WEEK_9_DAY_4_MORNING

MLOps

- Overview

- Why MLOps?

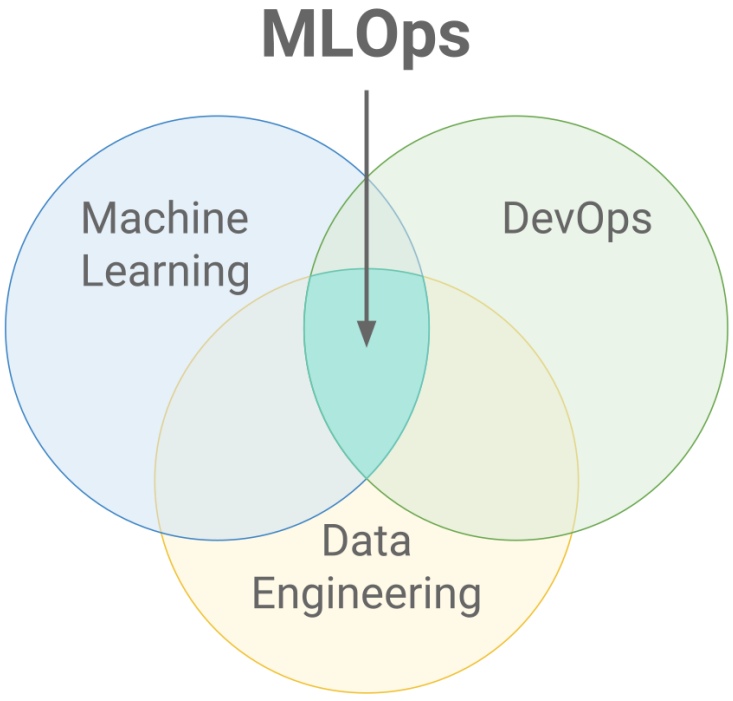
- ML pipeline

- Versioning

- Model registry

What is MLOps?

- MLOps stands for Machine Learning Operations.
- MLOps is a core function of Machine Learning engineering, focused on streamlining the process of taking machine learning models to production, and then maintaining and monitoring them.
- MLOps is a collaborative function, often comprising data scientists, devops engineers, and IT.
- MLOps can encompass everything from the data pipeline to machine learning model production.



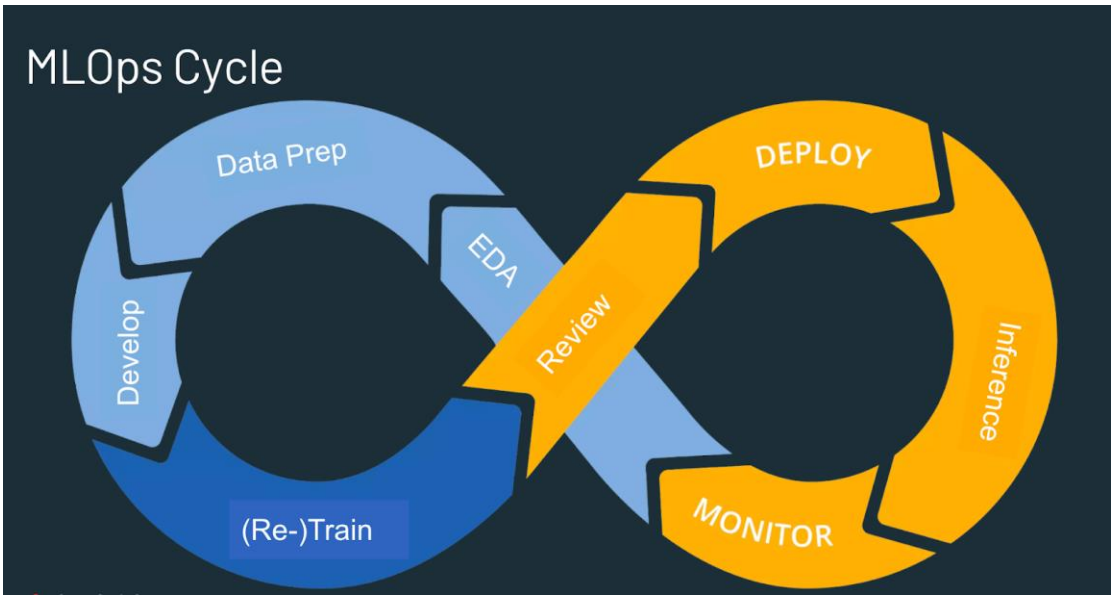
Why MLOps?

Productionizing machine learning is difficult.

