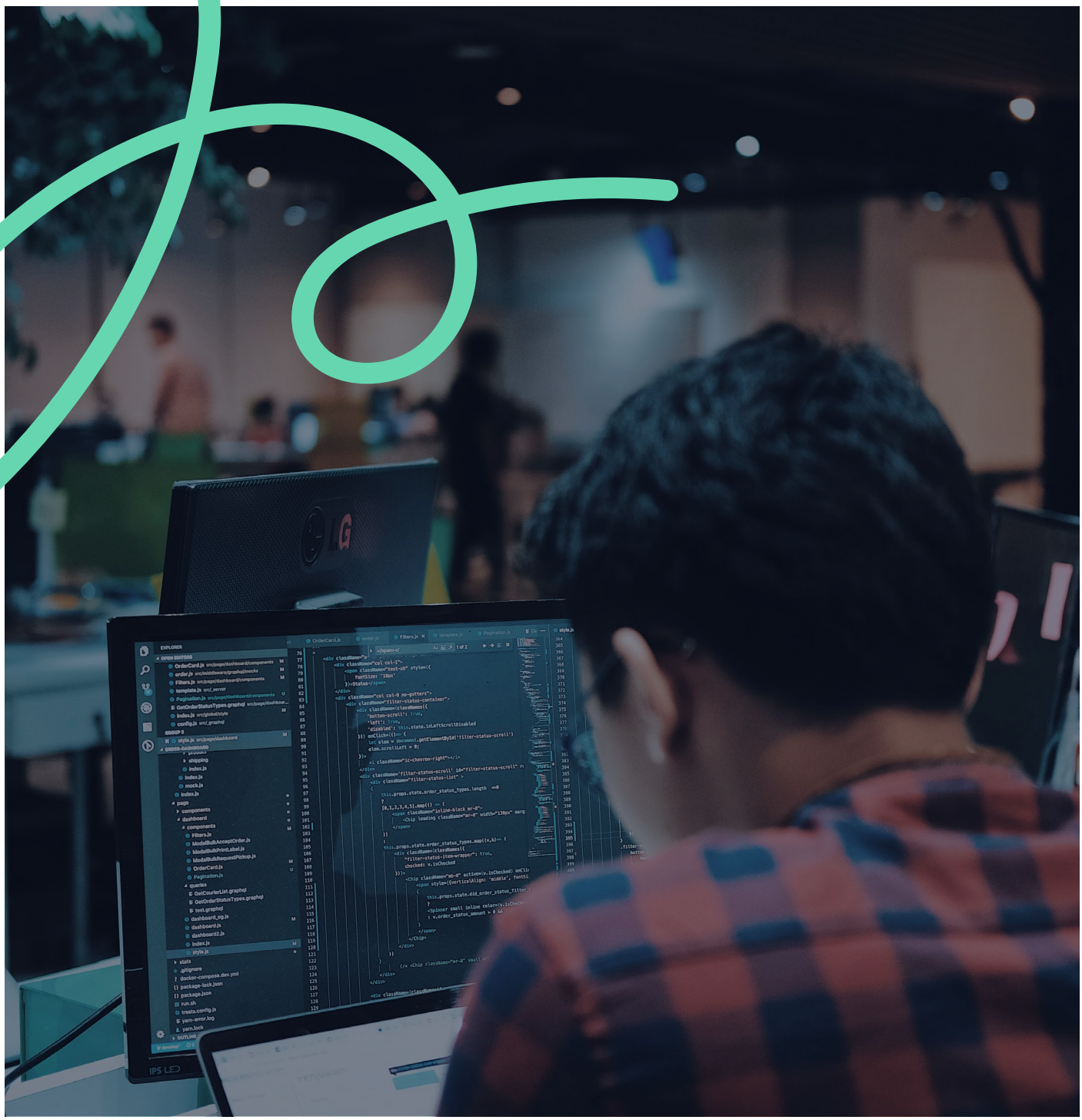


Evaluating MLOps & its aspects

nagarro

nagarro

Table of Contents

What is MLOps and why is it necessary for Machine Learning projects?

Machine Learning Operations (MLOps) is the mechanism of delivering Machine Learning (ML) projects through repeatable and efficient workflows.

MLOps considers the unique ML requirements to define a new Life Cycle that exists alongside the existing SDLC and CI/CD processes, generating a better workflow, with more effective results for ML.

According to Gartner, "While many organizations have experimented with AI proofs of concept (POCs), there are still major blockers to operationalizing its development. IT leaders must strive to move beyond the POC to ensure that more projects get to production and that they do so at scale to deliver business value."[2]

Many AI/ML projects fail due to a lack of a framework and architecture for model development, deployment, and monitoring. MLOps is the name given to such a framework. MLOps is a new method for data scientists and IT experts to collaborate and communicate in order to automate and productize machine-learning algorithms.

While most data science organisations have specified a method for developing, training, and testing machine learning models, they lack a common approach on how to continue from there once the model has been completed. As soon as ML models are put into production, integration, deployment, and monitoring become critical components for giving continuous feedback.

This is where the entire process of developing machine learning models resembles that of a classic software development project rather than a data analytics one. Many businesses believe that data science projects are restricted to simply the creation of models. But we must understand that once a model is developed and deployed, many other aspects are also needed to operationalize it:

- Management and monitoring to ensure the model performs best within the business's stated criteria
- A feedback loop to track model drift or degradation
- Model tweaking and retraining on a regular basis As a result, MLOps is required as a method for operationalizing the machine learning model building process. This generates a continuous delivery cycle of models that serve as the foundation for ML-based systems.

