O'REILLY®

What Is MLOps?

Generating Long-Term Value from Data Science & Machine Learning

Mark Treveil & Lynn Heidmann

REPORT

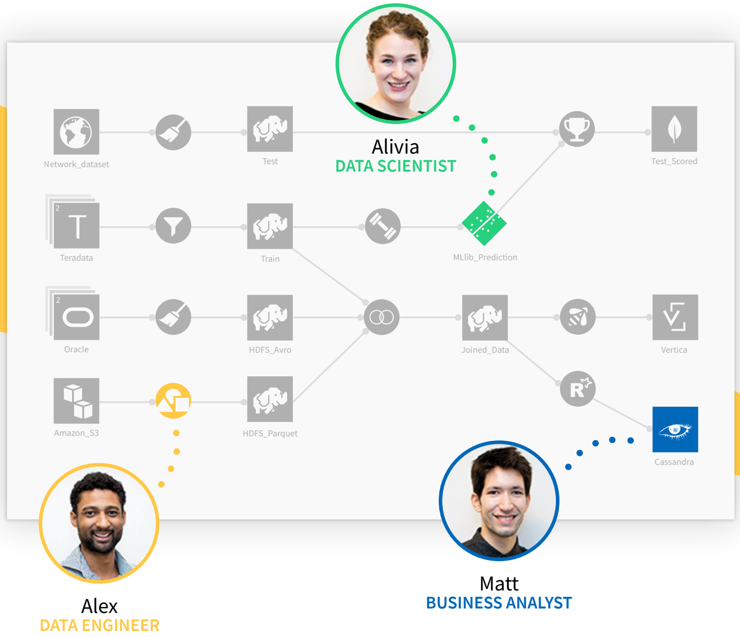
MASTERING MLOps
WITH DATAIKU

Dataiku is the only platform that provides one simple, consistent UI for data connection, wrangling, mining, visualization, machine learning, deployment, and model monitoring, all at enterprise scale.

Key features for a scalable MLOps strategy include:

1. Model input drift detection that looks at the recent data the model has had to score and statistically compares it with the data on which the model was evaluated.
2. Easier creation of validation feedback loops via Dataiku Evaluation Recipes to compute the true performance of a saved model against a new validation dataset, plus automated retraining and redeployment.
3. Dashboard interfaces dedicated to the monitoring of global pipelines.
4. ...and more! Go in-depth on all the features Dataiku has to offer with the complete data sheet.

→ GET THE DATAIKU DATA SHEET



What Is MLOps?

Generating Long-Term Value from Data Science and Machine Learning

Mark Treveil and Lynn Heidmann

Beijing · Boston · Farnham · Sebastopol · Tokyo

O'REILLY®

What Is MLOps?

by Mark Treveil and Lynn Heidmann

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
Development Editor: Angela Rufino
Production Editor: Kate Galloway
Copyeditor: Piper Editorial, LLC

Proofreader: Asa Tomash
Interior Designer: David Futato
Cover Designer: Randy Comer
Illustrator: Kate Dullea

November 2020: First Edition

Revision History for the First Edition

2020-10-29: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781492093619* for release details.

The views expressed in this work are those of the authors, and do not represent the publisher's views. While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

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

978-1-492-09359-6

[LSI]

Table of Contents

CHAPTER 1

Introduction to MLOps and the AI Life Cycle

In the wake of the global health crisis of 2020, the question of scaling AI in the enterprise has never been more pressing. As many industries try to cope with the instability of a changing landscape, data science, machine learning (ML), and AI have moved from experimental initiatives to necessities.

Despite the growing need for AI to bring a newfound agility to a post-pandemic world, businesses still struggle to pivot their operations around these technologies precisely because it's not simply a matter of technology; processes and people are also critically important. This report will introduce the data science, ML, and AI project life cycle so that readers can understand what (and who) drives these projects before covering MLOps (short for *machine learning operations*), a process that brings the required agility and allows for massive scaling of AI initiatives across the enterprise.

Why Are AI Projects So Complex to Execute?

It's important to understand the challenges that AI projects present in order to properly address and overcome them with good MLOps practices. So, why are AI projects so complex, and why do so many organizations struggle to execute them (even those that succeed in other complex processes and in software development)?

There are two fundamental reasons for this.

Business Needs (and Data) Are Not Static

Not only is data constantly changing, but business needs shift as well. Results of ML models (i.e., mathematical models based on sample data that output predictions—Chapter 2 covers what ML is more in detail) need to be continually relayed back to the business to ensure that the reality of the model aligns with expectations and—critically—addresses the original problem or meets the original goal.

For example, take this (unfortunately) common scenario: let's say a data team is presented with a business problem, and the team has six months to come up with a solution. The team spends months cleaning data, building models, and refining information into visualizations according to the initial project parameters.

Six months later, the data team presents their work to the business team, and the response is, "Great! Unfortunately, since the start of the project, the original data has changed and so has the behavior of our customers." That's six months of wasted effort and time, and it's back to the drawing board.

Perhaps four additional months pass as the data is refined and tweaked again, only to be told that the original project parameters have changed yet again. Rinse, repeat. The vicious circle has only just begun, and with no particular end in sight. The expiration of data (and the changing nature of business, especially in the context of the 2020 health crisis) constantly invalidates models' relevance. If data teams work in a bubble, then their solutions won't be relevant or provide value outside that bubble.

Not Everyone Speaks the Same Language

Even though AI projects involve people from the business, data science, and IT teams, none of these groups are using the same tools or even—in many cases—sharing the same fundamental skills to serve as a baseline of communication.

Some symptoms of serious communication barriers include:

First communication between teams at the end of the project

Successful data teams and data projects involve experts in IT, business, and data science from the start. Pulling in expertise at the last minute when most of the work is already done is extremely costly and is a sign of larger organizational issues around AI projects.

Lack of strong leadership

If team leaders don't support horizontal collaboration (between team members with the same profile or background—for example, data scientists) as well as vertical collaboration (between different types of profiles, like between business and IT), AI projects are doomed to fail.

Problems with tracking and versioning

It doesn't take long for email threads to grow in length. Using email to share files is a recipe for disaster when it comes to keeping track of content and for data versioning. Expect the loss of data and noninclusion of key stakeholders.

Lack of strong data governance policies

Organizations typically implement policies for the sharing of content and data protection, but "shadow IT," or the deployment of other policies or systems outside of a central team (which can differ widely across the organization) can, again, be a sign of deeper issues with the organizational structure around the AI project life cycle.

Other Challenges

In addition to these two primary challenges, there are many other smaller inefficiencies that prevent businesses from being able to scale AI projects (and for which, as we'll see later in this report, MLOps provides solutions). For example, the idea of reproducibility: when companies do not operate with clear and reproducible workflows, it's very common for people working in different parts of the company to unknowingly be working on creating exactly the same solution.

From a business perspective, getting to the 10th or 20th AI project or use case usually still has a positive impact on the balance sheet, but eventually, the marginal value of the next use case is lower than the marginal costs (see Figures 1-1 and 1-2).

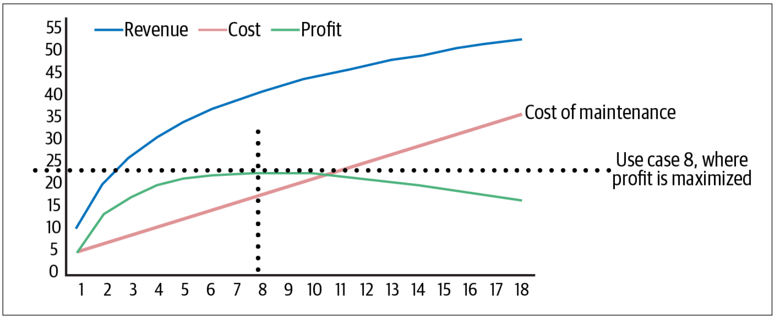


Figure 1-1. Cumulative revenues, costs, and profits over time (number of use cases). Note that after use case 8, profit is decreasing due to increased costs and stagnation of revenue.

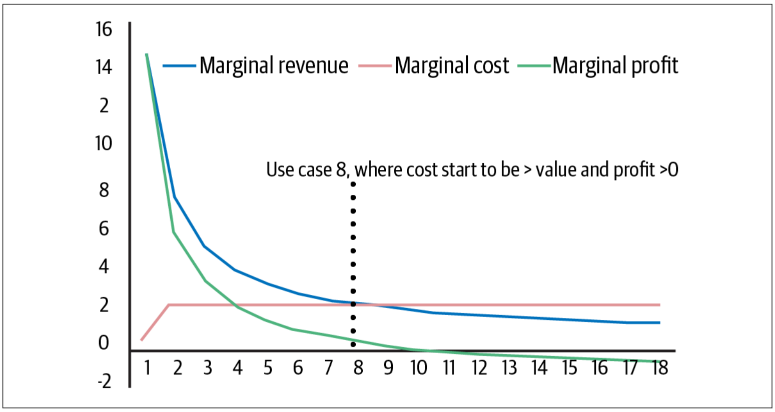


Figure 1-2. Marginal revenue, cost, and profit over time (number of use cases).

