comet

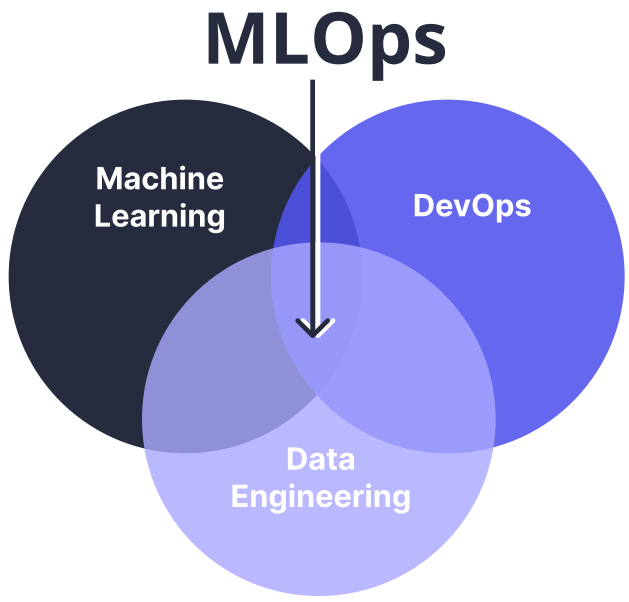
The Ultimate Guide to MLOps

Organizations are increasingly realizing the value of machine learning and looking for more ways to apply it to more innovative cases. Machine learning model operational management (MLOps) provides massive returns when organizations develop a robust and efficient system. It can help organizations streamline processes and launch evolvable ML-powered software. But how can teams know if they have a powerful MLOps workflow? Here's your ultimate guide to understanding and creating better MLOps.

What is MLOps?

MLOps stands for machine learning operations. It's a set of practices and processes streamlining ML model management, deployment, and monitoring. In machine learning, data scientists collaborate with other teams, including engineers, developers, business professionals, and operations, to push ML models into production.

The ML lifecycle comprises five stages: data collection, data preparation, model training, model deployment, and model monitoring. Model development and deployment are often separate processes handled by different teams, which creates a deployment gap and siloed tasks, introduces human error, and causes lengthy development cycles. With MLOps, each stage of the ML lifecycle is unified in a single workflow to facilitate collaboration and communication, aligning previously siloed teams.



The Principles of MLOps

Let's look at some of the essential concepts to understand and implement them better.

Versioning

Versioning is tracking and managing any changes to your models across their entire lifecycle. In machine learning, the development process is highly iterative. It entails testing several models, parameter optimization, feature tuning, and more. Many variables, like hyperparameter values, metrics, and model versions, are repeatedly tweaked between experiments. If you want reproducible experiments, you must implement versioning to manage and track the changes. Collaborating without proper versioning can be challenging, especially when working with multiple teams or on complex projects. Additionally, versioning is essential to implement AI governance, where documentation across all activities is imperative.

Automation

The entire machine learning lifecycle contains several highly iterative steps – automating workflow with as little manual intervention as possible results in faster deployment, easier problem detection, and more reliable processes..

In model development and training, teams can implement triggers for automated processes such as monitoring events, changes to data and code, messaging, and more. Automated testing aids in the early detection of issues and enables the likelihood of fixing errors quickly.

According to MLOps.org, there are three levels of automation in ML projects.:

1. **Manual Process** – The entire machine learning lifecycle is done manually, from data preparation to deployment.

2. **ML Pipeline Automation** – This level includes training models automatically. The process of model retraining is triggered whenever new data is available.

3. **CI/CD Pipeline Automation** – This stage introduces a CI/CD pipeline to perform fast and reliable model deployment. Teams automatically build, test, and deploy ML models and training pipeline components at this level.

CI/CD for Machine Learning

Continuous Integration and Continuous Deployment (CI/CD) are essential for effective MLOps. Continuous integration in machine learning means that every time data or code is updated, the ML pipeline reruns. Since everything is versioned and repeatable, teams can share the codebase across different projects. On the other hand, continuous deployment (CD) is a technique to deploy any new releases to production automatically. This technique allows teams to receive faster user feedback and new data for retraining and creating new models..

