Flight Delay Prediction Analysis

This project explores how machine learning can be used to predict whether a flight will be delayed, based on historical flight data and passenger information. By applying models such as Logistic Regression, XGBoost, and Neural Networks, we aim to uncover which factors contribute most to delays and how predictive analytics can support airport operations, airlines, and traveler planning.

Dataset Overview and Feature Engineering

The dataset used for this project consists of flight and passenger details, including variables like gender, age, nationality, departure and arrival airport, date of flight, and flight status (on time or delayed).

Initial preprocessing involved:
- Converting `Departure Date` to datetime and extracting time-based features such as month and weekday.
- Encoding categorical variables like gender, continent, and airport.
- Dropping columns such as names and IDs that do not influence prediction.

To address data imbalance, binary labels were created for `Flight Status`, and class weighting was applied during modeling.

Code Explanation and Technical Workflow

This section outlines the machine learning workflow implemented in the notebook. It describes the logic behind each component, from data loading to model evaluation.

1. Library Imports and Setup

Standard libraries for data handling (`pandas`, `numpy`) and visualization (`matplotlib`, `seaborn`) were imported. Machine learning components were sourced from `scikit-learn`, `xgboost`, and `tensorflow.keras`.

2. Data Loading and Target Engineering

Data is loaded using `pandas.read_csv()`. The `Flight Status` column is converted to a binary target variable. Temporal features like month and weekday are extracted from the departure date.

3. Categorical Encoding and Cleanup

Categorical features are encoded using `LabelEncoder`. Unnecessary columns (e.g.,

names, Pilot Name) are dropped. `Departure Date` is removed after extracting time-based features.

4. Train-Test Split and Feature Scaling

The data is split into training and testing sets. Features are scaled using `StandardScaler`, which is essential for convergence in models like Logistic Regression and Neural Networks.

5. Logistic Regression with Class Weighting

Logistic Regression is trained with `class_weight='balanced'` to handle data imbalance. It serves as a fast, interpretable baseline model.

6. XGBoost with Imbalance Handling and Tuning

XGBoost uses `scale_pos_weight` to adjust for imbalance. `RandomizedSearchCV` is employed to optimize key hyperparameters, resulting in improved precision and recall for delayed flights.

7. Neural Network with Keras and Class Weights

A feedforward neural network with two hidden layers is built using Keras. The model uses dropout for regularization and `class_weight` to improve sensitivity to delayed flights.

8. Model Evaluation and Metrics

Models are evaluated using accuracy, precision, recall, F1-score, and confusion matrices. XGBoost offers the best trade-off in predictive performance, particularly for the minority class.

Exploratory Data Analysis (EDA)

EDA uncovered key trends:
- Most flights were on time, revealing significant class imbalance.
- Delays were more frequent on certain weekdays and in specific months.
- Arrival airports and continents showed variation in delay frequency.
- Older passengers had slightly higher delay rates.

Visualizations included pie charts, delay bar plots by continent and age group, monthly trends, and top delayed airports.

Visual Analytics and Interpretability

Visual tools played a central role in interpreting model outputs:

- Pie chart of delay distribution
- Monthly and weekday delay bar plots
- Age group and continent delay analysis
- Confusion matrices to compare models

These visuals offered intuitive insights for stakeholders and helped diagnose model behavior.

Feature Importance and Key Insights

XGBoost's feature importance analysis highlighted:
- Arrival Airport
- Continent of Departure
- Month and Day of Week
- Passenger Age

These features proved critical in identifying delay-prone flights, especially in operational and seasonal contexts.

Limitations and Future Enhancements

Current limitations include:
- Lack of real-time weather or traffic data
- Potentially irrelevant features (e.g., `Pilot Name`)
- Missing hour-of-day granularity

Future enhancements may include:
- Integrating weather and air traffic APIs
- Using SHAP or LIME for explainable AI
- Deploying as a web app for real-time inference

Conclusion

This project shows how flight and passenger data can be used to build delay prediction models. XGBoost, enhanced with tuning and class weighting, yielded the most reliable performance. With broader data and real-time deployment, such models can significantly improve decision-making for airlines and travelers.