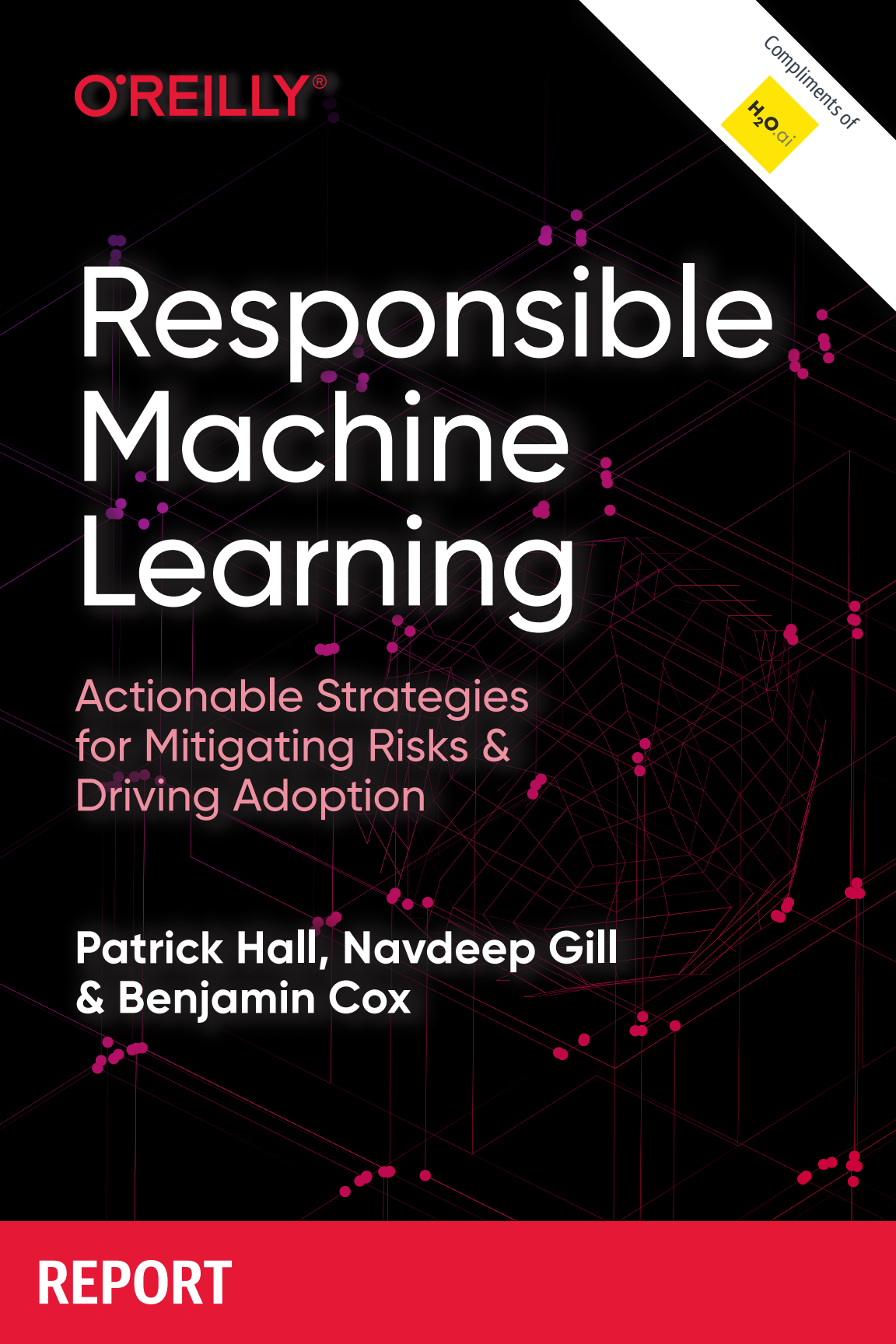
O'REILLY®
Compliments of
H2O.ai
Responsible Machine Learning
Actionable Strategies for Mitigating Risks & Driving Adoption
Patrick Hall, Navdeep Gill & Benjamin Cox
REPORT

Let us Help You Use AI to Get Results Responsibly

Find out How >>

In order to drive deeper insights, address privacy and security vulnerabilities, and prevent the perpetuation of historical human or data bias, organizations should consider how our core frameworks for responsible artificial intelligence (AI) enable the adoption of AI while accounting for its known risks.

H2O has the AI expertise and technology to help you innovate responsibly.

Learn More >>

H2O.ai

Responsible Machine Learning

Actionable Strategies for Mitigating Risks and Driving Adoption

Patrick Hall, Navdeep Gill, and Benjamin Cox

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Responsible Machine Learning
by Patrick Hall, Navdeep Gill, and Benjamin Cox

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Rebecca Novack
Developmental Editor: Michele Cronin
Production Editor: Beth Kelly
Copyeditor: Piper Editorial, LLC
Proofreader: Abby Wheeler
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Kate Dullea

October 2020: First Edition

Revision History for the First Edition
2020-10-02: First Release

This work is part of a collaboration between O'Reilly and H2O.ai. See our statement of editorial independence.

978-1-492-09084-7

[LSI]

Table of Contents

Preface

Machine learning (ML) sits at the cross section of business applications, statistics, and computer science. It's seen several waves of hype and disappointment in its roughly 60-year history. It's a big, technical subject, and it's also emerging as a powerful commercial technology. Yet, like other powerful technologies, it presents both opportunities and challenges. It can be a tool or a weapon. It can generate revenue, and, in some instances, be transformational for organizations. But it can also fail, be attacked, and cause significant incidents. From where we sit, most of the ML community appears too focused on ML's hype and upside. When we focus only on the upside of technology, we turn our backs on obvious realities that must be addressed to enable wider adoption. Perhaps this is why we've found it a little strange to write a report called *Responsible Machine Learning*. After all, we don't often hear about "responsible" aviation or "responsible" nuclear power. Responsibility and risk mitigation are usually baked into our notions of these technologies, and this apparently hasn't happened yet for ML. This could be because we know what happens when commercial jetliners or nuclear reactors fail or are attacked, but as a community, we're not yet so sure about the consequences of lousy ML.

If this is the case, it betrays a lack of effort by the practitioner community. While the consequences of ML failures and attacks are just beginning to filter into the news, if you know where to look, you can already find over 1,000 public reports of AI incidents. Furthermore, government and public awareness of creepy and discriminatory ML is also increasing. Digging deeper than surface level ML hype reveals an entire world of AI incidents, pending regulation of AI, and ML risk mitigation tools. This report provides an introduction to this

world. Since there are so many issues to cover, we don't dwell on any subject for very long. We hope that interested readers will engage in and explore our references to understand the real-world implications and the real, human consequences. We also hope that the sheer volume and diversity of presented material leave an indelible impression on how readers think about ML.

We break our discussions of risk mitigation and ML adoption strategies into three major chapters: people, processes, and technologies. The people and processes chapters (Chapters 2 and 3) describe actions people can take, and processes that organizations can employ to mitigate ML risks and increase ML adoption. These two chapters are meant to be approachable for non-technical audiences. While it includes no mathematical formulas, the technology chapter (Chapter 4) requires some technical background in ML, and it may be best suited for ML practitioners and their frontline managers. It's also important to say that the ML risks we're addressing are sophisticated, unsolved, and intersectional. Given the complexity of ML systems and how they interact with the world, there's no silver bullet to derisk an ML system completely.

Moreover, the serial nature of a printed report means that we address risks and mitigation strategies one by one. In truth, both the risks and strategies are inherently connected: compliance, discrimination, instability, privacy, and security risks are related, and so are the actions you could take to address them. Since deployment by an organization is often where risks become real for ML, proper risk mitigation is a key last-mile problem for ML's success. Although imperfect, we hope you'll find the proposed strategies helpful and actionable to decrease risk and maximize the long-term value of ML in your organization.

CHAPTER 1

Introduction to Responsible Machine Learning

> *"Success in creating effective AI, could be the biggest event in the history of our civilization. Or the worst."*
>
> —Stephen Hawking

Machine learning (ML) systems can make and save money for organizations across industries, and they're a critical aspect of many organization's digital transformation plans. For these reasons (and others), ML investments were increasing rapidly before the COVID-19 crisis, and they're expected to stay healthy as the situation unfolds. However, ML systems present risks for operators, consumers, and the general public. In many ways, this is similar to an older generation of transformational commercial technologies, like jetliners and nuclear reactors. Like these technologies, ML can fail on its own, or adversaries can attack it. Unlike some older transformational technologies, and despite growing evidence of ML's capability to do serious harm, ML practitioners don't seem to consider risk mitigation to be a primary directive of their work.[1]

Common ML failure modes include unaccountable black-box mechanisms, social discrimination, security vulnerabilities, privacy harms, and the decay of system quality over time. Most ML attacks involve insider manipulation of training data and model

1 See also *https://oreil.ly/9hzwC*, *https://oreil.ly/hFjRY*, and *https://oreil.ly/2T8Kt*.

mechanisms; manipulation of predictions or intellectual property extraction by external adversaries; or trojans hidden in third-party data, models, or other artifacts. When failures or attacks spiral out of control, they become full-blown AI incidents, creating significant adverse outcomes for the operator or the public. There have been over 1,000 reports of AI incidents to date.

While AI incidents are receiving more attention in the news and technology media of late, the hype around ML still seems to focus mostly on ML successes and not on ML risks. Subsequently, some decision makers and practitioners implement ML without a sober evaluation of its dangers. This report will cut through the hype to provide a high-level overview of ML's emerging risk mitigation practices—often called "responsible machine learning." This first chapter will give definitions of responsible AI and ML, and Chapters 2, 3, and 4 discuss viable ML risk mitigation steps for people, processes, and technologies, respectively. Chapter 5 closes this report with business-driven perspectives on risk and trust.

What Is Responsible Machine Learning?

What is responsible ML? It's not strictly defined yet, and the authors of this report don't seek to define it precisely. The concept of responsible ML needs time to evolve and grow with input from diverse practitioners, researchers, and decision makers. We hope that, like commercial aviation and energy production today, risk mitigation will eventually rise to the forefront of ML's practice, and there will be no need to differentiate between the general practice of ML and the responsible practice of ML. So, instead of putting forward a single definition, we present several potential definitions and discuss a few key similarities and differences between them to increase community awareness of this vital concept.

Responsible Artificial Intelligence

Several researchers and organizations have put forward helpful related definitions, particularly for "Responsible Artificial Intelligence." Given that ML is a subdiscipline of AI, and that the two terms are often used interchangeably, these definitions seem like an excellent place to start.

In her book, *Responsible Artificial Intelligence* (Springer), Virginia Dignum defines the eponymous concept: "Responsible Artificial

Intelligence is about human responsibility for the development of intelligent systems along fundamental human principles and values, to ensure human-flourishing and well-being in a sustainable world."

The Institute for Ethical AI & Machine Learning presents eight principles that "provide a practical framework to support technologists when designing, developing or maintaining systems that learn from data." The principles include:

Human augmentation
: Human review and assessment of risks

Bias evaluation
: Understanding, documenting, and monitoring sociological discrimination

Explainability by justification
: Transparency and explainability

Reproducible operations
: Processes and outcomes should be reproducible

Displacement strategy
: Consideration of the replacement of human jobs

Practical accuracy
: Real-world accuracy in addition to test data accuracy

Trust by privacy
: Addressing training data and consumer data privacy

Data risk awareness
: Reasonable security precautions for data and models

Google has also put forward Responsible AI Practices. These include using human-centered design principles, using multiple assessment metrics for any AI system, examining raw data, understanding the limitations of selected approaches, and thorough testing and monitoring of AI systems. Google is just one many organizations to publicize such guidance, and a brief summary of the many posted responsible AI guidelines boils down to the use of transparent technical mechanisms that create appealable decisions or outcomes, perform reliably over time, exhibit minimal social discrimination, and are designed by humans with diverse experiences, both in terms of demographics and professional backgrounds.

The authors of this text recently put forward two additional relevant definitions. Both are visual definitions. One is a higher-level conceptual summary, and the other is geared toward frontline practitioners. The higher-level description uses a Venn diagram, presented in Figure 1-1, to portray responsible AI as a combination of several preexisting and evolving disciplines.

