Report: Flight Delay Prediction Project

Hamza Jaffer

22i-0583

AI-B ML

Phase 1: Data Preprocessing and Feature Engineering

1. Data Integration:

- **Weather Data Integration**: The weather dataset was integrated with the flight data to enrich the model with weather-related features, which could influence flight delays. This included attributes like temperature, wind speed, and pressure, which were merged based on the flight date and time.

2. Data Cleaning and Transformation:

1. **Handling Missing Values**:
   - Missing values in the dataset were handled by imputing where possible (e.g., filling missing temperature or wind speed values with the mean) or dropping rows that had excessive missing values, especially in critical columns like delay time or departure scheduled time.
2. **Formatting Time Fields**:
   - Time fields such as `Scheduled Departure Time`, `Actual Departure Time`, and `Estimated Time of Arrival` were converted into a standard `datetime` format using the `pd.to_datetime()` function. This ensured consistency and easier extraction of temporal features (e.g., hour, day of the week).

3. Feature Engineering:

1. **Departure Delay**:

   - A new feature, `Departure Delay`, was created by calculating the difference between the `Scheduled Departure Time` and the `Actual Departure Time`. This was used as the target variable for prediction models.

2. **Merge Weather Data**:

   - Weather features, such as temperature and wind speed, were extracted from the integrated weather dataset and joined with the flight data on date and time. These features were expected to influence the delay duration.

3. **Extract Temporal Features**:

- Additional temporal features were derived to capture patterns in flight delays:
    - **Day of the Week**: Extracted from the `Scheduled Departure Time` to capture weekly patterns.
    - **Hour of the Day**: Extracted from the `Scheduled Departure Time` to analyze delays based on the time of day.
    - **Month of the Year**: Extracted from the `Scheduled Departure Time` to detect seasonal effects on flight delays.

Phase 2: Exploratory Data Analysis (EDA)

1. Visualizations:

- **Delay Distributions**:
    - A histogram of delay durations was created to visualize the distribution of delays in the dataset. This helped identify the proportion of flights with no delay vs. those with short, moderate, and long delays.
- **Temporal Analysis**:
    - Line plots and bar charts were used to show how delays varied across different hours of the day, days of the week, and months. This analysis revealed temporal patterns, such as higher delays in the afternoon or certain months.
- **Category-Wise Analysis**:
    - Delays were grouped by airline, departure airport, and flight status to understand how different categories affected delays. This helped identify trends such as certain airlines being more prone to delays.

2. Correlation Analysis:

- A series of visualizations (e.g., scatter plots and heatmaps) were created to explore the relationship between weather variables (e.g., temperature, wind speed) and flight delays. These visualizations highlighted potential correlations, like the effect of high winds or extreme temperatures on delays.

3. Comparison:

- A comparison was made between the delay distributions of the training and testing datasets. This ensured that the datasets were consistent, and no significant discrepancies would bias the model.

Phase 3: Analytical and Predictive Tasks

1. Classification Tasks:

- **Binary Classification**:
  - Flights were classified as "on-time" or "delayed" based on the delay criteria (`delay = 0: on-time`, `delay > 0: delayed`).
  - A model was trained for binary classification using features such as hour of the day, wind speed, temperature, etc.
  - **Model Evaluation**:
    - **Accuracy**, **Precision-Recall**, **F1-Score**, and **Class-wise Precision-Recall** were computed.
    - A confusion matrix was generated to evaluate the classification performance (true positives, false positives, etc.).

- **Multi-Class Classification**:
  - Flights were categorized into four classes:
    - **No Delay** (0 min)
    - **Short Delay** (<45 min)
    - **Moderate Delay** (45–175 min)
    - **Long Delay** (>175 min)
  - A model was trained to classify delays into these categories, and evaluation metrics similar to the binary classification task were used.

2. Regression Analysis:

- **Delay Duration Prediction**:
  - The exact delay duration for each flight was predicted using a regression model.
  - **Model Evaluation**:
    - Models were validated using cross-validation techniques, and their performance was assessed using **Mean Absolute Error (MAE)** and **Root Mean Square Error (RMSE)**.

Phase 4: Model Optimization and Evaluation

1. Hyperparameter Tuning:

- Both **Grid Search** and **Randomized Search** were employed to optimize the hyperparameters of the models. The parameter grid included options for the number of trees in Random Forests, tree depth, and split criteria, among others.

2. Validation:

- **K-fold cross-validation** was used to assess model performance. This technique helped ensure that the models were robust and generalizable by evaluating them on multiple subsets of the data.

3. Model Comparison:

- The performance of different models (e.g., Random Forest for classification, regression) was compared based on evaluation metrics such as MAE, RMSE, accuracy, and F1-Score. The model with the best performance across metrics was selected.

Phase 5: Model Testing

1. Model Predictions:

- After training and optimizing the models, predictions were made on the test dataset using the best performing models.

2. Kaggle Submission Format:

- For the **binary classification** task, predictions were stored in a CSV file with two columns: `index` (flight identifier) and `delay` (predicted value as "on-time" or "delayed").

- For **regression**, the exact delay durations were predicted and saved in the required format.

The predictions were saved in CSV format for submission to Kaggle or other platforms.

Conclusion

The project successfully integrated flight and weather data, performed data cleaning and feature engineering, and applied both classification and regression models to predict flight delays. The models were optimized using hyperparameter tuning and evaluated using various metrics, ensuring robust performance. Finally, the predictions were saved in the required Kaggle submission format for further analysis or competition purposes.