CI/CD are concepts borrowed from DevOps that can help ML teams build, test, and deploy applications continuously. It's also used for creating and maintaining applications incrementally. You can use CI/CD to improve the quality of your code base, reduce time to market, and ensure that your application meets customer expectations

Continuous Monitoring

In machine learning, deploying your models into production is only half the battle. The real-world environment often presents factors that make your model fall short of expectations. Once an ML model has been deployed, it must be monitored to ensure it performs as expected.

A model's performance can also degrade over time due to changes in data, environmental shifts, changes in consumer behavior, and more. Monitoring these issues before they can negatively affect a user's decisions establishes trust in the model and prevents regulatory and operational risks. Implementing a model production monitoring platform (MPM) can help you set alerts to detect model performance issues. MPMs can also provide drill-down capabilities to troubleshoot and detect anomalies.

What's the Difference Between MLOps and DevOps?

We defined MLOps in the earlier section. But what is DevOps?

DevOps is a set of practices and tools that integrate software development (Dev) and IT Operations (Ops). It aims to bridge the gap between the teams that write the code (Dev) and those that oversee the infrastructure and management of the tools used to run and manage the product (Ops). DevOps promote communication and collaboration to accelerate a system's development cycle and provide continuous delivery to ensure high-quality software.

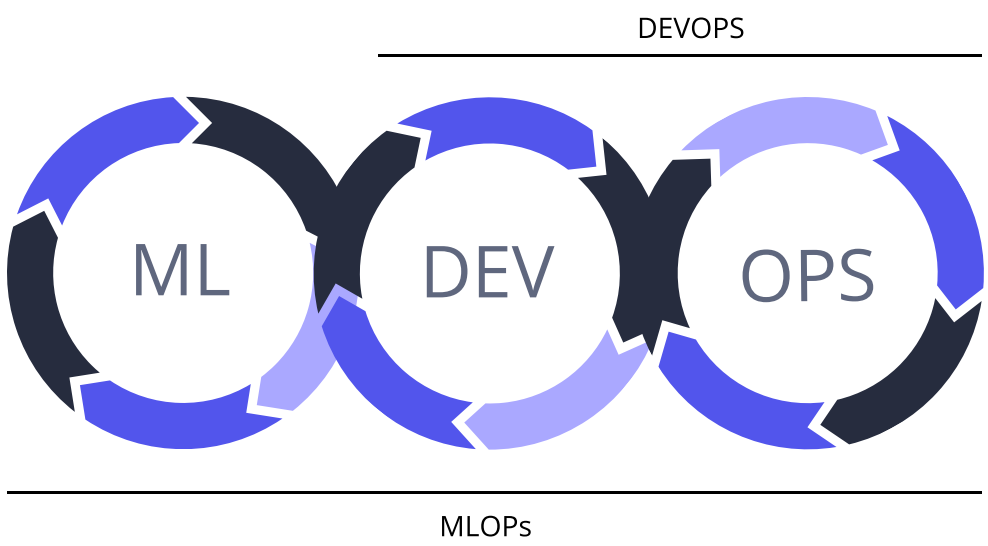
The wide adoption and rapid success of DevOps prompted the adoption of similar principles to streamline & improve the processes in machine learning (MLOps). Even though MLOps share the same principles as DevOps, they vary in execution. Let's take a look at their fundamental differences.

1. In DevOps, code version control ensures clear documentation of all changes or updates made to the project in development. In machine learning, code is just one of the many things to version. Data, iterative parameters, metadata, logs, and models are significant inputs that must also be tracked. All these inputs necessitate more complex version control.

2. Traditional software does not degrade the way an ML model does. Once engineers deploy the software, it will always serve its intended purpose. But, ML models need monitoring for model drift, data skews, negative feedback loops, and more. Data and environmental factors constantly change, potentially affecting the model's performance. Models require regular retraining to stay current and provide consistent value.