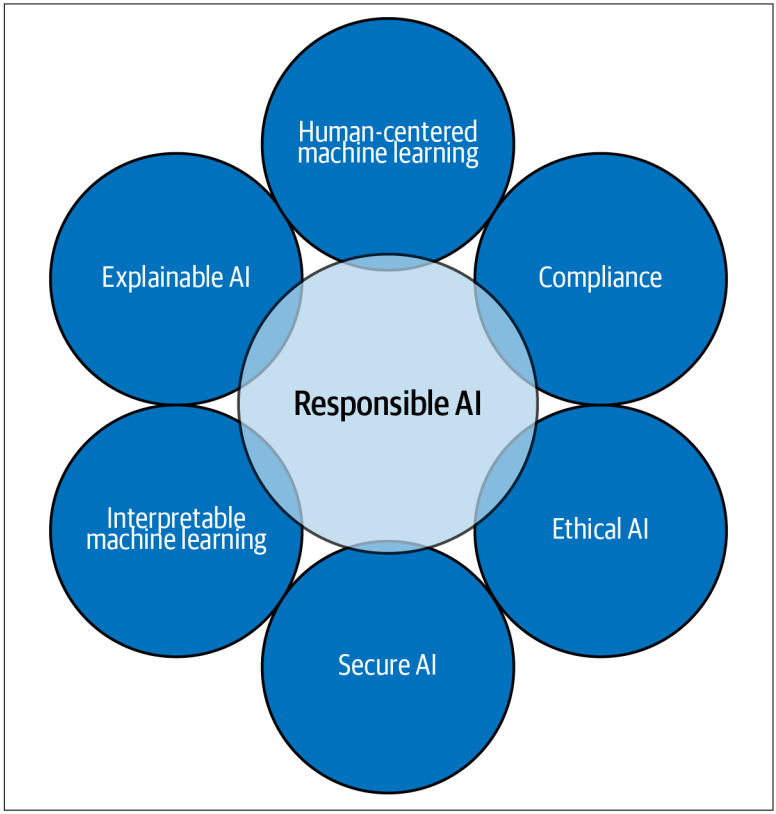

Figure 1-1. A responsible AI Venn diagram (courtesy of Benjamin Cox and H2O.ai).

Figure 1-1 claims that responsible AI is the combination of:

Ethical AI
: Sociological fairness in ML predictions (i.e., whether one category of person is being weighed unequally or unfavorably)

Explainable AI
: The ability to explain a model after it has been developed

Human-centered machine learning
: Meaningful user interactions with AI and ML systems

Interpretable machine learning
: Transparent model architectures and increasing how intuitive and comprehensible ML models can be

Secure AI
: Debugging and deploying ML models with similar counter measures against insider and cyber threats, as seen in traditional software

Compliance
: Aligning your ML systems with leading compliance guidance such as the EU GDPR, the Equal Credit Opportunity Act (ECOA), or the US Federal Reserve's SR 11-7 guidance on model governance

In the next section, a more technical definition is presented as a workflow in Figure 1-2 and adapted from the recent paper, *A Responsible Machine Learning Workflow with Focus on Interpretable Models, Post-hoc Explanation, and Discrimination Testing.* It specifically addresses details of Responsible ML.

A Responsible Machine Learning Definition

Most AI in the world today is likely based on ML. In trying to be as careful and realistic as possible, the Figure 1-2 workflow is designed specifically for today's ML systems. It walks practitioners through the processes required to mitigate many known risks associated with ML. In addition to traditional ML workflow steps, this diagram emphasizes transparency, human review, model end-of-life issues, and the evaluation of multiple key performance indicators (KPIs), including fairness, privacy, and security.

The many other available definitions for responsible AI and ML touch on a wide variety of topics, including everything from environmental impact to future unemployment. Common themes running through most definitions include human consideration and review of risks, enabling effective human interaction with ML systems, enhanced transparency and the treatment of discrimination, privacy harms, and security vulnerabilities. Notably, both the *Responsible Machine Learning Workflow* paper and the Venn diagram in Figure 1-1, bring compliance and legality into the fold of

responsible ML. Based on our experience as industry practitioners, we find that regulation and law can provide some of the clearest guidance for difficult ethical problems that arise in the implementation of ML systems. Moreover, legality is often the bottom-line concern for many high-stakes applications of ML. Compliance, legality, and regulation for ML, and several other concepts presented in the responsible AI and ML definitions will be discussed in the following chapters.

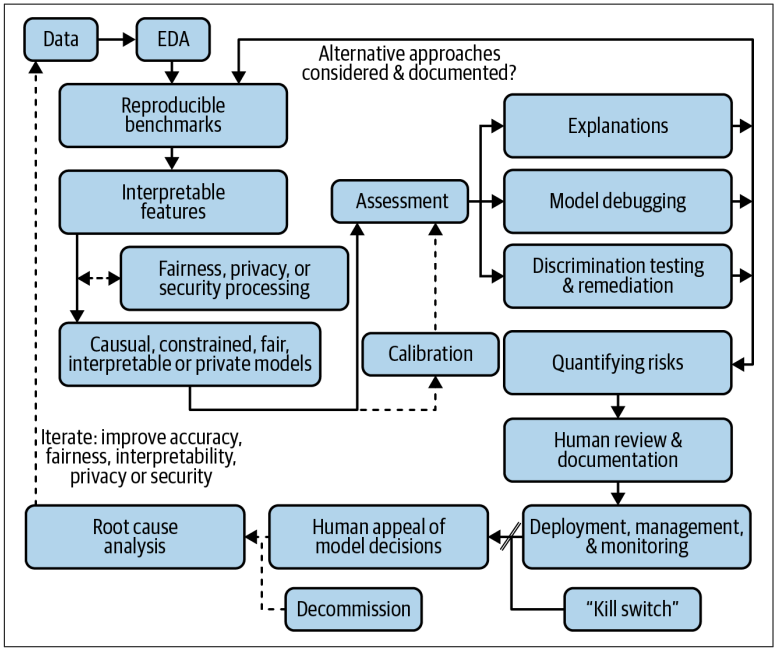

Figure 1-2. A responsible machine learning workflow diagram (adapted with permission of the authors).

CHAPTER 2

People: Humans in the Loop

> *"People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world."*
>
> —Pedro Domingos

Since its inception, there has been the temptation to give AI and ML increasingly more agency. However, this should *not* be the goal for organizations deploying ML today. Due to all the AI incidents we're seeing, we firmly believe the technology isn't mature enough. Instead, the goal should be to make sure humans are in the loop of ML-based decision making. Human involvement is imperative because an all too common mistake, as the quote above highlights, is for firms to assume their responsible ML duties lie solely in technological implementation. This chapter presents many of the human considerations that companies must address when building out their ML infrastructure. We'll start with organizational culture then shift the discussion to how practitioners and consumers can get more involved with the inner workings of ML systems. The chapter closes by highlighting some recent examples of employee activism and data journalism related to the responsible practice of ML.

Responsible Machine Learning Culture

An organization's ML culture is an essential aspect of responsible ML. This section will discuss the cultural notions of accountability, dogfooding, effective challenge, and demographic and professional

diversity. We'll also discuss the arguably stale adage, "go fast and break things."

Accountability

A key to the successful mitigation of ML risks is real accountability. Ask yourself: "Who tracks the way ML is developed and used at my organization? Who is responsible for auditing our ML systems? Do we have AI incident response plans?" For many organizations today, the answers may be, "no one" and, "no." If no one's job is on the line when an ML system fails or gets attacked, then it's possible that no one at that organization really cares about ML risks. This is a primary reason that many leading financial institutions now employ chief model risk officers. Smaller organizations may not be able to spare an entire full-time employee to monitor ML model risk. Still, it's essential to have an individual or group responsible and held accountable if ML systems misbehave. In our experience, if an organization assumes everyone is accountable for ML risk and AI incidents, the reality is that no one is accountable.

Dogfooding

Dogfooding is a term from software engineering that refers to an organization using its own software, i.e., "eating your own dog food." In the context of responsible ML, dogfooding brings an additional layer of alpha or prealpha testing that is often neglected in the mad dash to profit from a perceived ML gold rush. More importantly, dogfooding can bring legal and risk questions to the forefront. If an organization has developed an ML system that operates in a manner that, say, violates their own privacy policies, or is meant to be deceptive or manipulative, employees engaging in dogfooding might find this objectionable and raise concerns. Dogfooding can bring the Golden Rule into ML: if you wouldn't use an ML system on yourself, you probably should not use it on others. We'll discuss diversity in the next section, but it's worth mentioning here that if your team is more diverse, dogfooding is more likely to detect a wider variety of objectionable (or problematic) features.

Demographic and Professional Diversity

Many have documented the unfortunate outcomes that can arise as a result of ML engineers not considering demographic diversity in the training or results of ML systems. A potential solution to these kinds of oversights is increasing demographic diversity on ML teams from its current woeful levels. Another type of diversity that can also help to mitigate ML risk is a diversity of professional experience. According to Professor Karl Broman at the University of Wisconsin, "If you're analyzing data, you're doing statistics". ML is very much a data analysis field, and as such, it is a statistical discipline. Despite Kaggle leaderboards prizing single outcomes from single models, ML systems often benefit from the perspectives of other data analysis disciplines such as statistics, econometrics, or psychometrics. These fields have rich histories of learning that they can bring to bear on almost any ML project. Security personnel are another useful technical add-on to ML projects, as ML can present data privacy and security concerns.

Developing teams with deep cross-disciplinary professional experience can be invaluable as you look to deploy ML. Many of the most successful quantitative investment and advisory companies, such as McKinsey or Renaissance Technologies, pride themselves on how they have assembled an elite team with extremely diverse technical backgrounds from physics, biology, medicine, astronomy, and other fields. Legal, compliance, and audit personnel can also be necessary for ML projects. ML projects can run afoul of laws, regulations, or corporate privacy policies. Involving oversight professionals from the beginning is a great way to assess and mitigate these risks. ML can push ethical boundaries, too, and very few ML engineers have the education or experience necessary to guide a project through murky ethical waters. Bringing professional ethicists into the development of ML systems can help manage moral problems as they arise.

Cultural Effective Challenge

The notion of effective challenge was born out of the practice of model governance. When building complex ML systems, effective challenge roughly says that one of the best ways to guarantee good results is to actively challenge and question steps in the ML development process. There are more technical aspects of effective

challenge, which will be addressed in Chapter 4. Still, a culture that encourages serious questioning of ML design choices will be more likely to catch problems before they balloon into AI incidents. Of course, cultural effective challenge cannot be abusive, and it must apply to everyone developing an ML system, even so-called "rock-star" engineers and data scientists. In our experience, cultural effective challenge practices should be structured, such as weekly meetings where alternative design and implementation choices are questioned and discussed.

Going Fast and Breaking Things

The saying, "go fast and break things" is enshrined in the mindsets of many top engineers and data scientists. Unfortunately, when you go fast and break things, things tend to break. If you're working in the space of entertainment apps and advertisements for those apps, this may not be such a big deal. But suppose you're using ML in medicine, human resources, credit lending, criminal justice, the military, or other high-stakes applications. In these fields, going fast and breaking things can break the law or ruin people's lives. Practitioners must recognize the implications and downstream risks of their work instead of racing towards results for an outdated maxim.

Traditional model governance practices offer options to defend against these kinds of breakages, such as rigorous validation and monitoring. However, these practices require serious resources: lots of people, time, and technology. Standard model governance may not be feasible for young or smaller organizations under commercial pressure to move quickly. Common sense indicates that when going fast and breaking things, and without conventional model governance, AI incidents are even more likely. So, if you need to go fast and break things, AI incident response plans can be crucial for your organization. With AI incident response, as discussed in Chapter 3, smaller organizations without the resources for strict supervision of ML projects can spend their limited resources in ways that allow them to move quickly, but also confront the inevitability of AI incidents. In the long run, it's possible that being prepared for complex system failures may ultimately be the fastest development strategy.