One might see these figures and conclude that the most profitable way to approach AI projects is to only address the top 5 to 10 most valuable use cases and stop. But this does not take into account the continued cost of AI project maintenance.

Adding marginal cost to the maintenance costs will generate negative value and negative numbers on the balance sheet. It is, therefore, economically impossible to scale use cases, and it's a big mistake to think that the business will be able to easily generalize Enterprise AI everywhere by simply taking on more AI projects throughout the company.

Ultimately, to continue seeing returns on investment (ROI) in AI projects at scale, taking on exponentially more use cases, companies must find ways to decrease both the marginal costs and incremental maintenance costs of Enterprise AI. Robust MLOps practices, again, are one part of the solution.

On top of the challenges of scaling, a lack of transparency and lack of workflow reusability generally mean there are poor data governance practices happening. Imagine if no one understands or has clear access to work by other members of the data team—in case of an audit, figuring out how data has been treated and transformed as well as what data is being used for which models becomes nearly impossible. With members of the data team leaving and being hired, this becomes exponentially more complicated.

For those on the business side, taking a deeper look into the AI project life cycle and understanding how—and why—it works is the starting point to addressing many of these challenges. It helps bridge the gap between the needs and goals of the business and those of the technical sides of the equation to the benefit of the Enterprise AI efforts of the entire organization.

The AI Project Life Cycle

Looking at the *data science, ML, and AI project life cycle*—henceforth shortened to *AI project life cycle*—can help contextualize these challenges. In practice, how does one go from problem to solution? From raw data to AI project?

Surface level, it seems straightforward (see Figure 1-3): start with a business goal, get the data, build a model, deploy, and iterate. However, it's easy to see how managing multiple AI projects throughout their life cycle, especially given the aforementioned challenges, can quickly become difficult in and of itself.

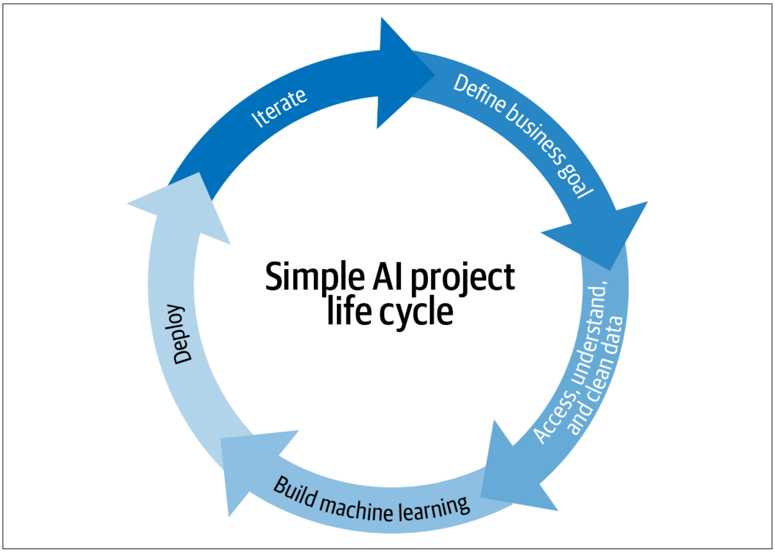

Simple AI project life cycle
Define business goal
Access, understand, and clean data
Build machine learning
Deploy
Iterate

Figure 1-3. A simple representation of the AI project life cycle.

Even though ML models are primarily built by data scientists, that doesn't mean that they own the entire AI project life cycle. In fact, there are many different types of roles that are critical to building AI projects, including most notably:

Subject matter experts on the business side
: While the data-oriented profiles (data scientist, engineer, architect, etc.) have expertise across many areas, one area where they tend to lack is a deep understanding of the business and the problems or questions at hand that need to be addressed using ML.

Data scientists
: Though most see data scientists' role in the ML model life cycle as strictly the model-building portion, it is actually—or at least, it should be—much wider. From the very beginning, data scientists need to be involved with subject matter experts, understanding and helping to frame business problems in such a way that they can build a viable ML solution.

Architects
: AI projects require resources, and architects help properly allocate those resources to ensure optimal performance of ML

models. Without the architect role, AI projects might not perform as expected once they are used.

Software engineers and traditional DevOps
Software engineers usually aren't building ML models, but on the other hand, most organizations are not producing only ML models. When it comes to deploying AI projects into the larger business and making sure they work with all the other non-AI systems, these roles are critically important.

After considering all these different roles plus breaking down the steps of the AI life cycle more granularly, the picture becomes much more complex (see Figure 1-4).

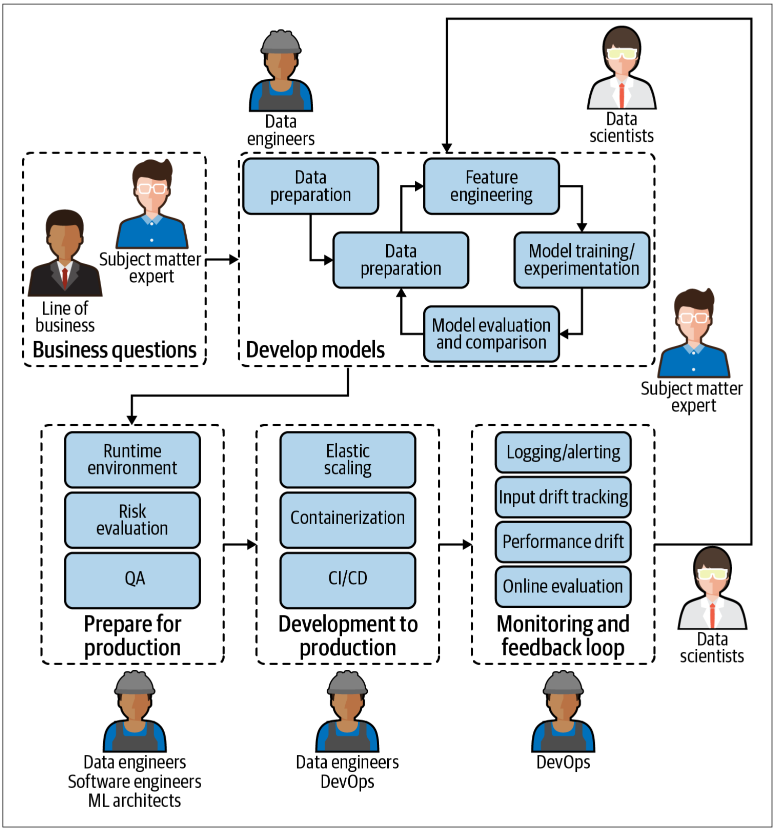


Figure 1-4. The realistic picture of an ML model life cycle inside an average organization today, which involves many different people with completely different skill sets and who are often using entirely different tools.

Given the complexity of the nature of AI projects themselves, the AI project life cycle in the organization, and the number of people across the business that are involved, companies looking to scale AI efforts need a system that keeps track of all the intricacies. That's where MLOps comes into play.

At its core, MLOps is the standardization and streamlining of data science, ML, and AI project life cycle management. For most traditional organizations, working with multiple ML models is still 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. Now, the tables are turning and organizations are increasingly looking for ways to formalize a multistage, multidiscipline, multiphase process with a heterogeneous environment and a framework for MLOps best practices, which is no small task.

The Role of MLOps in the AI Project Life Cycle

Some believe that deploying ML models in production (i.e., feeding them real data and making them a part of business operations) is the final step—or one of the final steps—of the AI project life cycle. This is far from the case; in fact, it's often just the beginning of monitoring their performance and ensuring that they behave as expected.

MLOps isn't one specific step in the life cycle or a check along the way before passing from one step to another. Rather, MLOps is an underlying process that encompasses and informs all of the steps in the AI project life cycle, helping the organization:

Reduce risk

Using ML models to drive automatic business decisions without MLOps infrastructure is risky for many reasons, first and foremost because fully assessing the performance of an ML 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 if—and more like when—data changes, the model performance is likely to decrease rapidly. This translates to any number of undesirable business results, from bad press to poor customer experience.

Introduce transparency

MLOps is a critical part of transparent strategies for ML. Upper management, the C-suite, and data scientists should all be able to understand what ML models are being used by the business and what effect they're having. Beyond that, they should arguably be able to drill down to understand the whole data pipeline behind those ML models. MLOps, as described in this report, can provide this level of transparency and accountability.

