramid entail? Translating the twelve general principles to a machine learning platform for social change implies a focus on appropriate functionality, adaptable user interfaces, deskilling, broad architecture, distribution methods, and education. You'll obtain appropriate functionality by paring down the machine learning capabilities to the core model, algorithm, or toolkit that is reusable by many different social change organizations with similar needs, as discussed earlier. Such foundational capabilities mean that the algorithms have to be created only once and can be improved by a dedicated machine learning team that is not reliant on, or part of, any one social change organization.

The remaining aspects touch on the last-mile problem. You can achieve adaptable user interfaces and deskilling by following the inclusive architecture presented in Chapter 16 for people with lived experience of marginalization. Such an architecture takes the scarce and expensive skill of data scientists out of the development lifecycle through low-code/no-code and auto ML. Low-code/no-code and auto ML should make it easy to configure and fine-tune the machine learning capability for the

[15]Andrew Burgess. *The Executive Guide to Artificial Intelligence: How to Identify and Implement Applications for AI in Your Organization*. London, England, UK: Palgrave Macmillan, 2017.

specific task being approached by the social change organization. It should also be easy to slightly change the definition of an outcome variable and apply the model to a new setting with slightly different features. The interface should also provide a data catalog and tools for managing data. Moreover, the interface should include meaningful and easy to consume visualizations of the output predictions. The focus should be to simplify, simplify, simplify, but not so much that you are left with something meaningless.

A web and cloud-based platform is specifically designed to support quick and easy incorporation of new capabilities. Any improvements to the machine learning diffuse to social change organizations automatically. Similarly, cloud-based platforms are designed in a way that allow broad distribution to any device anywhere there is an internet connection. This method of delivery is starting to lead to turnkey deployment and monitoring of machine learning systems.

Finally, the last component of a machine learning platform for social impact is education: teaching and reference materials, tutorials, how-to guides, examples, etc. presented in the language of social change organizations. It must be presented in a way that people starting at different skill levels all have an on-ramp to the content. An important part of the education for members of social change organizations is sparking the imagination of what's possible using machine learning in the social impact sector. A persona that has not come up in the book so far, a *broker* who bridges the gap between members of social change organizations and the data science world by translating and aligning the concepts used in each field, is very useful in the education component of the platform.[16]

Have you noticed something? All of the desirable attributes of a machine learning platform seem to be desirable not only for empowering social change organizations, but also desirable for any organization, including ones at the top of the pyramid. And that is the beauty of bottom-of-the-pyramid innovation: it is good old innovation that is useful for everyone including JCN Corporation's enterprise customers.

Beyond the design and the ease of use of the platform, a critical aspect for you to sustainably bring the platform and overall data science for social good program to fruition is winning the support of large grantmaking foundations that fund social change organizations. First, the foundations must give some sort of implicit permission to social change organizations to use the platform and provide them enough leeway in their budgets to get started. Second, in a similar vein as international development projects specified without the perspective of vulnerable people, there are many international development efforts whose funding did not provision for maintenance and long-term support beyond the initial headline. JCN Corporation will not be able to sustain a data science for social good platform you create without grants for its maintenance, so you'll need to line up funding. Foundations are beginning to see the need to support technology efforts among their grantees,[17] but are not yet ready to fund a platform operated by a private corporation.

You have your work cut out for you to launch a data science for social good program at JCN Corporation and push it along the lifecycle beyond just the initial set of projects to common algorithms

---

[16]Youyang Hou and Dakuo Wang. "Hacking with NPOs: Collaborative Analytics and Broker Roles in Civic Data Hackathons." In: *Proceedings of the ACM on Human-Computer Interaction* 1.CSCW (Nov. 2017), p. 53.