Get in the Loop

Now that we've touched on some cultural aspects of responsible ML, this section will describe concrete steps practitioners or managers can take to get more control over ML systems. As of today, humans still have a big role to play in the successful implementation of ML systems. As the quote at the beginning of this chapter highlights, many decision makers and practitioners may be putting too much faith in today's ML systems. Along with the serious questioning of ML design choices in effective challenge, a human's detailed review of ML systems is another viable risk mitigation strategy. Inventories and documentation are a staple of model governance, and many recent AI and ML best practice guidelines highlight the need for human audits of ML systems. Of course, all of this requires people with a deep understanding of the data and problem domain, and that ML systems are constructed to enable interactions with those domain experts. Without domain expertise, ML systems can be trained on incorrect data, results can be misinterpreted, audits are less meaningful, and data or programming errors may explode into full-blown AI incidents. ML systems should also typically be designed to allow users to provide meaningful feedback, particularly to appeal and override ML-based decisions, and, if necessary, flipping the kill switch!

Human Audit of Machine Learning Systems

With the advent of technologies that enable more behind-the-scenes transparency and better discrimination and security testing of ML systems, it is now possible to foster enhanced human understanding and trust of ML. These technologies will be discussed in Chapter 4, but the technologies still must be deployed by people. One of the best uses of these newer technologies is the human audit of ML systems. In a recent paper, researchers at Google put forward a framework for ML model audits. They've also put forward basic sample documentation for models and data. These developments come on the coattails of years of model governance in the financial services vertical, where governance, effective challenge, model inventories, model documentation, model validation, and multiple technical and decision maker review levels have been the norm for high-stakes applications of predictive models.

What can you and your organization do to promote human audits of ML systems? The basics are relatively straightforward:

- Create an inventory of ML systems
- Nominate accountable executive(s)
- Instate executive and technical review of documented ML systems
- Require technical and executive sign off before deploying ML systems
- Carefully document, validate, and monitor all ML systems

When you're ready to move beyond these basic steps, check out the referenced papers from Google Research and look into resources from public model risk management forums, e.g., The North American Chief Risk Officer Council.

Domain Expertise

Many were introduced to the human expertise in-the-loop concept by the Pandora recommendation algorithm or something similar, which has ultimately evolved into a multibillion-dollar industry of expert labeling and decision review of ML systems. More generally, real-world success in ML almost always requires some input from humans with a deep understanding of the problem domain. Of course, such experts can help with feature selection and engineering, and interpretation of ML results. But the experts can also serve as a sanity check mechanism. For instance, if you're developing a medical ML system, you should consult physicians and other medical professionals. How will generalist data scientists understand the subtlety and complexity inherent in medical data and the results of systems trained on such data? They might not be able to, which can lead to AI incidents when the system is deployed. The social sciences deserve a special callout in this regard as well. Described as "tech's quiet colonization of the social sciences", some organizations are pursuing ill-advised ML projects that either replace decisions that would be made by trained social scientists or they are using practices, such as facial recognition for criminal risk assessments that have been condemned by actual social scientists.

User Interactions with Machine Learning

Since ML is always conducted with software, people at your organization will likely be interacting with ML results and outcomes through software. For maximum impact, nontechnical and decision-maker users need to understand and act on ML system results. Unfortunately, many ML systems and software packages generate only numeric outputs or visuals designed by highly technical researchers and practitioners. At best, this limits the number of people inside an organization who can work with AI and ML technologies. At worst, people can misunderstand poorly designed outputs, leading to process failures, customer dissatisfaction, and even AI incidents. When constructing ML systems, it is wise to consider the different types of users and personas who will need to interact with the system. Your organization should probably also have qualified user interaction professionals to help build out comprehensible and useful interfaces for these different ML system users.

User Appeal and Operator Override

What if a computer unjustly kept you in prison? What if a computer erroneously accused you of a crime? What if a computer kept you or a loved one out of the college of your dreams? You'd probably like to know why and you'd probably like the ability to appeal such decisions. From the ML system operator's standpoint, the operator might even want to tell you how the decision was made. (In some cases, the operator may be legally obligated to provide this information.) Or maybe the operator would like to have the capability to override individual ML-system decisions. If the ML system in question is based on a black-box algorithm, its operators may not be able to tell you how that decision was made, and they might not be able to check or override the decision promptly. Given that "all models are wrong," at least at some point, all this seems like a recipe for disaster.[1] In the worst-case scenario, a black-box ML system will (inevitably) issue a wrong prediction and maybe many of them at high speeds. These wrong decisions will hurt consumers or the general

1 The famous statistician George Box is credited with saying, "all models are wrong, but some are useful".

public, and the ML system operator will be subject to reputational, if not regulatory, damages.

This topic, also known as "intervenability" in data privacy circles, is already fairly well understood. So, there are steps you can take to prevent your organization's ML systems from making unappealable, and potentially illegal, black-box decisions:

- Use of interpretable ML models or reliable post-hoc explanation techniques (preferably both)
- Proper documentation of the processes used in these systems
- Meticulous testing of ML system interpretability features before deployment

The underpinning issue for appeal, override, and intervenability is transparency. Hence, your organization should understand how ML decisions are made, enabling operators to override—and consumers to appeal—ML system decisions logically. Ideally, ML systems should enable overall transparency for consumers and the public, especially those impacted by the ML system. This can even involve users probing these systems, extracting the reasoning behind individual decisions, and negating the decisions when necessary. These types of appeal and override mechanisms can also stop unavoidable ML system errors from becoming full-blown AI incidents.

Kill Switches

The title of a recent Forbes article asks, "Will There Be a Kill Switch For AI?". If your organization wants to mitigate risks around ML and AI, we hope the answer for your ML systems will be, "yes." ML systems can make decisions very quickly—orders of magnitudes faster than humans. So, if your ML system goes seriously wrong, you will want to be able to turn it off fast. But how will you even know if your ML system is misbehaving? ML systems should be monitored for multiple kinds of problems, including inaccuracy, instability, discrimination, leakage of private data, and security vulnerabilities.

Once you've detected a severe problem, the question then becomes, can you turn off the ML system? ML system outputs often feed into downstream business processes, sometimes including other ML systems. These systems and business processes can be mission critical, as in the case of an ML system used for credit underwriting or

e-retail product recommendations. To turn off an ML system, you'll not only need the right technical know-how and personnel available, but you also need an understanding of the model's place inside of broader organizational processes. During an ongoing AI incident is not a great time to start thinking about turning off a fatally flawed ML system. So, kill processes and kill switches are a great addition to your ML system documentation and AI incident response plans (see Chapter 3). This way, when the time comes to kill an ML system, your organization can be ready to make an informed decision.

Going Nuclear: Public Protests, Data Journalism, and White-Hat Hacking

Sometimes working within the confines of organizational culture or getting yourself into the loop of an ML system isn't enough. Sometimes organizations can be so irresponsible with technology that employees, journalists, researchers, or others feel the need to take drastic action. The remainder of this chapter discusses some recent and relevant examples of walkouts, protests, investigative data journalism, and even white-hat hacking.

In recent years, employees at technology giants have staged protests to voice their dissatisfaction with company policies regarding misinformation, climate change, sexual harassment, and other critical issues. As highly-skilled employees are perhaps the most valuable assets to most technology companies, companies do seem to pay attention to these protest activities, both by responding to protester demands and with retribution against protesters. Another exciting type of ML oversight has surfaced in recent years; it can be described best as a mixture of extreme data journalism and white-hat hacking. The catalyst for these actions appears to be the 2016 ProPublica analysis of the criminal risk assessment instrument known as COMPAS. In what was essentially a model extraction attack, journalists at ProPublica made a rough copy of COMPAS's proprietary training data and black-box logic and used this analysis to make serious claims about discrimination in algorithmic criminal risk assessments. Although the analysis results are scientifically controversial, the work brought widespread attention to the problem of algorithmic discrimination, and the company that licensed COMPAS later changed its name.

In another example of external oversight of commercial technology, researchers at MIT, operating under the project name Gender Shades, tested several commercial facial recognition tools for racial and gender discrimination. The results of the study were made public in 2018, and they were damning. Some of the day's leading facial recognition technologies performed well on white males and very poorly on female persons of color. Once the Gender Shades results were made public, companies were forced to either correct their apparently discriminatory systems or defend them. While most companies chose to address the issue quickly, Amazon chose to defend its Rekognition system and continued to license it to law enforcement. In the face of growing public outcry, Amazon and IBM—both cited in Gender Shades research—canceled their surveillance facial recognition programs in the summer of 2020. Such public attempts to expose troubling uses of ML continue and are perhaps happening even more frequently. In January of 2020, Walmart employees tipped off journalists about an antitheft ML system they believe is error prone and unnecessarily increasing contact between customers and associates during the COVID-19 pandemic. In February of 2020, The Markup, a nonprofit news organization devoted to oversight of Big Tech, alleged in an analysis somewhat akin to the original COMPAS exposé, that Allstate Insurance uses algorithms to charge its highest paying customers higher rates, essentially creating a "suckers list."

Fortunately, as evidenced by the fallout from the original COMPAS journalism and Gender Shades project, the rising awareness of ML risks in governments and the public seem to have at least some effect on organizational uses of ML. Organizations are seeing that at least sometimes, these types of incidents can be damaging to brand reputation, if not to the bottom line. As we transition our discussion to processes in Chapter 3 and technology in Chapter 4, remember that as of today, people are still the key ingredient in nearly any deployment of ML or AI technologies. Whether it's through fostering a culture of responsibility, getting more involved in the inner workings of ML systems, or taking more drastic actions, you really can make a difference in how ML systems impact our world.

CHAPTER 3

Processes: Taming the Wild West of Machine Learning Workflows

> *"AI is in this critical moment where humankind is trying to decide whether this technology is good for us or not."*
>
> —Been Kim

Despite its long-term promise, ML is likely overhyped today just like other forms of AI have been in the past (see, for example, the first and second AI winters). Hype, cavalier attitudes, and lax regulatory oversight in the US have led to sloppy ML system implementations that frequently cause discrimination and privacy harms. Yet, we know that, at its core, ML is software. To help avoid failures in the future, all the documentation, testing, managing, and monitoring that organizations do with their existing software assets should be done with their ML projects, too. And that's just the beginning. Organizations also have to consider the specific risks for ML: discrimination, privacy harms, security vulnerabilities, drift toward failure, and unstable results. After introducing these primary drivers of AI incidents and proposing some lower-level process solutions, this chapter touches on the emergent issues of legal liability and compliance. We then offer higher-level risk mitigation proposals related to model governance, AI incident response plans, organizational ML principles, and corporate social responsibility (CSR). While this chapter focuses on ways organizations can update their processes to better address special risk considerations for ML, remember that ML needs basic software governance as well.

Discrimination In, Discrimination Out

We hear about many discriminatory algorithms these days, but discrimination tends to enter ML systems most often through poor experimental design or biased, unrepresentative, or mislabeled training data. This is a crucial process concern because business goals often define an ML model's inherent experiment, and training data is usually collected or purchased as part of some broader organizational mechanism. When an organization is designing an ML system or selecting data for an ML project, discrimination can enter into the system in many ways, including:

Problem framing (e.g., association or label bias)
: In ML, we essentially use a dataset to ask the question: is X predictive of y? Sometimes simply asking this question can set up a discriminatory premise. For instance, predicting criminal risk (y) based on facial characteristics (X), or using individual healthcare costs (y) as an inherently biased substitute for healthcare needs. Said another way, just because you have access to data on different topics, doesn't mean that ML can link the two topics without introducing or perpetuating discrimination.

Labeling or annotation (e.g., exclusion, sampling, reporting, label, or nonresponse bias)
: Data is often cleaned or preprocessed before it ever reaches an ML algorithm. These processes, if done without care, can introduce discrimination. For example, switching race to a numeric code, misinterpreting a coded value for a particular demographic group, or mislabeling sound or images due to unconscious human bias are just a few ways discrimination can seep into data cleaning or preprocessing.

Unrepresentative data (e.g., selection or coverage bias)
: ML models require highly representative training data. Consider training a facial recognition classifier on face images collected in one country, for example, the US, and then applying it in another country, like Japan or Kenya. The chances are that the model will be less accurate for the people it learned less about during training. This is yet another way ML can be discriminatory.

Accurate data that is correlated to demographic group membership (e.g., historical or prejudice bias)

Data like traditional credit scores are accurate predictors of credit default, but due to long-running systemic inequalities in the US, some minorities have lower average credit scores than Whites or Asians. It's not necessarily wrong to use a traditional credit score in an ML system, but you have to be aware that data like this encodes information about demographic groups into your ML model and may lead to discriminatory ML system outcomes.

