MLOps: What It Is, Challenges, Tools

Dheeraj Arremsetty, Data Scientist, Data & AI Elite at IBM, Krysten Thompson, Data Scientist, Data & AI Elite at IBM
MLOps...a new and burgeoning term in the machine learning and AI field. But what is it? And if you're a data scientist, should you care?

An increasing number of news stories highlight the brand and MLOps is a relatively new concept that has come about because more and more companies are adopting and integrating AI and ML into their businesses. Many companies learn (often the hard way) that it's one thing to say "Hey let's build a machine learning solution for this problem", and a completely different thing to get the solution working seamlessly in production. And then there's monitoring and managing the models in production.

Learning the hard way usually causes reflection about how things can be done differently (differently = better, more efficiently, less painful). And here's where MLOps comes in. Whether you're a software developer, engineer, data scientist or other techie person, you've likely heard of DevOps. As businesses began talking about the problems with getting ML solutions to production (in an efficient way that includes checks and balances throughout the process), the idea of leveraging what DevOps has been doing for many years came to mind. Would it be possible to incorporate some of the best practices of DevOps when building and deploying machine learning solutions?

What is MLOps?

If we were to sum up MLOps in one sentence, MLOps is about streamlining, automating, and monitoring the entire machine learning process, from ingesting data to running a model in production. Anyone who's built a machine learning solution, whether the solution was deployed to production or not, is familiar with the rigor and detail that goes into its creation.

First, you need data. Maybe you use a public data source, maybe the data comes from your company. We'll say it once: Data Cleaning. Because unless you're using the Kaggle Titanic dataset, it's unlikely you will ever receive perfect data.

Then there's feature engineering. What dataset comes with the perfect set of features to make a prediction? And you test different models to see which one performs best. While accuracy is important, you're working on a disease prediction model. Your False Negative rate is critically important in your model performance. You tune the hyper-parameters. The process is iterative...you create new features, test various models, evaluate the model metrics over and over again. Finally, you choose a model and get the solution deployed in production. Then what?

Enter MLOps...a set of processes and tools that:

- Helps train, build and orchestrate the models much faster in the development or experiment phase
- Consolidates and streamlines the process of moving ML models from development to production
- Tracks lineage and versioning
- Continually monitors & manages the model in production
- Flags model drift and bias
- Ensures the model is fair and robust to changing business conditions
- Collects meta data across all the stages of the process

The visual below compares DevOps to MLOps by highlighting their respective pipelines, from development through production. DevOps leverages something called Continuous Integration and Continuous Delivery (or Continuous Deployment), a.k.a. CI/CD. In DevOps, CI/CD is an automated process that allows software developers to update/edit code in an agile manner. Software developers can spend more time focusing on the business problem solution vs spending time dealing with code quality, compliance, and security.

MLOps strategically incorporates some of the best practices and processes from DevOps into the end-to-end machine learning lifecycle such as CI/CD. Automating machine learning processes combined with continuous integration and continuous delivery are critical components of MLOps.

![MLOps Diagram](image)

Fig 1.0: Shows comparison of DevOps and ML Ops

But MLOps adds one more element to its "continuous" processes...CM or Continuous Monitoring. Monitoring machine learning models in production not only ensures the pipeline doesn't break, but also helps flag drift or bias. (Drift is when a model's outputs change and shift in a certain direction over time. Bias can be introduced after a model is deployed to production when data inputs change unexpectedly. More information can be found here [https://www.ibm.com/blogs/journey-to-ai/2021/01/protect-against-model-drift-and-bias-to-ensure-your-ai-is-accurate-explainable-and-governed-on-any-cloud/]

"MLOps is a methodology of operation that aims to facilitate the process of bringing an experimental Machine Learning model into production and maintaining it efficiently."