[17]Michael Etzel and Hilary Pennington. "Time to Reboot Grantmaking." In: *Stanford Social Innovation Review*. URL: https://ssir.org/articles/entry/time_to_reboot_grantmaking, Jun. 2017.

and then a scalable platform. But with enough conviction, wherewithal, and luck, you just might be able to pull it off. Go forth, you genuine do-gooder![18]

17.4 Summary

- Data science for social good—using machine learning in a beneficent way—is not an application area for machine learning, but a paradigm and value system.
- The goal is to empower social change organizations in the development of machine learning systems that help uplift vulnerable people on their own terms.
- The decade-long experience with data science for social good has rarely yielded truly impactful results because individual projects fail to overcome the last-mile problem.
- Social change organizations are typically low-resourced and need much more than just code or a custom solution to be able to use machine learning in their operations.
- Machine learning platforms that are specifically designed to deskill data science needs and minimize the effort for deployment, maintenance, and support are the solution. Such platforms should be built around common algorithmic patterns in the social impact space that you start seeing by conducting several projects over a lifecycle.
- All the attributes of trustworthy machine learning are essential in applying machine learning for social impact, including fairness, robustness, explainability, and transparency.

[18]William D. Coplin. *How You Can Help: An Easy Guide to Doing Good Deeds in Your Everyday Life*. New York, New York, USA: Routledge, 2000.

18

Filter Bubbles and Disinformation

Imagine that you're a technology executive who is unhappy with the stranglehold that a handful of companies have on how people receive information via ad-supported social media timelines, recommendations, and search engines. Your main issue with these 'big tech' companies is the filter bubbles, disinformation, and hate speech festering on their platforms that threaten a functioning non-violent society. Many of these phenomena result from machine learning systems that help the platforms maximize engagement and revenue. Economists call these considerations that extend beyond revenue maximization for the company and are detrimental to society *negative externalities*. According to your values, recommendation and search to maximize engagement are problems that should not even be worked on in their currently prevailing paradigm because they have consequences on several of the items listed in Chapter 14 (e.g. disinformation, addiction, surveillance state, hate and crime).

> "The best minds of my generation are thinking about how to make people click ads. That sucks."
>
> —Jeff Hammerbacher, computer scientist at Facebook

In recent months, you have seen an upstart search engine enter the fray that is not ad-driven and is focused on 'you,' with 'you' referring to the user and the user's information needs. This upstart gives you a glimmer of hope that something new and different can possibly break through the existing monopolies. However, your vision for something new is not centered on the singular user 'you', but on plural society. Therefore, you start planning a (fictional) search engine and information recommendation site of your own with a paradigm that aims to keep the negative externalities of the current ad/engagement paradigm at bay. Recalling a phrase that the conductor of your symphonic band used to say before concerts: "I nod to you and up we come," you name your site Upwe.com.

Does Upwe.com have legs? Can a search engine company really focus on serving a broader and selfless purpose? Many would argue that it is irrational to neither focus on solely serving the user (to make it attractive for paying subscribers) nor maximizing the platform's engagement (to maximize the

company's ad revenue). However, as you learned in Chapter 15, corporations are already moving toward broadening their purpose from maximizing shareholder value to maximizing the value for a larger set of stakeholders. And by focusing on the collective 'we,' you are appealing to a different kind of ethics: relationality instead of rationality. *Relational ethics* asks people to include considerations beyond themselves (which is the scope of rational ethics), especially their relationships with other people and the environment in determining the right action. One effect of relational thinking is bringing negative externalities to the forefront and mitigating an extractive or colonial mindset, including in the context of machine learning.[1]

So coming back to the original question: is Upwe.com tenable? Does your vision for it have any hope? In this chapter, you'll work toward an answer by:

- sketching the reasons why society is so reliant on the digital platforms of 'big tech,'
- examining the paradigm that leads to echo chambers, disinformation, and hate speech in greater detail, and
- evaluating possible means for countering the negative externalities.

18.1 Epistemic Dependence and Institutional Trust

As you're well aware, the amount of knowledge being created in our world is outpacing our ability to understand it. And it is only growing more complex. The exponential increase in digital information has been a boon for machine learning, but perhaps not so much for individual people and society. There is so much information in the modern world that it is impossible for any one person, on their own, to have the expertise to really understand or judge the truth of even a sliver of it. These days, even expert scientists do not understand the intricacies of all parts of their large-scale experimental apparatus.[2] Known as *epistemic dependence*, people have to rely on others to interpret knowledge for them. You've already learned about epistemic uncertainty (lack of knowledge) and epistemic advantage (knowledge of harms possessed by people with lived experience of marginalization) in Chapter 3 and Chapter 16, respectively. Epistemic dependence is along the same lines: obtaining knowledge you lack from people who possess it, trusting them without being able to verify the truth of that knowledge yourself.