3. Machine learning teams are more hybrid than software teams. They consist of data scientists, data engineers, researchers, developers, and ML engineers, while a DevOps team typically consists of only software engineers.

4. DevOps teams only require a CI/CD (Continuous Integration and Deployment) pipeline. Machine learning, however, requires a CI/CD pipeline with a retraining approach. Future data may change, affecting the model's performance, so ML teams must add a retraining stage in their workflow to keep the model's results reliable.



Stages of the MLOps Cycle

Every machine learning project has to go through several stages of development before turning into a practical model. Optimizing work in each machine learning lifecycle can improve ML projects and produce better results. Some of the critical steps in the MLOps cycle are the following:

1. Identifying ML Goals and Plans

Every model development initiative begins with detailed planning, which includes defining the problems that need solving. It's also crucial to mark all the key points that will lead to project success. Setting a concrete, measurable success metric for the model is non-negotiable.

2. Data Collection, Preparation, and Data Exploration

The next stage in the MLOps cycle is collecting data and preparing it for model development. In machine learning, a model is only as good as its data. However, data can come in different formats and from various sources. It's crucial to establish a set of rules to define and label your data. Cleaning, versioning, and attaching attributes to your data are essential to determine any underlying patterns that will help you build your model. Data exploration is an additional step to verify the data for bias and completeness before passing it to the next stage.

3. Model Training

Model training is instrumental in understanding the various patterns, rules, and features. In this part of the process, an ML algorithm is fed with training data from which it can learn. Model training consists of the sample output data and the input data sets that will influence the outcome or prediction. This iterative training process, called model fitting, continues until the model learns.

4. Model Optimization

Model optimization is critical for ensuring smooth, consistent performance. This process involves tradeoffs between size, runtime performance, and accuracy, all of which impact the core user experience.

5. Model Evaluation

Model evaluation is typically an ongoing process throughout the machine learning lifecycle. It’s essential to ensure the efficacy of a model during the initial research phases, and it also plays a role in model monitoring.

6. Model Deployment

The next stage is model deployment. If the model produces an accurate result that meets requirements at an acceptable speed, it's deployed in the existing system. Deployment involves launching the model into live environments where consumers can use them.

7. Model Monitoring

Once deployed into production, monitoring the model's performance is necessary to ensure it is still performing as expected. A model's performance may deteriorate over time when the real world presents new and unknown data (data drift) or when there are changes in the environment and the model's learning pattern no longer holds (concept drift). These issues can negatively affect consumers and businesses if not detected in time.

Benefits of MLOps

1. Enables Scalability

There's been a considerable increase in ML investment across various industries. Along with it, many organizations have increased the number and complexity of ML projects. Organizations need a structured framework for training multiple models simultaneously while incorporating business and regulatory requirements and ensuring AI governance as they scale. MLOps provides a framework for managing the ML lifecycle efficiently, creating repeatable processes, and staying in compliance with regulations. Organizations can quickly scale by establishing MLOps.

2. Improves Collaboration and Breaks Organizational Silos

MLOps helps establish rules and practices that foster collaboration. It keeps everyone informed of each other's progress and improves the model hand-off process between development and deployment. Every ML project involves a development and deployment team and internal stakeholders like project managers, business owners, legal teams, and key decision-makers. To create the best ML product for the problem, ML teams must work with internal stakeholders to align business goals and strategies. MLOps ensures alignment and promotes frequent team communication to achieve business goals and hit KPIs.

3. Accelerates the ML Lifecycle

The increased demand for machine learning requires rapidly iterating ML processes like experimentation, training runs, and deployment. Borrowing the concept of DevOps, MLOps aims to meet this demand by implementing a set of practices to streamline, automate, and integrate the development and production phases of the ML lifecycle. Establishing a robust MLOps process helps teams speed up the model development process, enabling faster deployment of models into production.

