Machine Learning Operations (MLOps)

Overview, Definition, and Architecture

Abolfazl Yarian

Spring 2023

Outline

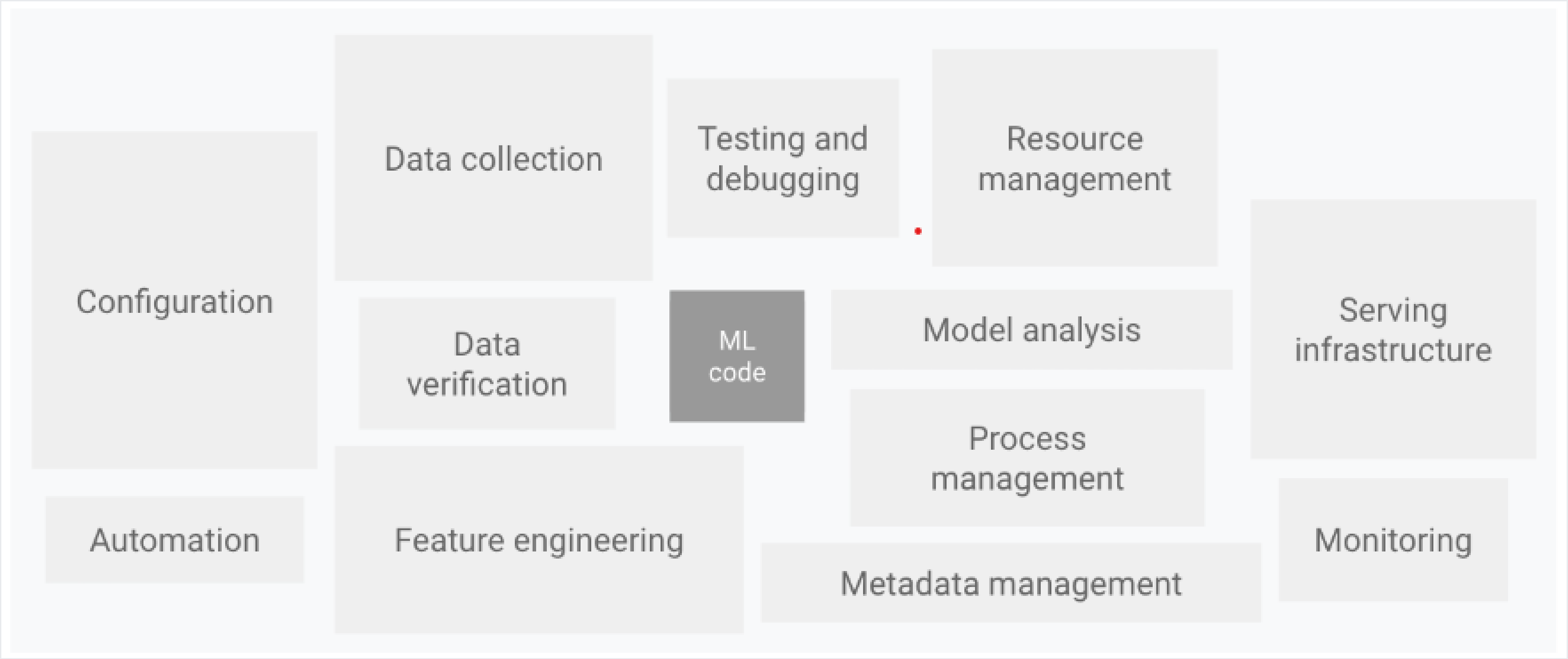
1. Introduction
2. Data science Overview
3. MLOps definition
4. Architecture
5. Open-source tools
6. References

