

[Paperspace is now part of DigitalOcean, and we've got a new look to match!](#)

[Learn more](#)

Products

Resources

[Pricing](#)

[We're hiring!](#)



[Sign in](#) [Sign up free](#)

[MLOps](#)

Introduction to MLOps

In this overview of MLOps, Paperspace contributor Joydip Kanjilal covers the basics of MLOps and building resilient machine learning applications.

4 years ago • 8 min read

 By [Joydip Kanjilal](#)

Table of contents [»](#)

Introduction

Most data scientists are not expert programmers. While they are adept at choosing or creating the best model to solve a machine learning problem, they don't necessarily have the expertise to package, test, deploy, and maintain this model in a production environment. That's exactly where MLOps comes to the rescue.

MLOps creates a bridge between data scientists and production teams. It is a practice that combines DevOps with Machine Learning. This article talks about MLOps, why it is needed, the challenges of MLOps, the tools available, and how MLOps pipelines work.

Specifically, we'll cover:

- What is MLOps?
- Why do we need MLOps?
- Similarities between MLOPs and DevOps
- Dissimilarities between MLOPs and DevOps
- What are MLOPs pipelines?
- Why do machine learning projects fail?
- MLOps: Core principles
- MLOps: Best practices

- Summary

Let's get started!

What is MLOps?

Machine Learning Operations (also known as MLOps) is a collection of tools and best practices for improving communication across teams and automating the end-to-end machine learning life cycle to improve continuous integration and deployment efficiency. It is a concept that refers to the merger of long-established DevOps methods with the growing science of Machine Learning.

MLOps encompasses more than model construction and design. It includes data management, automated model development, code generation, model training and retraining, continuous model development, deployment, and model monitoring. Incorporating DevOps ideas into machine learning offers a shorter development cycle, improved quality control, and the ability to adapt to changing business needs.

System administrators, data science teams, and other business units collaborate and communicate to foster a common understanding of how production models are created and maintained, much the same as DevOps does for software. DevOps is a proven practice that can provide rapid development life cycles, increase development velocity, improve code quality through proper testing, and help achieve faster time to market.

Why do we need MLOps?

Long-term value and reduced risks. MLOps helps organizations generate long-term value while reducing the risks that are associated with data science, machine learning, and AI initiatives.

Streamlined processes and improved customer experience. Machine learning can assist in deploying solutions that uncover previously untapped income streams, save time, and decrease resource costs by streamlining processes, using data analytics for decision making, and improving the customer experience.

Automation and faster time to market. MLOps automation provides quicker time-to-market and lower operational costs, enabling businesses to be more agile and strategic in their decision-making.

Similarities between MLOPs and DevOps

Both MLOps and DevOps share a need for process automation, continuous integration, and continuous delivery.

It also helps to have proper testing of the code base for both MLOps and DevOps.

In addition, there should be adequate collaboration between software developers and those who manage the infrastructure, as well as other stakeholders.

Dissimilarities between MLOps and DevOps

Although MLOps is derived from DevOps, there are subtle differences between the two.

In MLOps, data is a necessary input for developing the machine learning model. But in DevOps, data is an output of the program, not an input.

In MLOPs, the model must be validated continuously in production for performance deterioration caused by new data over time. The software system does not deteriorate in DevOps; it is merely monitored for health maintenance purposes.

Concepts such as model training, model testing, and validation are all unique to MLOps and irrelevant in the conventional software realm of DevOps. Moreover, the training model tends to be compute-intensive, hence requiring (typically) the use of powerful GPUs.

MLOps requires Continuous Training (CT), a process that automatically identifies scenarios that require a particular model to be retrained and re-deployed due to performance degradation in the currently deployed version of the model.

As you can see, despite their similarities you can't just use DevOps tools to run machine learning projects; there are too many requirements that are specific to machine learning.

What are MLOps pipelines?

