**By:** Nehal Taha Elzeftawy

**Project Report**

**Airline Delay Prediction System**

**Project Description**

Flight delays are a major concern for both airlines and passengers. They affect customer satisfaction, operational efficiency, and financial performance. This project aimed to build a machine learning system capable of **predicting** and **classifying** flight delays using historical flight data.

**The project was carried out with two main objectives:**

* **Classification**: Predict whether a scheduled flight will be delayed or not.
* **Regression**: For delayed flights, estimate the expected duration of the delay.

Throughout the project, multiple preprocessing techniques, feature engineering steps, and machine learning models were applied. Each step of the workflow was carefully tested and documented to achieve the best possible performance.

**Dataset and Features**

The dataset used was the [**2015 Flight Delays and Cancellations**](mailto:https://www.kaggle.com/datasets/usdot/flight-delays?&select=flights.csv) dataset, obtained from **Kaggle**. It contained flight schedules, carrier information, departure and arrival data, and detailed breakdowns of delays due to different causes.

**Key Steps:**

* Loaded the dataset into a Pandas DataFrame.
* Checked the dataset’s **shape, columns, and data types**.
* Inspected the **first rows** to understand the structure.
* Generated a **statistical summary** of numerical features.
* Listed and explained the dataset features, such as **YEAR**, **MONTH**, **DAY\_OF\_WEEK**, **AIRLINE**, **ORIGIN\_AIRPORT**, **DESTINATION\_AIRPORT**, **DEPARTURE\_DELAY**, **ARRIVAL\_DELAY**, and **delay categories** (weather, airline, security, etc.).
* Defined targets:
  + **Classification target** – Flight delayed or not.
  + **Regression target** – ARRIVAL\_DELAY in minutes.

**Project Workflow**

**1. Data Exploration**

* Explored first few data records of the dataset.
* Checked data Shape, Features & Data types.
* Generated statistical summary for numerical data.
* Checked for duplicated records.
* Explored the unique values for each categorical feature (for better understanding).

**2. Data Cleaning Before EDA**

* Dropped features with more than 85% missing values (Which were not important for our case study).
* Transform object data types into categories (for better memory usage).
* Dropped features that are out of the scope of the model study (Leakage Data).

**3. Exploratory Data Analysis (EDA)**

* Checked distributions of features such as scheduled departure time, and delays.
* Analyzed correlations between numerical features and the target variable.
* Check for Airlines with most delays.
* Box plotted features to check for outliers.
* Scatter plotting important features to check for their relationship with the regression target.

**4. Data Preprocessing & Feature Engineering**

* Addressed missing values by either imputing or dropping them (if so small amounts).
* Handled outliers by replacing them with the minimum & maximum quantile.
* Create the Classification Target variable **IS\_DELAYED** by specifying 1 to flights with more than 15 minutes delay.
* Checked for class target balance, which shows that the data is **imbalanced** by **18% to 82%**.
* Encoded categorical feature **AIRLINE** using one-hot encoding.
* Encoded categorical features (**ORIGIN\_AIRPORT**, **DESTINATION\_AIRPORT**) using Frequency Encoding.

**5. Handling Class Imbalance**

* Identified imbalance in the classification target (**delayed flights were fewer**).
* Tested different approaches:
  + **Undersampling** of the majority class.
    - Still the model performance was bad as some patterns from the major class were dropped with their records, which leads to significantly low accuracy (which is not the best way to evaluate model with imbalanced data), but also the F1-Score & ROC gave low scores.
  + **SMOTE (Synthetic Minority Oversampling Technique)**.

I did not include this step in the notebook because I did not use it after trying it

* + - It works by generating data points for the minor data between data points that already exist.
    - It took so much time and didn’t affect the model performance that much, it was better by only 7%.
  + I came to conclusion that it will be better to use methods that work better in such cases, such as ensemble methods (Bagging or Boosting).

**6. Clustering**

* Clean & Scale the Regression data.
* Applied MinBatchMeans for efficiency.
* Added Cluster labels on the data set.
* Used the added feature in model training.

**7. Model Training and Evaluation**

* **Classification Model**

The project tested several models step by step:

**Logistic Regression**

* Scaled the numerical features in training data before passing it to the model.
* Used as the baseline classifier.
* Trained on preprocessed data with balanced sampling (After Undersampling).
* Evaluated performance using accuracy, precision, recall, and F1-score.

**Random Forest Classifier**

* Implemented ensemble bagging with decision trees.
* Compared performance against Logistic Regression.
* Faced memory usage issues when using large feature sets (handled by dimensionality reduction).

**XGBoost Classifier (Best Model)**

* Configured with different tree\_method options for efficiency.
* Tested with undersampling strategies to balance the dataset.
* Achieved the best balance between F1-score and ROC-AUC compared to other models.
* **Regression Models**
* Focused on predicting **delay duration** (ARRIVAL\_DELAY).
* Applied Linear Regression model as the base line.
* Applied **regression** versions of **Random Forest** and **Gradient Boosting**.
* Evaluated using RMSE and R² metrics.
* **Model Comparison**
* Collected metrics across all models:
  + **Classification** – Accuracy, Precision, Recall, F1-score, ROC-AUC.
  + **Regression** – RMSE, R².
* XGBoost and Random Forest performed better than Logistic Regression.
* Logistic Regression provided a good baseline but was not suitable for complex patterns.

**Results and Observations**

* The dataset was highly **imbalanced**, which significantly affected baseline models.
* Undersampling slightly improved classification performance.
* Among classifiers:
* **XGBoost** achieved the highest F1-score (~0.45) and ROC-AUC (~0.69).
  + Logistic Regression had the lowest performance.
  + Regression models struggled with high variance due to the noisy delay distribution. Random Forest Regressor provided more stable predictions than linear approaches.
* Memory issues occurred when training on the full dataset locally.

**Conclusion and Future Work**

This project successfully built a system to **predict flight delays and estimate delay duration** using historical flight data. The process involved thorough data exploration, preprocessing, feature engineering, balancing techniques, and model experimentation.

**Key Takeaways:**

* Data imbalance was the primary challenge, requiring undersampling to improve results.
* Ensemble methods (Random Forest, XGBoost) consistently outperformed simpler models.
* Best results were achieved with **XGBoost**, which balanced accuracy and efficiency.
* Regression predictions of exact delay duration remain difficult due to high variability in real-world delays.