Data science steps for ML

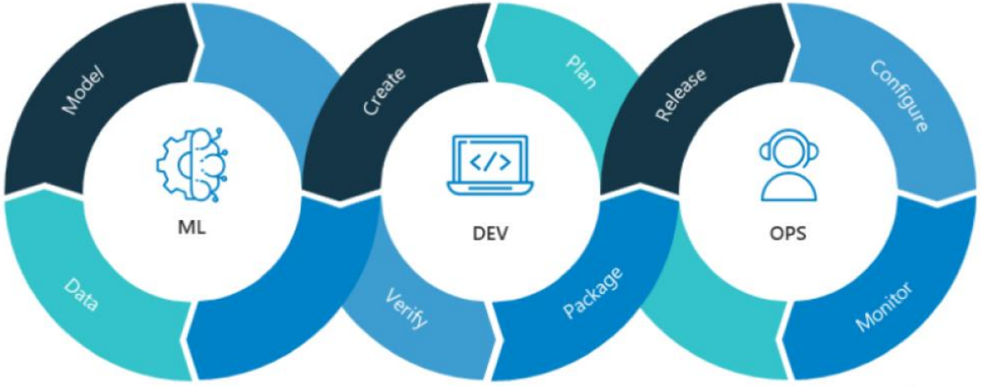
- Data extraction
- Data analysis
- Data preparation
  - Data cleaning
  - Data splitting
  - Transformation and feature engineering

Data science steps for ML

- Model training
  - Implement different algorithm
  - Hyperparameter tuning
- Model evaluation/validation
- Model serving
  - Microservices
  - Edge device
- Model monitoring


Configuration
Data collection
Testing and debugging
Resource management
Serving infrastructure
Data verification
ML code
Model analysis
Process management
Automation
Feature engineering
Metadata management
Monitoring

MLOps



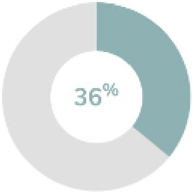
MLOps (Machine Learning Operations) refers to the practice of applying DevOps (Development Operations) principles to the machine learning workflow. It involves a set of processes, tools, and techniques to build, deploy, monitor, and manage machine learning models in production environments

MLOps

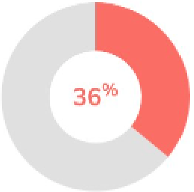
- MLOps manages and automates the end-to-end lifecycle of machine learning models.
- Combines DevOps and data science to streamline development, deployment, and monitoring.
- Improves collaboration, efficiency, and scalability by standardizing tools, processes, and infrastructure.
- Enables continuous integration and delivery, ensuring reliability, security, and performance in production.
- Involves monitoring, testing, and updating ML models to remain accurate and relevant.
- Accelerates innovation, reduces costs, and improves customer satisfaction.

Responses from 582 survey respondents

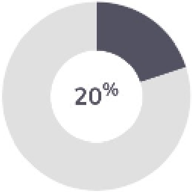
What percentage of your data scientists' time is spent deploying ML models?



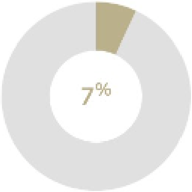
36% of survey participants said their data scientists spend **a quarter** of their time deploying ML models



36% of survey participants said their data scientists spend **a quarter to half** of their time deploying ML models



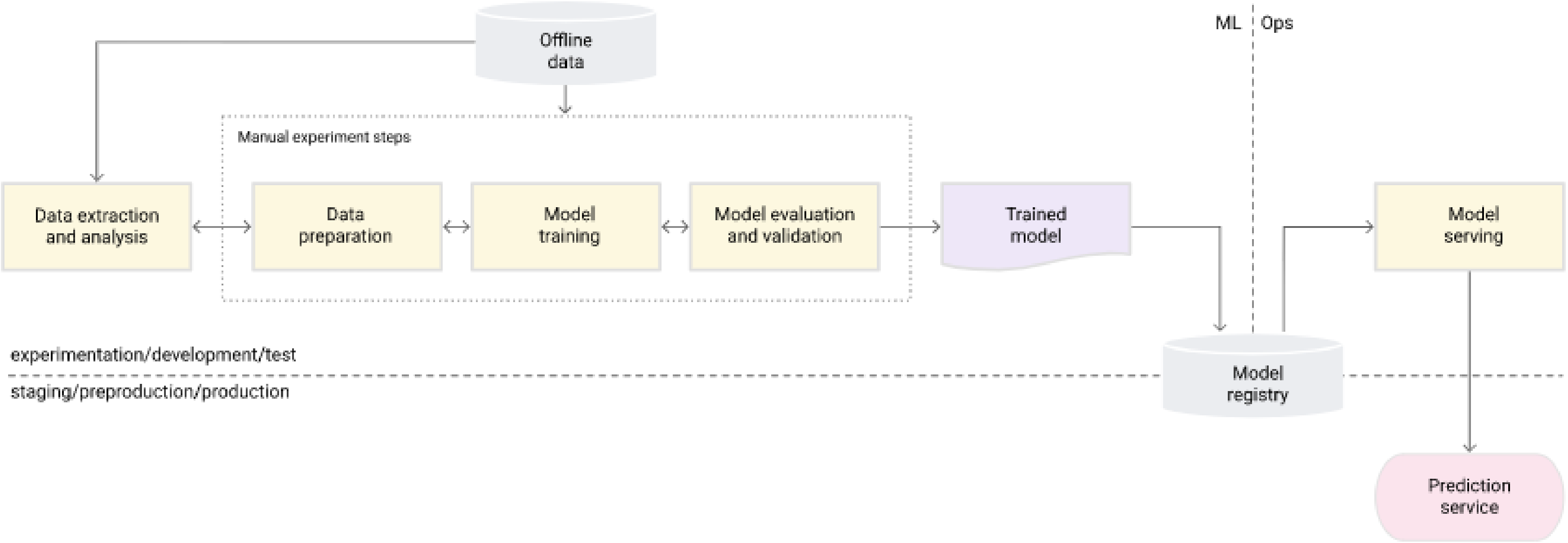
20% of survey participants said their data scientists spend **half to three-quarters** of their time deploying ML models



