MLOps Guide

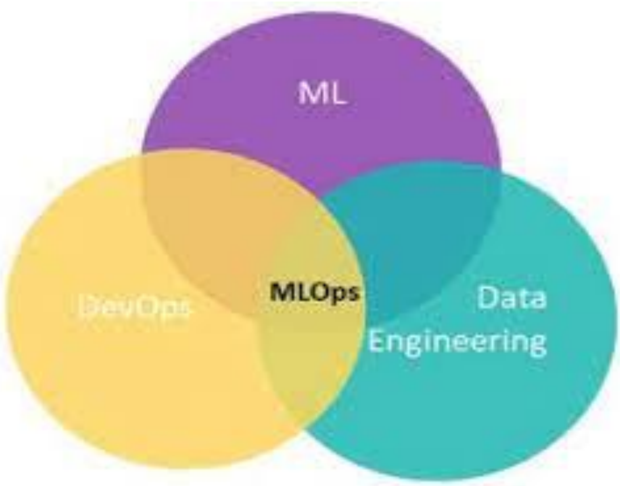
Raguram

Introduction MLOps:

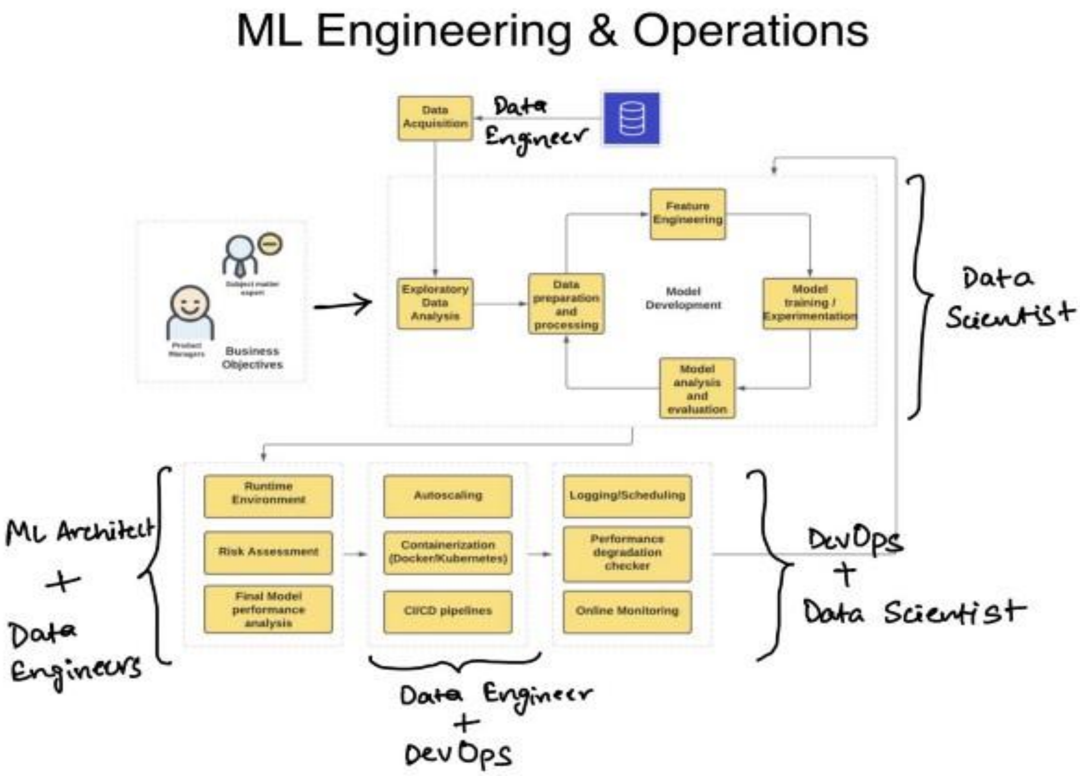
- ✓ MLOps is an abbreviation for Machine Learning Operations.
- ✓ It is a basic component of machine learning engineering that focuses on optimizing the process of deploying machine learning models and subsequently maintaining and monitoring them.
- ✓ Also, it is a valuable technique for developing and improving the quality of machine learning and AI solutions.
- ✓ By integrating continuous integration and deployment (CI/CD) procedures with adequate monitoring, validation, and governance of ML models, data scientists and machine learning engineers may cooperate and accelerate model development and production by using an MLOps strategy.

