01_Introduction_to_AIOps

Introduction

Objectives

1. **MLOps** is an ML engineering culture and practice that aims at unifying
   - ML system development (Dev)
   - ML system operations (Ops)

2. Techniques for implementing and automating
   - CI: Continuous Integration-
   - CD: Continuous Delivery- put it into production service
   - CT: Continuous Training for ML System - Dynamic model creation

3. MLOps advocates for-
   - Automation

- monitoring

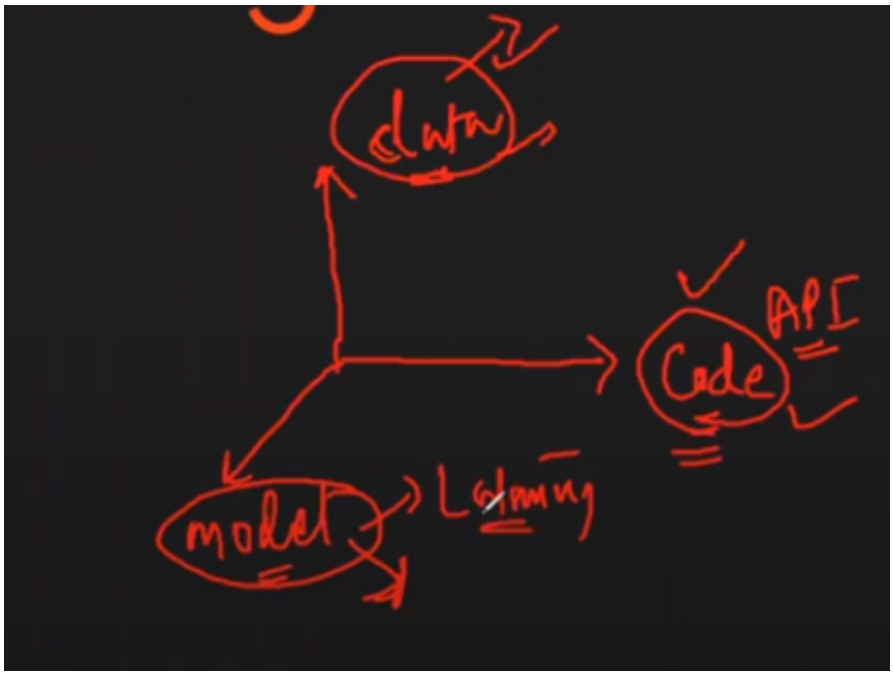
at all stages of ML system process includes