Accurate data that encodes discrimination (e.g., historical or prejudice bias)

For example, data sampled from police or court records may be highly accurate, but it likely also contains historical and current racism. Models simply make decisions from what they learn in training data, so this kind of data should only be used with the utmost care in ML systems.

NOTE

These topics are often discussed under the heading *ML fairness*. Of course, fairness has proven devilishly tricky to define mathematically, and the concept of fairness is subject to varying political, cultural, and ethical interpretations. We use the term *discrimination* more frequently in this text due to its seemingly more narrow and clearly negative interpretation.[1]

Once discriminatory data enters into an ML system, you can bet discriminatory predictions will quickly follow. A real difference between ML and human decision making is speed. ML systems can make decisions about a lot of people, very quickly. Moreover, the complexity of ML models can make finding discrimination more difficult than in traditional linear models. All of this can add up to a disastrous AI incident, like the one described in a recent *Science* article, where a major US insurer unintentionally used an allegedly discriminatory algorithm to allocate healthcare resources for perhaps millions of patients. Such discrimination can cause significant harm to consumers and regulatory and reputational problems for

1 See: *https://oreil.ly/K4HMW*; for a more broadly accepted academic treatment of the subject see: *https://fairmlbook.org*.

organizations. Luckily, you can test for many types of discrimination in ML system outputs before deployment. The major ways that discrimination can manifest in the output of ML systems are:

Explicit discrimination
: Demographic group membership, or direct proxies, are used directly in an ML system, resulting in unfavorable outcomes for historically disadvantaged groups. This is sometimes called "disparate treatment," and it's often an illegal business practice.

Group outcome disparities
: Learning different correlations between demographic groups and favorable model outcomes, resulting in disproportionately unfavorable outcomes for historically disadvantaged groups. This is sometimes called "disparate impact," and it can also be illegal in some settings.

Group accuracy disparities
: Exhibiting different accuracies across demographic groups, especially when an ML system is less accurate for historically disadvantaged groups. Sometimes known as "differential validity," this type of discrimination is also of interest to certain regulators in the US.

Individual disparities
: An ML system treats similarly situated individuals that differ only in terms of demographic group membership differently in terms of outcomes or accuracy.

Because of all the different vectors for discrimination to infect an ML system, it's always safest to test for these types of discrimination and attempt to remediate any discovered discrimination. Chapter 4 will provide more details on testing ML systems for discrimination and how to address discovered discrimination.

Algorithmic Discrimination and US Regulations

Before moving onto data privacy and security, it's important to mention laws and regulations on the books today that already address AI, ML, and discrimination. Given the recent rush of headlines on AI and ML discrimination, you could be forgiven for thinking this is a new problem. But that's not true. Discrimination in testing and decision making has likely always been around and it has been studied for decades. So long, in fact, that specific discrimination tests

and remediation tactics have become almost enshrined in US law and regulations. The key here is for practitioners to understand when those laws and regulations apply and when you might be more free to pursue your choice of discrimination testing and remediation strategies outside of regulatory frameworks. For practitioners in industry verticals like consumer credit, healthcare, employment, or education (and maybe others), testing for discrimination with a shiny new open-source package, instead of the legally required methods, could get your organization into a lot of trouble. Much like the data privacy and security topics in the next section, antidiscrimination regulatory and compliance concerns are a good reason to include legal personnel in ML projects.

Data Privacy and Security

Successful ML implementations require training data, and typically, lots of it. This means that data privacy and security concerns are relevant to ML workflows that consume and generate data. ML engineers and managers should acquaint themselves with the basics of their organization's security and privacy policies, in addition to the major features of applicable privacy and security regulations. While there is not yet a nationwide uniform regulation for data privacy in the US, the combination of data security requirements, the EU GDPR, the CCPA, and many industry-specific, local, or emerging laws make the US a heavily regulated region for data privacy and security. For ML practitioners, key concepts for data privacy to keep in mind include:

Consent for use
: Though sometimes burdensome for consumers, most current privacy laws put forward notions of consumer consent for data usage. Training ML systems or adding new ML system features without the appropriate consideration for consent could cause big problems.

Legal basis for data collection
: The GDPR lays out six valid reasons that consumer personal data can be collected and used: consumer consent, contractual obligation, legal obligation, public interest (i.e., public or governmental tasks), vital interest (i.e., to save the consumer's life), or legitimate business interests (e.g., appropriate marketing activities). ML training data should probably be collected for

one of these reasons too, or an ML system could raise unpleasant questions down the road.

Alignment with privacy policy
: Many organizations have privacy policies, and if you or your ML system violate that privacy policy, it may cause regulatory or reputational headaches for your organization.

Anonymization requirements
: Laws such as HIPAA and GDPR include data anonymization requirements applicable to ML training data. If you're working with personal or sensitive data in an ML project, it should probably be anonymized (even though true anonymization has proven difficult in practice).

Retention requirements and limitations
: Sensitive consumer data often comes with conditions on how long it must be stored or when it must be destroyed. These requirements and constraints should likely be considerations for data selection and generation in ML systems.

From a data security perspective, goals and failures are usually defined in terms of the confidentiality, integrity, and availability (CIA) triad. The CIA triad can be briefly summarized as: data should only be available to authorized users, data should be correct and up-to-date, and data should be promptly available when needed. If one of these tenets is broken, this is usually a security incident. To avoid incidents involving ML-related data, these basic best practices can be helpful:

Access control
: Limit access to training data, particularly personal or sensitive data.

Authentication
: For those with access to ML-related data, require strong passwords or other authentication to access training data.

User permissions
: Review and update user permissions frequently. Use the concept of "least privilege" in which all personnel receive the lowest possible access level. Strictly limit the number of administrative or "root" users.

Remote access
: Limit and monitor remote access to ML-related data.

Third parties
: Verify that third-party data providers and consumers follow reasonable security standards.

Physical media
: Protect documents, thumb drives, backup media, and other portable data sources.

The above principles and more may fall under the FTC's reasonable security purview. This means that violations can be a big deal for your organization. Moreover, privacy and security breaches must often be reported to the proper authorities. Of course, breach reporting and other incident response steps should be part of written incident response plans. These plans should be followed when incidents occur, and they should be reassessed and updated frequently. Later in Chapter 3, we'll address AI incident response specifically. Next, we'll go over dangerous new vectors for attackers to extract ML-training data and the problems it can cause.

Machine Learning Security

If "[t]he worst enemy of security is complexity," according to Bruce Schneier, ML may be innately insecure. Other researchers have also released numerous studies describing and confirming specific security vulnerabilities for ML systems. And we're now beginning to see how real-world attacks occur, like Islamic State operatives blurring their logos in online content to evade social media filters. Since organizations often take measures to secure valuable software and data assets, ML systems should be no different. Beyond specific incident response plans, several additional information security processes should be applied to ML systems. These include security audits, bug bounties, and red teaming.

The primary security threats for ML today appear to be:

- Insider manipulation of ML-system training data or software
- Manipulation of ML-system outcomes by external adversaries
- Extraction of ML-system logic or training data by external adversaries

- Trojans buried in third-party ML software, models, data, or other artifacts

For mission-critical, or otherwise high stakes, deployments of ML, systems should be audited for at least these known vulnerabilities. Audits can be conducted internally or by specialist teams in what's known as *red teaming*, as is done by Facebook. Bug bounties, or when organizations offer monetary rewards to the public for finding vulnerabilities, are another practice from general information security that should probably also be applied to ML systems. Moreover, audits, red teaming, and bug bounties need not be limited to security concerns alone. These types of processes can also be used to spot other ML-system problems, such as discrimination or instability, and spot them before they explode into AI incidents.

Legality and Compliance

ML can create a lot of value for organizations. But given the real discrimination, privacy, and security concerns, among others, it can also bring on serious reputational damage or legal liabilities. These include causing your organization to be slapped with a lawsuit or to run afoul of local, federal, or international regulations. Difficult questions about compliance and legality often slam the brakes on end-stage ML products and projects because oversight personnel are seldom involved in the build stages of ML endeavors. Moreover, ML, like many other powerful commercial technologies that came before it, is likely to be highly regulated in the future. With increasing international regulation, US government regulatory agencies, such as CFPB, FINRA, and FTC making announcements about ML guidance, and state regulators announcing various ML discrimination investigations, now is a good time to consider your ML systems in the current and evolving ML legal and compliance landscape.

As mentioned previously, some regulations can impact your ML system today, especially in healthcare, financial services, and employment. Under laws and regulations like ECOA, FCRA, FHA, SR 11-7, and under EEOC guidelines, ML systems are generally expected to be mathematically sound and stable, exhibit minimal discrimination, and be explainable. Outside these verticals, your ML system could still be held to local anti-discrimination laws, reasonable security standards, unfair and deceptive practice (UDAP) laws, and to scrutiny related to its terms of service or warranty (or lack thereof).

Today, violations of these laws and regulations can result in regulatory fines, litigation costs, reputational damage for your organization, and harm to consumers. Moreover, government agencies have telegraphed increased future ML regulation, or, outside of the US, started to implement such regulations. For a more detailed list of US and international ML guidance documents, see the Awesome Machine Learning Interpretability metalist.

As of today, US government agencies are generally advising that your ML be documented, explainable, managed, monitored, and minimally discriminatory. Of course, we're not lawyers or regulators, and it's not our place to give legal advice or determine what is compliant with regulations. So, take a look at the documents highlighted below to see for yourself what some US regulators and agencies are saying about ML:

- *Artificial Intelligence (AI) in the Securities Industry*
- *A Primer on Artificial Intelligence in Securities Markets*
- *Innovation spotlight: Providing adverse action notices when using AI/ML models*
- "Office of Management and Budget Draft Guidance for Regulation of Artificial Intelligence Applications"
- *Using Artificial Intelligence and Algorithms*

If you're feeling overwhelmed by the combination of ML, law, and regulation, there are current examples on which to pattern your organization's future ML endeavors. Organizations that operate under current ML-related regulations, or that already learned their lesson about playing fast and loose with ML, often ascribe to a practice known as model governance. The next section will attempt to summarize practices from that field.

Model Governance

The decision to move into the world of ML is not a simple undertaking and smart leaders can be left asking, "how can I mitigate the risks for my organization?" Luckily, there are mature model governance practices crafted by government agencies and private companies that your organization can use to get started. This section will highlight some of the governance structures and processes your organization can employ to ensure fairness, accountability, and

transparency in your ML functions. This discussion is split into three major sections: model monitoring, model documentation, and organizational concerns. We'll wrap up the discussion of model governance with some brief advice for practitioners looking for just the bare bones needed to get started on basic model governance.

Model Monitoring

Model monitoring is a stage in the ML lifecycle that involves keeping tabs on your ML system while it is making predictions or decisions on new, live data. There's lots to be aware of when monitoring ML models. First and foremost is model decay. Model decay is a common failure mode for ML systems. It happens when the characteristics of live data coming into an ML system drift away from those of the training data, making the underlying ML model less accurate. Model drift is most often described in terms of decreasing model accuracy, but can also affect the fairness or security of ML systems. Model drift is typically detected by monitoring the statistical properties of model inputs and predictions and comparing them to those recorded at training time. For fairness and security, monitoring could involve real-time discrimination testing and ongoing red-teaming or security audits of deployed ML systems, respectively. Anytime a significant drift is detected, system stakeholders should be alerted. To address accuracy drift, ML systems are typically retrained with new data when drift is detected, or at frequent intervals to avoid drift altogether. Addressing drifts in fairness or security is a more novel pursuit and standard practices are not yet established. However, the discrimintion testing and remediation and security countermeasures discussed elsewhere in this report could also be helpful in this regard.

Another major topic in model monitoring is anomaly detection. Strange input or output values from an ML system can be indicative of stability problems or security and privacy vulnerabilities. It's possible to use statistics, ML, and business rules to monitor anomalous behavior in both inputs and outputs, and across an entire ML system. Just like when model drift is detected, system stakeholders must be made aware of anomalous ML system inputs and outputs. Two additional and worrisome scenarios for which to monitor are error propagation and feedback loops. Error propagation refers to problems in the output of some data or ML system leading to worsening errors in the consuming ML system or in subsequent

downstream processes. Feedback loops can occur in applications like predictive policing or algorithmic trading, leading to serious external risks. Whenever an ML system can affect the real-world, and then have that outcome act as an input to the ML system again, feedback loops and AI incidents can ensue.