What is MLOps?

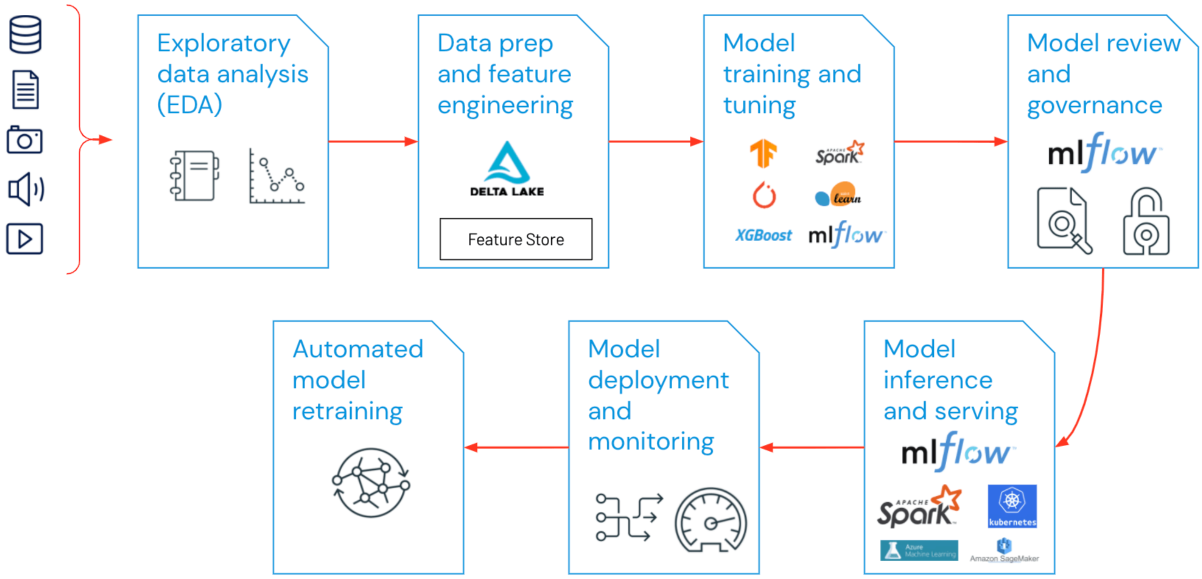
- ❖ MLOps or ML Ops is a set of practices that aims to deploy and maintain machine learning models in production reliably and efficiently using machine learning, DevOps & Data Engineering (OR) It is the process of taking an experimental ML model into a production system.
- ❖ Model creation must be scalable, collaborative, reproducible. The principles, tools and techniques that make model scalable, collaborative, reproducible are known as MLOps.



A complete walkthrough of ML systems development lifecycle and need for MLOps



Components of MLOps:



Let’s try to understand all the component in detail:

Exploratory data analysis (EDA) — Create repeatable, editable, and shareable datasets, tables, and visualizations to iteratively explore, share, and prepare data for the machine learning lifecycle.

Data Preparation and Feature Engineering — Transform, consolidate, and de-duplicate data iteratively to develop enhanced features. Most importantly, to make features accessible and shareable across data teams.

Model training and tuning — To train and enhance model performance, use popular open-source tools like scikit-learn, TensorFlow, and PyTorch. Use automated machine learning techniques like AutoML to execute trial runs and generate reviewable and deployable code as a more straightforward option.

Model review and governance — entails tracking model lineage and versions and managing model objects and transitions throughout their existence. Using an open-source MLOps platform like MLflow, you can discover, share, and collaborate across ML models.

Model inference and serving — Control the frequency of model refresh, inference request timings, and other testing and QA-specifics. To automate the pre-production workflow, use CI/CD technologies.

Model deployment and monitoring — Automate permissions and cluster building to make registered models’ production-ready. Allow REST API model endpoints to be enabled.

Automated model retraining — Create warnings and automation to take remedial action in model drift due to discrepancies in training and inference data.