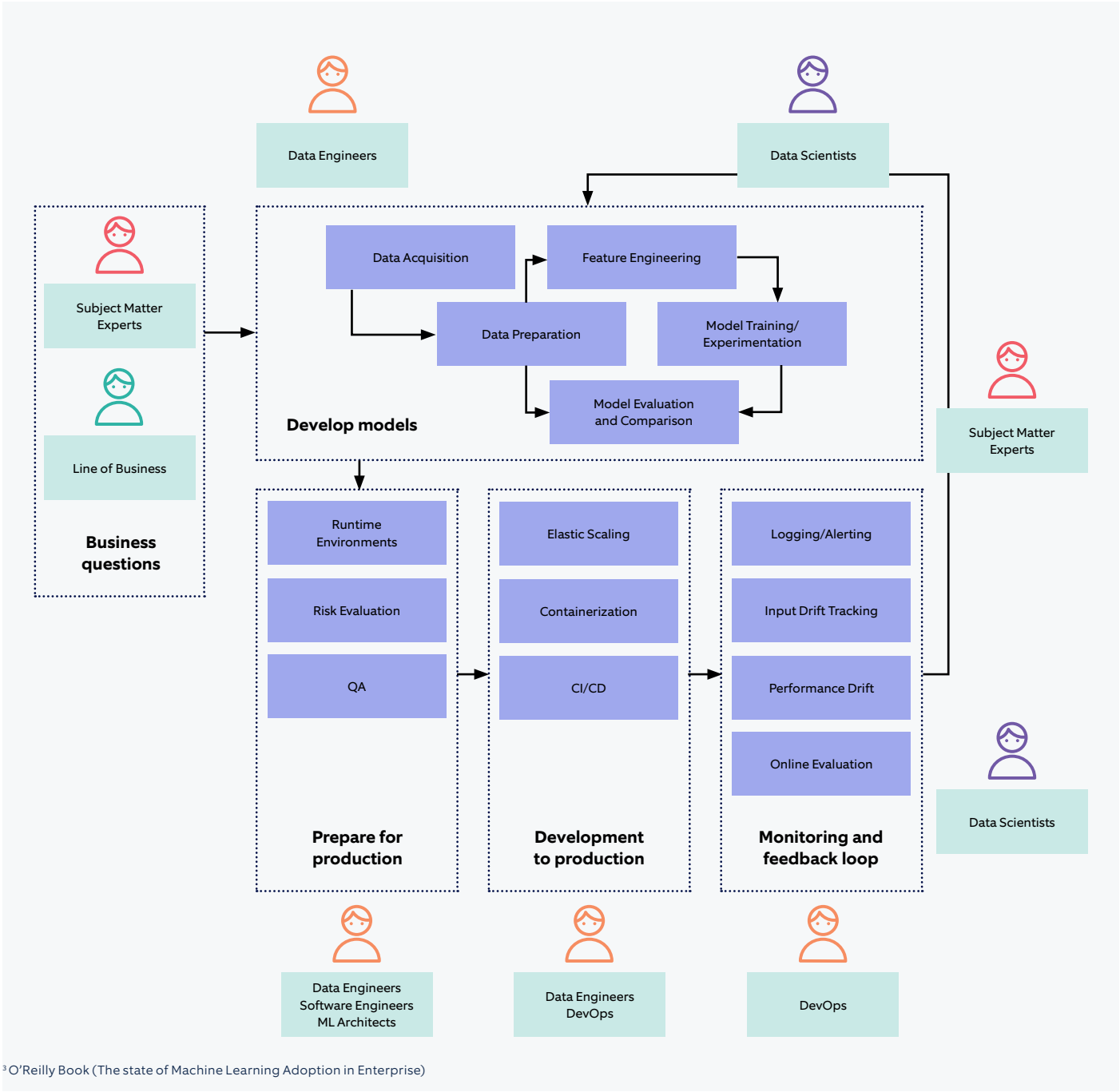
[2] Gartner Inc.

Role in ML model life cycle requirements

Developing successful machine learning projects requires multiple levels of expertise, making it an extremely collaborative job. In most projects, the operational aspects are enormous as compared to the actual model building. Thus, it often requires several roles beyond that of just being a data scientist.

Depending on the size of the organization and the nature of the project, data science teams can have one or many of these roles. In small and medium-sized teams, their responsibilities can often seem blurry or ambiguous. Yet, most large firms are able to manage distinct roles and responsibilities to a reasonable extent.

While ML has been around for a while, there's still some confusion about job titles and their associated responsibilities. Let's look at the different roles and their associated responsibilities in a typical ML team, and how they collaborate.



[3] O'Reilly Book (The state of Machine Learning Adoption in Enterprise)

Subject Matter Experts

- Provide business questions, goals, or KPIs around which ML models should be framed.
- Responsible for a mechanism or feedback loop to validate model results that don't align with business expectations - which can be a very tedious activity.
- Continually evaluate and ensure that model performance aligns with or resolves the initial need.
- They play a translator role between the business stakeholders and the technical team.

Data Scientists

- Build models that address the business question or needs brought by subject matter experts.
- Deliver operationalizable models so that they can be effectively use in the production environment and with production data.
- Assess model quality (of both original and tests) in tandem with subject matter experts to ensure they answer initial business questions or needs.
- Ensure automated model packaging and delivery for quick and easy (yet safe) deployment to production.
- Develop tests to determine the quality of deployed models and to make continual improvements.
- Assess the performance of all deployed models (including side-by-side for tests) from one central location.
- Investigate data pipelines of each model to make quick assessments and adjustments regardless of who originally built the model.

When developing an MLOps strategy, the needs of data scientists are the most important to consider. To be sure, they stand to benefit greatly; data scientists in most organisations today work with segregated data, processes, and technologies, making scaling their efforts challenging. MLOps is in an advantageous position to change that.

Data Engineers

- Integrate ML models in the company's applications and systems.
- Perform versioning and automatic tests.
- Optimize the retrieval and use of data to power ML models.
- Assess the performance of all deployed models.
- Look at the complete details of individual data pipelines to address underlying data plumbing issues.
- Work simultaneously on the same application.
- Ensure that ML models work seamlessly with other non-machine-learning-based applications.

DevOps

- Develop and test operational systems for security, performance, and availability.
- Perform Continuous Integration/Continuous Delivery (CI/CD) pipeline management.
- Seamless integration of MLOps into the larger DevOps strategy of the enterprise.
- Seamless deployment pipeline. Model risk managers/ auditors
- Ensure compliance with internal and external requirements before pushing ML models to production.
- Use comprehensive, potentially automated reporting tools for all models (in production now or in the future), including data lineage.
- Ensure a scalable and flexible environment for ML model pipelines, from design to development and monitoring.
- Introduce modern technologies that improve ML model performance in production.
- Create a high-level overview of models and their resources consumed.
- Drill down into data pipelines to assess and adjust infrastructure needs.