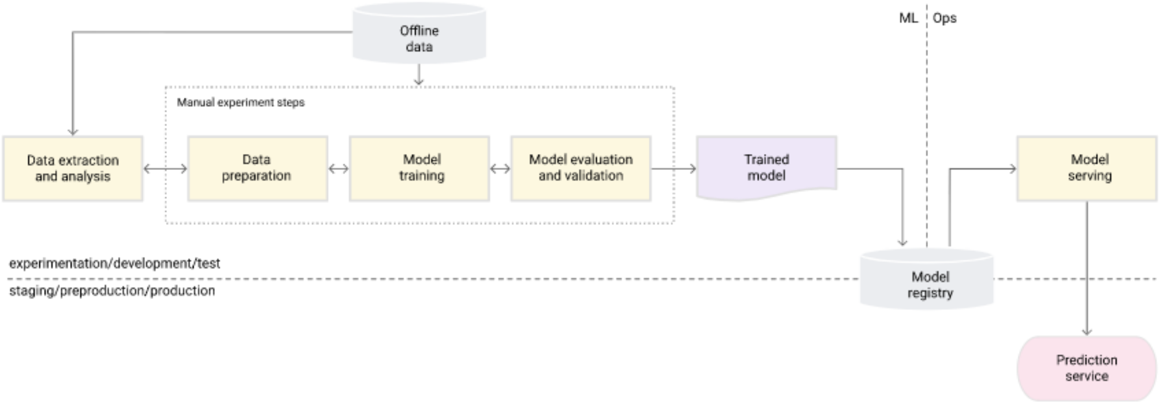
7% of survey participants said their data scientists spend **more than three-quarters** of their time deploying ML models

1% of respondents said they were unsure.

