Chapter 1. Why Now and Challenges

Machine learning operations (MLOps) is quickly becoming a critical component of successful data science project deployment in the enterprise (Figure 1-1). It's a process that helps organizations and business leaders generate long-term value and reduce risk associated with data science, machine learning, and AI initiatives. Yet it's a relatively new concept; so why has it seemingly skyrocketed into the data science lexicon overnight? This introductory chapter delves into what MLOps is at a high level, its challenges, why it has become essential to a successful data science strategy in the enterprise, and, critically, why it is coming to the forefront now.

MLOPS VERSUS MODELOPS VERSUS AIOPS

MLOps (or ModelOps) is a relatively new discipline, emerging under these names particularly in late 2018 and 2019. The two—MLOps and ModelOps—are, at the time this book is being written, largely being used interchangeably. However, some argue that ModelOps is more general than MLOps, as it's not only about machine learning models but any kind of model (e.g., rule-based models). For the purpose of this book, we'll be specifically discussing the machine learning model life cycle and will thus use the term "MLOps."

AIOps, though sometimes confused with MLOps, is another topic entirely and refers to the process of solving operational challenges through the use of artificial intelligence (i.e., AI for DevOps). An example would be a form of predictive maintenance for network failures, alerting DevOps teams to possible problems before they arise. While important and interesting in its own right, AIOps is outside the scope of this book.

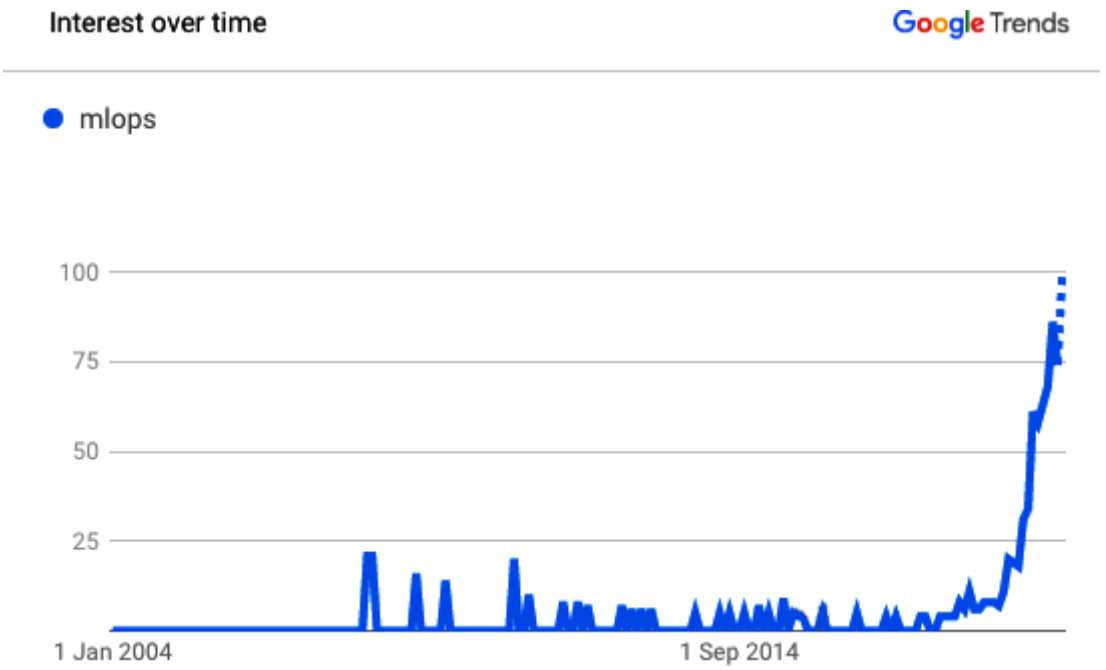

Figure 1-1. Representation of the exponential growth of MLOps (not the parallel growth of the term “ModelOps”)

Defining MLOps and Its Challenges

At its core, MLOps is the standardization and streamlining of machine learning life cycle management (Figure 1-2). But taking a step back, why does the machine learning life cycle need to be streamlined? On the surface, just looking at the steps to go from business problem to a machine learning model at a very high level, it seems straightforward.