Monitoring your ML in production environments is extremely important as this can quickly catch degradation in accuracy or changes in the fairness, security, and stability characteristics of your system—before they become an AI incident. To get the most out of model monitoring, you'll need to know what your system looked like at training time and who to call when things go wrong. The perfect place for those details is in model documentation, which we'll discuss next.

Model Documentation

All organizational predictive models should be inventoried and documented. When done correctly, model documentation should provide all pertinent technical, business, and personnel information about a model, enabling detailed human review, maintenance continuity, and some degree of incident response. Moreover, in some industries, model documentation is already a regulatory requirement. The main drawback of model documentation is that it is tedious and time consuming, sometimes taking longer to write the documentation than to train the ML model itself. One answer to this problem was provided by Google research in their recent model cards and datasheet work. Model cards and datasheets provide quick, summary information about the models and data used in ML systems, respectively. Another promising answer has started to emerge in the commercial analytics market: automatic model documentation. Purchasing or building ML software that creates model documents along with your ML-model training can be a great solution for ML teams looking to document their models and save time on resource-intensive model governance activities. Of course, even if model documentation is generated automatically, humans must still read the documentation and raise concerns when necessary.

Hierarchy and Teams for Model Governance

To guarantee models are designed, deployed, and managed responsibly, properly defining the organization that will execute these responsibilities is crucial. In Figure 3-1 we propose a simple

structure for organizations looking to build ML into their operational processes. This chart uses the acronym *D&A*, a common industry shorthand for *data and analytics groups*, especially in years past.

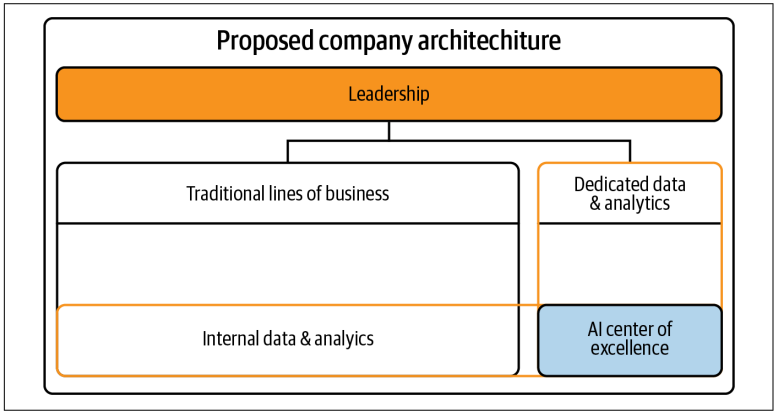

Figure 3-1. Basic overview of a proposed organizational structure for data and analytics (D&A) groups (courtesy of Ben Cox and H2O.ai).

One early problem in D&A adoption was that companies leaned into one of two less than ideal structures:

D&A as a unique group within every line of business (horizontal)
: D&A resources in each business unit spend too much time building redundant tools that exist in other areas of the firm, and because the teams are disparate, the organization is not taking advantage of cross-functional synergies and often create duplicate customer or transaction data that hinders operations.

D&A as a single line of business (LOB) reporting out to all other LOBs (vertical)
: A silo effect forces the D&A unit to spend too much time correcting disconnects between respective lines of business and responding to ad hoc requests to spend time on new value-driving opportunities. Also, friction can develop between incumbent areas of ML expertise, like credit risk and the silo.

Learning from these lessons by having analytics, data science, and ML functions operate at the cross section of different groups within an organization can minimize classic data management obstacles and technical debt through increased transparency and collaboration. Additionally, the cross-sectional option includes a centralized

AI and ML Center of Excellence, a specific unit focused on the highest value or novel ML pursuits across organizations and capabilities.

Figure 3-2 puts forward an idealized operating architecture for a contemporary D&A team that's ready to tackle ML adoption responsibly. In Figure 3-2, we can see that technical functions roll up to a single accountable executive while also serving all LOBs.

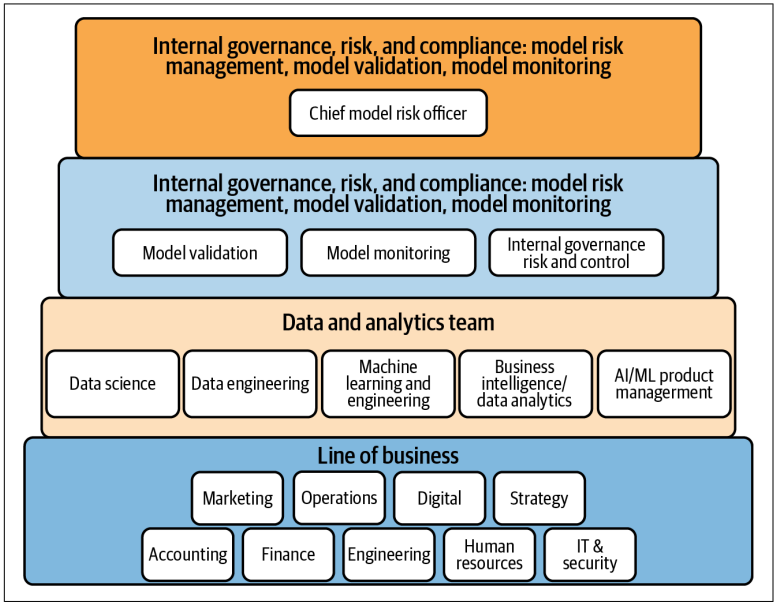

Figure 3-2. A proposed ML model governance reporting organizational hierarchy (courtesy of Ben Cox and H2O.ai).

Finally, Figure 3-3 illustrates how technical, responsible ML functions, defined in Figure 1-2, fit into the larger organizational hierarchy proposed in Figure 3-2. As is common in model governance, data science and analytics teams train ML models, while other groups act as defense lines to challenge, validate, and monitor those models.

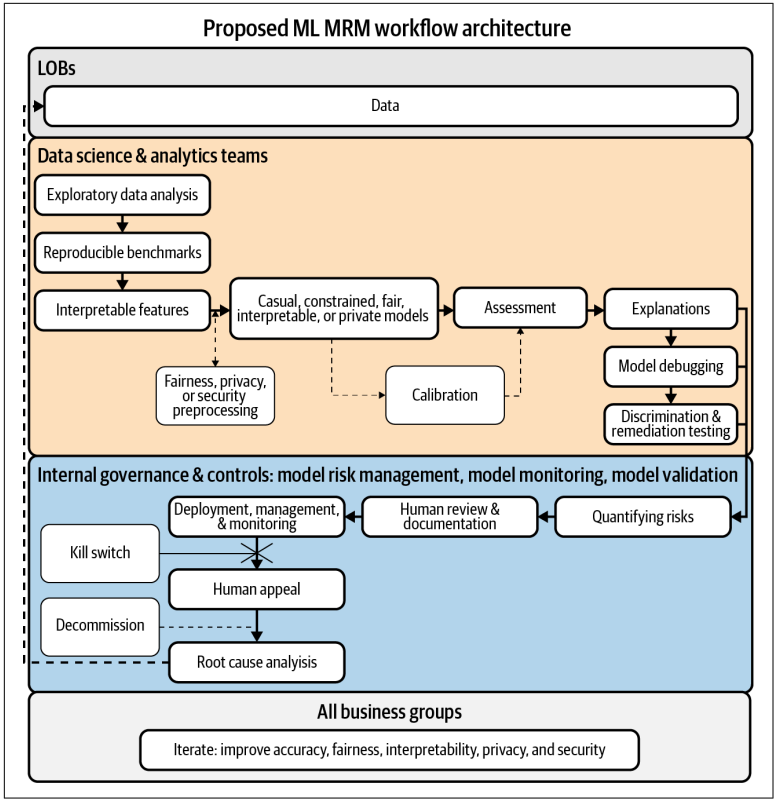

Figure 3-3. Proposed model governance workflow and organizational responsibility architecture (courtesy of Ben Cox and H2O.ai).

Model Governance for Beginners

If you're at a small or young organization, you may need just the bare bones of model governance. Two of the most crucial model governance processes are model documentation and model monitoring:

Basic model documentation

Model documentation should contain the who, what, when, where, and how for any personnel, hardware, data, or algorithms used in an ML system. These documents should enable new employees to understand how an ML system works so they can take over maintenance, or they should facilitate third parties in performing detailed investigations of the model in case of failures and attacks. And yes, these documents can be very long.

Basic model monitoring

: Even if you never touch your ML system's code, its outputs can change with the new data it's encountering. This rarely happens in a way that's beneficial for ML systems, and sadly, most ML systems are destined to drift toward failure. Hence, ML systems must be monitored. Typically, monitoring is used to watch the inputs or outputs of an ML system change over time, particularly for model accuracy. However, it can't hurt to monitor for drift in fairness or security characteristics as well.

If your organization is not in a place to devote large amounts of resources to model governance, whole-hearted investment in these two practices, plus preparing for AI incidents as discussed below, can take you a long way toward mitigating ML risks.

AI Incident Response

Like nearly all of the commercial technologies that came before it, ML systems fail and can be attacked. To date, there have been over 1,000 public reports of such incidents. Even our most secure, regulated, and monitored commercial technologies, like airliners and nuclear reactors, experience attacks and failures. Given that very few organizations are auditing and monitoring ML systems with the same rigor, and that regulatory interest in ML systems is on the rise, we'll probably hear more about AI incidents in the next few years. Furthermore, when a technology is important to an organization's mission, it's not uncommon to have built-in redundancy and incident response plans. ML systems should be no different. Having a plan in place for ML system failures or attacks can be the difference between a glitch in system behavior and a serious AI incident with negative consequences for both the organization and the public.

The stress and confusion of an active AI incident can make incident response difficult. Who has the authority to respond? Who has the budget? What are the commercial consequences of turning off an ML system? These basic questions and many more are why AI incident response requires advanced planning. The idea of being prepared for problems is not new for computer systems, or even for predictive modeling. Respected institutions such as SANS and NIST already publish computer security incident response plans. Model governance practices typically include inventories of ML systems with detailed documentation designed, among other goals, to help

respond to ML system failures. While conventional incident response plans and model governance are great places to start mitigating AI incident risks, neither are a perfect fit for AI incident response. Many conventional incident response plans do not yet address specialized ML attacks, and model governance often does not explicitly address incident response or ML security. To see a sample AI incident response plan that builds off both traditional incident response and model governance, and that incorporates the necessary specifics for ML, check out the free and open *Sample AI Incident Checklist*. And don't wait until it's too late to make your own AI incident response plan!

Organizational Machine Learning Principles

If your organization is working with ML, organizational ML principles are critical for many reasons. One primary reason is that ML systems present opportunities to scale, automate, and defer liability for harm. When you blend these capabilities with entrenched corporate or government power structures, history tells us bad things can happen. Actionable, public ML principles provide at least one mechanism to hold accountable those seeking to combine organizational and ML power for nefarious purposes. There's also the issue of ML openly designed to cause harm. Organizational ML principles can serve as a guiding compass for difficult decisions related to this topic. Imagine if your company's ML systems were performing so well that they attracted attention from military customers. Are you willing to sell into a situation where ML could be used to kill people at scale? Having some idea of how you will handle these situations before they arise can help your organization make the best decision. And whatever that decision is, it's probably better that it is based on previously agreed upon, responsible, human-centered ML tenets than on heat-of-the-moment inclinations. Another advantage of organizational ML principles is that they can educate nontechnical employees on ML risks. What if a group in your organization that was new to ML deployed a black-box image classifier without discrimination testing? If the organization had an ML principle that stated, "All ML products that could be used on humans will be tested for discrimination," perhaps someone somewhere in your organization would be more likely to catch the oversight before release.

Need some example principles to get started? AlgorithmWatch keeps a catalog of many corporate and government ML and AI principles. Nearly all these principles appear well meaning and serviceable. However, what is often missing is the nuts-and-bolts roadmap of how to implement them. So, if you do create ML principles, keep in mind that a common pratfall to avoid is ineffectiveness. It's easy to make organizational ML principles so high-level or abstract that they cannot be implemented or enforced. AlgorithmWatch recently published a report stating that only 10 of 160 reviewed sets of principles were enforceable. So, getting technologists, along with ethics, legal, oversight, and leadership perspectives into organizational ML principles is probably a best practice. We'll reference a set of example ML principles, published as part of a publicly traded bank annual report, in the next section on the broader topic of CSR.