MLOps level 0: Manual process



Characteristics



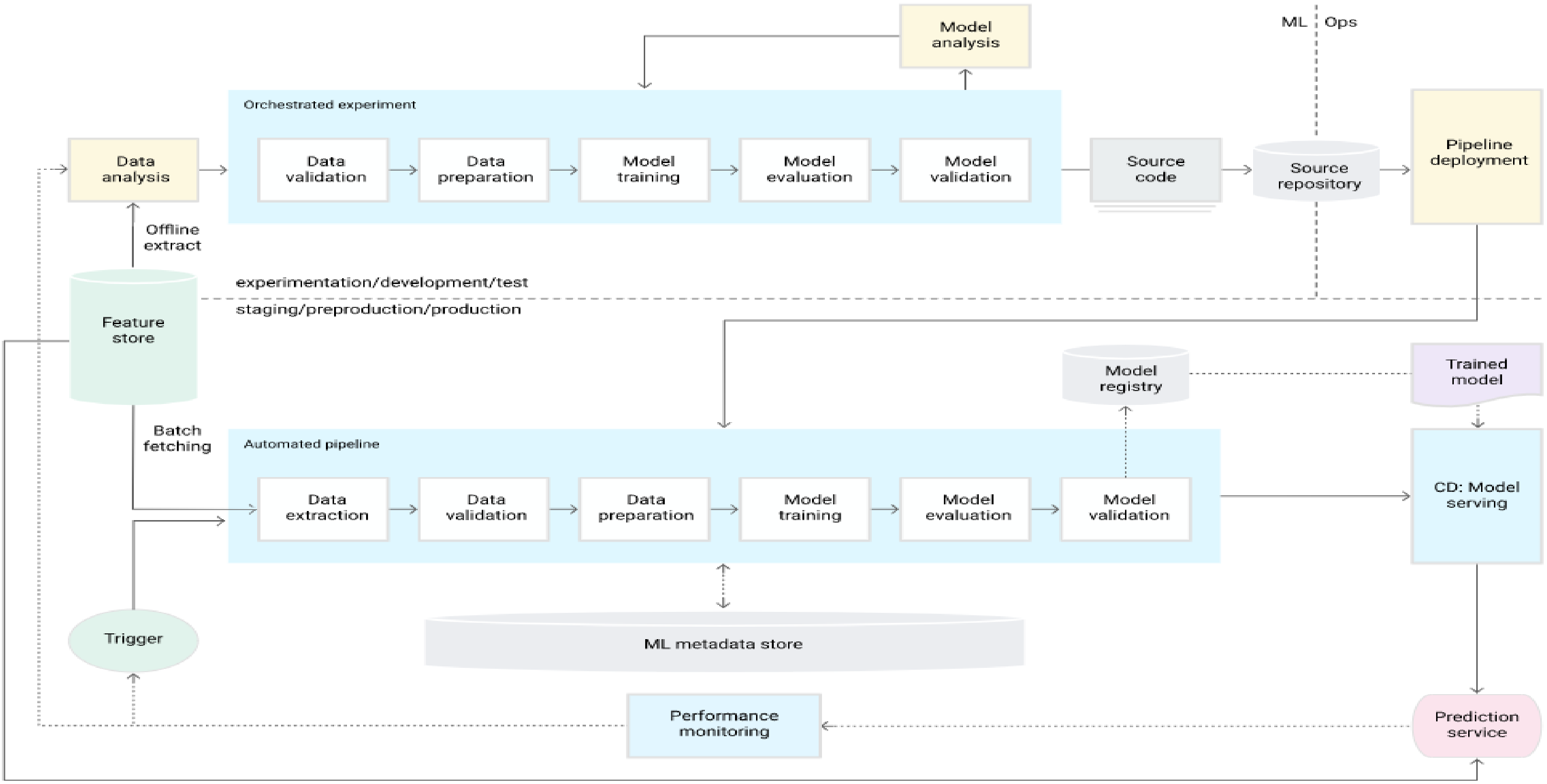
- Manual, script-driven, and interactive process
- Disconnection between ML and operations
- Infrequent release iterations
- No CI/CD
- Deployment refers to the prediction service
- Lack of active performance monitoring

Challenges

- Maintain model's accuracy in production
  - Actively monitor the quality of your model in production
  - Frequently retrain your production models
  - Continuously experiment with new implementations to produce the model

