MLOps Course

Guidelines

The course consists of three parts:

- Quick Review (Tutorials 1 & 4): You should review ML pipeline fundamentals and Docker setup; these topics will be briefly reviewed at the start of Session 1 and 2, respectively
- Hands-On Sessions (3 sessions): MLflow, FastAPI deployment, monitoring, and capstone project
- Bonus Materials (Tutorials 7 & 9): Testing and CI/CD - optional self-learning

1 Requirements

Prerequisites

You should have:

- Solid Python programming skills
- Basic machine learning knowledge (scikit-learn)
- Familiarity with Git version control
- Command line/terminal proficiency
- *Recommended:* Basic Docker knowledge

Tech Stack

- **Experiment Tracking:** MLflow
- **API Framework:** FastAPI
- **Containerization:** Docker & Docker Compose
- **Deep Learning:** PyTorch

2 Theory review

2.1 Tutorial 1: ML Pipeline Fundamentals

Part 1: Basic Pipeline

- Data loading & exploratory data analysis
- Train/test split
- Model training with multiple algorithms
- Manual evaluation & model saving

Part 2: Experiment Chaos

- Run experiments with different hyperparameters
- Manual tracking attempts
- Try to reproduce the best model

2.2 Tutorial 4: Docker Setup

Part 1: Docker Basics Refresher

- Dockerfile fundamentals
- Building and running containers
- Volume mounting & networking

Part 2: PostgreSQL Backend

- Setup PostgreSQL container
- Connecting MLflow to PostgreSQL

Part 3: MinIO Artifact Storage

- Setup MinIO
- Configure MLflow artifact location
- Testing artifact upload/download

Part 4: Docker Compose All-in-One

- Writing docker-compose.yml
- Multi-container orchestration
- Environment variables

3 Session 1: MLflow Foundations

3.1 Tutorial 2: MLflow Basics

We will cover:

- Install and configure MLflow locally
- Track experiments systematically
- Use MLflow UI for experiment comparison
- Understand Model Registry fundamentals

Theory Review

- MLflow overvie and philosophy
- Components: Tracking, Projects, Models, Registry

Part 1: Installation & Setup

- Virtual environment setup
- MLflow installation
- Exploring `mlruns/` directory structure
- Launching MLflow UI

Part 2: Experiment Tracking

- Logging parameters, metrics, and artifacts
- Creating custom metrics & plots
- Comparing runs in the UI

Part 3: Model Registry Basics

- Registering models
- Model versioning
- Model aliases (champion/challenger pattern)
- Loading models with `models:/` URI scheme

3.2 Tutorial 3: Model Registry

We will cover:

- Model Registry workflows
- Implement model lifecycle management with aliases
- Track models
- Production model transitions

Theory Review

- Model Registry architecture
- Model aliases
- Production workflow patterns

Part 1: Model Registration

- Programmatic model registration
- Adding descriptions & tags
- Model signatures & input examples

Part 2: Model Lifecycle Management

- Setting and updating model aliases
- Transition workflow

Part 3: Loading & Serving Models

- Loading models with different methods
- Understanding `models:/name/version` vs `models:/name@alias`
- Building batch prediction pipelines
- Tracking model lineage

4 Session 2: Production Deployment

4.1 Tutorial 5: FastAPI Model Serving

We will cover:

- Build production-ready REST APIs
- Load models from MLflow Registry
- Containerize ML services

Theory

- REST API fundamentals
- FastAPI overview and features
- Model serving patterns

Part 1: FastAPI Basics

- Creating a simple API
- Request/response models with Pydantic
- Path parameters & query parameters

Part 2: Model Loading & Inference

- Loading models from MLflow Registry
- Implementing prediction endpoints
- Building batch prediction endpoints

Part 3: Dockerize & Deploy

- Creating Dockerfile for API service
- Integrating with docker compose
- Testing workflow

Mini Exercise

You will complete an end-to-end integration:

1. Train and track a model with MLflow
2. Register model in Model Registry
3. Deploy model via FastAPI
4. Test with Postman or curl
5. View logs and metrics in MLflow UI

5 Session 3: Advanced Topics

5.1 Tutorial 6: PyTorch + MLflow Integration (Quick demo)

Theory

- Deep learning experiment tracking challenges
- MLflow integration patterns for DL

Hands-on Exercises

- PyTorch + MLflow demonstration
- Image classification with MNIST or CIFAR-10
- Logging model architecture and training curves
- Hyperparameter tuning with PyTorch Lightning (optional)

Self-study Material

HuggingFace Transformers + MLflow integration guide provided to you later.

5.2 Tutorial 8: Monitoring

Theory

- ML monitoring fundamentals
- Types of drift: data drift, concept drift, prediction drift

Part 1: Model Performance Monitoring

- Logging predictions to database
- Tracking accuracy, latency, and throughput
- Building monitoring dashboard with Streamlit

Part 2: Data Drift Detection with Tools

- Evidently AI
- Visualizing drift metrics

5.3 Tutorial 10: Capstone Project

Part 1: End-to-End Integration

- Complete pipeline: Data → Training → Registry → API
- Adding monitoring dashboard
- Full workflow demonstration
- Integration testing

Part 2: Final Project Requirements

- Documenting architecture & workflows
- Q&A

6 Bonus Materials

These materials are optional and designed to deepen understanding of testing and CI/CD practices in MLOps:

Tutorial 7: Testing with Pytest

Tutorial 9: CI/CD Fundamentals