Corporate Social Responsibility and External Risks

It is no secret that companies, both private and public, have experienced pressure to prioritize ethical and sustainable business practices in recent years. When thinking about the implications of this as it pertains to ML, we must note that ML sits at the intersection of multiple critical areas of concern for workers, consumers, and the public: automation, perpetuating discrimination and inequality, privacy harms, lack of accountability, and more. For this very reason, a focus on responsible deployment of ML and rigorous internal accountability should be a pillar of success for organizations adopting ML. This section will briefly touch on the importance of CSR for ML. It will also quickly discuss the mitigation of broad and general risks to organizations, consumers, and the public, i.e., external risks, that organizations can create through the irresponsible use of ML.

Corporate Social Responsibility

The news and technology media are replete with examples of why organizations should consider social responsibility when adopting ML. Let's consider the following:

- According to Forbes, "81 percent of Millennials expect companies to make a public commitment to good corporate citizenship."

- According to Cone Communications, "75 percent of millennials would take a pay cut to work for a socially responsible company."
- According to Crunchbase News, "More investors recognize that making money and making a positive impact on the world doesn't have to be mutually exclusive."
- According to Capgemini, "62% of consumers will place higher trust in the company if they perceive AI-enabled interactions as ethical."

As indicated in previous sections, responsible ML can minimize the long-term fines associated with regulatory noncompliance or costs stemming from large legal settlements. But, as the quotes here hint at, responsible ML and CSR processes do not act solely to protect brand reputation or as a shield from regulatory retribution. They can also impact an organization's ability to access customers, capital, funding, and talent, and impact employee satisfaction and retention. For a prime example of a company combining responsible ML with CSR reasonably early on in their AI transformation journey, see the Regions Bank 2019 *Annual Review and Environmental, Social and Governance Report*.

Mitigating External Risks

For many companies, answering the question, "Which of my customers get hurt if a model breaks?" is not straightforward. Suppose a company is systemically important, poses significant health risks to surrounding areas, or sells a product or service with network effects. In that case, they must be aware that their IT systems can cause problems beyond their own revenue and customers. One of the most prominent examples of this kind of serious hazard is the 2007–2008 Global Financial Crisis, where major financial institutions recklessly created synthetic financial instruments that mathematically disguised the risks entailed in these products. Ultimately when the dominos fell, the final result was a massive global recession that wreaked havoc in financial and real estate markets and created severe and lasting damage across much of the world economy.

In ML, we've just started to see these kinds of broad external risks emerge. For instance, in the US, health insurance companies sometimes use algorithms to allocate resources for entire insured populations—hundreds of millions of people. Unfortunately, at least one

such algorithm was found to harm the healthcare of large numbers of Black people. Another example of external algorithmic risk is the 2016 British Pound flash crash, in which algorithmic trading is suspected of crashing currency markets. Given even the possibility of such large-scale incidents, it seems evident that ML and algorithmic decision making must be treated more like commercial air travel or nuclear energy. In these verticals, operators are not only accountable for their successes and failures within the company, but they can also be held accountable for the impacts their decisions have on customers and tangential third parties. Perhaps one thing to keep in mind is that, in the already regulated verticals of ML, organizations and accountable executives can be penalized for their failures. Regardless of the presence of serious regulatory oversight in your industry, if you're doing big things with ML, it is probably a good time to start thinking about ML's external risks. To help your organization start tracking these kinds of ML risks, check out the high-level document created by the Future of Privacy Forum and bnh.ai, entitled Ten Questions on AI Risk.

CHAPTER 4

Technology: Engineering Machine Learning for Human Trust and Understanding

> *"If builders built houses the way programmers built programs, the first woodpecker to come along would destroy civilization."*
>
> —Gerald M. Weinberg

Human users of ML need to trust that any decision made by an ML system is maximally accurate, secure, and stable, and minimally discriminatory. We may also need to understand any decision made by an ML system for compliance, curiosity, debugging, appeal, or override purposes. This chapter discusses many technologies that can help organizations build human trust and understanding into their ML systems. We'll begin by touching on reproducibility, because without that, you'll never know if your ML system is any better or worse today than it was in the past. We'll then proceed to interpretable models and post hoc explanation because interpretability into ML system mechanisms enables debugging of quality, discrimination, security, and privacy problems. After presenting some of these debugging techniques, we'll close the chapter with a brief discussion of causality in ML.

Reproducibility

Establishing reproducible benchmarks to gauge improvements (or degradation) in accuracy, fairness, interpretability, privacy, or

security is crucial for applying the scientific method. Reproducibility can also be necessary for regulatory compliance in certain cases. Unfortunately, the complexity of ML workflows makes reproducibility a real challenge. This section presents a few pointers for increasing reproducibility in your organization's ML systems.

Metadata

Metadata about ML systems allows data scientists to track all model artifacts that lead to a deployed model (e.g., datasets, preprocessing steps, model, data and model validation results, human sign offs, and deployment details). Many of the additional reproducibility steps presented below are just specific ways to track ML system metadata. Tracking metadata also allows retracing of what went wrong, throughout the entire ML life cycle, when an AI incident occurs. For an open-source example of a nice tool for tracking metadata, checkout TensorFlow's MLMD.

Random Seeds

ML models are subject to something known as the "multiplicity of good models," or the "Rashomon effect." Unlike more traditional linear models, this means that there can be huge numbers of acceptable ML models for any given dataset. ML models also utilize randomness, which can cause unexpected results. These factors conspire to make reproducible outcomes in ML models more difficult than in traditional statistics and software engineering. Luckily, almost all contemporary, high-quality ML software comes with a "seed" parameter to help improve reproducibility. The seed typically starts the random number generator inside an algorithm at the same place every time. The key with seeds is to understand how they work in different packages and then use them consistently.

Version Control

ML code is often highly intricate and typically relies on many third-party libraries or packages. Of course, changes in your code and changes to third-party code can change the outcomes of an ML system. Systematically keeping track of these changes is another good way to increase reproducibility, transparency, and your sanity. Git and GitHub are free and ubiquitous resources for software version control, but there are plenty of other options to explore. Ensuring

correct versions of certain ML libraries is also very important in any ML application, as different versions of ML libraries can lead to differences in performance and accuracy. So, ensuring that versions of each library used are documented and controlled will often lead to better reproducibility. Also, remember that tracking changes to large datasets and other ML-related artifacts is different than tracking code changes. In addition to some of the environment tools we discuss in the next subsection, checkout Pachyderm or DVC for data versioning.

Environments

ML models are trained, tested, and deployed in an environment that is determined by software, hardware, and running programs. Ensuring a consistent environment for your ML model during training, testing, and deployment is critical. Different environments will most likely be detrimental to reproducibility (and just a huge pain to handle manually). Happily, many tools are now available to help data scientists and ML engineers preserve their computing environments. For instance, Python, sometimes called the lingua franca of ML, now includes virtual environments for preserving coding environments.

Virtual machines, and more recently, containers, provide a mechanism to replicate the entire software environment in which an ML system operates. When it comes to ML, the container framework is very popular. It can preserve the exact environment a model was trained in and be run later on different hardware—major pluses for reproducibility and easing ML system deployment! Moreover, specialized software has even been developed specifically to address environment reproducibility in data and ML workflows. Check out Domino Data Lab, Gigantum, KubeFlow Pipelines, and TensorFlow Extended to see what these specialized offerings look like.

Hardware

Hardware is the collection of different physical components that enable a computer to run, subsequently allowing ML code to run, which finally enables the training and deployment of ML systems. Of course, hardware can have a major impact on ML system reproducibility. Basic considerations for hardware and ML reproducibility include ensuring similarity of the hardware used between training

and deployment of ML systems and testing ML systems across different hardware with an eye toward reproducibility.

By taking stock of these factors, along with the benchmark models also discussed later in Chapter 4, data scientists, ML and data engineers, and other IT personnel should be able to enhance your organization's ML reproducibility capabilities. This is just a first step to being more responsible with ML, but should also lead to happier customers and faster ML product delivery over an ML system's lifespan. And once you know your ML system is standing on solid footing, then the next big technological step is to start applying interpretable and explainable ML techniques so you can know exactly how your system works.

Interpretable Machine Learning Models and Explainable AI

Interpretability is another basic requirement for mitigating risks in ML. It's just more difficult to mitigate risks in a black-box system that you don't understand. Hence, interpretability enables full debuggability. Interpretability is also crucial for human learning from ML results, enabling human appeal and override of ML outcomes, and often for regulatory compliance. Today, there are numerous methods for increasing ML's interpretability, but they usually fall into two major categories: interpretable ML models and post hoc explanation techniques.

Interpretable Models

For decades, an informal belief in a so-called "accuracy-interpretability tradeoff" led most researchers and practitioners in ML to treat their models as supposedly accurate, but inscrutable, black boxes. In recent years, papers from leading ML scholars and several empirical studies have begun to cast serious doubt on the perceived tradeoff.[1] There has been a flurry of papers and software for new ML algorithms that are nonlinear, highly accurate, and directly interpretable. Moreover, "interpretable" as a term has become more associated with these kinds of new models.

1 See *https://oreil.ly/gDhzh* and *https://oreil.ly/Fzilg*.

New interpretable models are often Bayesian or constrained variants of older ML algorithms, such as the explainable neural network (XNN) pictured in the online resources that accompany this report. In the example XNN, the model's architecture is constrained to make it more understandable to human operators.

Another key concept with interpretability is that it's not a binary on-off switch. And XNNs are probably some of the most complex kinds of interpretable models. Scalable Bayesian rule lists, like some other interpretable models, can create model architectures and results that are perhaps interpretable enough for business decision makers. Other interesting examples of these interpretable ML models include:

- Explainable boosting machines (EBMs, also known as GA2M)
- Monotonically constrained gradient boosting machines
- Skope-rules
- Supersparse linear integer models (SLIMs)
- RuleFit

Next time you're starting an ML project, especially if it involves standard structured data sources, evaluate one of these accurate and interpretable algorithms. We hope you'll be pleasantly surprised.

Post hoc Explanation

Post hoc explanations are summaries of model mechanisms and results that are typically generated after ML model training. These techniques are also sometimes called explainable AI (XAI). These techniques can be roughly broken down into:

Local feature importance measurements
: For example, Shapley values, integrated gradients, and counterfactual explanations. Sometimes also referred to as "local" explanations, these can tell users how each input feature in each row of data contributed to a model outcome. These measures can also be crucial for the generation of adverse action notices in the US financial services industry.

Surrogate models
: For example, local interpretable model-agnostic explanations (LIME) or anchors. These techniques are simpler models of more complex ML models that can be used to reason about the more complex ML models.

Visualizations of ML model results
: For example, variable importance, accumulated local effect, individual conditional expectation, and partial dependence plots. These plots help summarize many different aspects of ML model results into consumable visualizations. They're also helpful for model documentation requirements in US financial services applications.

Many of these post hoc explanation techniques can be applied to traditional ML black boxes to increase their interpretability, but these techniques also have to be used with care. They have known drawbacks involving fidelity, consistency, and comprehensibility. Fidelity refers to the accuracy of an explanation. Consistency refers to how much an explanation changes if small changes are made to training data or model specifications. And comprehensibility refers to human understanding of generated explanations. All of these drawbacks must be considered carefully when using XAI techniques. Likely one of the best ways to use post hoc explanations is with constrained and interpretable ML models. Both the constraints and the inherent interpretability can counterbalance concerns related to the validity of post hoc explanations. Pairing interpretable model architectures with post hoc explanation also sets the stage for effective model debugging and ML system testing.

Model Debugging and Testing Machine Learning Systems

Despite all the positive hype, there's nothing about ML systems that makes them immune to the bugs and attacks that affect traditional software systems. In fact, due to their complexity, drift characteristics, and inherently stochastic nature, ML systems may be even more likely than traditional software to suffer from these kinds of incidents. Put bluntly, current model assessment techniques, like cross validation or receiver operator characteristic (ROC) and lift curves, just don't tell us enough about all the incidents that can occur when

ML models are deployed as part of public facing, organizational IT systems.