MLOps pipelines are sequential steps triggered to automatically design, deploy, and manage model workflows. Machine Learning CI/CD pipelines are comprised of the following smaller pipelines:

- Data Pipeline: this pipeline is responsible for conducting ETL and automatically bringing the necessary data into the model.
- Environment Pipeline: this pipeline guarantees that the right dependencies are always loaded and available.
- Training Pipeline: this pipeline is responsible for training your model. Based on the design of your MLOps pipeline, it might need the data pipeline and training pipeline to import data and dependencies.
- Testing Pipeline: this pipeline is responsible for verifying the trained model. It might often take advantage of automated test cases that are triggered on a pre-defined schedule.
- Deployment Pipeline: this pipeline is used to deploy your Machine Learning model in pre-production or a production environment

Challenges of MLOps

Similar to other technologies, MLOps has its challenges as well. Here are a few such challenges you might face when implementing MLOps in your organization:

- Data Quality: The quality of your model is directly proportional to the quality of the data used for training the model. Data quality is one of the most critical factors for building and training models such that those models will be able to produce better insights or predictions.
- Data Volume: As the data set grows in size, the accuracy of the machine learning model improves since more data is utilized for training the model. Simultaneously, if the underlying resources are not adaptable to data upticks, such as adding more storage and processing capacity, the model's usefulness suffers significantly.
- Deployment: Deployment is yet another challenge since deploying a Machine Learning model involves deploying the model *and* the data used to train it. In addition to this, you might have to retrain your models and validate them as well. If you're trying to do this manually, it is a time-consuming process.

Various Teams in a Typical Machine Learning Project

Typically, you'll have the following teams in organizations that run Machine Learning projects:

- **Data Engineering Team:** this team is responsible for building data pipelines within various applications in your organization
- **IT Team:** this team ensures that IT security standards are properly enforced when working with Machine Learning projects
- **Testing Team:** this team is responsible for verifying the accuracy of the Machine Learning model created by the Data Scientists
- **Operations Team:** the Operations Team is in charge of keeping the system running in the production environment and keeping track of daily progress.
- **Data Scientist Team:** this team is responsible for creating and training predictive Machine Learning models. The Data Scientist Team also needs to collaborate with all other teams for the Machine Learning project to succeed.

Why do Machine Learning projects fail?

There are several reasons due to which machine learning projects can fail. Some of them are discussed in this section.

1. Lack of Team Collaboration

To be successful, it is imperative that the data scientists work with multiple teams in synergy and not in isolation. Lack of communication and collaboration can be detrimental to the success of the Team as a whole and hence can be a reason for failure.

2. Lack of Continuous Improvements

Since model performance deteriorates over time, you should upgrade your models as part of a maintenance cycle. A successful Machine learning project should be evaluated and improved as a continuous process based on the feedback received or when the data changes. However, data scientists often fail to deliver improvements promptly. For each data collection cycle, they need to build, test, and deploy the model again, which requires them to connect with multiple teams.

3. Lack of Expertise

Lack of expertise is yet another reason for the failure of Machine Learning projects. You need expert data scientists to manage your Machine Learning projects. Unfortunately, while companies have been desperately trying to hire expert data scientists, there is a massive shortage of the right talent.

4. The Quality and Quantity of Data

The quality and quantity of data are other problems. Typically, Machine Learning projects use large datasets since you need large amounts of data for better predictions. However, as the volume of data grows, the complexity and challenges associated with it magnify. Moreover, data is usually merged from multiple sources, and data might not be in sync. This might pose yet another challenge since you would not provide better insights with "bad" data. Data sourced from different locations can also have security constraints and can even be in various formats such as structured, unstructured, text, images, etc.

How can you prevent Machine Learning projects from failing?

While there are no specific guidelines for ensuring the success of a Machine Learning project, here are some ways that might help prevent their failure.

- It would help if you understood how Machine Learning works and how handling a Machine Learning project is different from other kinds of projects.
- The project should be appropriately scoped with realistic objectives, a reasonable budget, and leadership support.
- It would help if you had the resources necessary to successfully execute a Machine Learning project with good communication and collaboration between team members.
- Your team should be adept at collecting, storing, cleaning, and analyzing massive volumes of data.

MLOps: Core Principles

With the surge in the usage of Machine Learning and AI in software products and services, everyone should be careful to follow MLOps best practices and use the right tools for testing, deploying, managing, and monitoring machine learning models in real-time. **You can take advantage of MLOps to evade "technical debt" in your Machine Learning applications.**

Automation

Automation should be your priority if you want to use MLOps in your machine learning project effectively. The maturity of your machine learning process is determined by the level of automation you have. This, in turn, can increase the speed at which your machine learning models can be developed, trained, and deployed. In addition, this principle promotes the usage of ML workflows that are completely automated, with no need for human involvement.