4. Enables AI Observability

AI observability is a method that provides deep insights into an ML model's data, behavior, and performance across the model lifecycle. It goes beyond model monitoring since monitoring tells us "what" issues are happening, while observability explains "why" they occur. We must implement the MLOps principles of automation, continuous training and monitoring, and versioning to fully embrace AI observability.

5. Demonstrates Explainable AI

Explainability plays a crucial role in machine learning and AI. It aims to answer a user's or a stakeholder's question about how an ML model arrived at its decision. A lack of explainability poses risks across industries. In healthcare, where an ML model suggests patient care, providers must trust the model's reasoning since the stakes are exceptionally high.

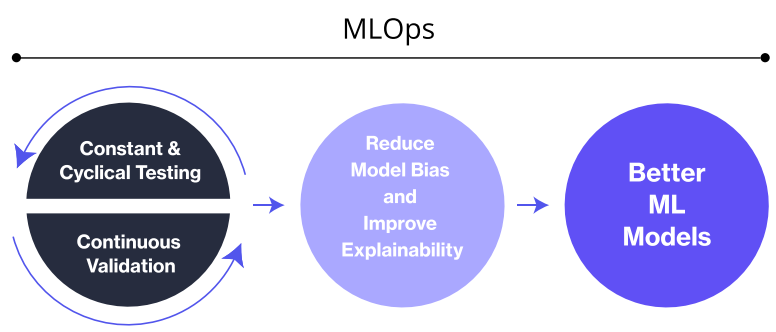
We must build responsible, trustworthy, reliable, robust, accountable, and transparent models to achieve explainable AI. Establishing MLOps helps accomplish this through well-defined frameworks, processes, and practices across the ML lifecycle. MLOps helps us understand the model's outcome and behavior and, in turn, enables us to explain it to others and build trust in the model. Continuous model training and monitoring help ensure that a model performs as intended.

6. Promotes AI Governance

AI governance refers to implementing a legal framework that ensures ML models and their applications are explainable, transparent, and ethical. In AI governance, organizations must define policies and establish accountability in creating and deploying these models. MLOps ensures that these policies are in place through well-documented activities on an ML project, keeping prior versions of models, testing models for biases, monitoring models in production to prevent concept drift, and more. Implementing MLOps protects your organization from legal, ethical, and regulatory risks that can harm your organization's reputation and financial performance.

7. Build Better Models That Can Improve Business ROI

Ultimately, the critical output of establishing an MLOps culture is to build a high-quality model that users can trust. With MLOps, teams can create better models because of continuous & focused feedback. Constant and cyclical testing and validation reduce model bias and improve explainability.



MLOps Tips and Best Practices

Machine learning is iterative and complex by nature, but the complexity isn't limited to the data science behind the technology. Efficient model deployment requires efficient processes, teamwork, and communication. Successful machine learning teams need to be highly functional when it comes to critical components, such as:

- The visibility to view, access, and react to ML processes and deliverables
- The ability to reproduce functions and achieve the same outcomes
- Collaboration across multiple work units and teams

Aside from the above, ML engineers must institute best practices to deliver machine learning systems consistently. Here are our recommendations:

1. Set Up Naming Conventions

As machine learning systems expand, so do the number of variables. Make sure to develop and establish a straightforward naming convention that every team member understands. As your project gets more complex, remember to stick to the convention to mitigate the challenge of the CACE principle – changing anything changes everything.

2. Code Quality Checks

High-quality code, according to Python's Alexander Van Tol, consists of three agreeable identifiers:

- It serves its intended purpose
- It does not contain defects
- It is easy to read, maintain, and extend

The discrepancy in model fitting happens when real-world data fed into training pipelines doesn’t provide the right outcome variable. That's why it's essential to check codes past unit testing. Use linters or formatters to enforce a particular code style throughout your ML project for better efficiency.

3. Experiment Tracking

Aspects of machine learning, like model architecture and hyperparameter search, are evolving. Delivering the best possible system means you always have to track the evolutions of patterns in your data. Experiment tracking is vital, as well as finding the right platform to do it. Use a powerful tool like Comet to track and reproduce experiments and improve productivity.