MLOps Maturity Models:

In January 2020, both Microsoft and Google released what they referred to as 'maturity' models for MLOps.

Microsoft's model is the most fine-grained in terms of how it distinguishes progress through the levels. It outlines five separate levels (starting at zero) that can be used to understand the extent of MLOps adoption and maturity in an organization.

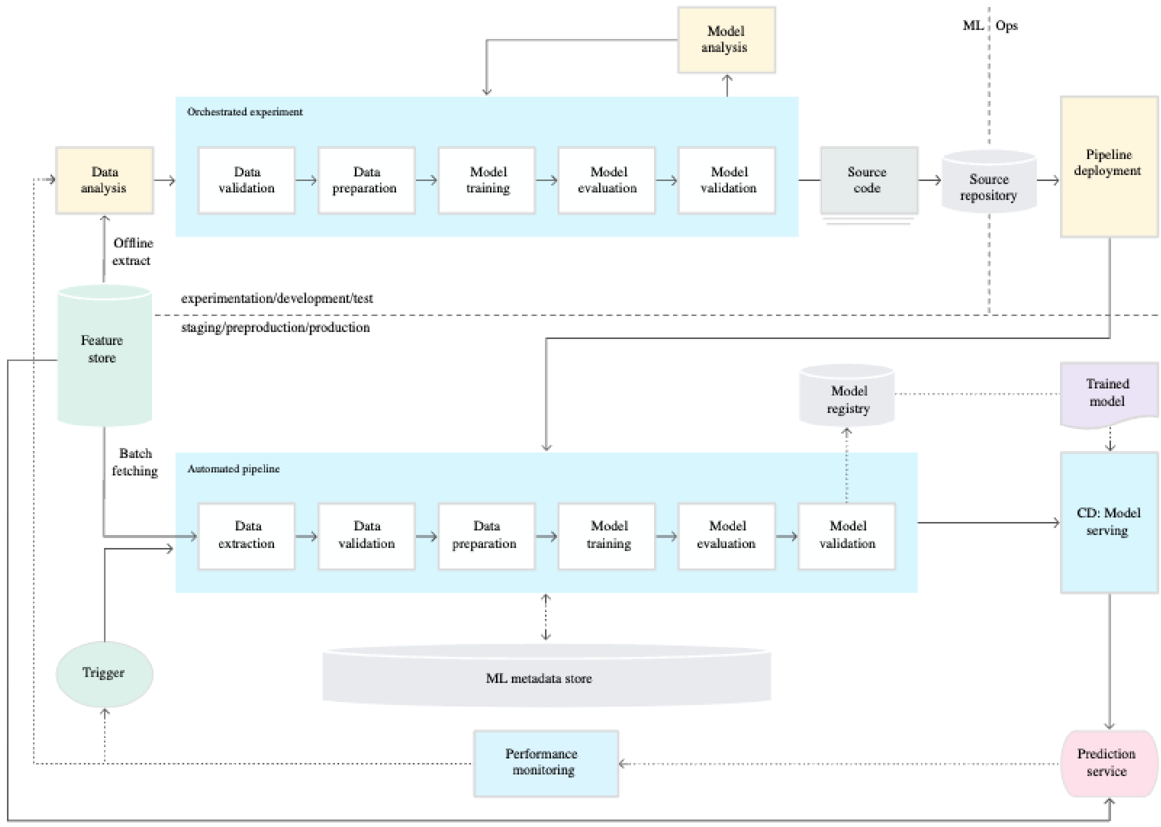
Maturity Level	Training Process	Release Process	Integration into app
Level 0 - No MLOps	Untracked, file is provided for handoff	Manual, hand-off	Manual, heavily DS driven
Level 1 - DevOps no MLOps	Untracked, file is provided for handoff	Manual, hand-off to SWE	Manual, heavily DS driven, basic integration tests added
Level 2 - Automated Training	Tracked, run results and model artifacts are captured in a repeatable way	Manual release, clean handoff process, managed by SWE team	Manual, heavily DS driven, basic integration tests added
Level 3 - Automated Model Deployment	Tracked, run results and model artifacts are captured in a repeatable way	Automated, CI/CD pipeline set up, everything is version controlled	Semi-automated, unit and integration tests added, still needs human signoff
Level 4 - Full MLOps Automated Retraining	Tracked, run results and model artifacts are captured in a repeatable way, retraining set up based on metrics from app	Automated, CI/CD pipeline set up, everything is version controlled, A/B testing has been added	Semi-automated, unit and integration tests added, may need human signoff

Google's assessment of 'maturity' in MLOps also relates to automation.

