Assignment 1

“Build, Track, Package, Deploy and Monitor an ML Model using MLOps Best Practices”

Group 32

Name	BITS ID
ATIK JAIN	2023ac05724@wilp.bits-pilani.ac.in
KURUMILLA SRILEKHHA	2022ac05127@wilp.bits-pilani.ac.in
MITHU DEB	2023ac05336@wilp.bits-pilani.ac.in
SAGAR MAHAPATRO	2023ac05796@wilp.bits-pilani.ac.in

Project Overview: mlop-assignment-githubaction by

This repository is a well-organized MLOps project that implements a complete pipeline for building, deploying, and monitoring a machine learning model, with a focus on automation via GitHub Actions.

Key Components

Directory Structure

.github/workflows/: Contains GitHub Actions YAML workflow(s) responsible for automating tasks like DVC data management, model training, Docker image building, and deployment.

data/, models/, dvcstore/: Organized folders for raw/processed data, trained model artifacts, and DVC storage folders respectively.

src/: Source code directory likely including data loading, preprocessing, model training, model registration, and API serving logic.

prometheus/: Dedicated directory for monitoring configurations—possibly containing prometheus.yml or related setup.

Dockerfile and docker-compose.yml: Configuration for containerizing the application and orchestrating services (FastAPI app, Prometheus, Grafana, etc.).

Functional Highlights

Git & DVC Integration: Version control for both code and data, with a DVC store for tracked datasets.

CI/CD with GitHub Actions: Automated processes include data processing (via DVC), experiment tracking (MLflow), Docker image builds, and container deployment—ensuring reproducible MLOps workflows.

Model Serving: API packaging using FastAPI (or Flask), containerized for consistent deployment via Docker.

Monitoring & Logging: Integration of Prometheus (and potentially Grafana) for collecting and visualizing app metrics.

Orchestration with Docker Compose: Easily run all services locally or in deployment environments using a single docker-compose.yml.

Repository: mlop-assignment-githubaction

Goal: Full-stack MLOps pipeline using California Housing dataset—covering model training, experiment tracking, deployment, and monitoring with industry best practices.

Features:

Data versioning with DVC

Experiment tracking and model registry via MLflow

API deployment using FastAPI + Docker

CI/CD automation with GitHub Actions

Monitoring via Prometheus (and Grafana)

Local orchestration via Docker Compose

Structure:

```
.

├── .github/workflows/     # CI/CD workflows

├── data/ & dvcstore/      # Dataset tracking (DVC)

├── src/                   # Data loading, model training, serving code

├── prometheus/            # Monitoring configuration

├── Dockerfile             # Container specification
```

```
└── docker-compose.yml     # Service orchestration (API, Prometheus, etc.)
```

Highlights: Automated pipeline, reproducible experiments, containerized deployment, observability, and scalability.