ML Architects/Software Engineers

- Machine Learning architects play a critical role in the ML model life cycle, ensuring a scalable and flexible environment for model pipelines.
- In addition, data teams need their expertise to introduce new technologies (when appropriate) that improve ML model performance in production.
- When it comes to MLOps, the ML architects' role is about having a centralized view of resource allocation. As they have a strategic, tactical role, they need an overview of the situation to identify bottlenecks and use that information to find long-term improvements.
- This is why the designation of data architect is insufficient; to fulfil this critical role in the ML model life cycle, they must have a thorough understanding of not just the business architecture but about ML as well. This role requires collaboration across the enterprise, from data scientists and engineers to DevOps and software engineers.
- Their job is to identify new technologies or infrastructure that could be worth investing in, not to provide operational quick fixes that don't address the system's scalability.

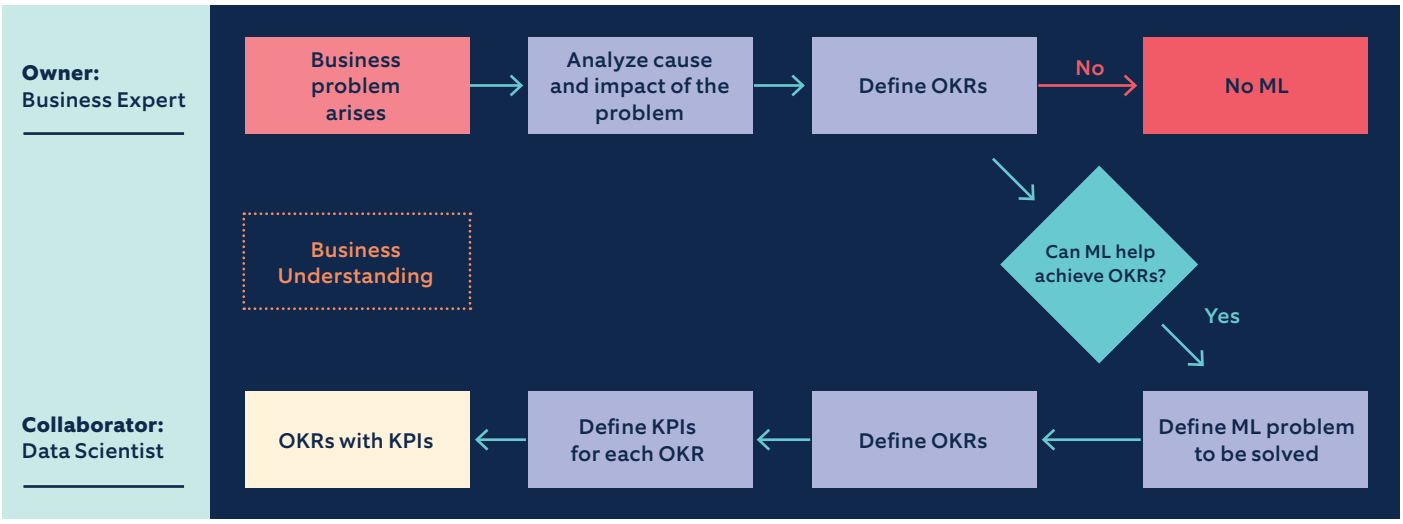
Steps involved in ML projects

There are five phases of an MLOps flow that are necessary for successful data science projects. This flow is inspired by some project frameworks for data science. Let us walk through the following 5 key steps/stages that we've seen consistently emerge across many organizations' data science Life Cycles:

- **Business Understanding**
- **Data Acquisition**
- **Model Development**
- **Model Deployment**
- **Model Monitoring**

Business Understanding

The development of a machine learning model often begins with a business goal, which can be as simple as eliminating fraudulent transactions to less than 0.1 percent or being able to recognise people's faces in social network photographs. Performance targets, technical infrastructure needs, and financial limits are all related to this goal; all these aspects can be represented as key performance indicators, or KPIs, which allow the business performance of models in production to be monitored. It's critical to remember that machine learning projects don't develop in a vacuum. They're usually part of a bigger project that has an impact on technologies, procedures, and people. As a result, setting objectives should also involve change management, which may provide some insight on how the ML model should be constructed. For example, the desired level of openness will have a significant impact on algorithm selection and may necessitate the provision of explanations along with predictions so that the predictions can be translated into useful business choices.