Build Responsible AI

The reality is that introducing automation vis-à-vis ML 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 machine. Given the potential risks of AI projects as well as their particular challenges, 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.

Scale

MLOps is important not only because it helps mitigate the risk, but also because it is an essential component to scaling ML efforts (and in turn benefiting from the corresponding economies of scale). To go from the business using one or a handful of models to tens, hundreds, or thousands of models that positively impact the business requires MLOps discipline.

Each of these points is an important yet challenging part of the transformation of the organization around data. The next section will go more in depth on the rise of MLOps and the role it plays in the organization's success in AI initiatives.

MLOps: What Is It, and Why Now?

Machine learning is not new, and neither is its use in business contexts. So why is MLOps—or the systematic streamlining of AI projects—becoming a popular topic now (see Figure 1-5)? Until recently, teams have been able to get by without defined and centralized MLOps processes mostly because, at an enterprise level, they weren't leveraging ML models on a large enough scale.

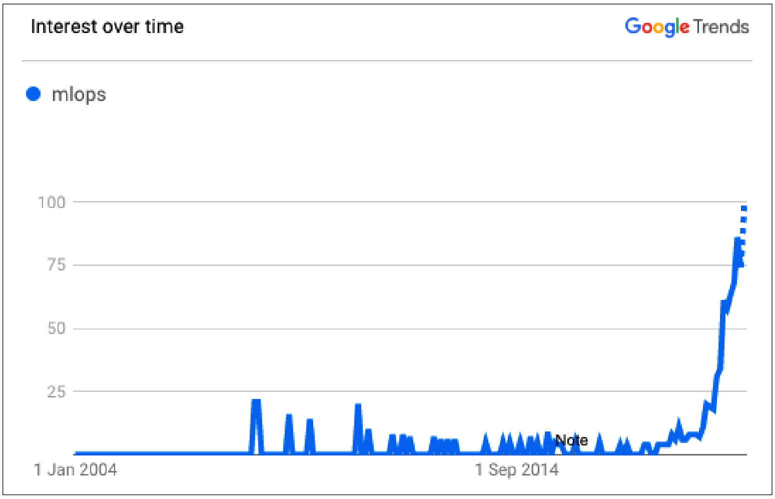


Figure 1-5. The exponential growth of MLOps. This represents only the growth of MLOps, not the parallel growth of the term ModelOps (subtle differences explained in the sidebar MLOps versus ModelOps versus AIOps).

That's not to say that MLOps is only important for organizations creating lots of AI projects. In fact, MLOps is important to any team that has even one model in production, as depending on the model, continuous monitoring and adjusting is essential. Think about a travel site whose pricing model would require top-notch MLOps to ensure that the model is continuously delivering business results and not causing the company to lose money.

However, 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 some standardization.

This Sounds Familiar...

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 as well as high quality

Yet there is one critical difference between MLOps and DevOps that makes the latter not immediately transferable to data science teams, and it relates to one of the challenges presented in the beginning of this chapter: deploying software code in production is fundamentally different than deploying ML models into production.

While software code is relatively static, data is always changing, which means ML 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 ML models are made up of both code and data, is what makes MLOps a new and unique discipline.

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 report is being written and published, 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 report, we'll be specifically discussing the ML model life cycle and will thus use 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 but 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.

Key Components of a Robust MLOps Practice

Good MLOps practices will help teams on both the business and tech side at a minimum:

- Keep track of different model versions, i.e., different variations of models with the same ultimate business goal to test and find the best one
- Understand if new versions of 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

At a more detailed level, there are five key components of MLOps: development, deployment, monitoring, iteration, and governance. The bulk of this report will cover at a high level the three components that are most important for those on the business side to understand (both conceptually and in terms of the role of the business in those components): development, monitoring, and governance.[1]

Closing Thoughts

MLOps is critical—and will only continue to become more so—to both scaling AI across an enterprise as well as ensuring it is deployed in a way that minimizes risk. Both of these are goals with which business leaders should be deeply concerned.

While certain parts of MLOps can be quite technical, it's only in streamlining the entire AI life cycle that the business will be able to develop AI capabilities to scale their operations. That's why business leaders should not only understand the components and complexities of MLOps, but have a seat at the table when deciding which tools or processes the organization will follow and use to execute.

[1] This report will cover the other two components (deployment and iteration) only at a high level. Those looking for more detail on each component should read *Introducing MLOps* (O'Reilly).

The next chapter is the first to dive into the detail of MLOps, starting with the development of ML models themselves. Again, the value of understanding MLOps systems at this level of detail for business leaders is to be able to drive efficiencies from business problems to solutions. This is something to keep in mind throughout Chapters 2–4.

CHAPTER 2

Developing and Deploying Models

To understand the key components of MLOps for business and subject matter experts, it's essential to first have a baseline understanding of how machine learning works. At its core, ML is the science of computer algorithms that automatically learns and improves from experience rather than being explicitly programmed. The algorithms analyze sample data—known as training data—to build a software model that can make predictions. ML algorithms can tackle problems that were either infeasible or too costly with previous software algorithms.

For example, an image recognition model might be able to identify the type of electricity meter from a photograph by searching for key patterns in the image that distinguish each type of meter. Another concrete example is an insurance recommender model, which might suggest additional insurance products that a specific existing customer is most likely to buy based on the previous behavior of similar customers.

When faced with unseen data, be it a photo or a customer, the ML model uses what it has learned from previous data to make the best prediction it can based on the assumption that the unseen data is somehow related to the previous data.

With the basics out of the way, this chapter will move on to more detailed components of ML model building and identify points in this process where business insights can provide particular value to a technical team.

Why the Development Process Matters to the Business

The process of developing an ML model typically starts with a business objective, which can be as simple as reducing fraudulent transactions to < 0.1% or having the ability to identify people's faces on their social media photos. Business objectives naturally come with performance targets, technical infrastructure requirements, and cost constraints; all of these factors can be captured as key performance indicators, or KPIs, which will ultimately enable the business performance of models in production to be monitored.

It's important to recognize that ML projects don't happen in a vacuum—they are generally part of a larger project that in turn impacts technologies, processes, and people. That means part of establishing objectives also includes change management, which may even provide some guidance for how the ML model should be built. For example, the required degree of transparency will strongly influence the choice of algorithms and may drive the need to provide explanations together with predictions so that predictions are turned into valuable decisions at the business level.

With clear business objectives defined, the next steps in the model development process include data selection, feature engineering, and model training, all of which will be covered in the remainder of this section.

Data Selection

Data selection sounds simple, but can often be the most arduous part of the journey once one delves into the details to see what's at stake and all the factors that influence data reliability and accuracy. Key questions for finding data to build ML models include (but are not limited to):

- What relevant datasets are available?
- Is this data sufficiently accurate and reliable?
- How can stakeholders get access to this data?
- What data properties (known as features) can be made available by combining multiple sources of data?
- Will this data be available in real time?

- Is there a need to label some of the data with the “ground truth” that is to be predicted, or does unsupervised learning make sense? If so, how much will this cost in terms of time and resources? What platform should be used?
- How will data be updated once the model is deployed?
- Will the use of the model itself reduce the representativeness of the data?
- How will the KPIs, which were established along with the business objectives, be measured?

The constraints of data governance bring even more questions, including:

- Can the selected datasets be used for this purpose?
- What are the terms of use?
- Is there personally identifiable information (PII) that must be redacted or anonymized?
- Are there features, such as gender, that legally cannot be used in this business context?
- Are minority populations sufficiently well represented so that the model has equivalent performances on each group?

For the business side, these questions are critical to building AI that is responsible and, by extension, doesn’t put the organization at risk.

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 that 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 ML models lack the transparency of traditional software code. In other words, it is much harder to understand what specific parts of datasets 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.

Feature Engineering

Feature engineering is the process of taking raw data from the selected datasets and transforming it into "features" that better represent the underlying problem to be solved. It includes data cleansing, which can represent the largest part of a project in terms of time spent.

When it comes to feature creation and selection, the question of how much and when to stop comes up regularly. Adding more features may produce a more accurate model or achieve more fairness. However, it also comes with downsides, all of which can have a significant impact on MLOps strategies down the line:

- The model can become more and more expensive to compute
- More features require more inputs and more maintenance down the line
- More features mean a loss of some model stability
- The sheer number of features can raise privacy concerns

Model Training

After data preparation by way of feature engineering and selection, the next step is model training. The process of training and optimizing a new ML model is iterative. In addition to—or in many cases because of—its iterative nature, training is also the most intensive step of the ML model life cycle when it comes to computing power.