- Integration
- Testing
- releasing
- deployment
- infrastructure management

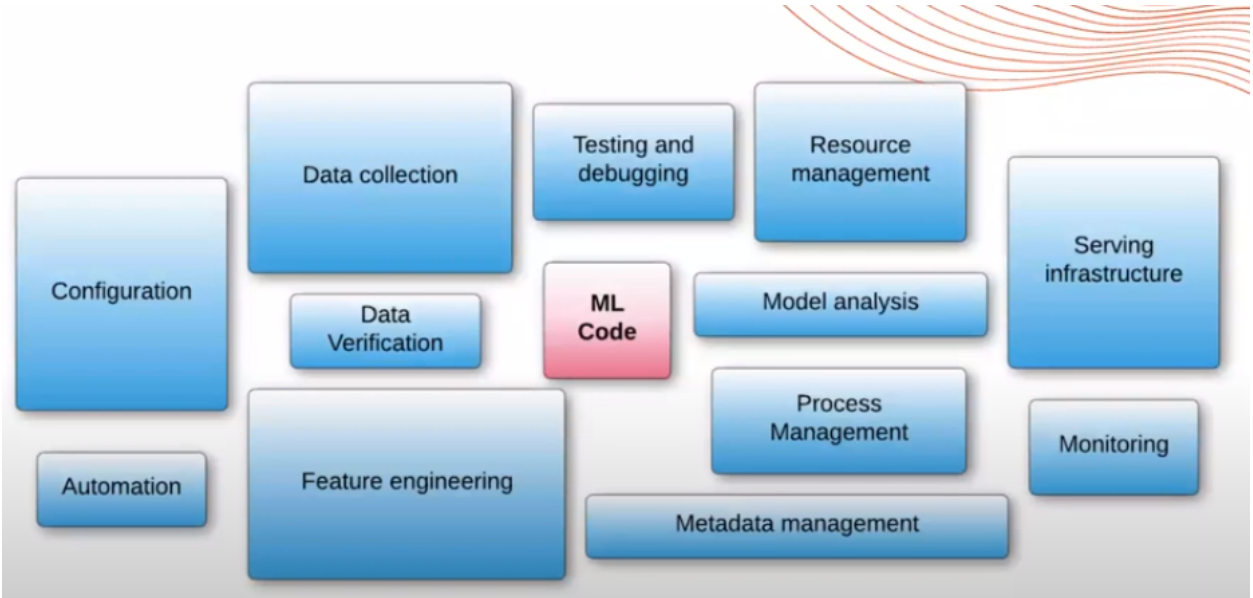
Challenges

1. Building an integrated ML system and continuously operate it in production with a vast array of surrounding infrastructure
2. To automate the process from beginning to end while managing-
   - different teams
   - using different technologies and
   - follow different routines
   - And also make them-
     - auditable
     - reproducible
3. Dependencies-
   - data dependency
   - model complexity
   - reproducibility
   - testing
   - monitoring

   These changes and dependencies in addition to code must be controlled and integrated into the software delivery process.



Hidden Technical Debt in ML System



Sculley et. al. 2015

DevOps Vs MLOps

DevOps

Development and operations of large scale software systems. DevOps is a widespread approach. Shortening the development cycle, boosting the deployment velocity and ensuring the reliable releases are all advantages of this strategy. It uses two concepts in software system development to get these benefits:

1. **Continuous Integration-** Can push the change at any point in time
2. **Continuous Delivery** -

Key Differences

MLOps:

1. **CI**: no longer about testing and validating **code and components, data, data schema, and models** must also be tested and validated.
2. **CD**: No longer about a single software package or service but about a system (an ML training pipeline) that automatically deploys another service(model prediction service).
3. **CT**: It is a novel attribute specific to a machine learning systems that deal with automatically retraining and servicing the models.
    a. If CT is not there then data science team will always need to be involved.
    b. Based on logs we infer if the model is performing well if not, then get DS team involved.
    c. When model performance is degraded then the CT is triggered.

Steps in AI/MLOps

Generic Steps

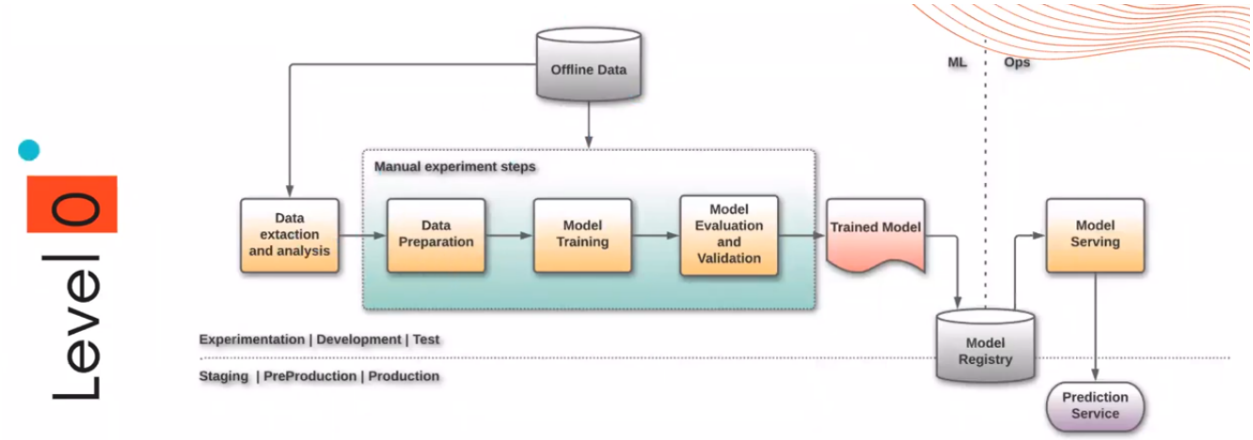
1. Data Extraction
2. Data Analysis
3. Data Preparation
4. Model Training

5. Model Evaluation
6. Model Validation
7. Model Serving- REST API
8. Model monitoring- whether model is performing well?