The people from whom you can obtain knowledge now includes anyone anywhere at lightning speed from their messages, articles, blog posts, comments, photos, podcasts, and videos on the internet. Epistemic dependence no longer has any bounds, but the space of knowledge is so vast that it requires search engines and recommendation algorithms to deal with retrieving the information. And what is going on behind the scenes is almost never clear to the user of a search engine. It is something abstract and mysterious in the ether. Even if the seeker of knowledge were aware of an information retrieval algorithm's existence, which is typically based on machine learning, its workings would not be comprehensible. So not only do you have to trust the source and content of the knowledge, but also the

[1]Sabelo Mhlambi. "From Rationality to Relationality: Ubuntu as an Ethical and Human Rights Framework for Artificial Intelligence Governance." Harvard University Carr Center Discussion Paper Series 2020-009, Jul. 2020.
[2]Matthew Hutson. "What Do You Know? The Unbearable Vicariousness of Knowledge." In: *MIT Technology Review* 123.6 (Nov./Dec. 2020), pp. 74–79.

closed-box system bringing it to you.[3] Nonetheless, people cannot entirely abdicate their epistemic responsibility to try to verify either the knowledge itself, its source, or the system bringing it forward.

From the very beginning of the book, the trustworthiness of machine learning systems has been equated to the trustworthiness of individual other people, such as coworkers, advisors, or decision makers. This framing has followed you throughout the journey of becoming familiar with trustworthy machine learning: going from competence and reliability to interaction and selflessness. However, when discussing the trustworthiness of the machine learning backing information filtering in digital platforms, this correspondence breaks down. To the general public, the machine learning is beyond the limits of their knowledge and interaction to such a degree that the machine learning model is not an individual person any longer, but an institution like a bank, post office, or judicial system. It is just there. Members of the public are not so much users of machine learning as they are subject to machine learning.[4] And institutional trust is different from interpersonal trust.

Public trust in institutions is not directed towards a specific aspect, component or interaction with the institution, but is an overarching feeling about something pervasive. The general public does not go in and test specific measures of the trustworthiness of an institution like they may with a person, i.e. assessing a person's ability, fairness, communication, beneficence, etc. (or even care to know the results of such an assessment). Members of the public rely on the system itself having the mechanisms in place to ensure that it is worthy of trust. The people's trust is built upon mechanisms such as governance and control described in Chapter 14, so these mechanisms need to be understandable and not require epistemic dependence. To understand governance, people need to understand and agree with the values that the system is working to align itself toward. Thus as you envision Upwe.com, you must give your utmost attention to getting the paradigm right and making the values understandable to anyone. Putting these two things in place will enable the public to make good on their epistemic responsibility. Remember from Chapter 15 that intervening on the paradigm is the most effective leverage point of a system and is why the focus of this chapter is on the paradigm rather than on tackling negative externalities more directly, such as methods for detecting hate speech.

18.2 Maximizing Engagement, or Not

So how can you get the paradigm and values right? There are many things that you can do, but the main one is to deprioritize engagement as the primary goal. Engagement or attention is often measured by a user's time on the platform and by their number of clicks. Maximizing engagement can lead to the extreme of the user becoming addicted to the platform.

---

[3]Boaz Miller and Isaac Record. "Justified Belief in a Digital Age: On the Epistemic Implications of Secret Internet Technologies." In: *Episteme* 10.2 (Jun. 2013), pp. 117–134.

[4]Bran Knowles and John T. Richards. "The Sanction of Authority: Promoting Public Trust in AI." In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Mar. 2021, pp. 262–271.

> "When you're in the business of maximizing engagement, you're not interested in truth. You're not interested in harm, divisiveness, conspiracy. In fact, those are your friends."
>
> —Hany Farid, computer scientist at University of California, Berkeley

First, let's see how single-mindedly valuing engagement leads to the harms of echo chambers, disinformation, and hate speech. The end of the section will briefly mention some alternatives to engagement maximization.