For most traditional organizations, the development of multiple machine learning models and their deployment in a production environment are relatively new. Until recently, the number of models may have been manageable at a small scale, or there was simply less interest in understanding these models and their dependencies at a company-wide level. With decision automation (that is, an increasing prevalence of decision making that happens without human intervention), models become more critical, and, in parallel, managing model risks becomes more important at the top level.

The reality of the machine learning life cycle in an enterprise setting is much more complex, in terms of needs and tooling (Figure 1-3).

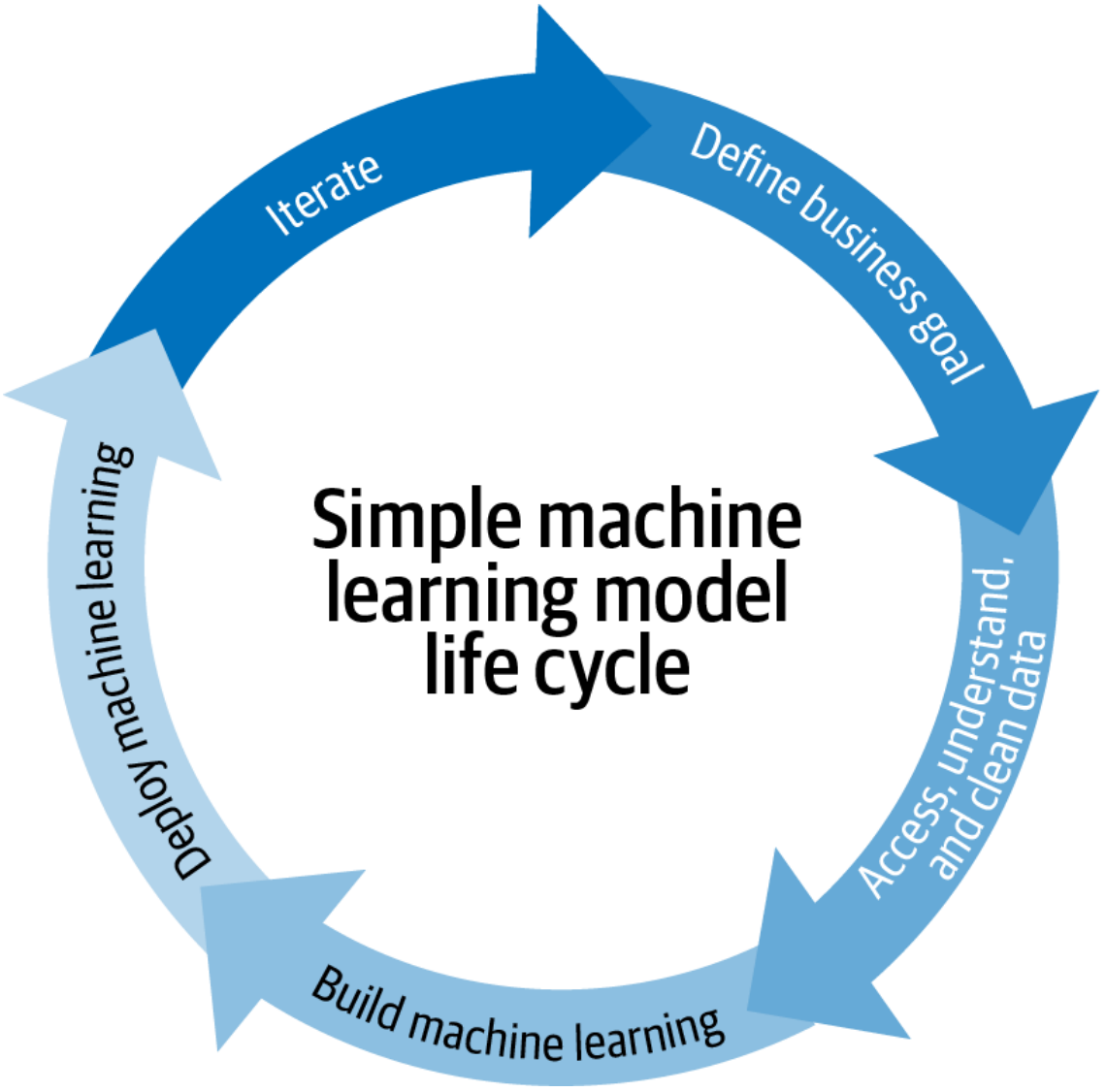

Figure 1-2. A simple representation of the machine learning model life cycle, which often underplays the need for MLOps, compared to Figure 1-3