This is where model debugging comes in. Model debugging is a practice that's focused on finding and fixing problems in ML systems. In addition to a few novel approaches, the discipline borrows from model governance, traditional model diagnostics, and software testing. Model debugging attempts to test ML systems like computer code because ML models are almost always made from code. And it uses diagnostic approaches to trace complex ML model response functions and decision boundaries to hunt down and address accuracy, fairness, security, and other problems in ML systems. This section will discuss two types of model debugging: porting software quality assurance (QA) techniques to ML, and specialized techniques needed to find and fix problems in the complex inner workings of ML systems.

Software Quality Assurance for Machine Learning

ML is software. All the testing that's done on traditional enterprise software assets should generally be done on ML as well.

> **NOTE**
>
> This is just the starting point! We give ample consideration to special ML risks in other sections of this report. This subsection simply aims to clarify that we recommend doing all the testing your organization is doing on traditional software assets on ML systems too—and then moving on to address the wide variety of risks presented by ML systems. Yes, that's a lot of work. With great power comes great responsibility.

Unit tests should be written for data processing, optimization, and training code. Integration testing should be applied to ML system APIs and interfaces to spot mismatches and other issues. And functional testing techniques should be applied to ML system user interfaces and endpoints to ensure that systems behave as expected. Wrapping some of these testing processes and benchmarking into continuous integration/continuous deployment (CI/CD) processes can lead to efficiency gains and even higher ML software quality. To learn more about getting started with simple QA and model debugging, check out Google's free course, Testing and Debugging in Machine Learning.

Specialized Debugging Techniques for Machine Learning

ML does present concerns above and beyond traditional software. As discussed in other report sections, ML poses some very specialized discrimination, privacy, and security concerns. ML systems can also just be wrong. In one famous case, a medical risk system asserted that asthma patients were at lower risk than others of dying from pneumonia. In another shocking instance, a self-driving car was found to be unprepared to handle jaywalking. It killed a human.

Finding these types of bugs does require some specialized approaches, but it's an absolute must for high-stakes ML deployments. Practical techniques for finding bugs in ML systems tend to be variants of sensitivity analysis and residual analysis. Sensitivity analysis involves simulating data and testing model performance on that data. Residual analysis is the careful study of model errors at training time. These two techniques, when combined with benchmark models, discrimination testing, and security audits can find logical errors, blinspots, and other problems in ML systems. Of course, once bugs are found, they must be fixed. There are lots of good options for that too. These include data augmentation and model assertions and editing, among others. For a summary of contemporary model debugging techniques, see *Why You Should Care About Debugging Machine Learning Models*. For code and examples of debugging an example consumer credit model, check out *Real-World Strategies for Model Debugging*. The next section will stay on the theme of model debugging and introduce benchmark models in more detail.

Benchmark Models

Benchmark models are simple, trusted, or transparent models to which ML systems can be compared. They serve myriad risk mitigation purposes in a typical ML workflow, including use in model debugging and model monitoring.

Model Debugging

First, it's always a good idea to check that a new complex ML model outperforms a simpler benchmark model. Once an ML model passes this baseline test, benchmark models can serve as debugging tools.

Use them to test your ML model by asking questions like, "What did my ML model get wrong that my benchmark model got right? And can I see why?" Another important function that benchmark models can serve is tracking changes in complex ML pipelines. Running a benchmark model at the beginning of a new training exercise can help you confirm that you are starting on solid ground. Running that same benchmark after making changes can help to confirm whether changes truly improved an ML model or pipeline. Moreover, automatically running benchmarks as part of a CI/CD process can be a great way to understand how code changes impact complex ML systems.

Model Monitoring

Comparing simpler benchmark models and ML system predictions as part of model monitoring can help to catch stability, fairness, or security anomalies in near real time. Due to their simple mechanisms, an interpretable benchmark model should be more stable, easier to confirm as minimally discriminatory, and should be harder to hack. So, the idea is to use a highly transparent benchmark model when scoring new data and your more complex ML system. Then compare your ML system predictions against a trusted benchmark model. If the difference between your more complex ML system and your benchmark model is above some reasonable threshold, then fall back to issuing the benchmark model's predictions or send the row of data for manual processing. Also, record the incident. It might turn out to be meaningful later. (It should be mentioned that one concern when comparing an ML model versus a benchmark model in production is the time it takes to score new data, i.e., increased latency.)

Given the host of benefits that benchmark models can provide, we hope you'll consider adding them into your training or deployment technology stack.

Discrimination Testing and Remediation

Another critical model debugging step is discrimination testing and remediation. A great deal of effort has gone into these practices over the past several decades. Discrimination tests run the gambit from simple arithmetic, to tests with long-standing legal precedent, to

cutting-edge ML research. Approaches used to remediate discrimination usually fall into two major categories:

1. Searching across possible algorithmic and feature specifications as part of standard ML model selection.
2. Attempting to minimize discrimination in training data, ML algorithms, and in ML system outputs.

These are discussed in more detail below. While picking the right tool for discrimination testing and remediation is often difficult and context sensitive, ML practitioners must make this effort. If you're using data about people, it probably encodes historical discrimination that will be reflected in your ML system outcomes, unless you find and fix it. This section will present the very basics of discrimination testing and remediation in hopes of helping your organization get a jump start on fighting this nasty problem.

Testing for Discrimination

In terms of testing for ML discrimination, there are two major problems for which to be on the lookout: group disparities and individual disparities. Group disparities occur when a model's outcome is unfair across demographic groups by some measure or when the model exhibits different accuracy or error characteristics across different demographic groups—most open source packages test for these kinds of disparities.

Individual disparity is a much trickier concept, and if you're just starting testing for discrimination in ML, it may not be your highest priority. Basically, individual disparity occurs when a model treats individual people who are similar in all respects except for some demographic information, differently. This can happen in ML for many reasons, such as overly complex decision boundaries, where a person in a historically marginalized demographic group is placed on the harmful side of an ML decision outcome without good reason. It can also happen when an ML system learns to combine some input data features to create proxy features for someone's unique demographic information.

While functionality to find counterfactual or adversarial examples is becoming more common (e.g., Google's "What-if" tool), testing for individual disparity is typically more involved than testing for group disparities. Today, it just takes some snooping around, looking at

many individuals in your data, training adversary models or using special training constraints, tracing decision boundaries, and using post hoc explanation techniques to understand if features in your models are local proxies for demographic variables. Of course, doing all this extra work is never a bad idea, as it can help you understand drivers of discrimination in your ML system, whether these are group disparities or local disparities. And these extra steps can be used later in your ML training process to confirm if any applied remediation measures were truly successful.

Remediating Discovered Discrimination

If you find discrimination in your ML system, what do you do? The good news is you have at least two major remediation strategies to apply—one tried and true and the others more cutting edge, but also potentially a little risky in regulated industries. We'll provide some details on these strategies below.

Strategy 1

Strategy 1 is the traditional strategy (and safest from a US regulatory perspective). Make sure to use no demographic features in your model training, and simply check standard discrimination metrics (like adverse impact ratio or standardized mean difference) across an array of candidate ML models. Then select the least discriminatory model that is accurate enough to meet your business needs. This is often the strategy used today in highly regulated areas like lending and insurance.

Figure 4-1 illustrates how simply considering a discrimination measure, adverse impact ratio (AIR) for African Americans versus Caucasians in this case, during ML model selection can help find accurate and less discriminatory models. AIR is usually accompanied by the four-fifths rule practical significance test, wherein the ratio of positive outcomes for a historically marginalized demographic versus the positive outcomes for a reference group, often Whites or males, should be greater than 0.8, or four-fifths.

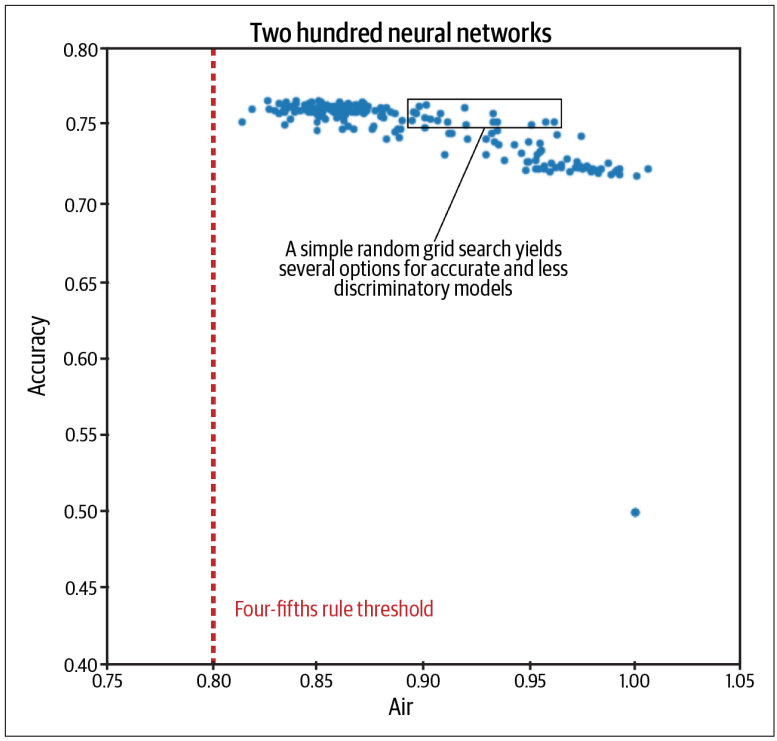

Figure 4-1. A random grid search for neural network models with results plotted in both quality and fairness dimensions (courtesy of Patrick Hall). The results show that high quality, lower discrimination models are available in this case. We just needed to look for them.

Strategy 2

Strategy 2 includes newer methods from the ML, computer science, and fairness research communities.

Fix your data. Today, in less regulated industrial sectors, you'll likely be able to use software packages that can help you resample or reweight your data so that it brings less discrimination into your ML model training to begin with. Another key consideration here is simply collecting representative data; if you plan to use an ML system on a certain population, you should collect data that accurately represents that population.

Fix your model. ML researchers have developed many interesting approaches to decrease discrimination during ML model training.

Some of these might even be permissible in highly regulated settings today but be sure to confer with your compliance or legal department before getting too invested in one of these techniques.

Regularization
: The most aggressive, and perhaps riskiest approach from a regulatory standpoint, is to leave demographic features in your ML model training and decision-making processes, but use specialized methods that attempt to regularize, or down weight, their importance in the model.

Dual optimization
: In a dual optimization approach, demographic features are not typically used in the ML system decision-making process. But, they are used during the ML model training process to down weight model mechanisms that could result in more discriminatory outcomes. If you're careful, dual optimization approaches may be acceptable in some US regulated settings since demographic information is not technically used in decision making.

Adversarial debiasing
: In adversarial debiasing, two models compete against one another. One ML model will be the model used inside your ML system for decision making. This model usually does not have access to any explicit demographic information. The other model is an adversary model that is discarded after training, and it does have access to explicit demographic information. Training proceeds by first fitting the main model, then seeing if the adversary can accurately predict demographic information from only the main model's predictions. If the adversary can, the main model uses information from the adversary, but not explicit demographic information, to down weight any hidden demographic information in its training data. This back-and-forth continues until the adversary can no longer predict demographic information based on the main model's predictions. Like a dual objective approach, adversarial debiasing may be acceptable in some US regulated settings.

Fix your predictions. Decisions based on low-confidence predictions or harmful decisions affecting historically marginalized demographic groups can be sent for human review. It's also possible to directly change your ML predictions to make ML systems less discriminatory by some measure. This can potentially be used for

already-in-flight ML systems with discrimination problems, as discrimination can be decreased without retraining the system in some cases. But this heavy-handed intervention may also raise regulatory eyebrows in the US consumer finance vertical.

As you can see, there are numerous ways to find and fix discrimination in your ML systems. Use them, but do so carefully. Without discrimination testing and remediation, it's possible that your ML system is perpetuating harmful, inaccurate, and even illegal discrimination. With these techniques, you'll still need to monitor your system outcomes for discrimination on ever-changing live data and be on the lookout for unintended side effects. As discussed in the next section, ML systems can be attacked, and in one famous example, hacked to be discriminatory. Or interventions that were intended to diminish discrimination can end up causing harm in the long run.

Securing Machine Learning

Various ML software artifacts, ML prediction APIs, and other ML endpoints can now be vectors for cyber and insider attacks. These ML attacks can negate *all* the hard work an ML team puts into mitigating risks. After all, once your model is attacked, it's not your model anymore. And the attackers could have their own agendas regarding accuracy, discrimination, privacy, or stability. This section will present a brief overview of the current known ML attacks and some basic defensive measures your team can use to protect your AI investments.