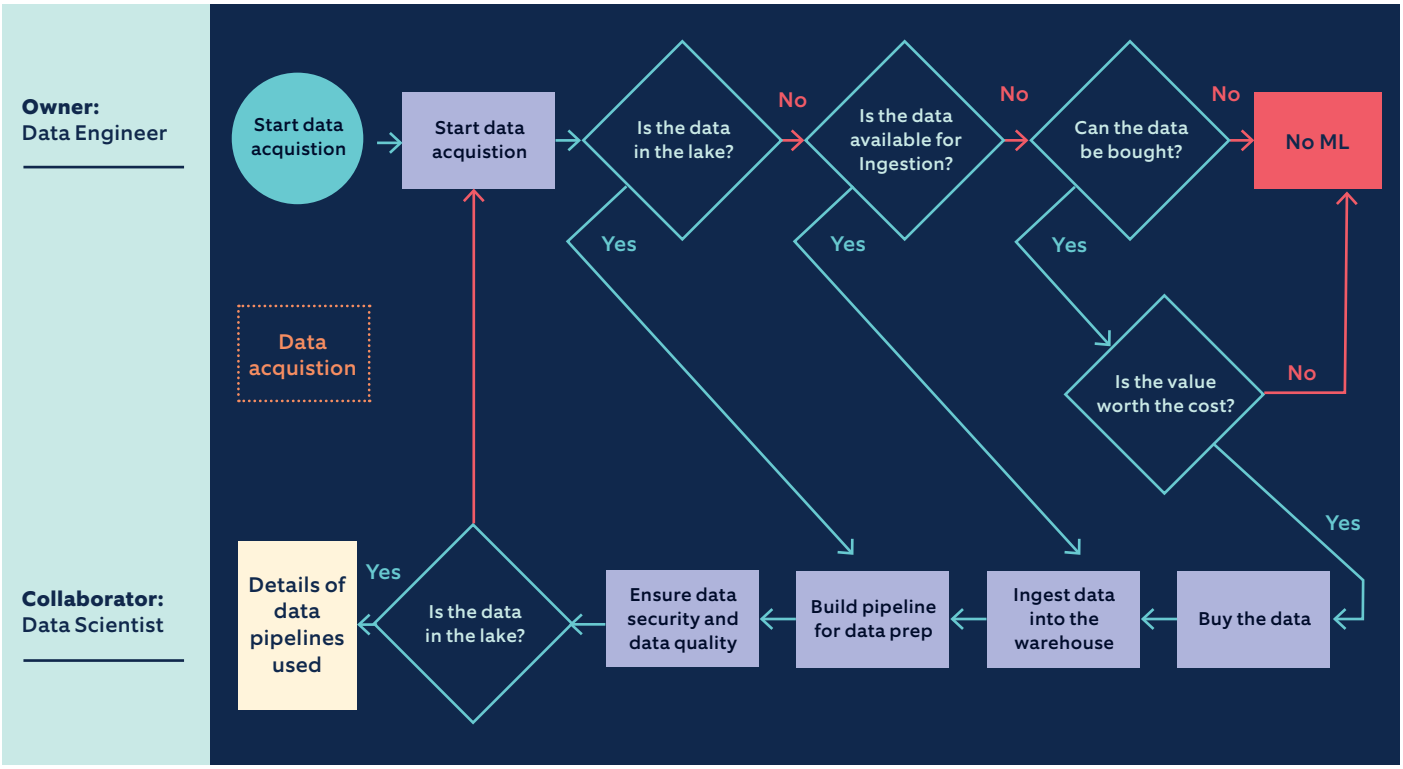
Data Acquisition

As we explore data and data sources with clear business goals in mind, it's also important to assemble a team of subject matter experts and data scientists to build the machine learning model. This is where the search for appropriate input data begins. While finding data may appear to be quite simple, it is often the most difficult aspect of the process.

Some questions or issues that must be considered for "good" data governance:

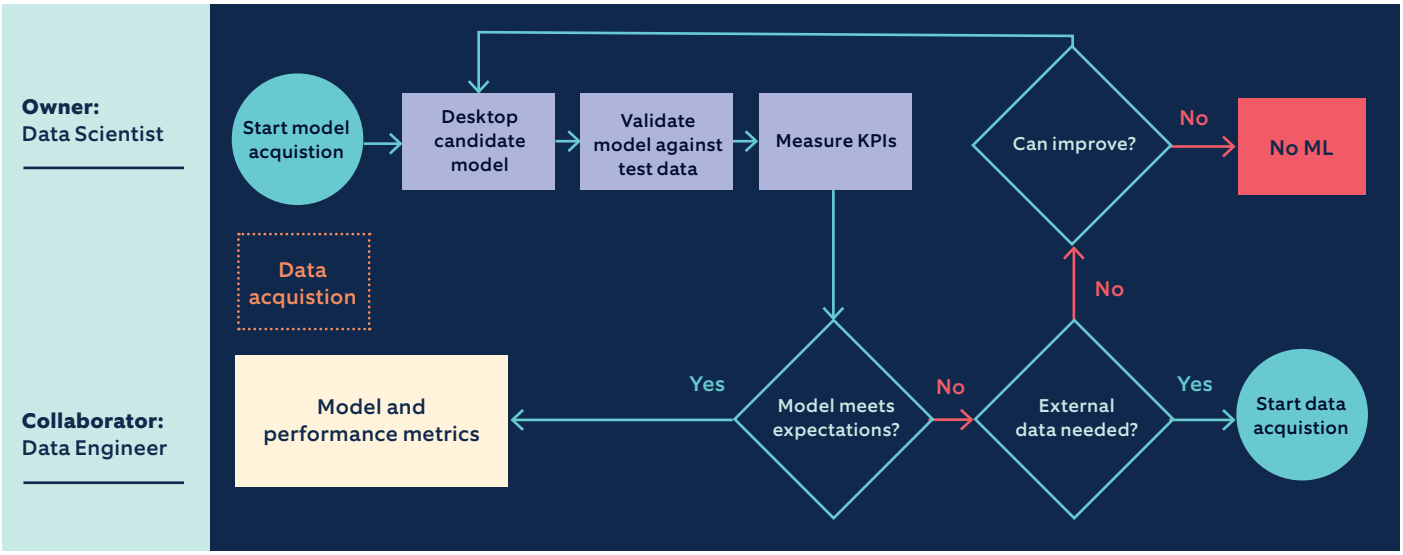
- What relevant datasets are available?
- Is this data sufficiently accurate and reliable?
- How can stakeholders get access to this data?
- Is there a need to label some of the data with the "ground truth" that is to be predicted, or does unsupervised learning make sense If so, how much will this cost in terms of time and resources?

- What data properties (known as features) can be made available by combining multiple sources of data?
- How will data be updated once the model is deployed?
- How will the KPIs, which were established along with the business objectives, be measured?
- What platform should be used?
- Can the selected datasets be used for this purpose?
- Will the use of the model itself reduce the representativeness of the data?
- Will this data be available in real time?
- Are there features, such as gender, that legally cannot be used in this business context?
- What are the terms of use?
- Are minority populations sufficiently well represented that the model has equivalent performances on each group?
- Is there personally identifiable information (PII) that must be redacted or anonymized?



Model Development

As we explore data and data sources with clear business goals in mind, it's also important to assemble a team of subject matter experts and data scientists to build the machine learning model. This is where the search for appropriate input data begins. While finding data may appear to be quite simple, it is often the most difficult aspect of the process.



Model Deployment

What about the production process between completing model development and physically deploying into production—what needs to be addressed?