Though many on the business side traditionally think that it's the responsibility of data scientists to develop ML models, as we've seen in this section, there are many steps in the process where things can go wrong without input from business specialists. Taking a more active approach to model development can smooth and maybe even shorten the overall process as well as result in models that are more closely aligned with business goals.

Model Deployment

For people on the business and not the technical side, it can be difficult to understand exactly what it means to deploy an ML model or AI project, and more importantly, why it matters. It's probably not necessary for most readers to understand the "how" of model deployment in detail—it is quite a complex process that is mostly handled by data engineers, software engineers, and/or DevOps.[1]

However, it is valuable to know the basics in order to be a more productive participant in the AI project life cycle, and more broadly, in MLOps processes. For example, when detailing business requirements, it's helpful to have a baseline understanding of the types of model deployment in order to have a richer discussion about what makes sense for the use case at hand.

Deploying an ML model simply means integrating it into an existing production environment. For example, a team might have spent several months building a model to detect fraudulent transactions. However, after that model is developed, it needs to actually be deployed. In this case, that means integrating it into existing processes so that it can actually start scoring transactions and returning the results.

[1] For readers who want to dive more into the details, we recommend the guidebook *Data Science Operationalization*.

There are two types of model deployment:

Model as a service, or live-scoring model
: Requests are handled in real time. For the fraudulent transaction example, this would mean as each transaction happens, it is scored. This method is best reserved for cases (like fraud) where predictions need to happen right away.

Embedded model
: Here the model is packaged into an application, which is then published. A common example is an application that provides batch scoring of requests. This type of deployment is good if the model is used on a consistent basis, but the business doesn't necessarily require the predictions in real time.

Again, being knowledgeable about these two types of model deployment on the business side can help inform more productive discussions with technical teams about how ML models should be released and how they can be used. The next section will explore this notion even further, specifically looking at how model development and deployment fit into the bigger picture of cross-organizational MLOps processes.

MLOps for Model Development and Deployment

The process of training and optimizing a new ML model is iterative, and several different algorithms may be tested. Why? Because some ML algorithms can best support specific use cases, and governance considerations may also play a part in the choice of algorithm. In particular, highly regulated environments where decisions must be explained (e.g., financial services) cannot use opaque algorithms and have to favor simpler techniques.

Keeping track of the results of each experiment when iterating becomes complex quickly. Nothing is more frustrating for the business than a data scientist not being able to recreate the best results because they cannot remember the precise configuration used. An experiment tracking tool can greatly simplify the process of remembering the data, features selection, and model parameters alongside the performance metrics. These enable experiments to be compared side by side, highlighting the differences in performance.

In addition, while many experiments may be short-lived, significant versions of a model need to be saved for possible later use. The challenge here is reproducibility—without reproducibility, data scientists have little chance of being able to confidently iterate on models.

The Role of MLOps in Explainability and Transparency

MLOps also plays a role in model development when it comes to explainability and transparency. ML models are fundamentally challenging to understand—it is a consequence of their statistical nature. While model algorithms come with standard performance measures to assess their efficacy, these don't explain how the predictions are made. However, understanding how predictions are made is one key way to check that the model is working as expected, to improve on feature engineering, and it also may be necessary to ensure that fairness requirements (e.g., around features like sex, age, or race) have been met.

Explainability techniques are becoming increasingly important as global concerns grow about the impact of unbridled AI. They offer a way to mitigate uncertainty and help prevent unintended consequences.

Ultimately, introducing MLOps ensures that during the model development process, both technical and business teams can document, keep track of, and be on the same page about different model versions. It will also ensure the models can be reproduced and explained. MLOps processes ensure that this early—yet critical—step in the AI project life cycle is executed in a way that will ensure success for the rest of the steps in the cycle, namely deployment and iteration.

MLOps to Mitigate Risk in Project Deployment

When it comes to model deployment, the role of MLOps is all about mitigating risk. Generally speaking, the broader the model deployment, the greater the risk. When risk impact is high enough, it is essential to control the deployment of new model versions, which is where tightly controlled MLOps processes come into play. Progressive, or canary, rollouts should be common practice, with models slowly served to parts of the organization or the customer base while simultaneously monitoring behavior and getting human feedback if appropriate.

Complex interactions between models is also a very real source of risk in the deployment stage. This class of issue will be a growing concern as ML models become pervasive, and it's an important potential area of focus for MLOps systems. Obviously, adding models will often add complexity to an organization, but the complexity does not necessarily grow linearly in proportion to the number of models; having two models is more complicated to understand than the sum since there are potential interactions between them.

Closing Thoughts

Though traditionally people on the business side, and especially business leaders, aren't the ones developing or deploying ML models, they have a vested interest in ensuring that they understand the processes and establish firm MLOps guidelines to steer them.

In these stages, carelessness (it's important to note that blunders are usually accidental and not intentional) can put the organization at risk—a poorly developed model can, at best, seriously affect revenue, customer service, or other processes. At worst, it can open the floodgates to a PR disaster.

CHAPTER 3
Model Monitoring and Iteration

Since ML models are effectively models of the data they were trained on, they can degrade over time. This is not a problem faced by traditional software, but it is inherent to machine learning. ML mathematics builds a concise representation of the important patterns in the training data with the hope that this is a good reflection of the real world. If the training data reflects the real world well, then the model should be accurate and, thus, useful.

But the real world doesn't stand still. The training data used to build a fraud detection model six months ago won't reflect a new type of fraud that started to occur in the last three months. If a given website starts to attract an increasingly younger user base, then a model that generates advertisements is likely to produce less and less relevant ads.

Once a model is in use, it is crucial that it continues to perform well over time. But good performance means different things to different people, in particular to data scientists and to the business. This chapter will take a closer look at the monitoring and iteration steps of the AI project life cycle and the role the business plays in the utility of both processes.

Why Model Monitoring Matters

Model moderating and iteration is the bread and butter of MLOps. And when it comes to monitoring the performance of models and of AI projects, it's important to recognize that everyone in the room (in other words, everyone involved with the AI project life cycle) has different priorities.

For IT or DevOps

The concerns of the DevOps team are very familiar and include questions like:

- Is the model getting the job done quickly enough?
- Is it using a sensible amount of memory and processing time?

This is traditional IT performance monitoring, and DevOps teams know how to do this well already. The resource demands of ML models are not so different from traditional software in this respect. However, as we'll see later, a model can meet both of these requirements and still not be effective, as defined by the business (which, ultimately, is the only definition that matters—if models are not useful for the business, why use them?).

For Data Scientists

The data scientist is interested in monitoring ML models for a new, more challenging reason: as alluded to in the beginning of this chapter, it's critical to understand that ML models—and thus, by extension, AI projects—can degrade over time.

How can data scientists tell a model's performance is degrading? It's not always easy. There are two common approaches, one based on ground truth and the other on input drift. Understanding each concept at a high level is important to facilitating conversations with data scientists about how to address the problem in a way that is best not just for the data scientist, but from the business perspective as well.

Ground truth

The ground truth is the correct answer to the question that the model was asked to solve, for example, "Is this credit card transaction actually fraudulent?" In knowing the ground truth for all predictions a model has made, one can judge how well that model is performing.

Sometimes ground truth is obtained rapidly after a prediction, for example, in models deciding which advertisements to display to a user on a web page. The user is likely to click on the advertisements within seconds, or not at all.

However, in many use cases, obtaining the ground truth is much slower. If a model predicts that a transaction is fraudulent, how can this be confirmed? In some cases, verification may only take a few minutes, such as a phone call placed to the cardholder. But what about the transactions the model thought were OK but actually weren't? The best hope is that they will be reported by the cardholder when they review their monthly transactions, but this could happen up to a month after the event (or not at all).

In the fraud example, ground truth isn't going to enable data science teams to monitor performance accurately on a daily basis. If the situation requires rapid feedback, then input drift may be a better approach.

Input drift

Input drift is based on the principle that a model is only going to predict accurately if the data it was trained on is an accurate reflection of the real world. So, if a comparison of recent requests to a deployed model against the training data shows distinct differences, then there is a strong likelihood that the model performance is compromised.

This is the basis of input drift monitoring. The beauty of this approach is that all the data required for this test already exists—no need to wait for ground truth or any other information.

For the Business

The business has the advantage of bringing a holistic outlook on monitoring, and some of their concerns might include questions like:

- Is the model delivering value to the enterprise?
- Do the benefits of the model outweigh the cost of developing and deploying the model? (And how can we measure this?)

The KPIs identified for the original business objective are one part of this process. Where possible, these should be monitored automatically, but this is rarely trivial. The previous example objective of reducing fraud to less than 0.1% of transactions is reliant on establishing the ground truth. But even monitoring this doesn't answer the question: what is the net gain to the business in dollars?