The machine learning lifecycle consists of many complex components such as data ingest, data prep, model training, model tuning, model deployment, model monitoring, explainability, and much more.

It also requires collaboration and hand-offs across teams, from Data Engineering to Data Science to ML Engineering.

Naturally, it requires stringent operational rigor to keep all these processes synchronous and working in tandem. MLOps encompasses the experimentation, iteration, and continuous improvement of the machine learning lifecycle.



The primary benefits of MLOps

Efficiency: MLOps allows data teams to achieve faster model development, deliver higher quality ML models, and faster deployment and production.

Scalability: MLOps also enables vast scalability and management where thousands of models can be overseen, controlled, managed, and monitored for continuous integration, continuous delivery, and continuous deployment. Specifically, MLOps provides reproducibility of ML pipelines, enabling more tightly-coupled collaboration across data teams, reducing conflict with devops and IT, and accelerating release velocity.

Risk reduction: Machine learning models often need regulatory scrutiny and drift-check, and MLOps enables greater transparency and faster response to such requests and ensures greater compliance with an organization's or industry's policies.

ML pipeline:

A Machine Learning pipeline is a process of automating the workflow of a complete machine learning task.

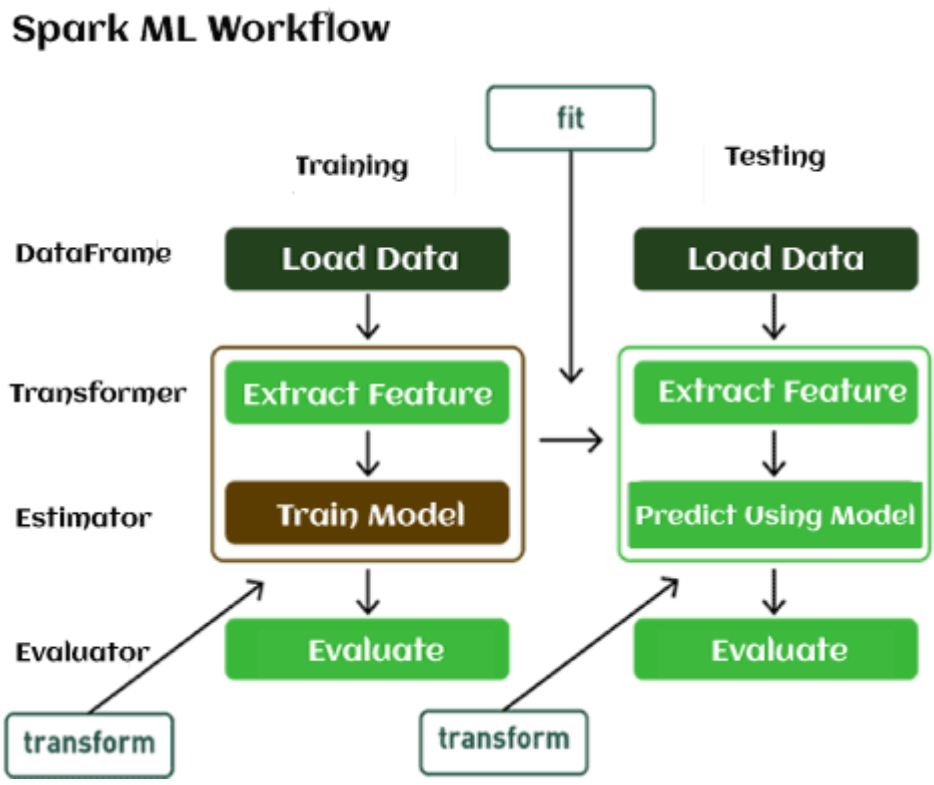
It can be done by enabling a sequence of data to be transformed and correlated together in a model that can be analyzed to get the output.

A typical pipeline includes raw data input, features, outputs, model parameters, ML models, and Predictions.