There are three key reasons that managing machine learning life cycles at scale is challenging:

- There are many dependencies. Not only is data constantly changing, but business needs shift as well. Results need to be continually relayed back to the business to ensure that the reality of the model in production and on production data aligns with expectations and, critically, addresses the original problem or meets the original goal.
- Not everyone speaks the same language. Even though the machine learning life cycle involves people from the business, data science, and IT teams, none of these groups are using the same tools or even, in many cases, share the same fundamental skills to serve as a baseline of communication.

- Data scientists are not software engineers. Most are specialized in model building and assessment, and they are not necessarily experts in writing applications. Though this may start to shift over time as some data scientists become specialists more on the deployment or operational side, for now many data scientists find themselves having to juggle many roles, making it challenging to do any of them thoroughly. Data scientists being stretched too thin becomes especially problematic at scale with increasingly more models to manage. The complexity becomes exponential when considering the turnover of staff on data teams and, suddenly, data scientists have to manage models they did not create.

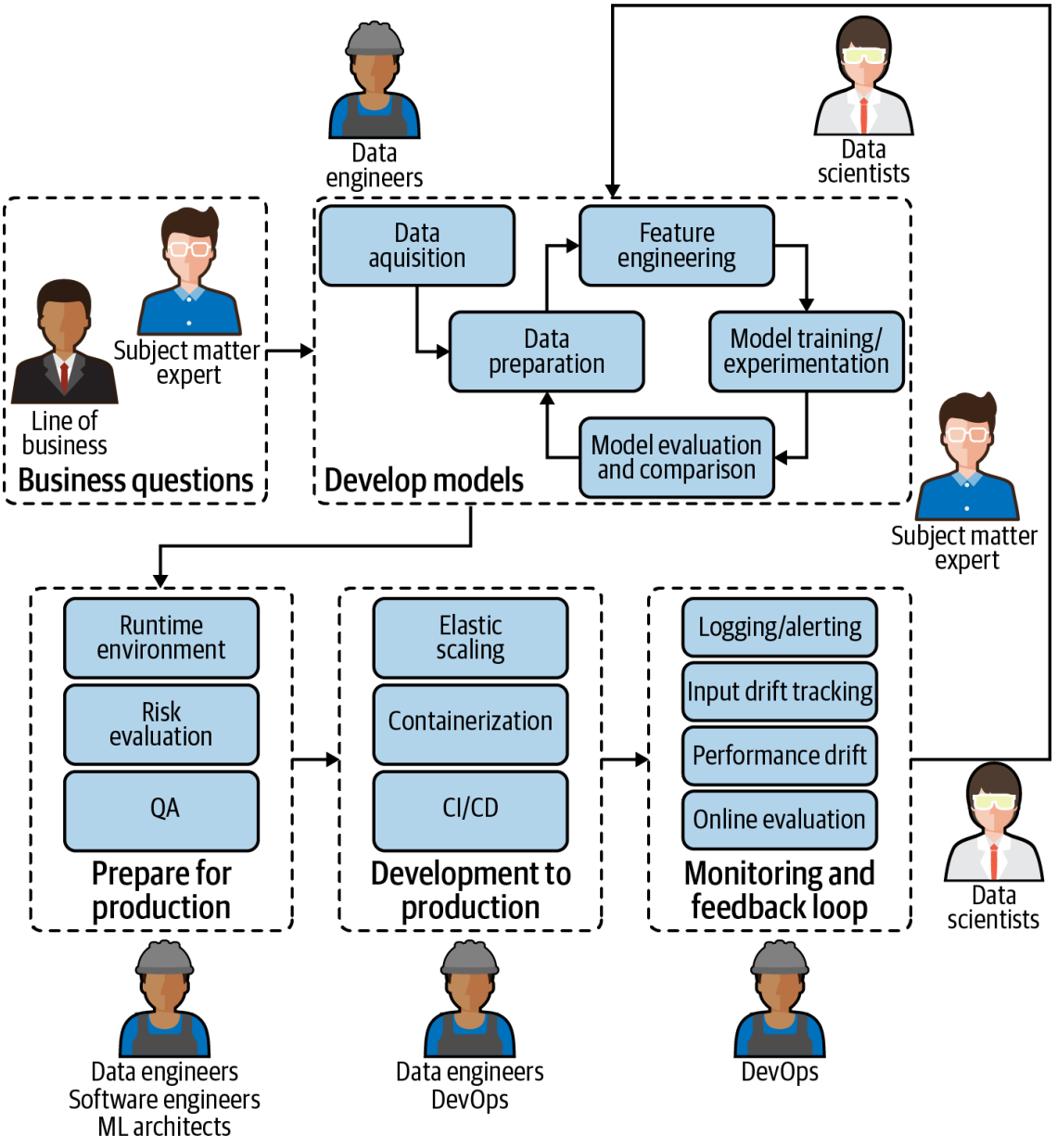

Figure 1-3. The realistic picture of a machine learning model life cycle inside an average organization today, which involves many different people with completely different skill sets and who are often using entirely different tools.

If the definition (or even the name MLOps) sounds familiar, that's because it pulls heavily from the concept of DevOps, which streamlines the practice of software changes and updates. Indeed, the two have quite a bit in common. For example, they both center around:

- Robust automation and trust between teams
- The idea of collaboration and increased communication between teams
- The end-to-end service life cycle (build, test, release)
- Prioritizing continuous delivery and high quality

Yet there is one critical difference between MLOps and DevOps that makes the latter not immediately transferable to data science teams: deploying software code into production is fundamentally different than deploying machine learning models into production. While software code is relatively static ("relatively" because many modern software-as-a-service [SaaS] companies *do* have DevOps teams that can iterate quite quickly and deploy in production multiple times per day), data is always changing, which means machine learning models are constantly learning and adapting—or not, as the case may be—to new inputs. The complexity of this environment, including the fact that machine learning models are made up of both code and data, is what makes MLOps a new and unique discipline.