This is an age-old challenge for software, but with ever-increasing expenditure on ML, the pressure for data scientists to demonstrate value is only going to grow. In the absence of a *dollar-o-meter*, effectively monitoring the business KPIs is the best option available. The choice of the baseline is important here and should ideally allow for differentiation of the value of the ML subproject specifically and not of the global project. For example, the ML performance can be assessed with respect to a rule-based decision model based on subject matter expertise to set apart the contribution of decision automation from ML.

The bottom line is that at some point, performance of AI projects in use will be unacceptable, and model retraining becomes necessary. How soon models need to be retrained will depend on how fast the real world is changing and how accurate the model needs to be (for example, by nature, an ecommerce recommendation engine does not need to be as accurate as fraud detection model), but also—importantly—how easy it is to build and deploy a better model. That's where (you guessed it) good MLOps practices come into play again.

The danger of models being reliant on history was perfectly demonstrated with the 2020 health crisis. An almost overnight change in the behavior of business and customers across the world rendered AI models in most industries useless. The patterns of activity had changed, the assumptions in the models were no longer valid, and a complete rethink was an urgent necessity.

At such points of discontinuity, models based on data history cannot simply be rebuilt with new data—there isn't enough available. Worse, the patterns are likely still in flux. The best possible outcome is the early identification of the degradation of the models and a swift replacement with simpler, rules-based strategies. Humans-in-the-loop are an essential part of managing such catastrophes, and swift action with the input of domain experts is essential to avoiding long-term damage to the business.

MLOps for Model Iteration

Developing and deploying improved versions of a model is an essential part of the MLOps life cycle, and one of the more challenging. There are various reasons to develop a new model version, one of which is model performance degradation due to model drift, as discussed in the prior section. Sometimes there is a need to reflect refined business objectives and KPIs, and other times, it's just that the data scientists have come up with a better way to design the model.

In some fast-moving business environments (think ecommerce, specifically in the case of a ML-driven recommendation engine to suggest related products), new training data becomes available every day.

Daily retraining and redeployment of the model is often automated to ensure that the model reflects recent experience as closely as possible. In the ecommerce example, consumer preferences and needs change so frequently that it's easy to see—especially in the context of the 2020 business environment—why the model trained last month probably won't work as well again this month.

Retraining an existing model with the latest training data is the simplest scenario for iterating a new model version. But while there are no changes to feature selection or algorithm, there are still plenty of pitfalls. In particular:

- Is the new training data in line with what's expected? Automated validation of the new data through predefined metrics and checks is essential.
- Is the data complete and consistent?
- Are the distributions of features broadly similar to those in the previous training set? Remember that the goal is to refine the model, not radically change it.

With a new model version built, the next step is to compare the metrics with the current live model version. Doing so requires evaluating both models on the same development dataset, whether it be the previous or latest version (this ensures an apples-to-apples comparison of the models, so to speak). Of course, if metrics and checks suggest a wide variation between the models, data scientists should intervene manually and likely consult the business rather than deploying the new model automatically.

However, to give an idea of the complexity, even in the "simplest" automated retraining scenario with new training data, there is a need for multiple development datasets based on scoring data reconciliation (with ground truth when it becomes available), data cleaning and validation, the previous model version, and a set of carefully considered checks. Retraining in other scenarios is likely to be even more complicated, rendering automated redeployment unlikely.

As an example, consider retraining motivated by the detection of significant input drift. How can the model be improved? If new training data is available, then retraining with this data is the action with the highest benefit-cost ratio, and it may suffice. However, in environments where it's slow to obtain the ground truth, there may be little new labeled data.

This case requires direct invention from data scientists who need to understand the cause of the drift and work out how the existing training data could be adjusted to more accurately reflect the latest input data. Evaluating a model generated by such changes is difficult. The data scientist will have to spend time assessing the situation—time that increases with the amount of modeling debt—as well as estimate the potential impact on performance and design custom mitigation measures. For example, removing a specific feature or sampling the existing rows of training data may lead to a better-tuned model.

The Feedback Loop

From a business perspective, it's important to understand why the AI project feedback loop is challenging. In a nutshell, it's because traditional DevOps best practice inside large enterprises will typically dictate that the live model scoring environment and the model retraining environment are distinct. As a result, the evaluation of a new model version on the retraining environment is likely to be compromised.

One approach to mitigating this uncertainty is shadow testing, where the new model version is deployed alongside the existing deployed model. All live scoring is handled by the incumbent model version, but each new request is then scored again by the new model version and the results logged, but not returned to the requestor. Once sufficient requests have been scored by both versions, the results can be compared statistically. Shadow scoring also gives more visibility to the subject matter experts on the future versions of the model and may allow for a smoother transition (see Figure 3-1, left).

For some use cases (like a model that generates and serves the right advertisement for a given user on a given site), it is impossible to tell if the ads selected by the model are good or bad without allowing the end user the chance to click on them. In this use case, shadow testing has limited benefits, and A/B Testing (see Figure 3-1, right) is more common.

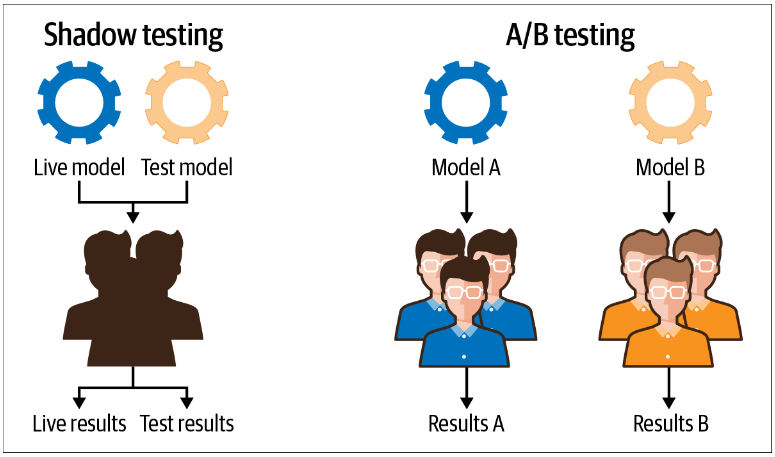


Figure 3-1. The difference between shadow testing and A/B testing

In A/B testing, both models are deployed into the live environment, but input requests are split between the two models. Any request is processed by one or the other model, not both. Results from the two models are logged for analysis (but never for the same request). Note that drawing statistically meaningful conclusions from an A/B test requires careful planning on the part of the data scientist. In particular, the A/B test cannot be stopped early, but must reach its predetermined end point, potentially making it slow and inflexible.

Multi-armed bandit tests are an increasingly popular alternative to the fixed-duration A/B test, with the aim of drawing conclusions more quickly. Multi-armed bandit testing is adaptive—the algorithm that decides the split between models adapts according to live results and reduces the workload of underperforming models. While multi-armed bandit testing is more complex, it can reduce the business cost of sending traffic to a poorly performing model.

Closing Thoughts

Model monitoring and iteration is what many people naturally think of when they hear MLOps. And while it's just one small part of a much larger process, it is undoubtedly important. Many on the business side see AI projects as something that can be built, implemented, and will then just *work*. However, as seen in this section, this often isn't the case.

Unlike static software code, ML models—because of constantly changing data—need to be carefully monitored and possibly tweaked in order to achieve the expected business results.

CHAPTER 4

Governance

In many ways, governance is the backbone of MLOps. It is the set of controls placed on a business to ensure that it delivers on its responsibilities to all stakeholders, from shareholders and employees to the public and national governments. These responsibilities include financial, legal, and ethical obligations. Underpinning all three of these responsibilities is the fundamental principle of fairness. All of these components are critical parts of a robust MLOps system.

This chapter will explore the many components of a modern AI governance strategy and how it's inherently intertwined with MLOps efforts. It will close out with a template for governance in the context of MLOps, which may be particularly useful for business leaders looking to spearhead governance strategies in their own organizations.

Why Governance Matters to the Business

What most businesses want from governance is to safeguard shareholder investment and to help ensure a suitable return on investment (ROI), both now and in the future. That means the business has to perform effectively, profitably, and sustainably. The shareholders need clear visibility that customers, employees, and regulatory bodies are happy, and they want reassurances that appropriate measures are in place to detect and manage any difficulties that could occur in the future.

If businesses and governments want to reap the benefits of ML, then they have to safeguard the public trust in it as well as proactively address the risks. For businesses, this means developing strong governance of their MLOps process. They must assess the risks and determine their own set of fairness values, and they must implement the necessary process to manage these. Much of this is simply about good housekeeping with an added focus on mitigating the inherent risks of ML, addressing topics such as data provenance, transparency, bias, performance management, and reproducibility.