18.2.1 Filter Bubbles and Echo Chambers

When a recommendation system shows a user only content related to their interests, connections, and worldview, they are in a *filter bubble*. But how do filter bubbles relate to maximizing engagement with a digital platform? This kind of curation and personalization keeps serving the user content that they enjoy, which keeps them coming back for more of the same. Pleasant and fun things attract our attention.

> "When you see perspectives that are different from yours, it requires thinking and creates aggravations. As a for-profit company that's selling attention to advertisers, Facebook doesn't want that, so there's a risk of algorithmic reinforcement of homogeneity, and filter bubbles."
>
> —Jennifer Stromer-Galley, information scientist at Syracuse University

In an *echo chamber*, a person is repeatedly presented with the same information without any differences of opinion. This situation leads to their believing in that information to an extreme degree, even when it is false. Filter bubbles often lead to echo chambers. Although filter bubbles may be considered a helpful act of curation, by being in one, the user is not exposed to a diversity of ideas. They suffer from *epistemic inequality*.[5] Recall from Chapter 16 that diversity leads to information elaboration—slowing down to think about contentious issues. Thus, by being in a filter bubble, people are apt to take shortcuts, which can lead to a variety of harms.

18.2.2 Misinformation and Disinformation

What are those fun things that attract us? Anything that is surprising attracts our attention.[6] There are only so many ways that you can make the truth surprising before it becomes old hat.[7] Permutations and combinations of falsehoods can continue to be surprising for much longer and thus keep a user more

---

[5]Shoshana Zuboff. "Caveat Usor: Surveillance Capitalism as Epistemic Inequality." In: *After the Digital Tornado*. Ed. by Kevin Werbach. Cambridge, England, UK: Cambridge University Press, 2020.
[6]Laurent Itti and Pierre Baldi. "Bayesian Surprise Attracts Human Attention." In: *Vision Research* 49.10 (Jun. 2009), pp. 1295–1306.
[7]Lav R. Varshney. "Limit Theorems for Creativity with Intentionality." In: *Proceedings of the International Conference on Computational Creativity*. Sep. 2020, pp. 390–393.

engaged on a platform. Moreover, people spread false news significantly faster on social media platforms than true news.[8]

> "Having constructed a technological apparatus that disseminates information instantaneously and globally without regard to its veracity, we shouldn't be surprised that this apparatus has left us drowning in lies."
>
> —Mark Pesce, futurist

Clickbait is one example of false, surprising, and attractive content that drives engagement. It is a kind of *misinformation* (a falsehood that may or may not have been deliberately created to mislead) and also a kind of *disinformation* (a falsehood that was purposefully created to mislead). In fact, 'big tech' companies have been found to finance so-called clickbait farms to drive up their platforms' engagement.[9]

> "Misinformation tends to be more compelling than journalistic content, as it's easy to make something interesting and fun if you have no commitment to the truth."
>
> —Patricia Rossini, communications researcher at University of Liverpool

Another type of disinformation enabled by machine learning is *deepfakes*. These are images or videos created with the help of generative modeling that make it seem as though a known personality is saying or doing something that they did not say or do. Deepfakes are used to create credible messaging that is false.

Although some kinds of misinformation can be harmless, many kinds of disinformation can be extremely harmful to individuals and societies. For example, Covid-19 anti-vaccination disinformation on social media in 2021 led to vaccination hesitancy in many countries, which led to greater spread of the disease and death. Other disinformation has political motives that are meant to destabilize a nation.

18.2.3 Hate Speech and Inciting Violence

Whether false or true (disinformation or not), hate speech (abusive language against a particular group) attracts attention. Traditional media typically does not disseminate hate speech. The terms and conditions of many social media platforms also do not allow for hate speech and provide mechanisms for users to flag it. Nevertheless, since the problem of defining and moderating hate speech at the scale of worldwide digital platforms is difficult, much hate speech does get posted in social media platforms and then amplified via information filtering algorithms because it is so engaging.

---

[8]Soroush Vosoughi, Deb Roy, and Sinan Aral. "The Spread of True and Fake News Online." In: *Science* 359.6380 (Mar. 2018), pp. 1146–1151.