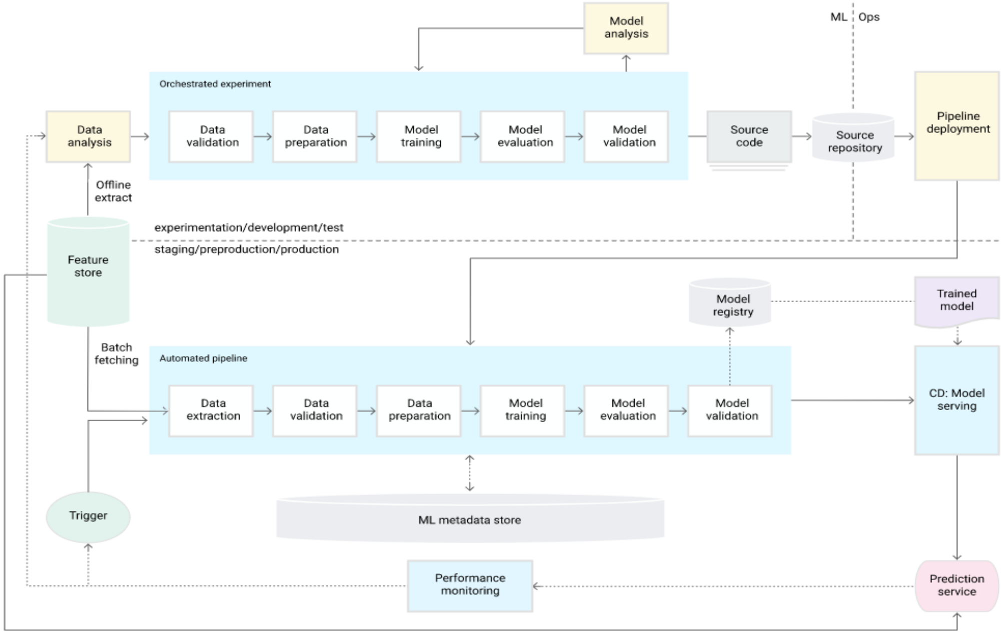
- Set up CT/CI/CD to rapidly test, build and deploy new implementation of the ML pipeline

MLOps level 1: ML pipeline automation(CT)

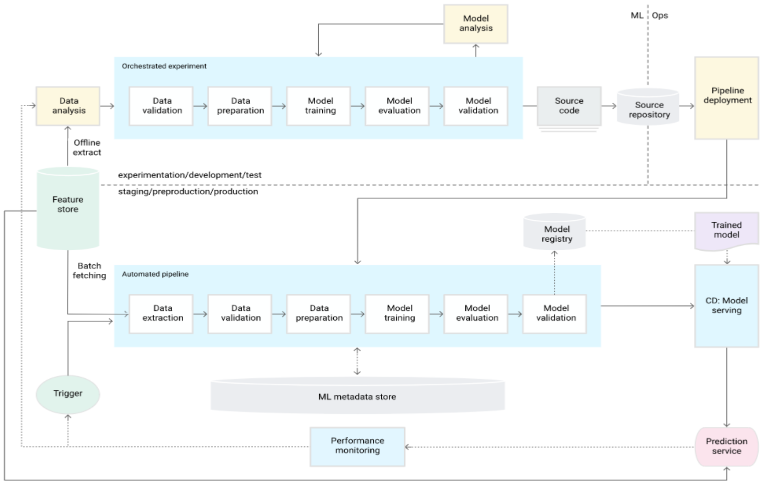


Data validation

- Data schema skews (stop)
  - Unexpected features
  - Unexpected values
  - Lack of all expected features
- Data value skew (retrain)



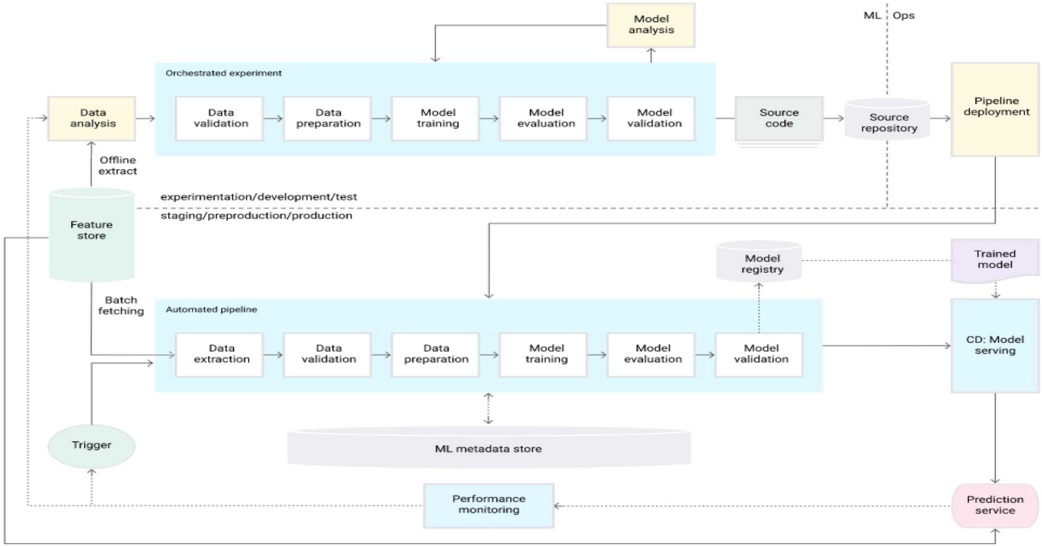
Model validation (offline)