But governance isn't a free lunch; it takes effort, discipline, and time.

From the perspective of the business stakeholders, governance is likely to slow down the delivery of new models, which may cost the business money. But it's also important for the business side to recognize what governance looks like to a data scientist, which is a lot of bureaucracy that erodes their ability to get things done.

Types of Governance

Applying good governance to MLOps is challenging. The processes are complex, the technology is opaque, and the dependence on data is fundamental. Governance initiatives in MLOps broadly fall into one of two categories:

Data governance
: A framework for ensuring appropriate use and management of data.

Process governance
: The use of well-defined processes to ensure that all governance considerations have been addressed at the correct point in the life cycle of the model, and that a full and accurate record has been kept.

Data Governance

Data governance, which concerns itself with the data being used—especially for model training—addresses questions like:

- What is the data's provenance?
- How was the original data collected and under what terms of use?

- Is the data accurate and up to date?
- Is there Personally Identifiable Information (PII) or other forms of sensitive data that should not be used?

AI projects usually involve significant pipelines of data cleaning, combination, and transformation. Understanding the data lineage is complex, and anonymizing or pseudo-anonymizing data is not always a sufficient solution to managing personal information. If not performed correctly, it can still be possible to single out an individual and their data.

In addition, inappropriate biases in models can arise quite accidentally despite the best intentions. The point is that making predictions based on past experience is a powerful technique, but sometimes the consequences are not only counterproductive, they are illegal.

Process Governance

The second type of governance is process governance, which focuses on formalizing the steps in the MLOps process and associating actions with those.

Today, process governance is most commonly found in organizations with a traditionally heavy burden of regulation and compliance, such as finance. Outside of these organizations, it is rare. With ML creeping into all spheres of commercial activity, and with rising concern about Responsible AI, we will need new and innovative solutions that can work for all businesses.

Those responsible for MLOps must manage the inherent tension between different user profiles, striking a balance between getting the job done efficiently, and protecting against all possible threats. This balance can be found by assessing the specific risk of each project and matching the governance process to that risk level. There are several dimensions to consider when assessing risk, including:

- The audience for the model
- The lifetime of the model and its outcomes
- The impact of the outcome

This assessment should determine not only the governance measures applied, but it should also drive the complete MLOps development and deployment toolchain.

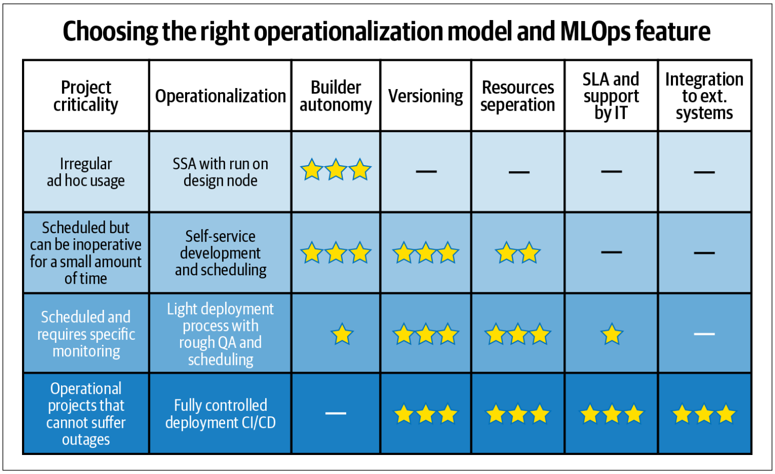
The Right Level of Governance for the Job

A Self Service Analytics (SSA) project, consumed by a small internal-only audience, calls for relatively lightweight governance. Conversely, a model deployed to a public-facing website making decisions that impact people's lives or company finances requires a very thorough process.

This process would consider the type of KPIs chosen by the business, the type of model-building algorithm used for the required level of explainability, the coding tools used, the level of documentation and reproducibility, the level of automated testing, the resilience of the hardware platform, and the type of monitoring implemented.

But the business risk is not always so clear-cut. An SSA project that makes a decision that has a long-term impact can also be high risk and can justify stronger governance measures. That's why across the board, teams need well-thought-out, regularly reviewed strategies for MLOps risk assessment (see Figure 4-1 for a breakdown of project criticality and operationalization approaches).

Ultimately, it's important to understand from the business side that in many ways, governance is not an overarching set of restrictions; rather, it's a balance that depends on the use case at hand. It's up to business and tech experts to work together to determine the proper governance standards for projects under an MLOps framework.



Choosing the right operationalization model and MLOps feature

Project criticality	Operationalization	Builder autonomy	Versioning	Resources seperation	SLA and support by IT	Integration to ext. systems
Irregular ad hoc usage	SSA with run on design node	★★★	—	—	—	—
Scheduled but can be inoperative for a small amount of time	Self-service development and scheduling	★★★	★★★	★★	—	—
Scheduled and requires specific monitoring	Light deployment process with rough QA and scheduling	★	★★★	★★★	★	—
Operational projects that cannot suffer outages	Fully controlled deployment CI/CD	—	★★★	★★★	★★★	★★★

Figure 4-1. Choosing the right kind of operationalization model and MLOps features depending on the project's criticality.

A Template for MLOps Governance

There is no one-size-fits-all solution across businesses, and different use cases within a business justify different levels of management, but the step-by-step approach outlined can be applied in any organization to guide the implementation process.

The process has seven steps:

1. Understand and classify the analytics use cases.
2. Establish responsibilities.
3. Determine governance policies.
4. Integrate policies into MLOps process.
5. Select the tools for centralized governance management.
6. Engage and educate.
7. Monitor and refine.

This section will go through each of the steps in detail, including a simple definition and the "how" of actually implementing the step.

Step 1: Understand and Classify the Analytics Use Cases

This step defines what the different classes of analytics use cases are and, subsequently, what the governance needs are for each.

Consider the answers to the following questions for a representative cross section of analytics use cases. Identify the key distinguishing features of the different use cases and categorize these features. Conflate categories where appropriate. Typically, it will be necessary to associate several categories to each use case to fully describe it:

- What regulations are each use case subject to, and what are the implications? Sector-specific regulations, regional, PII?
- Who consumes the results of the model? The public? One of many internal users?
- What are the availability requirements for the deployed model? 24-7 real-time scoring, scheduled batch scoring, ad hoc runs (self-service analytics)?
- What is the impact of any errors and deficiencies? Legal, financial, personal, public trust?
- What is the cadence and urgency of releases?
- What is the lifetime of the model and the lifetime of the impact of its decision?
- What is the likely rate of model quality decay?
- What is the need for explainability and transparency?

Step 2: Who Is Responsible?

Identify the groups of people responsible for overseeing MLOps governance as well as their roles:

- Engage the whole organization, across departments, from top to bottom of the management hierarchy.
- Peter Drucker's famous line "Culture eats strategy for breakfast" highlights the power of broad engagement and shared beliefs.
- Avoid creating all new governance structures—look at what structures exist already and try to incorporate MLOps governance into them.
- Get senior management sponsorship for the governance process.

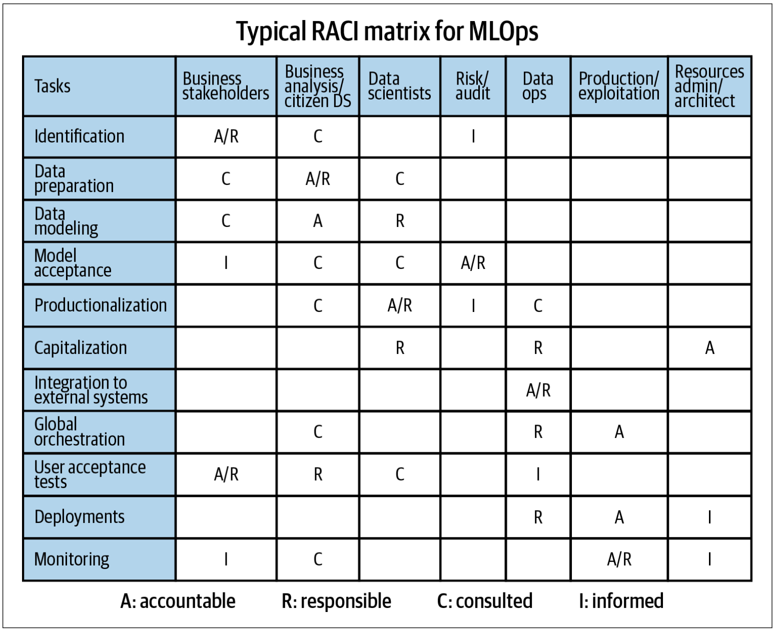
- Think in terms of separate levels of responsibility:

  Strategic
  : Set out the vision

  Tactical
  : Implement and enforce the vision

  Operational
  : Execute on a daily basis

- Consider building a RACI matrix for the complete MLOps process (see Figure 4-2). RACI stands for *Responsible, Accountable, Consulted, Informed,* and it highlights the roles of different stakeholders in the overall MLOps process. It is quite likely that any matrix you create at this stage will need to be refined later on in the process.