Basic ML models

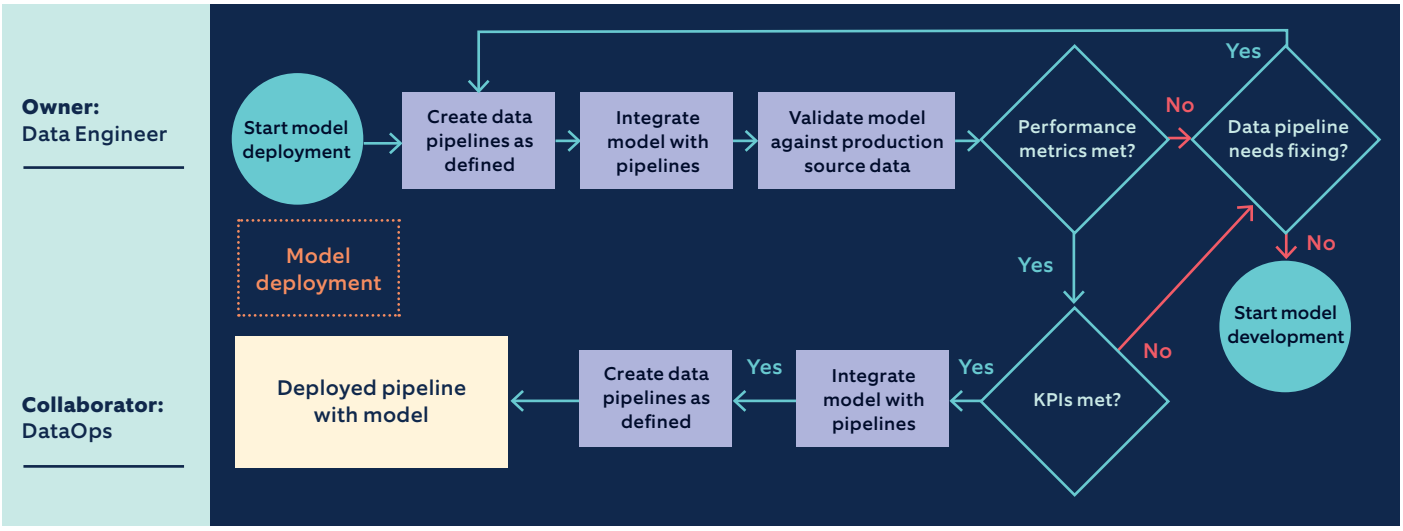
One thing is certain: When selecting the problem to be addressed, labour-intensive operations are always preferred over other tasks for speedy, automated deployment. If the model's highest resource demands can be safely capped using technologies like Linux cgroups, a completely automated single-step push-to-production may be sufficient. When using this lightweight deployment option, frameworks like Flask (micro web framework written in Python, where pre-existing third-party libraries provide common functions) can even handle simple user interfaces. Some business rule management systems, in addition to integrating data science and machine learning platforms, may also allow for autonomous deployment of basic machine learning models.

In customer-facing, mission-critical use cases, a more robust CI/CD pipeline is required.

In customer facing applications, we must ensure that all coding, documentation, and sign-off standards have been met. For additional agility, we can also automate the task of collecting the analysis of performance test results. Load Test provides key insights to continuously augment your processes.

In heavily regulated industries (e.g., finance and pharmaceuticals), governance and regulatory checks will be extensive and are likely to involve manual intervention.

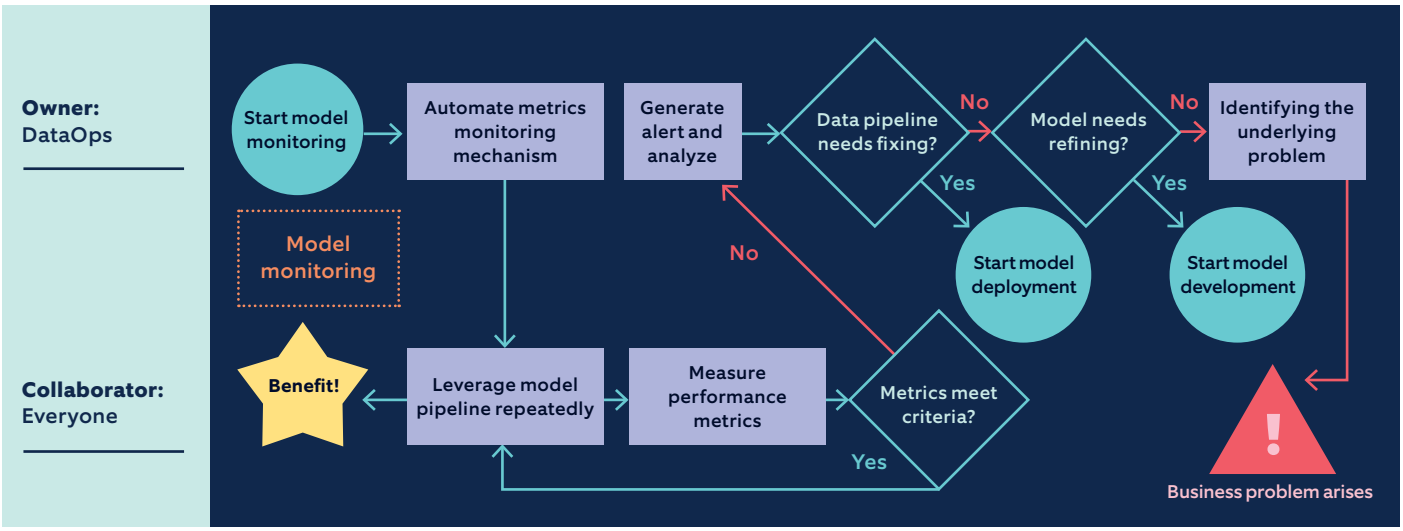
As is the case with DevOps, even in MLOps, there is this desire to automate the CI/CD pipeline as much as possible. Apart from expediting the deployment process, it also enables more extensive regression testing and reduces the likelihood of errors in the deployment.



Model Monitoring

Once the machine learning model has been deployed, it must be checked to ensure that it is performing as planned. Monitor dependency changes throughout the complete pipeline, with any changes triggering a notification.

- Send an alert if the data does not match the schema, which has been specified in the training step.
- Monitor the processes of feature generation as they have impact on the model.
- Monitor computational performance of an ML system. Both dramatic and slow-leak regression in computational performance should be notified.
- Check to see if the training and serving features provide the same result.
- Monitor the numerical stability of the ML model.
- Monitor how stale the system in production is.
- Track the ML model's predictive quality on served data as it deteriorates.