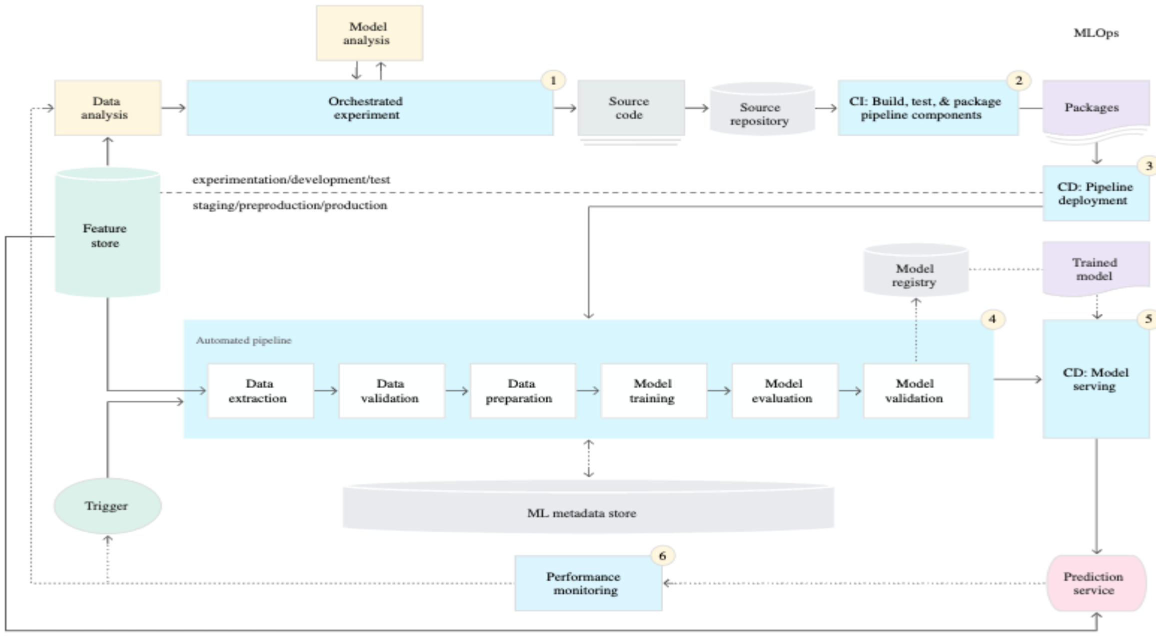
To Google, MLOps is therefore about how agile you can be when you're going through your process. The real challenge, as they see it, "isn't building an ML model, the challenge is building an integrated ML system and to continuously operate it in production".

Level 1 focuses on the automation of the whole machine learning workflow and work cycle for a single pipeline.

At this level, everything is completely automated, including the provision for retraining the entire pipeline when needed, full validation across data, code and models and so on.



Level 2 takes this one step further, abstracting another level higher and includes automation for many pipelines. It's quite similar to level one, just with the ability to handle many models being trained, deployed and handled at the same time.



Similarities of MLOps and DevOps (CI\CD)

- ✓ Continuous integration (CI) of code base amongst developers, data scientists, and data engineer.
- ✓ Testing of code and components of the machine learning system code.
- ✓ Continuous delivery (CD) of the system into production.

Differences between MLOps and DevOps (CI/CD/CT)

- ✓ In MLOps, addition to testing the code you also need to ensure data quality is maintained across the machine learning project life cycle.

- ✓ Deployment of a machine learning system can require a machine learning pipeline that involves data extraction, data processing, feature engineering, model training, model registry and model deployment.