4. Data Validation

Data can make or break your ML models. Your sampling process needs fixing if your data's statistical properties don't match its training properties. Data drift could also ensue. Improve your MLOps using a data validation library that helps you detect errors and perform statistical validation like hypothesis testing.

5. Periodic MLOps Scoring

An excellent way to gauge whether you have good MLOps is to score it. You can develop a system, given that you consider everything from features and data to infrastructure and monitoring. Starting with a transparent scoring system is a good step in drastically improving your workflow.

How to Choose the Right MLOps Stack

The best platform for you will depend on your use case since different capabilities are needed for different use cases. For example, testing a proof of concept requires data preparation, feature engineering, model prototyping, and validation using experimentation and data processing. However, if you need frequent retraining, such as in fraud detection, you need model training and ML pipelines to connect additional steps like data extraction and preprocessing.

When evaluating MLOps platforms, you should first define your use case to ensure any platform you consider has the right features to support your needs. When you've narrowed down your platform list according to use case, conduct a proof of concept with multiple vendors. This will help evaluate each one, understand their differences, and identify any issues before making a commitment.

Questions to Ask Your MLOps Provider

By leveraging people, processes, and technology, data science can drive your business forward in multiple ways.

Organizations that invest in MLOps and other data science initiatives see significant gains.

- Netflix reported its ML algorithm that drives its personalization engine is worth $1 billion
- Amazon's ML and AI apps that power robots and their pick, pack, and ship process in warehouses reduced the click-to-ship time by 225%. By automating the inventory flow, Amazon estimates it improved productivity by 20%
- A study by McKinsey found that companies successfully implementing machine learning and AI now report that 27% of their earnings are attributable to the technology

An investment in MLOps should lead to demonstrable improvements. When discussing ML solutions with a provider, ask for case studies demonstrating ROI and impact and references from current clients who can talk about workflow and ease of use.

Asking targeted questions to ensure you understand the benefits and limitations of any MLOps provider across the following areas is also essential:

Model Metadata Storage and Management

- What infrastructure is necessary, and will it integrate with my current workflow?
- Can you customize the metadata structure?
- Can you version and reproduce models and experiments with a complete model lineage?
- Can you customize the UI and visualizations?
- Can you use it inside orchestration and pipeline tools?

Data and Pipeline Versioning

- How is data modality supported, and can you see previews of tabular data?
- Can you compare diverse datasets?

Production Model Monitoring

- How do you monitor input data, feature, concept, or model drift?
- How easy is it to connect to model serving tools?
- Can you compare multiple models that are running simultaneously?
- Do you provide automated alerts if someone goes awry?

Hyperparameter Tuning

- What does it take to connect to my codebase?
- Can it be run in a distributed infrastructure?
- Can you stop trials that do not appear promising?
- What happens when trials fail on parameter configurations?
- Can you distribute training on multiple machines?
- Can I visualize sweeps?

Orchestration and Workflow Pipelines

- Can you abstract away execution to any infrastructure?
- Can you speed up pipeline execution by caching outputs in intermediate steps and only running specific steps?
- Can you rerun steps that failed without crashing the entire pipeline?
- Can you schedule pipeline execution based on events or time?
- Can you visualize the pipeline structure?

Model Deployment

- Is there a limit to infrastructure scalability?
- Do you have built-in monitoring functionality?
- What compatibility do you have with model packaging frameworks and utilities?

Ready for better MLOps?

Effectively developing, deploying, and continuously improving ML models requires automation of the MLOps pipeline. The most critical MLOps best practices that organizations must adopt to achieve this include:

- Pipeline to create advanced machine learning systems collaboratively
- Versioning to ensure the reproducibility of models
- Testing to ensure the production models meet the required standards
- Automation to save time and provide efficient systems.

Successfully implementing MLOps requires different tools. Comet is a machine learning platform that allows you to track, monitor, and optimize ML models. It will enable you to see and compare all your experiences in a single place. Organizations, teams, individuals, and anyone else can easily visualize experiments, facilitate work, and run experiments. Try for free today.