1-Month Expert Architecture Training Plan

Week 1, Day 4: MLOps & Model Lifecycle

Prepared by: Sami Mbarki
Solution Architect, Java Expert
ISYGO Consulting Services 18 September 2025

ISYGO Consulting Services
Delivering Advanced Architectural Training for Enterprise Solutions

Document ID: 28f69273-7825-4ce5-af65-034dd404ca6c
Version: 1.0

1-Month Expert Training in Microservices and Polyglot Persistence

Sami Mbarki
Solution Architect, Java Expert
ISYGO Consulting Services

18 September 2025

Contents

1 Preface: Operationalizing Machine Learning

This comprehensive chapter on MLOps turns Day 4 into an educational tome, defining terms like "model serving", clarifying lifecycle stages, and exploring technologies (MLflow, Triton) with alternatives (Sagemaker, Kubeflow). Exercises promote hands-on discovery.

1.1 How to Use This Book

- Diagram lifecycle for visual learning. - Compare tools for decision-making.

2 Introduction: From Models to Production

What is "technical debt" in ML, and how does MLOps mitigate it? MLOps combines ML, DevOps, and data engineering. Historical: Coined in 2015 by Google in "Hidden Technical Debt in Machine Learning Systems". Alternatives: DataOps for data focus.

2.1 Learning Objectives

Including understanding drift detection.

3 The Machine Learning Lifecycle

3.1 Stages in Detail

1. Business Understanding: Define metrics (e.g., F1-score for classification). 2. Data Engineering: Feature engineering (e.g., TF-IDF for documents). 3. Model Development: Hyperparameter tuning with grid search. 4. Evaluation: Bias detection using fairness metrics. 5. Deployment: A/B testing. 6. Monitoring: Detect data drift (change in input distribution) vs concept drift (change in target relationship).

Diagram:

Figure 1: ML Lifecycle with Feedback Loops

Clarification: CI/CD/CT - Continuous Training for retraining.

Pitfalls: Ignoring drift causes model decay.

4 MLOps Principles and Tools

Definition: MLOps is practices for reliable ML deployment.

Terminology: "Feature store" (e.g., Feast) for reusable features.

4.1 MLflow: End-to-End Management

Definition: Open-source platform for tracking, packaging, models.

Components: Tracking (log params/metrics), Projects (packaging), Models (serving), Registry (versioning).

Code Example for Tracking:

Alternatives: DVC for version control, Kubeflow for Kubernetes-native.

Best Practices: Use UI for experiment comparison.

Pitfalls: Poor logging leads to irreproducible models.

5 Model Deployment with Triton

Definition: Triton is NVIDIA's inference server for multi-framework models.

Terminology: "Dynamic batching" - groups requests for GPU efficiency.

Clarification: Supports ONNX, TensorFlow, PyTorch.

Alternatives: TensorFlow Serving (TF-specific), TorchServe (PyTorch-focused).

Configuration Example:

```
name: "document_model"
backend: "tensorrt"
max_batch_size: 16
input: [
  {
    name: "input"
    data_type: TYPE_FP32
    dims: [ -1, 512 ]
  }
]
output: [
  {
    name: "output"
    data_type: TYPE_FP32
    dims: [ -1, 10 ]
  }
]
dynamic_batching: { }
```

Advanced: Model ensemble for chaining (e.g., OCR + extraction).

Case Study: Fraud detection with low-latency inference.

6 Testing ML Pipelines

Definition: "Behavioral testing" verifies model fairness.

Terminology: "Shadow deployment" - test new models in parallel.

Alternatives: Great Expectations for data validation.

7 Lab: MLOps in Practice

Detailed: Train a simple model, log with MLflow, deploy to Triton.

8 Wrap-Up: Socratic Discussion

Discuss: How does concept drift affect our capstone? What alternative to MLflow?

9 Student Exercises and Review Questions

1. Log an experiment in MLflow. 2. Compare Triton and TF Serving.

10 Glossary

Expanded:

- **Data Drift**: Input change over time.
- **Concept Drift**: Target relationship change.
- **ONNX**: Open Neural Network Exchange format.
- **Dynamic Batching**: Request grouping.
- **Model Registry**: Versioned model store.

11 References and Further Reading

Books:

- Sculley, D., et al. "Hidden Technical Debt in Machine Learning Systems." NIPS 2015.
- Baylor, D., et al. "TFX: Production ML Platform." KDD 2017.
- Ameisen, E. *Building ML Powered Applications*. O'Reilly, 2020.

Online:

- MLflow: https://mlflow.org.
- Triton: https://developer.nvidia.com/triton-inference-server.