Role of DevOps in MLOps

To implement MLOps, we see three degrees of automation, starting with manual model training and deployment to pipeline automation, before progressing to fully automated ML and CI/CD pipelines.

- **Manual process**
  This is a common data science procedure that is carried out at the beginning of ML implementation. This is an experimental and iterative stage. Each pipeline step is carried out manually, including data preparation and validation, model training, and testing. Rapid Application Development (RAD) technologies, such as Jupyter Notebooks, are commonly used to process data.

- **ML pipeline automation**
  The model training is carried out automatically at the next level. This is where we introduce the model's ongoing training. The process of model retraining begins whenever fresh data becomes available. Data and model validation stages are included in this level of automation.

- **CI/CD pipeline automation**
  We introduce a CI/CD system for rapid and reliable ML model deployments in production. The main difference from the previous stage is that the Data, ML Model, and ML training pipeline components are now built, tested, and deployed automatically.

How DevOps role is different in MLOps

ML and other software systems are similar in continuous integration of source control, unit testing, integration testing, and continuous delivery of the software module or the package. However, in ML, there are a few notable differences:

- CI is no longer only about testing and validating code and components, but also about testing and validating data, data schemas, and models.

- CD is no longer about a single software package or a service, but also about a system (an ML training pipeline) that should automatically deploy another service (model prediction service).

- CT (Continuous Training) is a new property, unique to ML systems, that's concerned with automatically retraining and serving the models.

Now let's look at the typical steps for training and evaluating an ML model to serve as a prediction service.

Maturity level of MLOps projects

The DevOps maturity model determines growth through continuous learning from both teams and organizational perspectives. More the capabilities and skills, more will be the ability to handle issues of scale and complexities.

A perfect DevOps maturity model determines DevOps maturity in three ways:

- Assessing the current state of capabilities
- Identifying areas of improvement
- Outlining the steps required to achieve the desired DevOps goals[4]

Maturity level of MLOps can be categorized into 3 levels

MLOps Maturity Level
This is the same traditional machine learning Life Cycle, so this level is also known as no MLOps at all. It is considered a fundamental degree of maturity and is a manual process.

MLOps Maturity Level 1
This degree of maturity is concerned with continuous training in order to accomplish continuous supply of model prediction service, which is distinct from the entire ML application that acts on model inferences.

MLOps Maturity Level 2
While Level 1 has most of the items in place, there is still an opportunity for further improvement. Enter MLOps 2, which contains all the components of MLOps 1 plus a little something extra. MLOps 2 is MLOps' maximum level of automation and the pinnacle of any excellent machine learning platform.

[4] Based on Google Cloud

Managing ML life cycles at scale is challenging

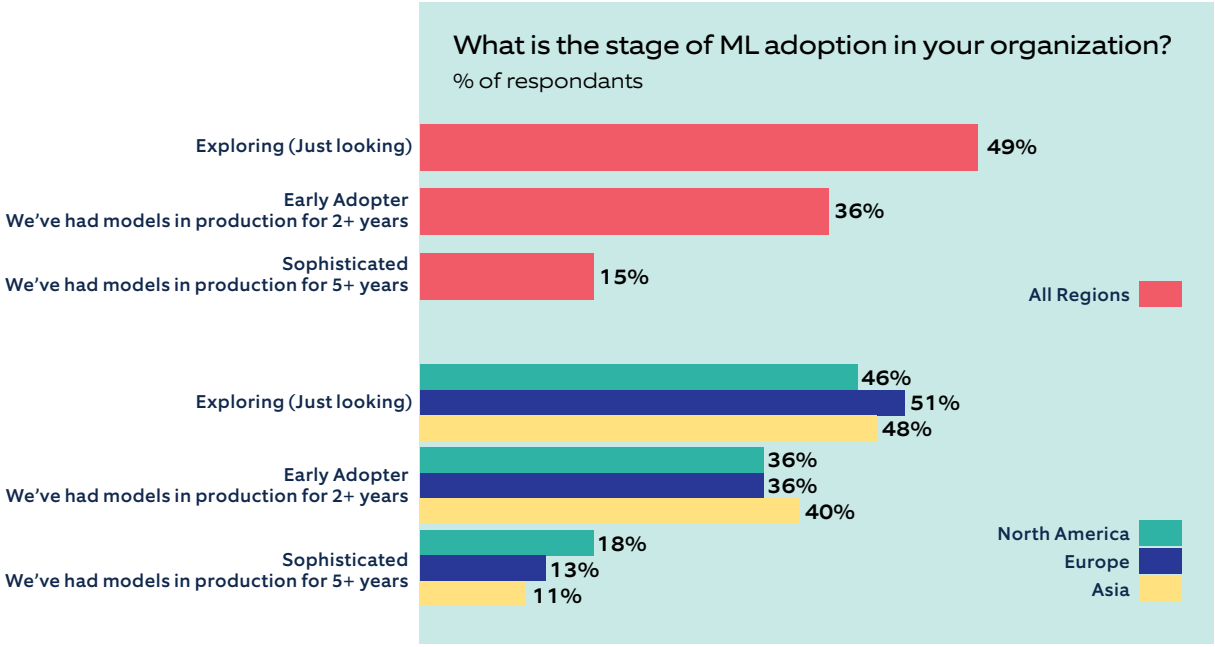
Data scientists are not software engineers
Most data scientists are specialists in model construction and evaluation but they aren't necessarily experts in application writing. Though this may change in the future as some data scientists specialise in deployment or operations, many data scientists currently find themselves juggling multiple roles, making it difficult to accomplish any of them thoroughly. When there are more models to handle, data scientists get overworked, which is obviously a recipe for trouble. And then, if you add the high turnover of data team members, the complexity increases exponentially, and data scientists are suddenly responsible for models they did not design.

Everyone does not speak the same language
Despite the fact that different personnel from the business, data science, and IT teams are involved in the machine learning life cycle, none of these groups use the same technologies or, in many cases, even share the same fundamental conceptual skills to serve as a communication baseline.