Typical RACI matrix for MLOps

Tasks	Business stakeholders	Business analysis/ citizen DS	Data scientists	Risk/ audit	Data ops	Production/ exploitation	Resources admin/ architect
Identification	A/R	C		I			
Data preparation	C	A/R	C				
Data modeling	C	A	R				
Model acceptance	I	C	C	A/R			
Productionalization		C	A/R	I	C		
Capitalization			R		R		A
Integration to external systems					A/R		
Global orchestration		C			R	A	
User acceptance tests	A/R	R	C		I		
Deployments					R	A	I
Monitoring	I	C				A/R	I

A: accountable R: responsible C: consulted I: informed

Figure 4-2. A typical RACI matrix for MLOps.

Step 3: Determine the Governance Policies

With an understanding of the scope and objectives for governance now established and the engagement of the responsible governance leaders, it is time to consider the core policies for the MLOps process. This is no small task, and it is unlikely to be achieved in one iteration. Focus on establishing the broad areas of policy and accept that experience will help to evolve the details.

Consider the classification of initiatives from Step 1. What governance measures does the team or organization need in each case?

In initiatives where there is less concern about the risk or regulatory compliance, lighter-weight, cheaper measures may be appropriate. For example, "what if" calculations to determine the number of in-flight meals of different types has relatively little impact—after all, the mix was never right even before the introduction of ML.

Even such a seemingly insignificant use case may have ethical implications as meals are likely to be correlated to religion or gender, which are protected attributes in many countries. On the other hand, the implications of calculations to determine the level of fueling of planes carry substantially greater risk.

Governance considerations can be broadly grouped under the headings in Table 4-1. For each heading, there is a range of measures to consider for each class.

Table 4-1. Governance considerations. Example measures that businesses can take to ensure that they address important governance considerations.

Governance consideration	Example measures
Reproducibility and traceability	Full data snapshot for precise and rapid model reinstantiation *or* ability to recreate the environment and retrain with a data sample *or* only record metrics of models deployed
Audit and documentation	Full log of all changes during development including experiments run and reasons for choices made *or* automated documentation of deployed model only *or* no documentation at all
Human-in-the-loop sign-off	Multiple sign-offs for every environment move (dev, QA, pre-Prod, Prod)
Pre-production verification	Verify model documentation by hand coding the model and comparing results *or* full automated test pipeline recreating in production-like environment with extensive unit and end-to-end test cases *or* automated checks on database, software version, and naming standards only
Transparency and explainability	Use manually coded decision tree for maximum explainability *or* use regression algorithms explainability tools such as Shapley values *or* accept opaque algorithms such as neural networks

Governance consideration	Example measures
Bias and harm testing	"Red Team" adversarial manual testing using multiple tools and attack vectors *or* automated bias checking on specific subpopulations
Production deployment modes	Containerized deployment to elastic scalable HA multinode configuration with automated stress/load testing prior to deployment *or* a single production server
Production monitoring	Real-time alerting of errors, dynamic multi-arm bandit model balancing, automated nightly retraining, model evaluation, and redeployment *or* weekly input drift monitoring and manual retraining *or* basic infrastructure alerts, no monitoring, no feedback-based retraining
Data quality and compliance	PII considerations including anonymization Documented and reviewed column-level lineage to understand the source, quality, and appropriateness of the data Automated data quality checks for anomalies

The finalized governance policies should provide:

1. A process for determining the classification of any analytics initiative. This could be implemented as a checklist or a risk assessment application.
2. A matrix of initiative classification against governance consideration, where each cell identifies the measures required.

Step 4: Integrate Policies into the MLOps Process

Having identified the governance policies for the different classes of initiatives, the measures to implement these need to be incorporated into the MLOps process and the responsibilities for actioning the measures assigned.

While most businesses will have an existing MLOps process, it is quite likely that this has not been defined explicitly but rather has evolved in response to individual needs. Now is the time to revisit, enhance, and document the process. Successful adoption of the governance process can only happen if it is communicated clearly and buy-in is sought from each stakeholder group.

Understand all of the steps in the existing process by interviewing those responsible. Where there is no previous formal process, this is often harder than it sounds—the process steps are often not explicitly defined, and ownership is unclear.

Attempting to map the policy-driven governance measures into the understanding of the process will identify issues in the process very quickly. Within one business there may be a range of different styles of project and governance needs, such as:

- One-off self-service analytics
- Internally consumed models
- Models embedded in public websites
- Models deployed to IoT devices

In these cases, the differences between some processes may be so great it is best to think in terms of several parallel processes. Ultimately, every governance measure for each use case should be associated with a process step and with a team that is ultimately responsible (see Table 4-2).

Table 4-2. Governance steps throughout the AI life cycle process. Example activities and governance considerations for each step in the raw data to ML model process.

Process step	Example activities and governance considerations
Business scoping	Record objectives, define KPIs, and record sign-off: for internal governance considerations
Ideation	Data discovery: data quality and regulatory compliance constraints Algorithm choice: impacted by explainability requirements
Development	Data preparation: consider PII compliance, separation of legal regional scopes, avoid input bias Model development: consider model reproducibility and audibility Model testing and verification: bias and harm testing, explainability, sign
Preproduction	Verify performance/bias with production data Production-ready testing: verify scalability
Deployment	Deployment strategy: driven by the level of operationalization Deployment verification tests Use of shadow challenger or A/B test techniques for in-production verification
Monitoring and feedback	Performance metrics and alerting Prediction log analysis for input drift with alerting

Step 5: Tools for Centralized Governance Management

The MLOps governance process impacts both the complete ML life cycle as well as many teams across the organization. Each step requires a specific sequence of actions and checks to be executed.

Traceability of both the development of the model and the execution of governance activities is a complex challenge.

Most organizations still have a “paper form” mindset for process management, where forms are filled in, circulated, signed, and filed. The forms may be text documents, circulated via email, and filed electronically, but the limitations of paper remain. It is hard to track progress, review many projects at once, prompt for action, and remind teams of responsibilities. The complete record of events is typically spread across multiple systems and owned by individual teams, making a simple overview of analytics projects effectively impossible.

While teams will always have tools specific to their roles, MLOps governance is much more effective if the overarching process is managed and tracked from one system. This system should:

- Centralize the definition of the governance process flows for each class of analytics use cases
- Enable tracking and enforcement of the complete governance process
- Provide a single point of reference for the discovery of analytics projects
- Enable collaboration between teams, in particular, the transfer of work between teams
- Integrate with existing tools used for project execution

The workflow, project management, and MLOps tools currently in use can only partially support these objectives. A new category of ML governance tools is emerging to support this need directly and more fully. These tools focus on the specific challenges of ML governance, including:

- A single view on the status of all models (otherwise known as a Model Registry).
- Process gates with a sign-off mechanism to allow ready traceability of the history of decision making.
- Ability to track all versions of a model.
- Ability to link to artifact stores, metrics snapshots, and documentation.

- Ability to tailor processes specifically for each class of analytics use cases.
- Ability to integrate health monitoring from production systems and to track the performance of models against the original business KPIs.

Step 6: Engage and Educate

Without a program of engagement and training for the groups involved in overseeing and executing the governance process, the chances of it being even partially adopted are slim. It is essential that the importance of MLOps governance to the business, and the necessity of each team's contribution, is communicated. Building on this understanding, every individual needs to learn what they must do, when, and how. This exercise will require considerable documentation, training—and most of all—time.

Start by communicating the broad vision for MLOps governance in the business. Highlight the dangers of the status quo, an outline of the process, and how it is tailored to the range of use cases.

Engage directly with each team involved and build a training program with them. Do not be afraid to leverage their experience to shape not only the training, but also the detailed implementation of their governance responsibilities. The result will be much stronger buy-in and more effective governance.

Step 7: Monitor and Refine

Is the newly implemented governance working? Are the prescribed steps being implemented, and are the objectives being met? What actions should be taken if things are going poorly? How do we measure the gap between today's reality and where the business needs to be?

Measuring success requires metrics and checks. It requires people to be tasked with monitoring and a way to address problems. The governance process and the way it is implemented will need to be refined over time, based both on lessons learned and evolving requirements (including, as discussed earlier in this chapter, evolving regulatory requirements).