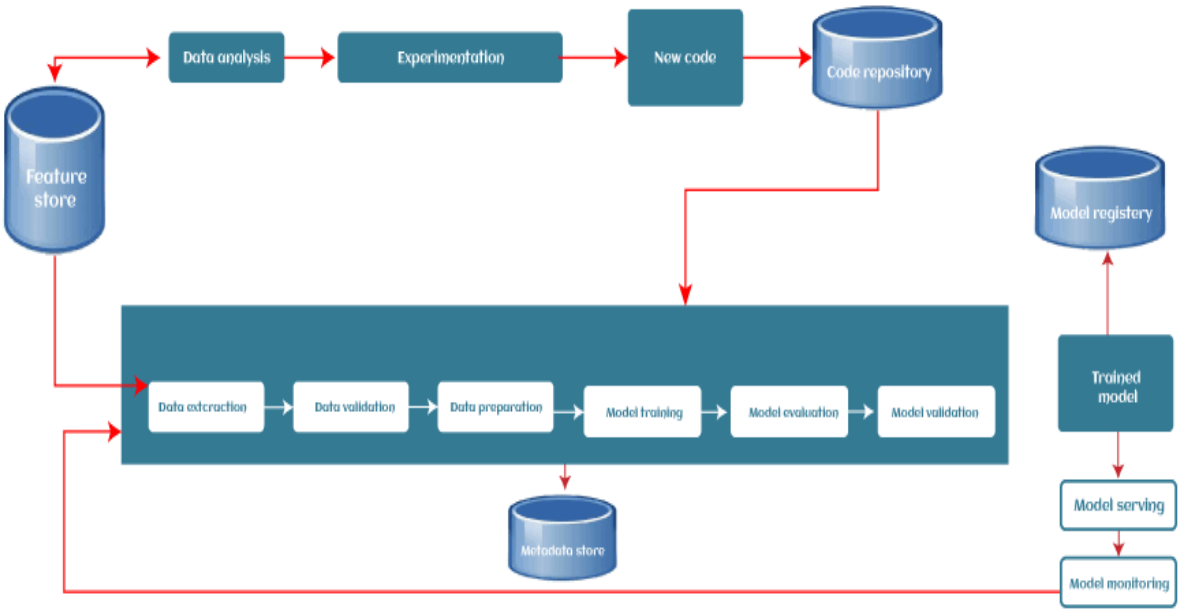
ML Pipeline contains multiple sequential steps that perform everything ranging from data extraction and pre-processing to model training and deployment in Machine learning in a modular approach.

in the pipeline, each step is designed as an independent module, and all these modules are tied together to get the final result.

A typical pipeline contains various stages. However, there are two main pipeline stages:



1. **Transformer:** It takes a dataset as an input and creates an augmented dataset as output. For example, A tokenizer works as Transformer, which takes a text dataset, and transforms it into tokenized words.
2. **Estimator:** An estimator is an algorithm that fits on the input dataset to generate a model, which is a transformer. For example, regression is an Estimator that trains on a dataset with labels and features and produces a logistic regression model.



1. Data Ingestion

Each ML pipeline starts with the Data ingestion step. In this step, the data is processed into a well-organized format, which could be suitable to apply for further steps.

2. Data Validation

Data validation focuses on statistics of the new data, e.g., range, number of categories, distribution of categories, etc. In this step, data scientists can detect if any anomaly present in the data.

3. Data Pre-processing

The pre-processing step involves preparing the raw data and making it suitable for the ML model. The process includes Data cleaning, feature scaling, etc. The product or output of the data pre-processing step becomes the final dataset that can be used for model training and testing.

4. Model Training & Tuning

In this step, the model is trained to take the input (pre-processed dataset) and predicts an output with the highest possible accuracy.

However, there could be some difficulties with larger models or with large training data sets. So, for this, efficient distribution of the model training or model tuning is required. This issue of the model training stage can be solved with pipelines as they are scalable, and a large number of models can be processed concurrently.

5. Model Analysis

After model training, need to determine the optimal set of parameters by using the loss of accuracy metrics. an in-depth analysis of the model's performance is crucial for the final version of the model.

6. Model Versioning

The model versioning step keeps track of which model, set of hyperparameters, and datasets have been selected as the next version to be deployed.

7. Model Deployment

After training and analyzing the model, it's time to deploy the model. An ML model can be deployed in three ways, which are:

- Using the Model server,
- In a Browser
- On Edge device

8. Feedback Loop

Each pipeline forms a closed-loop to provide feedback. With this close loop, data scientists can determine the effectiveness and performance of the deployed models. This step could be automated or manual depending on the requirement. **Except for the two manual review steps (the model analysis and the feedback step), we can automate the entire pipeline.**

Versioning

The use of code versioning tools is vital in the software development industry. The possibility of replicating the same code base so that several people can work on the same project simultaneously is a great benefit. In addition, versioning these bases allows them to work in different sections in an organized manner and without compromising the integrity of the code in production.

In a machine learning project, data scientists are continuously working on the development of new models. This process relies on trying different combinations of data, parameters, and algorithms. It's extremely positive to create an environment where it's possible to go back and forth on older or new experiments.

Model registry

The Model Registry is a system that allows machine learning engineers and data scientists to publish, test, monitor, govern and share them for collaboration with other teams. Essentially, the model registry is used when we done with our experimentation phase, and ready to share with the team and stakeholders.