There are numerous interdependencies
Not only does data change frequently, but so do business needs. These changes must be communicated to the business on a regular basis to ensure that the model's actuality in production and on production data matches expectations and, more importantly, addresses the original problem or achieves the original aim.

Trends and analysis

While the use of Machine Learning in production started near the turn of the century, it's taken roughly 20 years for the practice to become mainstream throughout the industry. And its impact is quite prevalent: More than 11,000 data specialists responded to a recent O'Reilly survey about their organization's approach—or intended approach—to machine learning, as shown below.



[5] Oreilly Book (The State of Machine Learning Adoption in Enterprise)

The MLOps scene is a demanding environment. This is evident not just in the hundreds of new tools that are becoming available but also in the wide variety of substantial new features that the major companies are introducing. On the one hand, these new tools and features provide users with fresh options. On the other hand, they also pose a threat to individuals attempting to standardise work practises across a company. We hope to keep future rework to a minimum, but as industry practises evolve, some rework may become necessary.

- Deployment and monitoring of the system: Azure's serving features was made available in 2021, and Databricks was available in 2020. In 2021, Google launched major serving-related monitoring features. In July 2021, DataRobot purchased Algorithmia, a deployment platform.

- Feature stores: Databricks and Google both launched feature stores in 2021. AWS announced theirs at the end of 2020.

The clearest contemporary trends (at the time of writing) evident in the big platforms are:

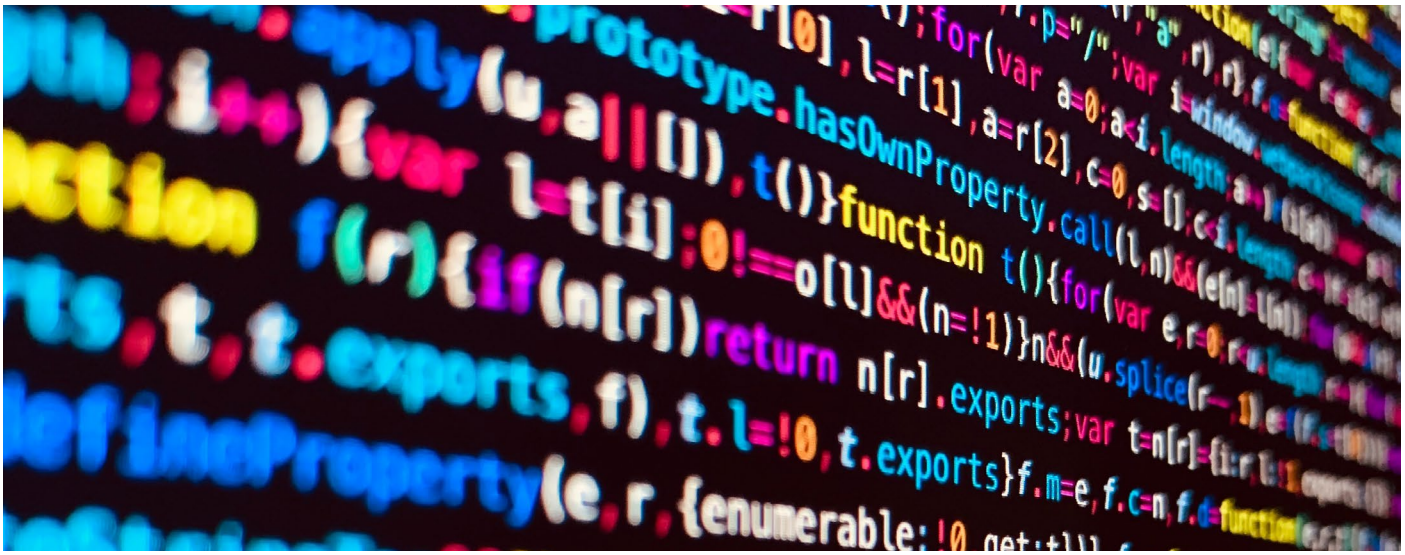
- API-fication of ML models is another key trend we are witnessingg, AI stores with pre-trained models, exposed via APIs, give a drag-and-drop option for AI development across companies.

- Tooling focused on data wrangling: AWS Data Wrangler was announced at the end of 2020 and Microsoft's Synapse integration is in preview. Most of the vendors featured in comparisons are offering ways to make data wrangling visual and interactive.

- AutoML's capabilities are expanding, including a variety of aided machine learning features for professional data scientists as well as approaches aimed at enabling citizen data scientists.

- AutoML offerings are getting more diverse, branching out into a range of assisted ML features for experienced data scientists as well as approaches aimed at empowering citizen data scientists.

These are trends to watch for in the future as vendors will likely deepen these new offerings and other platforms will likely follow.



Choosing the right MLOps platform for you - avoiding the pitfalls

Comparing Platform maturity

An MLOps platform isn't used by every organisation that benefits from machine learning. It is undoubtedly possible to deploy models without a platform – according to an O'Reilly survey, up to 46% of models may have been delivered without using any available MLOps deployment tool. Selecting and implementing a platform for a single project can be a waste of time in some situations. Where there are several projects, platforms are most useful since skills and knowledge can be exchanged more easily. It can be tough to decide on the ideal platform buying criteria, and it can be tempting to look for "feature completeness".