The benefits to an organization implementing an MLOps workflow include:

- Avoiding technical debt (think short-term gains for longer-term costs; faster software releases can result in code re-work down the road)
- Reproducibility, reliability, and efficiency of machine learning pipelines
- Enhanced collaboration across groups
- Improved code and knowledge sharing

This all sounds great doesn't it? Why is it difficult to do?
Implementation Challenges

The list of challenges to automate the lifecycle for a machine learning solution is long and varied.

Hardware resources may be a limiting factor - an ML solution will process terabytes of data daily, but the data scientist can only build the solution training megabytes of data due to lack of infrastructure.

People resources and/or skill sets - it's not just data scientists that build a machine learning solution. It takes data engineers, IT, dev ops, among others to get a solution into production, running smoothly, and continually monitor the entire pipeline's performance.

What about "tools"? There is no established MLOps framework (yet). And the "how to's" can vary by company, solution, and amount of money a company is willing to invest.

More specifically, some of the challenges around MLOps organized by Data, Models, Knowledge, and Automation/Integration are noted below.

DATA
- Ensure consistent and reliable data connector
- Data connectors connecting to both dev and prod data sources
- Data connectors accessing data from multiple data sources
- Maintaining data versioning and data permissions access levels to different teams

MODELS
- Conduct conventional unit testing for features
- Conduct model reproducibility tests, training crash tests, (re)-training performance validation
- Register, version, and deploy models as batch or REST API on cloud, on premise, or edge
- Ensure models align with the business objectives and monitor for fairness, bias, and drift

KNOWLEDGE
- Enable feature reusability through central storage
- Enable pre-trained model reusability with well documented models and logs
- Track experiments across multiple runs to compare and evaluate models
- Ensure repeatability of model building pipelines for boosting confidence in predictions

AUTOMATION & INTEGRATION
- Manually or automatically trigger key actions, such as re-training, re-deployment, and update of monitors
- Periodically or conditionally trigger pipelines, or components in a pipeline
- Pre-built analysis for MLOps team to communicate insights on production data back to model developers who only have access to dev data

IBM Products/Architecture

One of the key takeaways from above is that MLOps is a combination of processes and tools that help an organization streamline the machine learning and AI lifecycle.

The diagram below reflects a complete and detailed MLOps flow on Cloud Pak for Data. The visual is organized according to the processes and steps implemented in Development vs. Production. The pipeline components are modular, allowing a customer to choose the tools they need based on project preferences and use cases.

The lifecycle begins with an understanding of what business problem is to be solved along with the data to build the machine learning model.

1. Relevant data from the data source is extracted and validated. Watson Studio and Watson Knowledge Catalog connect data to Development and Production environments.
2. Data then moves to the "Training Experiment Pipeline" where it's processed, and new features are potentially created. Model experimentation and training begins, and code can be stored in a repo. Models are tested, validated, and a winner is selected. Code from this process is collected in a repo. The model and training data are ready for the Production pipeline.
3. In the Production pipeline, the model is saved to the model registry in Watson Machine Learning. The model will be tested and validated prior to being deployed.
4. To make changes to the model more frequently and efficiently, all stages in this flow are automated with an orchestrated pipeline using a CI/CD tool. Watson Studio Pipelines are utilized for this purpose.
5. After final testing and validation, the model is deployed and monitoring the model begins. Monitoring helps ensure that the metrics used to evaluate the model are still producing the same results. If the model was selected because the False Positive Rate yielded .001% (let's say the model predicts disease), the implications if model drift appeared could be quite severe.
6. Finally, given the involvement of various stakeholders at each stage, metadata must be stored throughout the process to share the knowledge and information across teams. IBM has built AI Factsheets to enable smooth knowledge transfer.

Figure 1.2: MLOps End-to-End Lifecycle

Model deployment isn't the "end of the line". Putting machine learning solutions in production is an iterative process for which there isn't an end (unless of course the solution is shut down).
When machine learning solutions are built, deployed, and monitored efficiently and in a manner that elicits trust at each stage, a cadence is established that can result in an increased number of solutions implemented.