Workflow

LEVELS OF MATURITY

1. **LEVEL 0: ML System Deployment : Mostly manual process**



- Basic level of Maturity
- Manual Training, Building, Deployment etc

  Characteristics:

  - Manual Script driven process
  - Disconnection between ML team and operations team.
    - May lead to training and serving issue.
  - No continuous delivery process of models

- CI is also ignored most of the times.
- Deployment here means just the prediction service. —> No deployment of ML pipeline?
- Lack of active performance monitoring

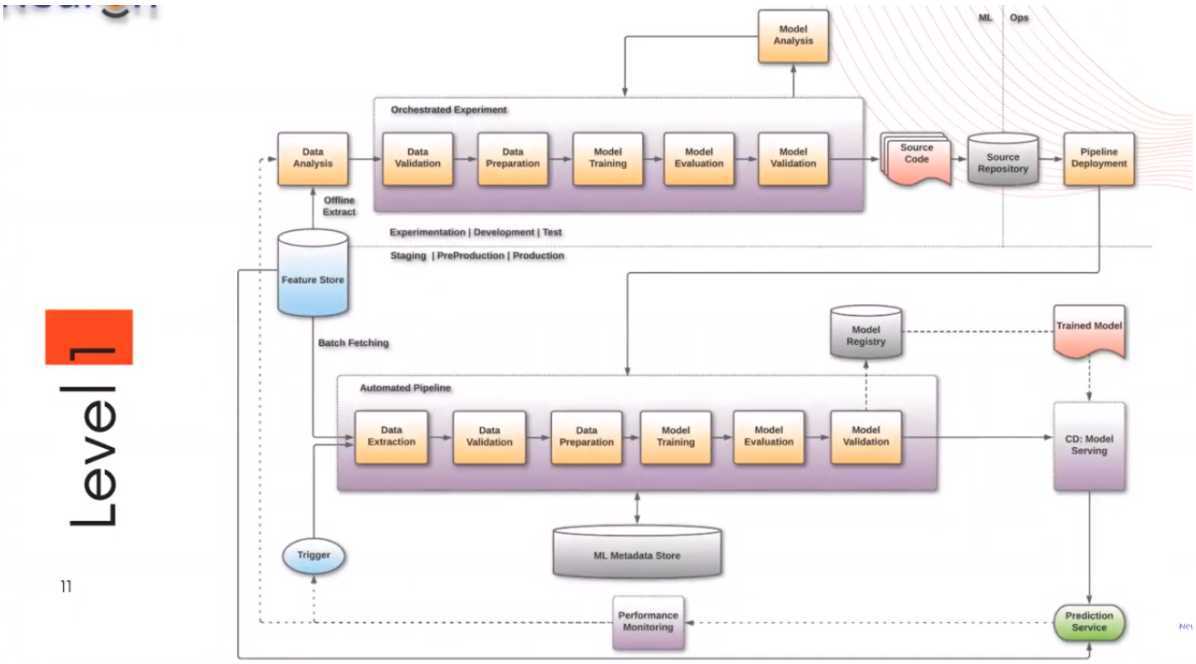
Challenges:

- Latency
- no frequent releases
- Data Science team intervention
- Teams working in silos
- Load Balancing
- Model Degradation
- Cost?

Possible Solutions:

- Active monitoring the quality of model in production.
- Frequent retraining the model in production on new data.
- Continuous experimentation with the new implementation to produce the model: Timeline Building
- Rigorous Testing of each module: CI/CD/CT is required here.

2. **LEVEL 1: ML pipeline Automation**



- Feature Store: Common store of data where we can find the schema and data- used for all the purpose training and testing
- Entire Pipeline is source code that can be pushed to git/source repository and entire pipeline is deployed in pre-production environment
- Pre-production pipeline fetch new data called batch fetching.
- Prediction Service —> Performance monitoring → logs → if there is performance degradation then retrain the model again.
- ML Metadata store: how many times the model is triggered and related metadata

- Orchestration: Flow of environment
  - Tools like : Apache Airflow/ KubeFlow
  - Create the pipeline using these tools.
  - DAG: Directed Acyclic Graphs: Step by Step Orchestration.
  - [Covered in later notes]

Characteristics

- Rapid Experiment
- CT of model in production
- Experimental Operational symmetry- pipeline implementation is same in both development and production environment
- Modular code for components and pipeline —> Reusability!
- Continuous delivery of the model is achieved