Versioning

Versioning aims to treat machine learning training scripts, models, and datasets for model training as first-class citizens in DevOps processes and ensure that any changes made in the datasets and codebase are tracked correctly. The version control system saves and tracks different versions of the model. This makes it easy and seamless to revert to a previous version if needed.

Testing

Testing is an essential aspect of the MLOps life cycle. MLOps supports a structured testing technique for Machine Learning systems based on the data pipeline, ML model pipeline, and application pipeline. As such, you should have methods in place for testing features and data, model development, and machine learning infrastructure.

Monitoring

Once a Machine Learning model has been deployed successfully, it should be monitored to know if it is performing as expected. In addition, the dependencies, data version, usage, and changes made to the model are monitored from time to time.

MLOps: Best Practices

Communication and collaboration

While the data scientist directs how the model should be built, you need a team of engineers and strategists to be successful. You should hire subject matter experts (also called SMEs), data scientists, software engineers, and business analysts to your team. Hence it is imperative that proper communication and collaboration are maintained.

Validating the Dataset

Data Validation is one of the most important practices you should adopt. Once the model has been pushed to production, performance might degrade, and you might not get the correct predictions. This is why you should retrain the model even if it is a costly affair, both in terms of time and resources.

Set up clear Business Objectives

Business objectives should be defined appropriately, and you should have a clear goal in mind. You should know the business problem you're trying to solve with Machine Learning and determine how to solve it. Most importantly, you should know what value your model is providing to the organization.

Containerization

Containerization is where programs are executed in separate user spaces known as containers, all of which utilize a standard operating system. It would be best to leverage containerization to automate the whole process, starting from model development to staging to production. You can use [Docker](#) for this.

Summary

MLOps comprises a collection of proven techniques to automate the machine learning life cycle to eliminate the divide between model design, development, and operations. Machine learning initiatives are no longer focused on pure data science. Instead, data and cloud engineering skills are becoming more critical for managing the whole machine learning lifecycle. That's where MLOps comes into the picture.

Add speed and simplicity to your Machine Learning workflow
today

[Get started](#)

[Contact Sales](#)

- Tags:
- [MLOps](#)
- [ML](#)
- [Machine Learning](#)

Spread the word

- [Share](#)
- [Tweet](#)
- [Share](#)
- [Copy](#)
- [Email](#)

<https://blog.paperspace.com/>
[public](#)
[Next article](#)

[Welcoming back applications for ATG Fellowships!](#)

[public](#)
[Previous article](#)

[New ML Showcase Entry: Creating Interactive Web Apps with NVIDIA RAPIDS and Plotly](#)

Keep reading

public

[Building an MLOps Community: Interview with Demetrios Brinkmann](#)

4 years ago • 6 min read
public

[What is MLOps?](#)

5 years ago • 6 min read

Subscribe to our newsletter

Stay updated with Paperspace by DigitalOcean Blog by signing up for our newsletter.

Your email address Your email address [JOIN NOW](#)

 Awesome! Now check your inbox and click the link to confirm your subscription.

Please enter a valid email address

Oops! There was an error sending the email, please try later

Paperspace

Solutions

[Machine Learning](#) [GPU Infrastructure](#) [Cloud Desktops \(VDI\)](#) [3D Workstations](#) [Visual Computing](#) [Gaming](#)

Product

[Docs](#) [Changelog](#) [Status Page](#) [Referral Program](#) [Download App](#) [Customers](#) [Media Kit](#)

Resources

[Support](#) [Talk to an expert](#) [Forum](#) [Business](#) [Security](#) [Cloud GPU Comparison](#) [NVIDIA Cloud Partner](#) [Graphcore IPUs](#) [Media Kit](#)

Company

[About](#) [Blog](#) [Careers](#) [Shop](#) [Get Paid to Write ATG \(Research\)](#).

Part of the



family

© Copyright by [Paperspace](#) • All rights reserved

[Terms of Service](#)

•

[Privacy Policy](#)