WHAT ABOUT DATAOPS?

To add to the complexity of MLOps versus DevOps, there is also DataOps, a term introduced in 2014 by IBM. DataOps seeks to provide business-ready data that is quickly available for use, with a large focus on data quality and metadata management. For example, if there's a sudden change in data that a model relies on, a DataOps system would alert the business team to deal more carefully with the latest insights, and the data team would be notified to investigate the change or revert a library upgrade and rebuild the related partition.

The rise of MLOps, therefore, intersects with DataOps at some level, though MLOps goes a step further and brings even more robustness through additional key features (discussed in more detail in Chapter 3).

As was the case with DevOps and later DataOps, until recently teams have been able to get by without defined and centralized processes mostly because—at an enterprise level—they weren't deploying machine learning models into production at a large enough scale. Now, the tables are turning and teams are increasingly looking for ways to formalize a multi-stage, multi-discipline, multi-phase process with a heterogeneous environment and a framework for MLOps best practices, which is no small task. Part II of this book, "MLOps: How," will provide this guidance.

MLOps to Mitigate Risk

MLOps is important to any team that has even one model in production because, depending on the model, continuous performance monitoring and adjusting is essential. By allowing safe and reliable operations, MLOps is key in mitigating the risks induced by the use of ML models. However, MLOps practices do come at a cost, so a proper cost-benefit evaluation should be performed for each use case.

Risk Assessment

When it comes to machine learning models, risks vary widely. For example, the stakes are much lower for a recommendation engine used once a month to decide which marketing offer to send a customer than for a travel site whose pricing and revenue depend on a machine learning model. Therefore, when looking at MLOps as a way to mitigate risk, an analysis should cover:

- The risk that the model is unavailable for a given period of time
- The risk that the model returns a bad prediction for a given sample
- The risk that the model accuracy or fairness decreases over time
- The risk that the skills necessary to maintain the model (i.e., data science talent) are lost

Risks are usually larger for models that are deployed widely and used outside of the organization. As shown in Figure 1-4, risk assessment is generally based on two metrics: the probability and the impact of the adverse event. Mitigation measures are generally based on the combination of the two, i.e., the model's severity. Risk assessment should be performed at the beginning of each project and reassessed periodically, as models may be used in ways that were not foreseen initially.