- Evaluate predictive quality on test dataset
- Compare with current model performance
- Check for consistency across data segments
- Test for deployment and infrastructure compatibility
- Conduct online validation through canary or A/B testing

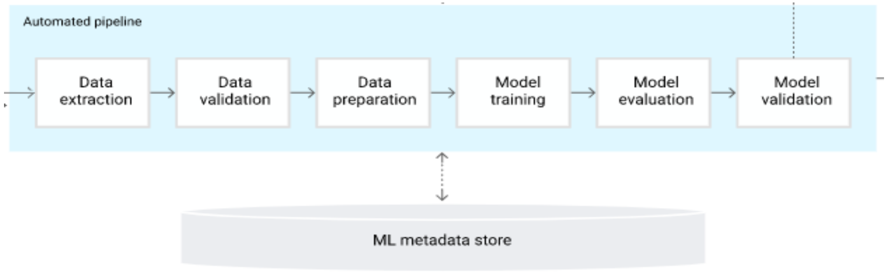
Feature store

- Discover and reuse existing feature sets to avoid duplication
- Serve up-to-date feature values from the feature store.
- Use the feature store for experimentation, CT, and online serving to avoid training-serving skew.
- avoid training-serving skew for:
  - Experimentation (offline)
  - continuous training
  - Online prediction

