# Introduction to MLOps

## Table of Contents

The course is designed for students with minimal exposure to production-level machine learning workflows.

This study plan assumes **12 to 15 sessions**, each lasting **150-180 minutes** and includes both lectures and hands-on practice. The focus is on teaching the full modern MLOps lifecycle and the industry-standard tools used in production-grade machine learning.

---

## Course Overview

- **Title**: Introduction to MLOps
- **Duration**: 12 to 15 sessions (2.5 hours per session)
- **Audience**: Beginners with basic Python knowledge
- **Primary Goal**: Equip students with the knowledge and hands-on skills to understand and implement production-grade machine learning workflows in real-world industry settings.

---

## Learning Objectives

1. Understand the fundamentals of MLOps and its importance in production-grade machine learning.
2. Explore the MLOps lifecycle: data engineering, model development, deployment, monitoring, and continuous improvement.
3. Learn and practice tools like Docker, Kubernetes, CI/CD, MLflow, Airflow, and more.
4. Build and deploy a production-grade ML pipeline end-to-end.
5. Gain hands-on experience with real-world MLOps scenarios.

---

## Session Breakdown

**General Format for Each Session:**

- **Lecture (75 minutes):** Introduce concepts and tools with examples.
- **Hands-on Practice (60 minutes):** Implement concepts using Jupyter Notebooks and tools installed locally.
- **Q&A and Recap (15 minutes):** Address questions and solidify understanding.

---

## Session 1: Introduction to MLOps and Environment Setup

- **Lecture**:
    - What is MLOps? Why is it important?
    - Overview of the MLOps lifecycle.
    - Key roles: Data Scientists, ML Engineers, DevOps.
    - Tools and technologies in MLOps.
    - Introduction to the project we'll build throughout the course.
- **Hands-on Practice**:
    - Set up Python environment: Anaconda, virtual environments.
    - Install necessary libraries: pandas, scikit-learn, numpy, etc.
    - Install Docker and set up a simple container.
    - Introduction to Jupyter Notebooks.
- **Homework**:
    - Research: What is the difference between MLOps and DevOps?
    - Install VSCode and learn basic usage.

---

## Session 2: Introduction to the MLOps Lifecycle & Version Control

- **Lecture**:
    - The stages of the MLOps lifecycle: Data, Model, Code, Deployment, Monitoring.
    - Importance of version control in MLOps.
    - Overview of Git and GitHub.
- **Hands-on Practice**:
    - Create a GitHub repository.
    - Practice git add, commit, push, and pull.
    - Organize an ML project repository structure.
- **Homework**:
    - Read about Git branching strategies (e.g., GitFlow, feature branches).
    - Fork and clone an open-source ML repository.

---

## Session 3: Data Engineering for MLOps

- **Lecture**:
    - Overview of data engineering in MLOps.
    - Data cleaning, validation, and transformation.
    - Tools: pandas, great_expectations, Apache Spark.
- **Hands-on Practice**:
    - Load and clean a dataset using pandas.
    - Perform exploratory data analysis (EDA).
    - Validate data using great_expectations.
- **Homework**:
    - Research data pipelines and ETL tools.
    - Explore Kaggle datasets for practice.

## Session 4: Building an ML Model

- **Lecture**:
    - Overview of model development in MLOps.
    - Best practices for reproducibility.
    - Introduction to MLflow for experiment tracking.
- **Hands-on Practice**:
    - Train a basic ML model (e.g., linear regression) on a cleaned dataset.
    - Use MLflow to log experiments (parameters, metrics, artifacts).
- **Homework**:
    - Read about hyperparameter tuning techniques.
    - Install scikit-optimize or optuna.

---

## Session 5: Containerization with Docker

- **Lecture**:
    - Introduction to Docker and containerization.
    - Benefits of containers in MLOps.
    - Building a Dockerfile for ML applications.
- **Hands-on Practice**:
    - Write a Dockerfile for the ML project.
    - Build and run a Docker container.
    - Share the container on DockerHub.
- **Homework**:
    - Research Kubernetes and its role in orchestration.
    - Experiment with Docker Compose.