Machine Learning Attacks

ML systems today are subject to general attacks that can affect any public facing IT system; specialized attacks that exploit insider access to data and ML code; external access to ML prediction APIs and endpoints; and trojans that can hide in third-party ML artifacts.

General attacks

ML systems are subject to hacks like distributed denial of service (DDOS) attacks and man-in-the-middle attacks.

Insider attacks

Malicious or extorted insiders can change ML training data to manipulate ML system outcomes. This is known as data poisoning. They can also alter code used to score new data, including creating back doors, to impact ML system outputs. (These attacks can also be performed by unauthorized external adversaries but are often seen as more realistic attack vectors for insiders.)

External attacks

Several types of external attacks involve hitting ML endpoints with weird data to change the system's output. This can be as simple as using strange input data, known as adversarial examples, to game the ML system's results. Or these attacks can be more specific, say impersonating another person's data, or using tweaks to your own data to evade certain ML-based security measures. Another kind of external ML attack involves using ML prediction endpoints as designed, meaning simply submitting data to—and receiving predictions from—ML endpoints. But instead of using the submitted data and received predictions for legitimate business purposes, this information is used to steal ML model logic and to reason about, or even replicate, sensitive ML training data.

Trojans

ML systems are often dependent on numerous third-party and open-source software packages, and, more recently, large pretrained architectures. Any of these can contain malicious payloads.

Illustrations of some ML attacks are provided in the online resources that accompany this report. These illustrations are visual summaries of the discussed insider and external ML attacks. For an excellent overview of most known attacks, see the Berryville Machine Learning Institute's Interactive Machine Learning Risk Framework.

Countermeasures

Given the variety of attacks for ML systems, you may now be wondering about how to protect your organization's ML and AI models. There are several countermeasures you can use and, when paired with the processes proposed in Chapter 3—bug bounties, security audits, and red teaming—such measures are more likely to be

effective. Moreover, there are the newer subdisciplines of adversarial ML and robust ML that are giving the full academic treatment to these subjects.

This section of the report will outline some of the most basic defensive measures you can use to help make your ML system more secure, including general measures, model monitoring for security, and defenses for insider attacks. Also, be sure to follow new work in secure, adversarial, and robust ML, as this subject is evolving quickly.

The basics

Whenever possible, require consumer authentication to access predictions or use ML systems. Also, throttle system response times for large or anomalous requests. Both of these basic IT security measures go a long way in hindering external attacks.

Model debugging

Use sensitivity analysis and adversarial example searches to profile how your ML system responds to different types of data. If you find that your model may be subject to manipulation by certain kinds of input data, either retrain your model with more data, constraints and regularization, or alert those responsible for model monitoring to be on the lookout for the discovered vulnerabilities.

Model monitoring

As discussed elsewhere in the report, models are often monitored for decaying accuracy. But models should also be monitored for an adversarial attack. Because a model could be attacked to be made discriminatory, real-time discrimination testing should be conducted if possible. In addition to monitoring for accuracy and discrimination, watching for strange inputs such as unrealistic data, random data, duplicate data, and training data can help to catch external adversarial attacks as they occur. Finally, a general strategy that has also been discussed in other sections is the real-time comparison of the ML system results to simpler benchmark model results.

Thwarting malicious insiders

A strict application of the notion of least privilege, i.e., ensuring all personnel—even "rockstar" data scientists and ML engineers—

receive the absolute minimum IT system permissions, is one of the best ways to guard against insider ML attacks. Other strategies include careful control and documentation of data and code for ML systems and residual analysis to find strange predictions for insiders or their close associates.

Other key points in ML security include privacy-enhancing technologies (PETs) to obscure and protect training data and organizational preparation with AI incident response plans. As touched on in Chapter 3, incorporating some defensive strategies—and training on how and when to use them—into your organization's AI incident response plans can improve your overall ML security. As for PETs, the next section will address them.

Privacy-Enhancing Technologies for Machine Learning

Privacy-preserving ML is yet another research subdiscipline with direct ramifications for the responsible practice of ML. Some of the most promising and practical techniques from this field include federated learning and differential privacy.

Federated Learning

Federated learning is an approach to training ML algorithms across multiple decentralized edge devices or servers holding local data samples, without exchanging raw data. This approach is different from traditional centralized ML techniques where all datasets are uploaded to a single server. The main benefit of federated learning is that it enables the construction of robust ML models without sharing data among many parties. Federated learning avoids sharing data by training local models on local data samples and exchanging parameters between servers or edge devices to generate a global model, which is then shared by all servers or edge devices. Assuming a secure aggregation process is used, federated learning helps address fundamental data privacy and data security concerns.

Differential Privacy

Differential privacy is a system for sharing information about a dataset by describing patterns about groups in the dataset without disclosing information about specific individuals. In ML tools, this

is often accomplished using specialized types of differentially private learning algorithms.[2] This makes it more difficult to extract sensitive data from training data or the trained model. In fact, an ML model is said to be differentially private if an outside observer cannot tell if an individual's information was used to train the model. (This sounds great for preventing those data extraction attacks described in the previous section!)

Federated learning, differential privacy, and ML security measures can go hand in hand to add an extra layer of privacy and security to your ML systems. While they will be extra work, they're very likely worth considering for high-stakes or mission-critical ML deployments.

Causality

We'll close our responsible ML technology discussion with causality, because modeling causal drivers of some phenomenon, instead of complex correlations, could help address many of the risks we've brought up. Correlation is not causation. And nearly all of today's popular ML approaches rely on correlation, or some more localized variant of the same concept, to learn from data. Yet, data can be both correlated and misleading. For instance, in the famous asthma patient example discussed earlier, having asthma is correlated with greater medical attention, not being at a lower risk of death from pneumonia. Furthermore, a major concern in discrimination testing and remediation is ML models learning complex correlations to demographic features, instead of real relationships. Until ML algorithms can learn such causal relationships, they will be subject to these kinds of basic logical flaws and other problems. Fortunately, techniques like Markov Chain Monte Carlo (MCMC) sampling, Bayesian networks, and various frameworks for causal inference are beginning to pop up in commercial and open-source software ML packages. More innovations are likely on the way, so keep an eye on this important corner of the data world.

Aside from rigorous causal inference approaches, there are steps you can take right now to incorporate causal concepts into your ML projects. For instance, enhanced interpretability and model debugging can lead to a type of "poor man's causality" where debugging is

2 See also *https://oreil.ly/ESyqR*.

used to find logical flaws in ML models and remediation techniques such as model assertions, model editing, monotonicity constraints, or interaction constraints are used to fix the flaw with human domain knowledge. Root cause analysis is also a great addition to high-stakes ML workflows. Interpretable ML models and post hoc explanation techniques can now indicate reasons for ML model behaviors, which human caseworkers can confirm or deny. These findings can then be incorporated into the next iteration of the ML system in hopes of improving multiple system KPIs. Of course, all of these different suggestions are not a substitute for true causal inference approaches, but they can help you make progress toward this goal.

CHAPTER 5

Driving Value with Responsible Machine Learning Innovation

> *"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."*
>
> —Eliezer Yudkowsky

"Why do 87% of data science projects never make it into production?" asks a recent VentureBeat article. For many companies, getting ML models into production is where the rubber meets the road in terms of ML risks. And to many, the entire purpose of building a model is to ultimately deploy it for making live predictions, and anything else is a failure. For others, the ultimate goal of an ML model can simply be ad hoc predictions, valuations, categorizations, or alerts. This short chapter aims to provide an overview of key concepts companies should be aware of as they look to adopt and drive value from ML. Generally, there are much more significant implications for companies looking to make material, corporate decisions based on predictive algorithms, versus simply experimenting or prototyping exploratory ML exercises.

Trust and Risk

For smart organizations adopting AI, there are often two major questions that get asked: "How can I trust this model?" and "How risky is it?" These are critical questions for firms to ask before they put ML models into production. However, the thing to understand is there is a flywheel effect between the answers to these questions.

The more you understand an ML system's risks, the more you can trust it. We often find that executives and leaders jump to ask, "What is the risk?" whereas the data science practitioners are more focused on, "Can I trust this prediction?" But in the end, they are asking the same question.

The first and most obvious metrics to be analyzed are those around the risk that a given ML model may manifest. Below are a few questions informed decision makers need to ask regarding ML deployments:

- What is the quality of the model? (Accuracy, AUC/ROC, F1)
- What is the cost of an inevitable wrong outcome?
  - Are there secondary cost considerations? Legal or compliance concerns? Customer lifetime value? Operational risk? Brand or reputational risk? Harm to end users or the general public?
  - Are we watching our models in real time for accuracy, discrimination, privacy, or security problems?
  - Do we have specific AI incident response plans?
- How many predictions is this model making?
  - What is the velocity of these predictions? How quickly does your company need to respond to errors or anomalous situations?
  - What is the materiality of this model to your company?

As of today, there's no silver bullet for ML risks. Much of the true nuance when it comes to data science and ML risk stems from understanding what happens when predictions go wrong. For the same reason, that is why having deep domain knowledge or contextual business background is imperative when designing and developing ML systems. The downside risk of an incorrect prediction is not simply lost profit or increased cost. It is a multilayered consideration that firms need to be rigorous in analyzing. The statistics quotation, "all models are wrong, but some are useful," by George Box, should be a starting point for organizations with ML. Know that your model will be incorrect and understand thoroughly what that means for your organization.

Signal and Simplicity

The most recent wave of popularity in ML and AI, starting roughly in the mid to late aughts, is often attributed to the explosion of deep neural networks and deep learning accelerated by high-performance computing. Originally, the big idea was that an appropriately tuned deep learning model with enough data could outperform any other method. The problem with this idea is that these models were arguably the most black-box of any leading method. This created a tradeoff: do I have to sacrifice accuracy (signal) for simplicity? How much? However, new research shows that with tabular data and methods like XNN and EBM, this tradeoff is probably small. For the time being, white-box methodologies can perform at the same accuracy as their black-box counterparts on standard business data sources. Remember that interpretability enables all kinds of risk mitigation processes around an ML system: improved effective challenge, better documentation, customer appeals, and more model debugging. If you're a business leader presented with an ML system that your team can't explain, this is a major red flag. You can have both signal and simplicity, especially for the most common types of traditional data mining problems.

The Future of Responsible Machine Learning

Over the last few years, there has been more demand for better understanding and trust of our ML systems—and that is a very good thing. In parts of Europe and Asia, governments have been more active in requiring organizations to consider these factors when deploying ML. In the US, responsible ML has largely been a grassroots push by data scientists, researchers, and industry practitioners aiming to encourage responsible innovation in AI and ML. Either way, the goal is increasing consumer trust and driving better practices in the field. We believe that responsibility in ML is going to continue to evolve and improve over time, particularly with serious regulatory oversight from government agencies in the US. With both the grassroots and future regulatory pressures increasing, organizations would be remiss to simply check off a series of boxes and consider responsible ML "done." We hope organizations will aim to continually improve the practices they use to better understand and trust their ML systems until responsible machine learning is just machine learning.

Further Reading

UK ICO AI Auditing Framework

Singapore PDPC Model AI Governance Framework

Berryville Institute Interactive Machine Learning Risk Assessment

Acknowledgments

Thanks to our colleagues at H2O.ai, especially Ingrid Burton. Thanks to Michele Cronin, Michelle Houston, Beth Kelly, Mike Loukides, and Rebecca Novack at O'Reilly Media. Thanks also to our colleagues in the broader data science community, Andrew Burt, Hannes Hapke, Catherine Nelson, and Nicholas Schimdt.

About the Authors

Patrick Hall is principal scientist at bnh.ai, a boutique law firm focused on data analytics and AI. Patrick also serves as a visiting professor in the Department of Decision Sciences at the George Washington University and as an advisor to H2O.ai.

Navdeep Gill is a senior data scientist and software engineer at H2O.ai where he focuses mainly on responsible machine learning. Navdeep has also contributed to H2O.ai's efforts in GPU-accelerated machine learning, automated machine learning, and to the core H2O-3 machine learning platform.

Ben Cox is a director of product marketing at H2O.ai where he leads responsible AI market research and thought leadership. Prior to H2O.ai, Ben held data science roles in high-profile teams at Ernst & Young, Nike, and NTT Data.