



FACULTY OF COMPUTER SCIENCE AND ENGINEERING
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi

AI321L Machine Learning

Lab No: OEL

Instructor: Asim Shah

Marks Weight = 10%

End-to-End Machine Learning Deployment & MLOps Pipeline

Project Title:

Design and Deploy an End-to-End Machine Learning System with FastAPI, CI/CD, Prefect, Automated Testing, and Docker Containerization

Domain Selection

Each student must choose ONE domain from the list below and build their complete ML Engineering pipeline around it:

Available Domains:

1. Healthcare
2. Economics & Finance
3. Entertainment & Media
4. Earth & Environmental Intelligence

Students must clearly mention their selected domain in the introduction of their project report.

Note: You have to include multiple machine learning tasks (classification, regression, dimensionality reduction, recommendation systems, time series analysis, clustering and association,) in same the work flow. Choose datasets wisely so you can showcase your project as real world problem solver.

Project Overview:

This project challenges students to build a full-stack ML Engineering system that mirrors professional MLOps workflows used in industry-leading companies such as Netflix, Airbnb, and Google.

Students must develop, test, containerize, orchestrate, and deploy a machine learning model using modern production-grade tools.

The goal is to evaluate the students' ability to work with machine learning, software engineering, automation, and DevOps, all integrated into one coherent project.

Project Objectives

Students will:

1. Build and Deploy ML Models with FastAPI

- Train a machine learning model (regression, classification, or deep learning).
- Serve real-time model predictions using FastAPI.
- Implement endpoints handling different input types (JSON, file uploads, numeric features).
- Ensure efficient model loading, logging, and maintainable code structure.

2. Implement CI/CD Pipeline Using GitHub Actions

- Automate:
 - Code checks
 - Unit tests and ML tests
 - Data validation
 - Model training triggers
 - Container image building
 - Deployment pipeline
- Enable continuous integration and continuous delivery for the full ML system.

3. Orchestrate ML Workflows Using Prefect

- Build a Prefect pipeline that includes:
 - Data ingestion
 - Feature engineering
 - Model training
 - Evaluation
 - Saving and versioning the model
- Implement error handling, retry logic, and success/failure notifications (Discord/Email/Slack).

4. Implement Automated Testing for ML Models

Using DeepChecks or equivalent ML testing framework:

- Test data integrity
- Identify drift

- Validate performance metrics
- Detect issues during CI/CD automatically before deployment

5. Containerize the Entire System

Using Docker:

- Create a Dockerfile for the FastAPI service
- Build and optimize the image
- Run all services in containers
- (Optional bonus: use Docker Compose to orchestrate API + Prefect + database)

6. ML Experimentation & Observations

Students must:

- Run multiple ML experiments
- Log results (accuracy, RMSE, F1-score, etc.)
- Compare model versions (baseline vs improved)
- Provide observations on:
 - Best-performing model
 - Data quality issues
 - Overfitting/underfitting patterns
 - Deployment speed improvements with CI/CD
 - Reliability improvements via Prefect orchestration

Expected Deliverables

Students will submit:

1. Source Code Repository (GitHub) containing:

- FastAPI app
- Prefect workflow
- Dockerfile + docker-compose
- ML model training scripts
- Automated tests
- GitHub Actions workflow file

2. Demonstration Video (5–10 minutes)

Showing:

- Running API

- CI/CD workflow in action
- Prefect flow execution
- Dockerized services

3. Project Report

Must include:

- Introduction, problem statement
- ML experiments & comparison
- System architecture diagram
- Containerization workflow
- CI/CD pipeline explanation
- Prefect orchestration flow
- Complete methodology flow diagram
- Final observations, limitations, and future work
- **Each part should be included in this project, you can use different tools based on open source and available resources**