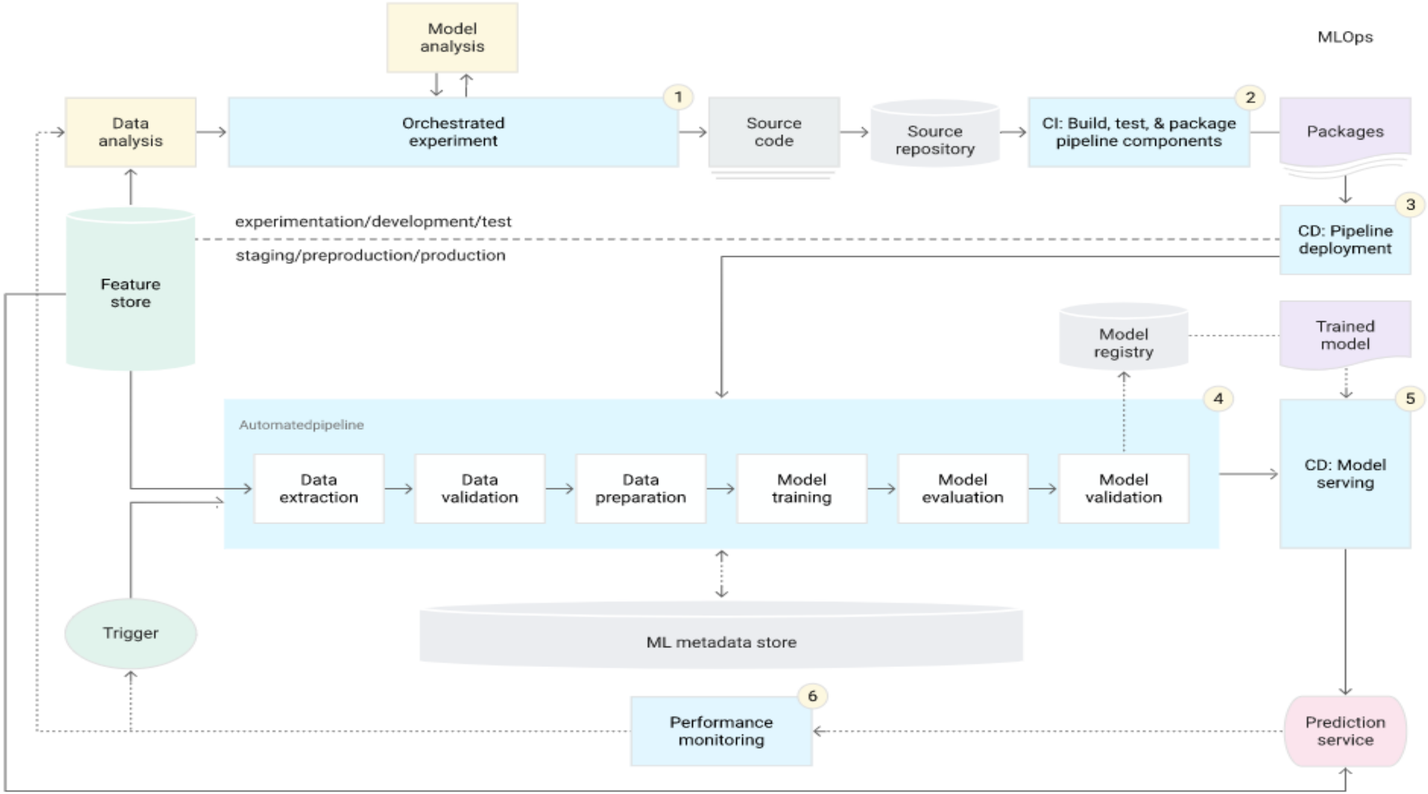


Metadata management

- Record pipeline versions, timestamps, and executor for lineage, reproducibility, and debugging
- Store parameter arguments passed to the pipeline
- Store pointers to the artifacts produced by each step of the pipeline
- Store pointers to previous models and evaluation metrics for comparison

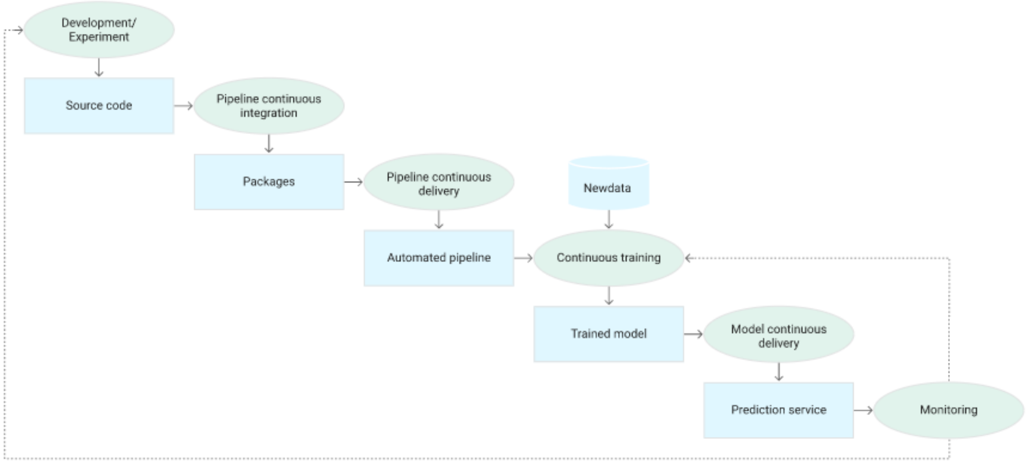


MLOps level 2: CI/CD pipeline automation



Characteristics

- Development and experimentation
- Pipeline continuous integration
- Pipeline continuous delivery/deployment
- Automated triggering
- Model continuous delivery
- Monitoring



Continuous integration

- Unit tests for feature engineering and model methods in implementation
- Tests for model convergence and avoiding NaN values
- Testing that each component in the pipeline produces the expected artifacts
- Testing integration between pipeline components.

Continuous delivery

- Verify the compatibility of the model with the target infrastructure before deployment
- Test the prediction service by calling the service API with expected inputs
- Test prediction service performance by load testing to capture metrics such as QPS and model latency
- Validate data for retraining or batch prediction
- Verify that models meet predictive performance targets before deployment
- Automate deployment to a test environment triggered by code push to development branch

Open-source libraries

MLOps Stage	Open-source Tool	Alternatives
Source Code	Github	Bitbucket
Feature Store	Feast	Hopsworks
ML Pipeline	Kubeflow	Polyaxon
Model Validation Testing/Maintenance	Deepchecks	Etiq AI, Great Expectations
Model Registry	MLflow	Neptune
Model Serving	Cortex	Seldon Core
Model Monitoring	Deepchecks	Prometheus, Grafana

Thanks!

Any questions?