[9]Karen Hao. "How Facebook and Google Fund Global Misinformation." In: *MIT Technology Review*. URL: https://www.technologyreview.com/2021/11/20/1039076/facebook-google-disinformation-clickbait, 2021.

Messages on social media platforms and actions in the real world are closely intertwined.[10] Hate speech, offensive speech, and messages inciting violence on digital platforms foment many harms in the physical world. Several recent instances of hateful violence, such as against the Rohingya minority in Myanmar in 2018 and the United States Capitol Building in 2021, have been traced back to social media.

18.2.4 Alternatives

You've seen how maximizing engagement leads to negative externalities in the form of real-world harms. But are there proven alternatives you could use in the machine learning algorithm running Upwe.com's information retrieval system instead? Partly because there are few incentives to work on the problem among researchers within 'big tech,' and because researchers elsewhere do not have the ability to try out or implement any ideas that they may have, the development of alternatives has been few and far between.[11]

Nevertheless, as you develop the paradigm for Upwe.com, the following are a few concepts that you may include. You may want the platform to maximize the truth of the factual information that the user receives. You may want the platform to always return content from a diversity of perspectives and expose users to new relations with which they may form a diverse social network.[12] You may wish to maximize some longer-term enjoyment for the user that they themselves might not realize is appropriate for them at the moment; this paradigm is known as *extrapolated volition*. Such concepts may be pursued as pre-processing, during model training, or as post-processing, but they would be limited to only those that you yourself came up with.[13] A participatory value alignment process that includes members of marginalized groups would be even better to come up with all of the concepts you should include in Upwe.com's paradigm.

Furthermore, you need to have transparency in the paradigm you adopt so that all members of society can understand it. Facts and factsheets (covered in Chapter 13) are useful for presenting the lower-level test results of individual machine learning models, but not so much for institutional trust (except as a means for trained auditors to certify a system). CP-nets (covered in Chapter 14) are understandable representations of values, but do not reach all the way back to the value system or paradigm. It is unclear how to document and report the paradigm itself, and is a topic you should experiment with as you work on Upwe.com.

---

[10]Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush R. Varshney. "The Effect of Extremist Violence on Hateful Speech Online." In: *Proceedings of the AAAI International Conference on Web and Social Media*. Stanford, California, USA, Jun. 2018, pp. 221–230.

[11]Ivan Vendrov and Jeremy Nixon. "Aligning Recommender Systems as Cause Area." In: *Effective Altruism Forum*. May 2019.

[12]Jianshan Sun, Jian Song, Yuanchun Jiang, Yezheng Liu, and Jun Li. "Prick the Filter Bubble: A Novel Cross Domain Recommendation Model with Adaptive Diversity Regularization." In: *Electronic Markets* (Jul. 2021).

[13]Jonathan Stray, Ivan Vendrov, Jeremy Nixon, Steven Adler, and Dylan Hadfield-Menell. "What Are You Optimizing For? Aligning Recommender Systems with Human Values." In: *Proceedings of the ICML Participatory Approaches to Machine Learning Workshop*. Jul. 2020.

18.3 Taxes and Regulations

There are few incentives for existing, entrenched platforms to pursue paradigms different from engagement maximization in the capitalist world we live in. Upwe.com will find it very difficult to break in without other changes. Short of completely upending society to be more relational via structures such as village-level democracy and self-reliance promoted by Mahatma Gandhi or anarchism,[14] the primary ways to control the harms of maximizing engagement are through government-imposed taxes and regulation spurred by a change in societal norms.[15] The norms should value the wellbeing of all people above all else. Viewing machine learning for information filtering as an institution rather than an individual, it is not surprising that the people who support interventions for controlling the negative externalities of the systems are those who have strong trust in institutions.[16] Society may already be on a path to demanding greater control of digital media platforms.[17]

While building up and developing Upwe.com, you should take a page out of Henry Heinz's playbook (remember from the preface that in addition to developing trustworthy processed food products, he lobbied for the passage of the Pure Food and Drug Act) and push for stronger regulations. Some possible regulations recommended by the Aspen Institute are:[18]

1. *High reach content disclosure*. Companies must regularly report on the content, source, and reach of pieces of knowledge that receive high engagement on their platform.
2. *Content moderation disclosure*. Companies must report the content moderation policies of their platform and provide examples of moderated content to qualified individuals.
3. *Ad transparency*. Companies must regularly report key information about every ad that appears on their platform.
4. *Superspreader accountability*. People who spread disinformation that leads to real-world negative consequences are penalized.
5. *Communications decency control on ads and recommendation systems*. Make companies liable for hateful content that spreads on their platform due to the information filtering algorithm, even if it is an ad.

