Observability in the MLOps Lifecycle with Prometheus

Shivay Lamba
KubeFlow Maintainer
WASMEdge Ambassador

@howdevelop
MLOps - DevOps Engineer?
SRE - Machine Learning
Reliability Engineering
- Making sure that machine learning infrastructure is highly available, reliable, and meets the service-level agreements (SLAs).
- Setting up a system to proactively monitor compute, memory, network latency, etc.
- Controlling costs of machine learning infrastructure by optimizing design and workflow.
Monitoring

➔ SLOs

➔ System Failures

➔ …
ML Model LifeCycle

- Model Building
- Model Evaluation and Experimentation
- Productionize Model
- Testing
- Deployment
- Monitoring and Observability

- Code
  - Training code
  - Test code
  - Application code

- Model
  - Candidate models
  - Chosen model
  - Productionized model
  - Model
  - Code and model in production

- Data
  - Training data
  - Test data
  - Production data
Monitoring in context of ML

- Different challenges
Monitoring in context of ML

➔ Different challenges
  ○ Model edge cases
Monitoring in context of ML

Different challenges

○ Model edge cases

○ Data distribution has shifted
Monitoring in context of ML

- Different challenges
  - Model edge cases
  - Data distribution has shifted
  - Misconfigured models
Monitoring in context of ML

➔ Different challenges
  ○ Model edge cases
  ○ Data distribution has shifted
  ○ Misconfigured models
➔ Model still makes a prediction
Monitoring in context of ML

➔ Different challenges
  ○ Model edge cases
  ○ Data distribution has shifted
  ○ Misconfigured models

➔ Model still makes a prediction but predictions are not useful
Monitor what?

➔ Model metrics
➔ System metrics
➔ Resource metrics

ML Metrics

Operational - Is it working?
- Latencies
- Memory size
- CPU usage

Are the predictions accurate?
- Model Outputs

Is the data what is expected?
- Model Inputs
Monitor what?

- Model metrics
- System metrics
- Resource metrics
Monitor what?

- Model metrics
- System metrics
  - Request throughput
  - Error rate
  - Request latencies
  - Request body size
  - Response body size
Monitor what?

- Model metrics
- System metrics
- Resource metrics
Monitor what?

➤ Model metrics

➤ System metrics

➤ **Resource metrics**
  - CPU utilization
  - Memory utilization
  - Network data transfer
  - Disk I/O
Monitor what?

➔ Model metrics
➔ System metrics
➔ Resource metrics
Model Drift

- Environment changes affect model
Model Drift

➔ Environment changes affect model
➔ Change in data distribution
Use of Prometheus

One of the most popular open-source stacks for monitoring metrics is the combination of Prometheus and Grafana.

Prometheus scrapes metrics from instrumented jobs, either directly or via an intermediary push gateway for short-lived jobs. It stores all scraped samples locally and runs rules over this data to either aggregate and record new time series from existing data or generate alerts.
Fast API Demo
Create a REST service to expose the model
FastAPI Demo

➔ Create a REST service to expose the model

➔ Instrument the server to collect metrics which are exposed via a separate metrics endpoint

prometheus-fastapi-instrumentator
Fast API Demo

➔ Create a REST service to expose the model

➔ Instrument the server to collect metrics which are exposed via a separate metrics endpoint

    prometheus-fastapi-instrumentator

➔ Deploy Prometheus to collect and store metrics
Create a REST service to expose the model

Instrument the server to collect metrics which are exposed via a separate metrics endpoint

`prometheus-fastapi-instrumentator`

Deploy Prometheus to collect and store metrics

Deploy Grafana to visualize the collected metrics
Fast API Demo

➔ Create a REST service to expose the model

➔ Instrument the server to collect metrics which are exposed via a separate metrics endpoint
  prometheus-fastapi-instrumentator

➔ Deploy Prometheus to collect and store metrics

➔ Deploy Grafana to visualize the collected metrics

➔ Locus to Simulate
What is Seldon?

Seldon Core, an open-source framework, makes it easier and faster to deploy our machine learning models and experiments at scale on Kubernetes. Seldon Core serves models built in any open-source or commercial model building framework.

Seldon Core exposes metrics that can be scraped by Prometheus. The core metrics are exposed by the service orchestrator (executor).
The analytics component is configured with the Prometheus integration. The monitoring for Seldon Deploy is based on the Prometheus Operator and the related PodMonitor and PrometheusRule resources.
Thank You!