IBM has developed several tools that assist with MLOps, from collecting and preparing data, to building/deploying/monitoring models. Here is a brief overview.

**DATA TOOLS**

IBM Cloud Pak for Data [https://www.ibm.com/products/cloud-pak-for-data] is a data and AI platform that makes all data available for AI and analytics, both on-premise and on any cloud. Data can be gathered from multiple data sources using single SQL statements via Data Virtualization [https://www.ibm.com/analytics/data-virtualization]. Data Virtualization provides access to data at the source without moving data, accelerating time to value with faster and more accurate queries. And many out-of-the-box connectors are available to make set-up fast and easy.

Figure 1.3: Data Virtualization - Various out of the box in built data connectors

Keeping track of each version of the datasets is easy with Watson Knowledge Catalog [https://www.ibm.com/cloud/watson-knowledge-catalog]. Measure data quality using 11 dimensions out of the box. Customize across every value of every row of every record to reflect a column’s quality for your business and compliance. Track where data originated and how it’s consumed, increasing trust when accessing data across many sources and destinations.

In addition, an IBM research tool called DQAI [https://www.ibm.com/products/dqaiapi] helps with data in the following ways:

- Data Validation: provides quality scores for your data and insights on the quality scores
- Data Remediation: execute the recommendations provided by the quality analysis done during the Data Validation process
- Data Synthesis: easily generate new data having the same characteristics and distributions
- Pipelining: combine validators and remediation with constraints for a specific use case or application workflow
- Reporting: changes in quality metrics and data transformations can be automatically documented

**MODELING TOOLS**


Figure 1.4: AutoAI - AutoAI helps you experiment with different models and assess the results by automatically processing structured data to generate model-candidate pipelines. The best-performing pipelines can be saved as a machine learning model and deployed for scoring.

AI Factsheets [https://aifs360.mybluemix.net/] automatically capture model metadata across the model lifecycle in a singular view. This is especially useful for companies that need to meet regulatory requirements and help with external validation. Model validators need details from the model development, testing and validation phases to approve models for production use. AI Factsheets enables validators and approvers to get an accurate, always up-to-date view of the model lifecycle details. The automated collection of model metadata also saves valuable data scientist and ML engineering resources to focus on model building, instead of writing lengthy model documentations.

Figure 1.5: AI Factsheets capturing model meta data

**MONITORING TOOLS**

Watson OpenScale [https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=services-watson-openscale] tracks and measures outcomes from your AI models, and helps ensure they remain fair, explainable, and compliant no matter where your models were built or are running. The details and insights from OpenScale provide transparency about how your model determined a prediction. Listing some of the most important factors that led to the predictions allows you to trust the outcomes and easily explain outputs within an organization.
Watson OpenScale also monitors model performance drift, fairness and bias, and helps correct the drift in accuracy when an AI model is in production.

**AUTOMATION TOOLS**

Watson Studio Pipelines

https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ml-orchestration-overview.html provides a graphical interface for creating an end-to-end machine learning pipeline, from importing data, modeling training, through deployment. The drag-and-drop interface lets you quickly and easily build a flow. Select data nodes, model nodes, Jupyter notebooks, and even AutoAI models (thanks to Watson Studio Pipelines full integration with Watson Studio and Watson Studio's product offerings).

Automating the pipeline makes it simpler to build, run, and evaluate a model, and also shortens the time you work on a solution from conception to production. Once the pipeline is created, you can rapidly update and test modifications.

The Pipelines canvas provides tools to visualize the pipeline, customize it at run time with pipeline parameter variables, and run it as a trial job or on a schedule.

The Pipelines editor also allows for more cohesive collaboration between a data scientist and a ModelOps engineer. A data scientist can create and train a model. A ModelOps engineer can then automate the process of training, deploying, and evaluating the model after it is published to a production environment.

*Note: Watson Studio Pipelines tool is currently in beta release and is not supported for use in production environments.*