Many of these recommended regulations enforce transparency since it is a good way of building institutional trust. However, they do not provide governance on the paradigm underlying the platform because it is difficult to measure the paradigm. Nevertheless, they will control the paradigm to some extent. If social media platforms are deemed *public utilities* or *common carriers*, like telephone and electricity providers, then even more strict regulations are possible. Importantly, if you have designed

---

[14]Brian Martin. *Nonviolence versus Capitalism*. London, England, UK: War Resisters' International, 2001.
[15]Daron Acemoglu. "AI's Future Doesn't Have to Be Dystopian." In: *Boston Review*. URL: https://bostonreview.net/forum/ais-future-doesnt-have-to-be-dystopian/, 2021.
[16]Emily Saltz, Soubhik Barari, Claire Leibowicz, and Claire Wardle. "Misinformation Interventions are Common, Divisive, and Poorly Understood." In: *Harvard Kennedy School Misinformation Review* 2.5 (Sep. 2021).
[17]Throughout the chapter, the governance of platforms is centered on the needs of the general public, but the needs of legitimate content creators are just as important. See: Li Jin and Katie Parrott. "Legitimacy Lost: How Creator Platforms Are Eroding Their Most Important Resource." URL: https://every.to/means-of-creation/legitimacy-lost, 2021.
[18]Katie Couric, Chris Krebs, and Rashad Robinson. *Aspen Digital Commission on Information Disorder Final Report*. Nov. 2021.

Upwe.com to already be on the right side of regulations when they become binding, you will have a leg up on other platforms and might have a chance of being sustainable.

In parallel, you should also try to push for direct ways of controlling the paradigm rather than controlling the negative externalities because doing so will be more powerful. Regulations are one recognized way of limiting negative externalities; Pigouvian taxes are the other main method recognized by economists. A *Pigouvian tax* is precisely a tax on a negative externality to discourage the behaviors that lead to it. A prominent example is a tax on carbon emissions levied on companies that pollute the air. In the context of social media platforms, the tax would be on every ad that was delivered based on a targeting model driven by machine learning.[19] Such a tax would directly push 'big tech' companies to change their paradigm while leaving the Upwe.com paradigm alone.

Seeing out your vision of an Upwe.com that contributes to the wellbeing of all members of society may seem like an insurmountable challenge, but do not lose hope. Societal norms are starting to push for what you want to build, and that is the key.

18.4 Conclusion

- There is so much and such complicated knowledge in our world today that it is impossible for anyone to understand it all, or even to verify it. We all have epistemic dependence on others.
- Much of that dependence is satisfied by content on the internet that comes to us on information platforms filtered by machine learning algorithms. The paradigm driving those algorithms is maximizing the engagement of the user on the platform.
- The engagement maximization paradigm inherently leads to side effects such as filter bubbles, disinformation, and hate speech, which have real-world negative consequences.
- The machine learning models supporting content recommendation on the platforms is so disconnected from the experiences of the general public that it does not make sense to focus on models' interpersonal trustworthiness, which has been the definition of trustworthiness throughout the book. An alternative notion of institutional trustworthiness is required.
- Institutional trustworthiness is based on governance mechanisms and their transparency, which can be required by government regulations if there is enough societal pressure for them. Transparency may help change the underlying paradigm, but taxes may be a stronger direct push.
- A new paradigm based on relational ethics is needed, which centers truth, a diversity of perspectives, and wellbeing for all.