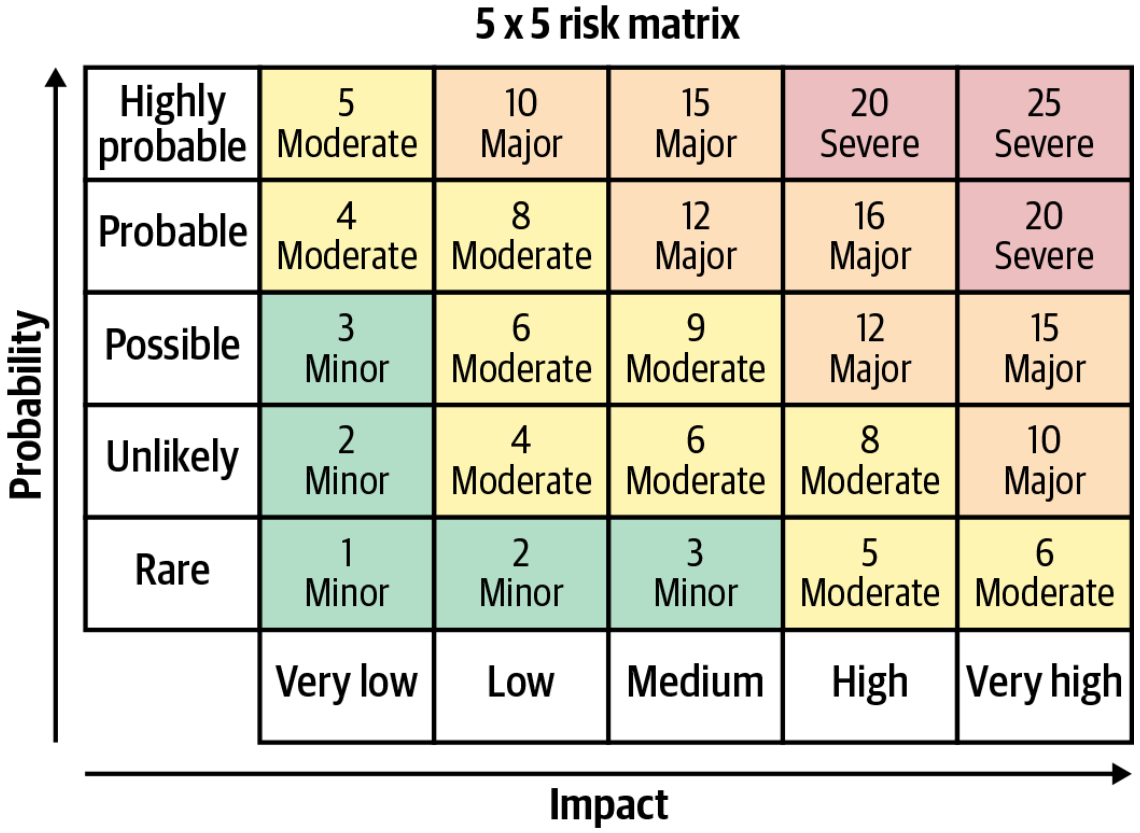

Figure 1-4. A table that helps decision makers with quantitative risk analysis

Risk Mitigation

MLOps really tips the scales as critical for risk mitigation when a centralized team (with unique reporting of its activities, meaning that there can be multiple such teams at any given enterprise) has more than a handful of operational models. At this point, it becomes difficult to have a global view of the states of these models without the standardization that allows the appropriate mitigation measures to be taken for each of them (see "Matching Governance with Risk Level").

Pushing machine learning models into production without MLOps infrastructure is risky for many reasons, but first and foremost because fully assessing the performance of a machine learning model can often only be done in the production environment. Why? Because prediction models are only as good as the data they are trained on, which means the training data must be a good reflection of the data encountered in the production environment. If the production environment changes, then the model performance is likely to decrease rapidly (see Chapter 5 for details).