---

## Session 6: CI/CD for MLOps

- **Lecture**:
    - Introduction to CI/CD and its importance in MLOps.
    - Tools: GitHub Actions, Jenkins, CircleCI.
    - Automating tests and deployments.
- **Hands-on Practice**:
    - Set up a GitHub Actions workflow to test code.
    - Create a CI/CD pipeline to automate Docker container builds.
- **Homework**:
    - Research deployment strategies: blue-green, canary, shadow.

---

## Session 7: Model Deployment Basics

- **Lecture**:
    - Introduction to model deployment.
    - REST APIs for ML models using Flask or FastAPI.
    - Deployment strategies.
- **Hands-on Practice**:
    - Create a Flask API to serve the ML model.
    - Test the API locally using requests.
- **Homework**:
    - Explore cloud platforms (AWS, GCP, Azure) for deployment.

---

## Session 8: Advanced Model Deployment with Kubernetes

- **Lecture**:
    - Introduction to Kubernetes.
    - Deploying ML models on Kubernetes clusters.
    - Concepts: Pods, Services, Deployments.
- **Hands-on Practice**:
    - Deploy a containerized ML model on Minikube or a managed Kubernetes service.
    - Scale the deployment.
- **Homework**:
    - Research Helm for Kubernetes configuration management.

---

## Session 9: Workflow Orchestration

- **Lecture**:
    - Workflow orchestration in MLOps.
    - Tools: Apache Airflow, Prefect, Kubeflow Pipelines.
- **Hands-on Practice**:
    - Set up Apache Airflow.
    - Create a simple ML workflow DAG (e.g., data preprocessing, model training).
- **Homework**:
    - Research alternative orchestration tools like Prefect.

---

## Session 10: Monitoring and Logging

- **Lecture**:
  - Importance of monitoring ML models in production.
  - Tools: Prometheus, Grafana, ELK stack.
  - Concepts: Model drift, data drift.
- **Hands-on Practice**:
  - Set up Prometheus and Grafana.
  - Log model predictions and monitor metrics.
- **Homework**:
  - Explore open-source tools for model monitoring (e.g., EvidentlyAI).

---

## Session 11: Model Retraining and Automation

- **Lecture**:
  - Automating the retraining of ML models.
  - Continuous Training (CT) pipelines.
  - Tools: TFX, Kubeflow Pipelines.
- **Hands-on Practice**:
  - Create a pipeline to retrain the model with new data.
  - Automate the pipeline using Kubeflow or Airflow.
- **Homework**:
  - Research A/B testing for ML models.

---

## Session 12-15: End-to-End Project

- **Lecture**:
    - Recap of the entire MLOps lifecycle.
    - Best practices for production-grade systems.
- **Hands-on Practice**:
    - Build and deploy an end-to-end MLOps project:
        - Data ingestion and validation.
        - Model training and tracking.
        - Deployment with CI/CD.
        - Monitoring and retraining.
- **Homework**:
    - Document the project as a portfolio piece.

---

## Additional Sessions (Optional)

- **Session 13: Advanced Topics in MLOps**
    - Concepts: Feature stores, model explainability, bias detection.
    - Tools: Feast, SHAP, LIME.
- **Session 14: Exploring MLOps on Cloud Platforms**
    - AWS SageMaker, GCP AI Platform, Azure ML.
- **Session 15: MLOps Case Studies**
    - Analyze real-world MLOps implementations.

---

# References and Resources

1. **Courses**:
    - Coursera: Machine Learning Engineering for Production (MLOps) by Andrew Ng
    - Udemy: Comprehensive MLOps Bootcamp
    - DeepLearning.AI: MLOps Specialization
2. **Books**:
    - *Introducing MLOps: How to Scale Machine Learning in the Enterprise* by Mark Treveil
    - *Designing Machine Learning Systems* by Chip Huyen.
    - *Building Machine Learning Pipelines* by Hannes Hapke and Catherine Nelson.
3. **Online Articles and Blogs**:
    - Google Cloud MLOps Guide
    - MLflow Documentation
4. **Open-Source Tools**:
    - MLflow
    - Kubeflow
    - EvidentlyAI
5. **Datasets for Practice**:
    - Kaggle Datasets
    - UCI Machine Learning Repository