MLOps Assignment 1
ML Experiment Tracking with MLflow

Submitted by: Luv Goel
Course: AWS & MLOps
Date: October 2025

1. Introduction

This project demonstrates the use of MLflow for experiment tracking and model management. By integrating MLflow with multiple machine learning models, the project shows how to log, compare, and register models in a structured workflow. This approach represents a key part of MLOps — automating and streamlining machine learning model lifecycle management.

2. Objective

The main objective of this assignment is to perform machine learning experiment tracking using MLflow. It involves training multiple models on the Iris dataset, recording their performance metrics, and registering the best-performing model for version control and deployment purposes.

3. Dataset Description

The Iris dataset is a classic dataset in machine learning containing 150 samples of iris flowers, divided into three species: Setosa, Versicolor, and Virginica. Each sample has four features — sepal length, sepal width, petal length, and petal width. The task is to classify the flower species based on these measurements.

4. System Requirements

The following software and libraries were used: - Python 3.10+ - scikit-learn - pandas - numpy - mlflow - matplotlib (optional, for plots) The project can be executed entirely in Google Colab or any Python environment with MLflow installed.

5. Workflow Overview

1. Load and preprocess the Iris dataset. 2. Train multiple machine learning models (Logistic Regression, Random Forest, Decision Tree, SVM, KNN). 3. Track metrics such as Accuracy, Precision, Recall, and F1-Score using MLflow. 4. Compare results and identify the best-performing model. 5. Register the best model in the MLflow Model Registry. 6. Transition the model through stages: None → Staging → Production.

6. Implementation Details

Each model was trained separately within an MLflow run context. MLflow logged parameters, metrics, and serialized model artifacts. MLflow features used: - `mlflow.start_run()`: To start experiment tracking. - `mlflow.log_param()` and `mlflow.log_metric()`: To record model details and scores. - `mlflow.sklearn.log_model()`: To save the trained model. - `mlflow.register_model()`: To register the best model in the Model Registry. Model training and tracking were performed in a loop, evaluating five different algorithms. The best model (based on F1-score) was automatically registered and promoted to the Production stage.

7. Results & Discussion

Each model achieved the following approximate results: - Logistic Regression: Accuracy ~96.7% - Random Forest: Accuracy ~96.7% - Decision Tree: Accuracy ~93.3% - SVM: Accuracy ~100% - KNN: Accuracy ~96.7% The SVM model produced the highest F1-score and was therefore selected as the best model. It was registered in MLflow's Model Registry as version 1 and promoted to the Production stage.

8. Key Learnings

Through this project, the following concepts were applied and understood: - Practical use of MLflow for tracking experiments. - Automatic metric logging and model comparison. - Model versioning and registry management. - Stage transitions to manage model lifecycle. - Reproducibility of ML experiments.

9. Conclusion

This project successfully demonstrates the complete MLOps pipeline for model experiment tracking using MLflow. By automating training, evaluation, and registration, MLflow helps ensure consistent and transparent model development. Such workflows are essential for deploying reliable and maintainable machine learning systems in production environments.