A big factor in the success of the process will be the diligence of the individuals responsible for the individual measures in the process, and incentivizing them is key.

Monitoring the governance process starts with a clear understanding of the key performance metrics and targets—KPIs for governance. These should aim to measure both whether the process is being enacted and if the objectives are being achieved. Monitoring and auditing can be time consuming, so look to automate metrics as far as possible and encourage individual teams to own the monitoring of metrics that relate to their area of responsibility.

It is hard to make people carry out tasks that seemingly deliver nothing concrete to those doing the work. One popular tactic to address this is gamification. This is not about making everything look like a video game, but about introducing incentives for people to carry out tasks where the main benefit is derived by others.

Look to gamify the governance process in simple ways—publishing KPI results widely is the simplest place to start. Just being able to see targets being met is a source of satisfaction and motivation. Leaderboards, whether at the team or individual level, can add some constructive element of competition. For example, people whose work consistently passes compliance checks the first time, or meets deadlines for tasks, should be able to feel their efforts are visible.

For example, GE Aviation developed a low-cost program to have individuals contribute to data quality by rolling out a point system such that each time someone tagged a dataset, created new documentation, created a new dataset, etc., that person would receive a certain number of points. More points unlocked the possibility to pass levels and get exclusive laptop stickers, and they took the competition to the next level by adding a leaderboard so people could see the accumulated points of others. The interest and involvement due to this gamification was undoubtedly a huge piece of the organization's overall success in removing data silos and becoming a data-driven company.[1]

[1] See "GE Aviation: From Data Silos to Self-Service Analytics" for the story of why (and how) the company upended its approach to analytics.

Excessive competition can be disruptive and demotivating. A balance must be struck, and this is best achieved by building up gamification elements slowly over time. Start with the least competition oriented and add new elements one by one, measuring their effectiveness before adding the next.

Monitoring changes in the governance landscape is essential. This might be regulatory, or it might be about public opinion. Those with responsibility for the strategic vision must continue to monitor this as well as have a process to evaluate potential changes.

Finally, monitoring of the process is only worthwhile if issues are acted upon. Establish a process for agreeing on change and for enacting it. Iteration is inevitable and necessary, but the balance between efficiency and effectiveness is hard to find, and many lessons can only be learned the hard way. Build a culture where people see iteration and refinement as a measure of a successful process, not a failed one.

Closing Thoughts

It is hard to separate MLOps from its governance. It is not possible to successfully manage the model life cycle, mitigate the risks, and deliver value at scale without governance. Governance impacts everything from how the business can acceptably exploit ML, the data and algorithms that can be used, to the style of operationalization, monitoring, and retraining.

MLOps at scale is in its infancy. Few businesses are doing it, and even fewer are doing it well—meaning it's an opportunity for businesses to set themselves apart and get ahead in the race to AI. When planning to scale MLOps, start with governance and use it to drive the process. Don't bolt it on at the end. Think through the policies; think about using tooling to give a centralized view; engage across the organization. It will take time and iteration, but ultimately the business will be able to look back and be proud that it took its responsibilities seriously.

CHAPTER 5

Get Started with MLOps

The previous chapters have only scratched the surface on the details and nuance behind an effective MLOps system, but they do provide a good introduction to understanding why it matters and how it can affect the success of a business with data science, ML, and AI initiatives.

However, MLOps is not possible if people aren't aligned, processes aren't well defined, and the right technology isn't in place to facilitate and underpin efforts. This chapter will dive into each of these areas, offering some practical lessons for getting started with MLOps in your organization.

People

As touched on in Chapter 1, the AI project life cycle must involve different types of profiles with a wide range of skills in order to be successful, and each of those people has a role to play in MLOps. But the involvement of various stakeholders isn't about passing the project from team to team at each step—collaboration between people is critical.

For example, subject matter experts usually come to the table—or at least, they *should* come to the table—with clearly defined goals, business questions, and/or key performance indicators (KPIs) that they want to achieve or address. In some cases, they might be extremely well defined (e.g., "In order to hit our numbers for the quarter, we need to reduce customer churn by 10%," or "We're losing *n* dollars per quarter due to unscheduled maintenance, how can we better predict downtime?”). In other cases, less so (e.g., “Our service staff needs to better understand our customers to upsell them” or “How can we get people to buy more widgets?”).

In organizations with healthy processes, starting the ML model life cycle with a more-defined business question isn’t necessarily always an imperative, or even an ideal scenario. Working with a less-defined business goal can be a good opportunity for subject matter experts to work directly with data scientists up front to better frame the problem and brainstorm possible solutions before even beginning any data exploration or model experimentation.

Subject matter experts have a role to play not only at the beginning of the AI project life cycle, but the end (postproduction) as well. Oftentimes, to understand if an ML model is performing well or as expected, data scientists need subject matter experts to close the feedback loop—traditional metrics (accuracy, precision, recall, etc.) are not enough.

For example, data scientists could build a simple churn prediction model that has very high accuracy in a production environment; however, marketing does not manage to prevent anyone from churning. From a business perspective, that means the model didn’t work, and that’s important information that needs to make its way back to those building the ML model so that they can find another possible solution—e.g., introducing uplift modeling that helps marketing better target potential churners who might be receptive to marketing messaging.

To get started on a strong foundation with MLOps, it might be worth looking at the steps AI projects must take at your organization and who needs to be involved. This can be a good starting point to making sure the right stakeholders not only have a seat at the table, but that they can effectively work with each other to develop, monitor, and govern models that will not put the business at risk. For example, are these people even using the same tools and speaking the same language? (More on this in “Technology” on page 48.)

Processes

MLOps is essentially an underlying system of processes—essential tasks for not only efficiently scaling data science and ML 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, 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 ML 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 behind those models. MLOps, as described in this report, can provide this level of transparency and accountability.

That being said, getting started involves formalizing the steps in the MLOps process and associating actions with those. Typically, these actions are reviews, sign-offs, and the capture of supporting materials such as documentation. The aim is twofold:

1. To ensure every governance-related consideration is made at the correct time, and correctly acted upon. For example, models should not be deployed to production until all validation checks have been passed.
2. To enable oversight from outside of the strict MLOps process. Auditors, risk managers, compliance officers, and the business as a whole all have an interest in being able to track progress and review decisions at a later stage.

Effectively defining MLOps processes is challenging, however, because:

- Formal processes for the ML life cycle are rarely easy to define accurately. The understanding of the complete process is usually spread across the many teams involved, often with no one person having a detailed understanding of it as a whole.

- For the process to be applied successfully, every team must be willing to adopt it wholeheartedly.
- If the process is just too heavyweight for some use cases, teams will certainly subvert it, and much of the benefit will be lost.

Technology

Unfortunately (but unsurprisingly), there is no magic-bullet solution: one MLOps tool that can make all processes work perfectly. That being said, technology can help ensure that people work together (the importance of which was described in "People " on page 45) as well as guide processes.

Fortunately, more and more data science and ML platforms allow for one system that checks all of these boxes and more, including making other parts of the AI project life cycle easier, like automating workflows and preserving processing operations for repeatability. Some also allow for the use of version control and experimental branch spin-off to test out theories, then merge, discard, or keep them, as well.

The bottom line when it comes to getting started and technology is that it's important not to further fragment the AI project life cycle with a slew of different tools that further complicate the process, requiring additional work to cobble together different technologies. MLOps is one unified process, so tooling should unite all different people and parts of processes into one place.

Closing Thoughts

In order for AI to become truly scalable and enact holistic organizational change, enterprises must achieve alignment across people, processes, and technology, as described specifically in this section, but also throughout the entire report. This task is far from a turnkey undertaking.

While this alignment is critical, building robust MLOps practices doesn't happen overnight, and it requires a significant time investment from everyone within an organization. Change management is an often overlooked, but critical—and admittedly challenging—part of pivoting an organization's strategy around data. That is one area of AI transformation, and of MLOps, where the business can bring particular value and strengths that technical teams might not be able to lead on their own. This fact further underscores the need for business and technology experts to work together toward common goals, of which MLOps is just the beginning.

About the Authors

Mark Treveil has designed products in fields as diverse as telecommunications, banking, and online trading. His own startup led a revolution in governance in the UK local government, where it still dominates. He is now part of the Dataiku Product Team based in Paris.

Lynn Heidmann received her BA in journalism/mass communications and anthropology from the University of Wisconsin–Madison in 2008 and decided to bring her passion for research and writing into the world of tech. She spent seven years in the San Francisco Bay Area writing and running operations with Google and subsequently Niantic before moving to Paris to head content initiatives at Dataiku. In her current role, Lynn follows and writes about technological trends and developments in the world of data and AI.