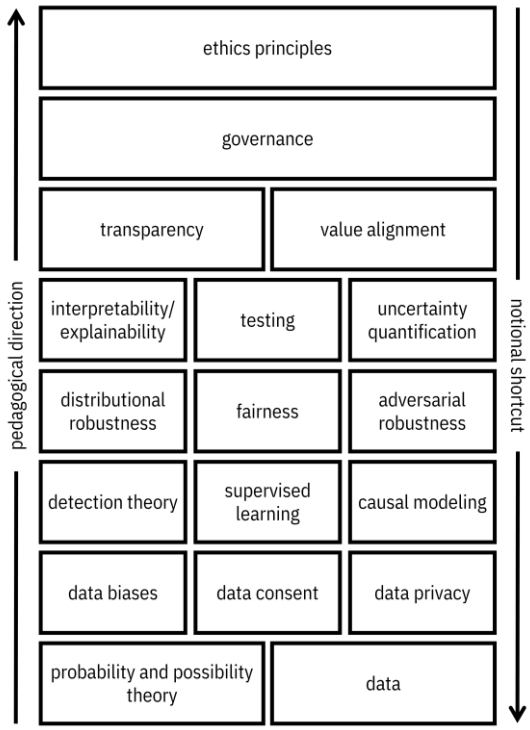
> "I nod to you and up we come."
>
> —Norbert Buskey, band teacher at Fayetteville-Manlius High School

[19]Paul Romer. "A Tax To Fix Big Tech." In: *New York Times* (7 May 2019), p. 23.

Shortcut

Even though I have admonished you throughout the entire book to slow down, think, and not take shortcuts, I know some of you will still want to take shortcuts. Don't do it. But if you're adamant about it and are going take a shortcut anyway, I might as well equip you properly.

Here is a picture showing how I structured the book, going from bottom to top. This direction makes sense pedagogically because you need to understand the concepts at the bottom before you can understand the nuances of the concepts that are higher up. For example, it is difficult to understand fairness metrics without first covering detection theory, and it is difficult to understand value elicitation about fairness metrics without first covering fairness. However, if you want to jump right into things, you should notionally start at the top and learn things from below as you go along.



Accessible caption. A stack of items in 8 layers. Top layer: ethics principles; layer 2: governance; layer 3: transparency, value alignment; layer 4: interpretability/explainability, testing, uncertainty quantification; layer 5: distributional robustness, fairness, adversarial robustness; layer 6: detection theory, supervised learning, causal modeling; layer 7: data biases, data consent, data privacy; bottom layer: probability and possibility theory, data. An upward arrow is labeled pedagogical direction. A downward arrow is labeled notional shortcut.

The ultimate shortcut is to give you a recipe to follow.

Preparation Steps:

1. Assemble socioculturally diverse team of problem owners, data engineers and model validators including members with lived experience of marginalization.
2. Determine ethics principles, making sure to center the most vulnerable people.
3. Set up data science development and deployment environment that includes fact flow tool to automatically collect and version-control digital artifacts.
4. Install software libraries in environment for testing and mitigating issues related to fairness and robustness, and computing explanations and uncertainties.

Lifecycle Steps:

1. Identify problem.
2. Conduct facilitated participatory design session including panel of diverse stakeholders to answer the following four questions according to ethics principles:
    a. Should the team work on the problem?
    b. Which pillars of trustworthiness are of concern?
    c. What are appropriate metrics?
    d. What are acceptable ranges of metric values?
3. Set up quantitative facts for the identified pillars of trustworthiness and their metrics.
4. If the problem should be worked on, identify relevant dataset.
5. Ensure that dataset has been obtained with consent and does not violate privacy standards.
6. Understand semantics of dataset in detail, including potential unwanted biases.
7. Prepare data and conduct exploratory data analysis with a particular focus on unwanted biases.
8. Train machine learning model.
9. Evaluate model for metrics of trustworthiness of concern, including tests that cover edge cases. Compute explanations or uncertainties if of concern.
10. If metric values are outside acceptable ranges, try other data, try other learning algorithms, or apply mitigation algorithms until metric values are within acceptable ranges.
11. Deploy model, compute explanations or uncertainties along with predictions if of concern, and keep monitoring model for metrics of trustworthiness of concern.