Here's why we believe it is risky to allow a breadth of features to dominate considerations:

- All platforms are adding features over time
- The varied use cases make it 'completely' impossible for them to be defined in a neutral and detailed way.
- The wide range of features can come at the cost of inflexibility
- Where any required features are missing, it is often sufficient to stitch together tools for them
- Many use cases don't require a wide range of features from a platform

You could be concerned that your company's caseload is so diverse that you'll need everything. Relax! Even at the largest companies, we find it best to first conduct research and write down the cases. If it becomes a lengthy list, it can always be prioritised. The breadth of a platform's features, on the other hand, is not irrelevant. If no single platform can match all of your requirements, you'll have to piece together various tools. It's fine to sew tools together as long as it goes smoothly. But tools that aren't built to function together don't always get along. Some businesses have a wide range of use cases, making it challenging for a single platform to support them all. Imagine that you have some combination of:

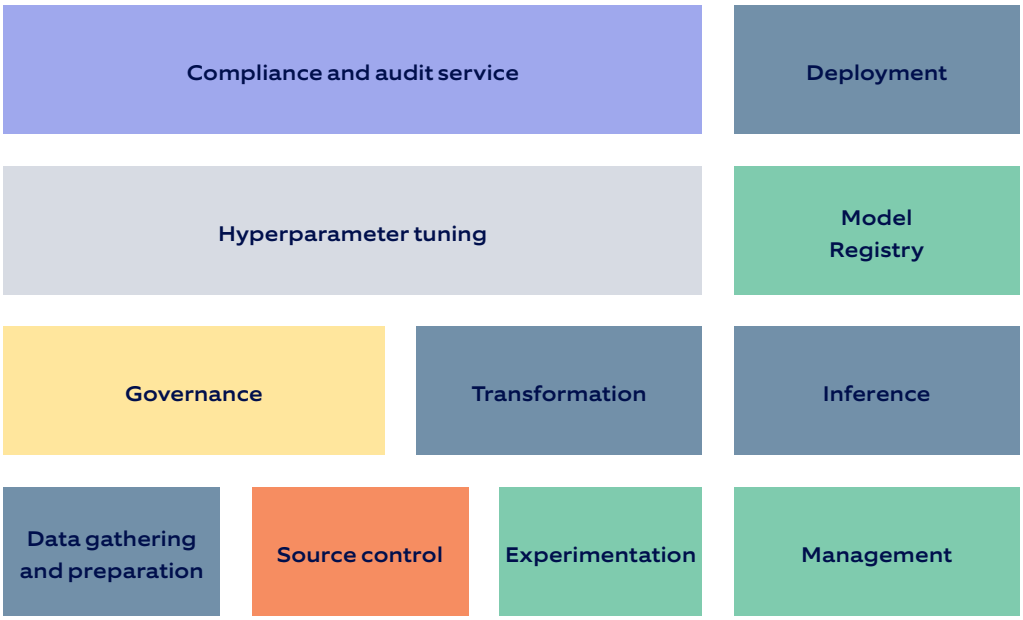
- Financial use cases requiring an elevated level of governance and transparency
- Use cases for long-running batch text processing Reinforcement learning use cases
- Models used in a high-traffic website with the need to A/B test versions in production
- AutoML use cases
- Low-latency prediction use cases that require GPUs for inference

In such a case, you could go for a general-purpose option. While MLOps platforms While MLOps platforms can cover much of this, you might need a plugin to a specific serving/monitoring solution for the high traffic. For financial cases, batch parts might be a better fit for Spark or Airflow and if the platform does not have adequate AutoML, you can get that from a different platform.

Features to consider in MLOps platforms

The beauty (and the problem) of machine learning is that you may experiment with a wide range of methodologies and approaches.

But at the same time, you must also keep a close eye on how your model performs under various settings, such as training methods, datasets, validation methods, and so on.



Let us look at some of the platforms in the market. These platforms can be classified into four groups:

- **Model training platforms**
  A few years ago, several prominent data platforms emerged to address data preparation and training requirements. As the industry progressed, the difficulty switched to operational challenges, and some of these platforms that originally focused on model training have now expanded to include MLOps features.

- **Platform with MLOps as the core offering**
  This new category of MLOps systems is specifically designed to address machine learning operations, monitoring, and deployment challenges. These systems deliver specialised functionality and architectural concepts purpose-built for executing machine learning at scale, as MLOps is their primary expertise

- **Open-source**
  There are several open-source options available, like kubeflow, ModelDB, Seldon, and MLflow. You can pick the open-source products and stitch them together. Companies looking to build their own custom MLOps solution may choose this route. However, be wary of the implementation effort and time-to-market impact. If possible, do sign up for the enterprise support plan to successfully run these products at scale.

- **Cloud providers**
  Most of the popular cloud providers also provide tools to help manage the entire ML life cycle, including MLOps, from data preparation to annotation to model training to deployment. The kitchen sink technique may be overly complicated, but if you are seeking for a one-stop-shop solution, it's worth exploring.

Offerings by different providers

Here is a list of some of the offerings being provided in the market:

	Data gathering, Transformation	Experimentation Testing, Tuning, Training	Deployment, Inference	Monitoring, Audting, MGMT, Retraining
Pachyderm	◆	◆	◆	
Algorithmia			◆	◆
MLflow		◆	◆	◆
Kubeflow	◆	◆	◆	
Polyaxon		◆	◆	◆
Valohai	◆	◆	◆	◆
Allegro	◆	◆	◆	◆
Azure ML	◆	◆	◆	◆
Google Cloud Auto ML	◆	◆	◆	◆

The way forward

MLOps is a complex and continuously evolving ecosystem. With ML tools, we have the tooling needed to address pain points around source control, model validation, model versioning / storage / sharing, and model deployment. In future, we will evaluate templates, tasks, triggers and processes with different tools. This will effectively reduce the friction between app developers and data scientists, and improve overall agility in AI-infused engineering.

References

Industry Forecast: oreilly-ml-ops.pdf

MLOps Levels: MLOps: Continuous delivery and automation pipelines in machine learning | Google Cloud

MLOps Life Cycle: AI-INFRASTRUCTURE.ORG

Author

Arun Madan is a DevOps enthusiast at Nagarro. He enjoys an open, value-adding culture, trust, and continuous learning. Thanks to his involvement in diverse project activities, he has invaluable experience in the optimal handling of migrating IT projects.

About Nagarro

Nagarro is a global digital engineering leader with a full-service offering, including digital product engineering, digital commerce, customer experience, AI and ML-based solutions, Cloud, immersive technologies, IoT solutions, and consulting on next-generation ERP. We help our clients become innovative, digital-first companies through our entrepreneurial and agile mindset, and we deliver on our promise of thinking breakthroughs.

We have a broad and long-standing international customer base, primarily in Europe and North America. This includes many global blue-chip companies, leading independent software vendors (ISVs), other market and industry leaders, and public sector clients.

Today, we are over 18,000 experts across 33 countries, forming a Nation of Nagarrians, ready to help our customers succeed.

For more information, visit www.nagarro.com.