- ✓ In MLOps, there is a third concept that does not exist in DevOps which is continuous training (CT) which indicates automatically identifying scenarios/events that requires a model to be re-trained and re-deployed into production due to a performance degradation in the currently deployed machine learning model/system. **Example**: Feedback Loop

Challenges addressed by MLOps:

- ✓ **Versioning** - Tools such as Git and Github are used in code version control. Also, data and artifacts are versioned to ensure reproducibility.

- ✓ **Model tracking** - Models in production can be degraded over time due to data drift.

- ✓ **Feature Generation** - It requires a lot of resources. MLOps allows to reuse functions. So, you can focus on the design/test of the model.

Different tools to assists MLOps:

- ✓ **MLflow**: This is a open source platform that aids and assists in the ML model management with model tracking, model registry and model deployment steps.

- ✓ **KubeFLow**: It is a platform for data scientists who want to build and experience with ML pipeline. It is also for ML engineers and operational teams who want to deploy ML systems to various environment for developments, testing and production-level serving.

- ✓ **Version control system such as**:
  Git hub - code version control
  Gitlab - build a CI/CD pipeline
  DVC & CML - Data version control, continuous machine learning

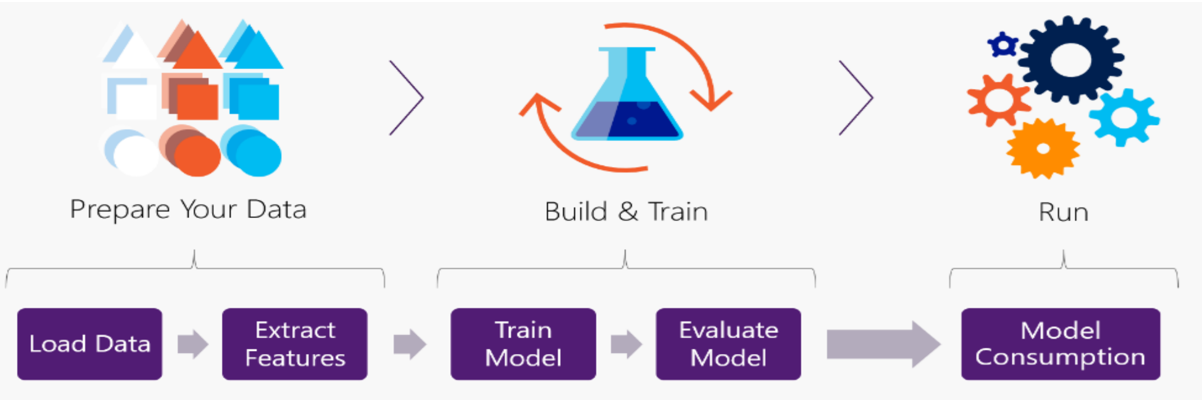
- ✓ **Cloud service** to conduct experiments and deploy the ML model pipelines:
  - AWS Segamaker
  - AZURE Machine Learning
  - Databricks

MLOps stages:

Let’s understand how to perform the action in each stage:

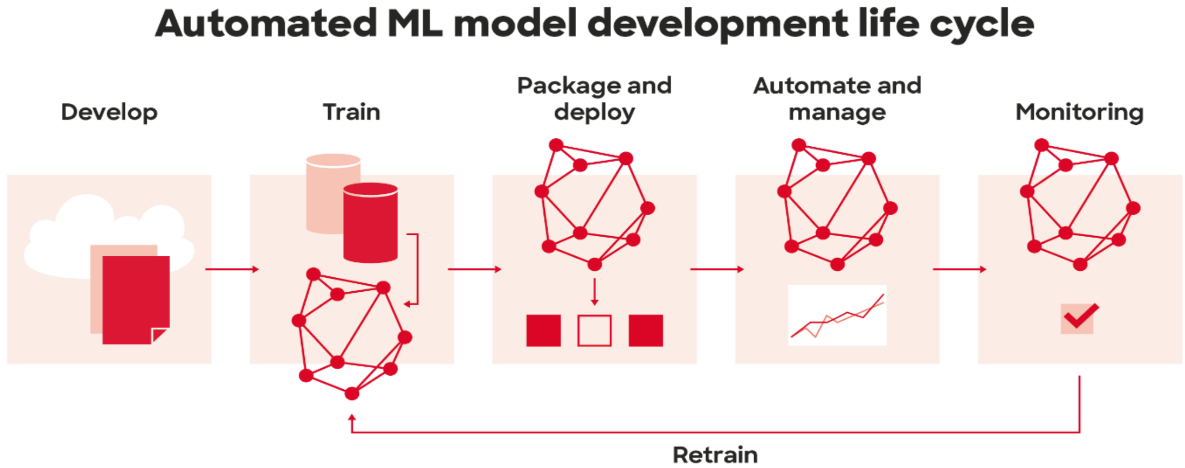
Stage1: Data collection and preparation

- ✓ There is no ML without data.
- ✓ ML teams need access to historical and /or online data from multiple sources.
- ✓ They must catalog and organize this data. Raw data cannot be used, they need to process this data.



Stage 02: Automated deployment

Model deployment generally follows the same process.



Note - All runs, along with their data, metadata, code, and results must be versioned and logged.

Stage03: Create ML service

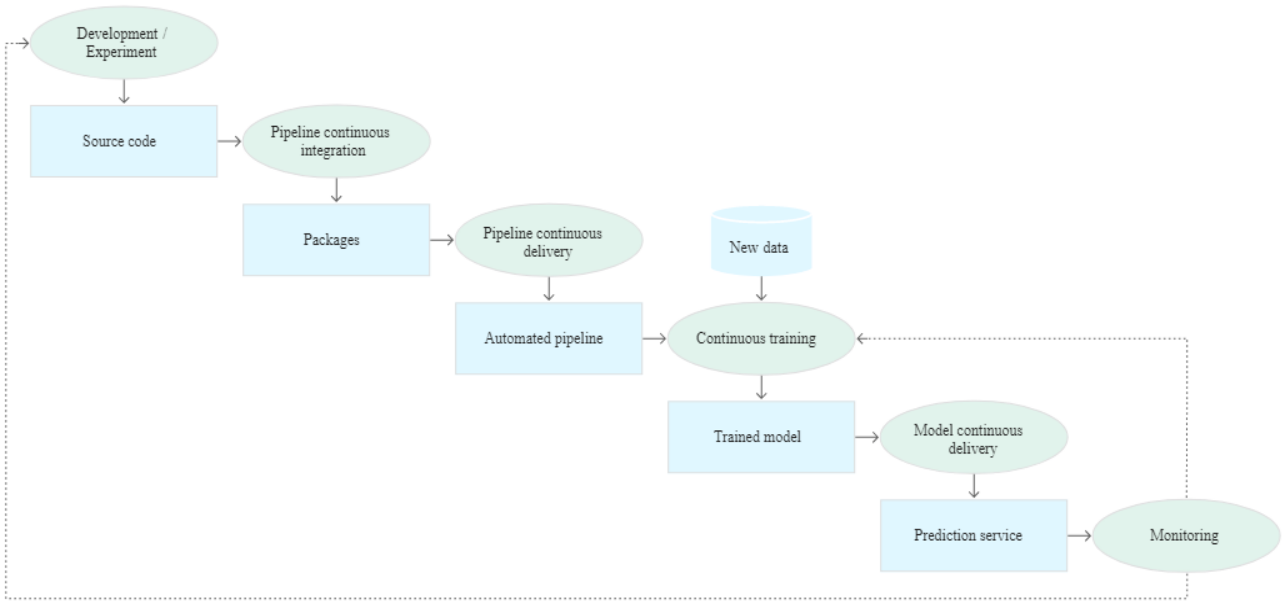
Once a model has been created, it must be integrated with the business application or front-end services. They must be implemented without interrupting service.

Production pipeline implement

- ✓ Real-time data collection, data validation and feature engineering.
- ✓ API services or application integration
- ✓ Data and model monitoring services
- ✓ Resource monitoring and alert services
- ✓ Telemetry and event logging services

Stage04: Monitoring, Governance and Retraining

Model monitoring is the core component of MLOps to keep model up-to-date and predicting with maximum accuracy. It guarantees the validity of the model in the long term.



Thanks for reading this article ☺