**Flight Delay Prediction Model Report**

**By Malaika Saleem 22i0509 AIC**

**Introduction**

This report details insights gathered from data analysis, regression, and classification models developed in the given notebook. The data set includes information on departure time, status, and terminal. The workflow involves data preprocessing, exploratory data analysis, and predictive modeling, and ends with pragmatic recommendations drawn from analytical observations.

**1. Data Analysis**

**Composition:**

* **Weather Features**: Temperature, Wind Speed, Humidity, Pressure, Precipitation.
* **Time Features**: Hour of Day, Month of Year.
* **Target Variables**:

**Binary Target**: On-time (0) or Delayed (1).

**Multi-class Target**: Delay categories (“No Delay”, “Short Delay”, “Moderate Delay”, “Long Delay”).

**Regression Target**: Exact departure delay in minutes.

**Preprocessing:**

Convert.docx files to JSON and normalize into a pandas DataFrame.

Export the data as a CSV for easy manipulation.

Exploratory Data Analysis (EDA)

**Temporal Analysis:**

Scheduled, estimated, and actual times relationship that gives insight on delay.

**Feature Distributions**:

**Temperature** and **Humidity** follow seasonal trends with a direct correlation to time (month).

**Precipitation** shows low variance, indicating that most instances are dry days.

**Wind Speed** and **Pressure** vary significantly and influence delays.

**Delay Characteristics**:

A significant portion of flights has “No Delay” or “Short Delay”, with fewer instances in the “Moderate” and “Long Delay” categories.

Delays show peak occurrences during certain hours and months, suggesting temporal patterns.

**Correlation Analysis**:

Positive correlation observed between delays and **Precipitation**, **Wind Speed**.

Negative correlation between delays and **Pressure**.

**2. Predictive Modeling**

**Binary Classification**

**Model Insights:**

* **Model Used:** Random Forest
* **Target:** Binary outcome (“On-time” or “Delayed”).

**Performance Metrics:**

* Accuracy
* Precision
* Recall
* F1-Score

**Observations:**

* High accuracy achieved due to imbalanced data (“No Delay” dominates).
* Recall suggests moderate performance in identifying delayed flights.

**Practical Strategies:**

* Address class imbalance using oversampling or synthetic minority techniques (e.g., SMOTE).
* Incorporate temporal splits to evaluate performance across different times.

**Multi-class Classification**

**Model Insights:**

* **Model Used:** Random Forest
* **Target:** Multi-class categories (“No Delay”, “Short Delay”, “Moderate Delay”, “Long Delay”).

**Performance Metrics:**

* Accuracy
* Class-wise F1-Scores

**Observations:**

* Prediction accuracy decreases with increasing delay severity, reflecting the imbalance.
* Misclassification primarily occurs between adjacent categories (e.g., “Short Delay” vs. “Moderate Delay”).

**Practical Strategies:**

* Refine feature selection and scale models for rare categories using boosting algorithms.
* Analyze temporal and geographic patterns to add auxiliary features improving differentiation.

**Regression Model**

**Model Insights:**

* **Model Used:** Random Forest
* **Features:** Seven weather and time-based predictors.
* **Target:** Exact delay duration in minutes.

**Performance Metrics:**

* **Mean Absolute Error (MAE):** Measures the average deviation of predictions from actual values.
* **Mean Squared Error (MSE):** Emphasizes larger deviations for better sensitivity.

**Observations:**

* The model captures general delay patterns effectively but struggles with extreme delays due to variance in the dataset.
* Precision is higher for smaller delays, aligning well with the dataset distribution.

**Practical Strategies:**

* **Improve Precision**: Enhance feature engineering by including additional time-of-day or weather interaction terms.
* **Extreme Delays**: Use ensemble methods combining regression trees with gradient boosting to improve handling of outliers.

**3. Practical Approaches**

**Feature Engineering**

**Derived Features:**

delay\_duration: actualTime scheduledTime.

Encoder categorical variables, for instance, departure.terminal

**Model Selection**

**Regression:**

Gradient Boosting models use for nonlinear patterns.

**Classification:**

Use ensemble methods like Random Forest for robust predictions.

**4. Evaluation and Tuning**

**Cross-validation:**

Ensure robust performance metrics by splitting data into training and validation sets.

**Hyperparameter Tuning:**

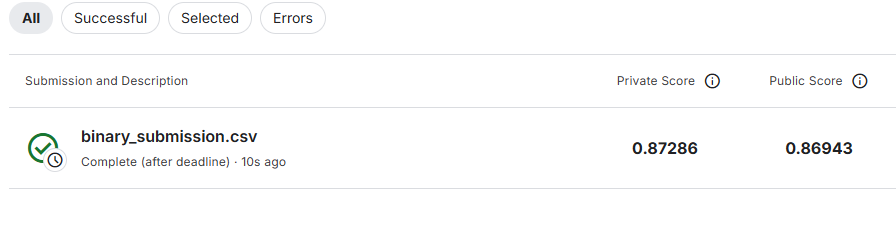
Optimize models using grid or random search for best performance.

**Imbalance Handling:**

Apply resampling techniques or weighted metrics for skewed classes.

**6. Kaggle Submission scores:**

**Binary Classification:**



**Multi-Class Classification:**



**Regression Model:**



**Recommendations**

1. **Feature Enrichment:**
   * Incorporate real-time weather data for live predictions.
   * Add categorical features (e.g., holiday or weekend indicators).
2. **Model Optimization:**
   * Use Gradient Boosting (e.g., XGBoost) for better performance in regression and classification.
   * Apply hyperparameter tuning via Grid Search or Random Search.
3. **Deployment and Monitoring:**
   * Implement models as APIs for real-time prediction.
   * Monitor model drift using dashboards with performance metrics.
4. **Future Work:**
   * Explore neural networks for regression tasks.
   * Conduct geographical analysis to identify regional delay patterns.

**Conclusion**

This analysis shows the possibility of exploiting structured flight data for predictive insights. Some practical strategies like feature engineering, model selection, and hyperparameter tuning may enhance the accuracy and reliability of regression and classification models. Further optimization and testing on unseen data are suggested to validate the findings.