Another major risk factor is that machine learning model performance is often very sensitive to the production environment it is running in, including the versions of software and operating systems in use. They tend not to be buggy in the classic software sense, because most weren't written by hand, but rather were machine-generated. Instead, the problem is that they are often built on a pile of open source software (e.g., libraries, like scikit-learn, Python, or Linux), and having versions of this software in production that match those that the model was verified on is critically important.

Ultimately, pushing models into production is not the final step of the machine learning life cycle—far from it. It's often just the beginning of monitoring its performance and ensuring that it behaves as expected. As more data scientists start pushing more machine learning models into production, MLOps becomes critical in mitigating the potential risks, which (depending on the model) can be devastating for the business if things go

wrong. Monitoring is also essential so that the organization has a precise knowledge of how broadly each model is used.

MLOps for Responsible AI

A responsible use of machine learning (more commonly referred to as Responsible AI) covers two main dimensions:

Intentionality

> Ensuring that models are designed and behave in ways aligned with their purpose. This includes assurance that data used for AI projects comes from compliant and unbiased sources plus a collaborative approach to AI projects that ensures multiple checks and balances on potential model bias. Intentionality also includes explainability, meaning the results of AI systems should be explainable by humans (ideally, not just the humans who created the system).

Accountability

> Centrally controlling, managing, and auditing the enterprise AI effort—no shadow IT! Accountability is about having an overall view of which teams are using what data, how, and in which models. It also includes the need for trust that data is reliable and being collected in accordance with regulations as well as a centralized understanding of which models are used for what business processes. This is closely tied to traceability: if something goes wrong, is it easy to find where in the pipeline it happened?

These principles may seem obvious, but it's important to consider that machine learning models lack the transparency of traditional imperative code. In other words, it is much harder to understand what features are used to determine a prediction, which in turn can make it much harder to demonstrate that models comply with the necessary regulatory or internal governance requirements.

The reality is that introducing automation vis-à-vis machine learning models shifts the fundamental onus of accountability from the bottom of the hierarchy to the top. That is, decisions that were perhaps previously made by individual contributors who operated within a margin of guidelines (for example, what the price of a given product should be or whether or not a person should be accepted for a loan) are now being

made by a model. The person responsible for the automated decisions of said model is likely a data team manager or even executive, and that brings the concept of Responsible AI even more to the forefront.

Given the previously discussed risks as well as these particular challenges and principles, it's easy to see the interplay between MLOps and Responsible AI. Teams must have good MLOps principles to practice Responsible AI, and Responsible AI necessitates MLOps strategies. Given the gravity of this topic, we'll come back to it multiple times throughout this book, examining how it should be addressed at each stage of the ML model life cycle.

MLOps for Scale

MLOps isn't just important because it helps mitigate the risk of machine learning models in production; it is also an essential component to massively deploying machine learning efforts (and in turn benefiting from the corresponding economies of scale). Going from one or a handful of models in production to tens, hundreds, or thousands that have a positive business impact requires MLOps discipline.

Good MLOps practices will help teams at a minimum:

- Keep track of versioning, especially with experiments in the design phase
- Understand whether retrained models are better than the previous versions (and promoting models to production that are performing better)
- Ensure (at defined periods—daily, monthly, etc.) that model performance is not degrading in production

Closing Thoughts

Key features will be discussed at length in Chapter 3, but the point here is that these are not optional practices. They are essential tasks for not only

efficiently scaling data science and machine learning at the enterprise level, but also doing it in a way that doesn't put the business at risk. Teams that attempt to deploy data science without proper MLOps practices in place will face issues with model quality and continuity—or, worse, they will introduce models that have a real, negative impact on the business (e.g., a model that makes biased predictions that reflect poorly on the company).

MLOps is also, at a higher level, a critical part of transparent strategies for machine learning. Upper management and the C-suite should be able to understand as well as data scientists what machine learning models are deployed in production and what effect they're having on the business. Beyond that, they should arguably be able to drill down to understand the whole data pipeline (i.e., the steps taken to go from raw data to final output) behind those machine learning models. MLOps, as described in this book, can provide this level of transparency and accountability.