**Evaluation IV – Assignment – Experiential Learning**

**Project Title: Real-Time Air Traffic Delay Prediction**

**Date: 12-10-2025**

**Name of the Faculty           : Mr. Yogesh Murumkar**

**Subject                                : MLOps**

**Signature of the Faculty             :**

**Submitted by:**

**Swati Sharma**

**24030242071**

**MBA – DSDA (2024 – 2026)**

1. **Problem statement**

Air traffic delays have a cascading impact on airline operations, passenger satisfaction, and airport efficiency.  
Predicting these delays in advance allows airlines and airports to proactively reschedule flights, optimize crew allocation, and improve customer experience.

The challenge lies in integrating **real-time weather data**, **historical flight records**, and **operational variables** to accurately predict whether a flight will be delayed — and by how much.

**Objective:**

To build a **machine learning model** that predicts **flight delays** (either as a regression problem – predicting delay duration, or as a classification problem – predicting delayed vs. on-time).

Then, deploy this model as a **real-time API** using **Docker** and **cloud infrastructure**, and track its entire lifecycle using **MLOps tools** such as **MLflow**, **Prometheus**, and **Whylogs**.

**Why this matters (business impact):**

* **Airlines** can reduce costs from crew overtime and fuel wastage.
* **Passengers** receive real-time delay information for better planning.
* **Airports** can manage gate usage, runways, and air traffic more efficiently.

1. **ROI & risk analysis**

**ROI = (Predicted Revenue - Actual Revenue) / Cost**

**Estimated ROI:**

* Up to **15–20% improvement** in on-time performance.
* **Reduced operational losses** by ~₹2–3 crores annually (for a mid-size airline).
* Enhanced **customer satisfaction** and **brand reputation**.

**Risk Analysis:**

| **Risk** | **Description** | **Mitigation** |
| --- | --- | --- |
| Data Quality | Missing or incorrect flight data | Data cleaning, imputation |
| Model Drift | Weather and traffic patterns change over time | Continuous monitoring, retraining |
| Cloud Downtime | Cloud deployment failure | Multi-region deployment |
| Security | Unauthorized API access | API key-based authentication, SSL |

**3. MLOps tools used (project-specific and how we used each)**

| **Stage** | **Tool** | **Purpose** |
| --- | --- | --- |
| **Data Versioning** | **DVC / Git** | **Manage dataset versions** |
| **Experiment Tracking** | **MLflow** | **Log metrics, models, parameters** |
| **Model Packaging** | **Docker** | **Create reproducible container image** |
| **Deployment** | **AWS EC2 / GCP Cloud Run / Azure App Service** | **Host model API** |
| **Monitoring** | **Prometheus / Whylogs** | **Track latency, model drift, accuracy** |
| **CI/CD** | **GitHub Actions** | **Automate testing and deployment** |

**4. Model Lifecycle Tracking**

1. **Data Ingestion:** Load flight + weather data (e.g., from Kaggle’s Flight Delay Dataset).
2. **Feature Engineering:** Extract features like temperature, wind speed, departure time, route, etc.
3. **Model Training:** Train a regression/classification model (e.g., XGBoost, RandomForest, or Linear Regression).
4. **Experiment Tracking (MLflow):**
5. Log metrics like accuracy, MAE, RMSE.
6. Register best model in MLflow Model Registry.
7. **Model Packaging (Docker):**
8. Build Docker image with dependencies.
9. Expose /predict API endpoint using Flask/FastAPI.
10. **Cloud Deployment:**
11. Deploy container to AWS/GCP/Azure.
12. **Monitoring:**
13. Use Prometheus or Whylogs to track latency, request volume, and model drift.

**5. Deployment Architecture**

Historical Flight + Weather Datasets

Data Preprocessing & Feature Engineering

ML Model Training

ML Flow Tracking & Model Registry

Docker Container  
(FastAPI/ FLASK)

Cloud Deployment

Monitoring (Prom.) Drift (Whylogs)

**6. Monitoring**

* **Latency (Prometheus):** Average response time of API requests.
* **Drift (Whylogs):** Statistical difference between training data and live input data.
* **Accuracy (MLflow):** Degradation in performance over time.

**7. Conclusion**

This MLOps-based Air Traffic Delay Prediction system ensures **reliable, scalable, and continuous learning** from flight data. It demonstrates how modern DevOps principles can be applied to machine learning, enabling:

* Faster deployment cycles
* Real-time predictions
* Automatic monitoring and retraining

The project successfully bridges the gap between **data science and operations** — achieving end-to-end automation from model development to deployment and monitoring.