MACHINE LEARNING

AI-3002

Semester Project Report

Fall-2024

"Advanced Flight Departure Delay Analysis Project"

Submitted By

ZOHA BINTE WAJAHAT

22I-0569

SECTION: C

DEPARTMENT OF COMPUTING, FAST-NUCES
ISLAMABAD

09th DECEMBER 2024

TABLE OF CONTENTS:

ABSTRACT:

The **Advanced Flight Departure Delay Analysis Project** aims to provide a comprehensive framework for understanding and predicting flight delays using modern data analysis and machine learning techniques. The project incorporates five key phases: data preprocessing and feature engineering, exploratory data analysis (EDA), analytical and predictive tasks, Model Optimization and Evaluation, and Model Testing.

The process of data preprocessing includes cleaning, merging flight data and weather data preparing them for any further analysis.

The phase of exploratory data analysis (EDA) allows to use visualizations and statistical insights to uncover patterns in delay distributions, temporal trends, and correlations between weather and flight characteristics,

The predictive modelling phase addresses three main objective:

a) Binary Classification
b) Multi-class Classification
c) Regression Analysis

The model evaluation phase introduces us to hyper parameters tuning techniques as well as K-fold cross validation.

In the model testing phase, we used trained model to test on the test dataset, predicting delay times and we format the predictions according to Kaggle submission.

I. INTRODUCTION:

Overview:

The project showcases a flight departure delay analysis which focuses on addressing the critical challenge of predicting and analyzing flight delays, which significantly impact the aviation industry's operational efficiency, customer satisfaction, and resource planning. This project combines data-driven techniques, machine learning models, and advanced analytics to uncover patterns and trends that influence flight delays.

Objective Statement:

This is a comprehensive and detailed introduction of the project, where the designers and implementers will use a broad data processing and machine learning methodologies to carry out a flight departure delay analysis. The key goals are to:

1. Learn and use data preprocessing, exploratory data analysis (EDA) techniques along with predictive modeling methodologies to aviation dataset.
2. Build the regression and classification models for delay forecasting based on machine learning algorithms.
3. Reproduce operational challenges by creating a model that will use weather and flight data to predict the delay more accurately.
4. Enhance the study of model optimization techniques which include the tuning of hyper parameters, k-fold cross-validation, and comparing the model evaluation techniques.

Scope

The project is about the development of the methodology, implementing and evaluating the predictive modeling of flight departure delays through the structured phases. The project involves the following main components:

Implementation:

a. **Framework Development:** A step by step analytical pipeline for flight delay, analysis involves data preprocessing, data modeling and evaluation
b. **Core Phases:** Conducting the binary and multi-class classification tasks of the delay and regression analysis for the duration of the delay prediction, to Kaggle.

II. PROBLEM STATEMENT:

Statement:

Flight departure delays are a critical challenge in the aviation industry. Such delays affect passenger satisfaction, airline operations, and overall efficiency. You are provided with raw Excel files including test dataset, train dataset, and weather dataset and are tasked with calculating departure delays. Using these datasets, you will analyze delay patterns and build predictive models to identify key factors contributing to these delays.

III. IMPLEMENTATION STRATEGY:

Explanation:

The implementation of the model is structured into the following phases:

Preprocessing Phase:

During the preprocessing phase, our task was to integrate the training data (provided in a DOC file) with the weather data (given in an Excel file with a single-row format). This process involved handling missing values, standardizing time fields into a uniform datetime format, and ensuring data consistency across both datasets.

I implemented a loop to read all the training data files and convert them into data frames. However, the initial structure of the data was not properly normalized, which made it challenging to work with. To resolve this, I transformed the data into a normalized format, facilitating easier analysis and further processing.

The test data, provided in CSV format, was loaded into a data frame to prepare it for future prediction tasks.

To address missing values in the training dataset, I replaced them with the mode of their respective columns. This approach is especially effective for categorical data, as it preserves the frequency distribution by filling in missing entries with the most commonly occurring category.

Additionally, I calculated the delay time in minutes within the training dataset by subtracting the scheduled departure time from the actual departure time and converting the result into minutes. I also parsed the scheduled departure date into separate columns for day, month, and year, converting these into integer data types. These extracted fields were then used to merge the training dataset with the weather dataset based on these three date-related columns, ensuring proper alignment between the two datasets.

Exploratory Data Analysis Phase:

During the exploratory data analysis (EDA) phase, we were assigned several tasks, including visualizations, correlation analysis, and comparisons between the training and testing datasets.

Visualizations

The visualization tasks were further divided into three sub-tasks:

Delay Distribution Analysis: We plotted the distributions of delays along with their frequencies to better understand the patterns of delays.
Temporal Analysis: This involved analyzing delays across different time periods using both line graphs and bar graphs. Specifically, we explored patterns of delays across hours, days, and months.
Category-wise Analysis: We conducted analysis by categories such as airline, departure airport, and flight status. This analysis was visualized using line graphs and bar graphs to identify trends and patterns in delays across these categories.

Correlation Analysis

We calculated the correlation of key variables such as *Temperature Average*, *Humidity Average*, *Wind Speed Average*, and *Flight Delay* and visualized these relationships using heatmaps, scatter plots, and boxplots.

Comparison Task

We compared the features of the training and testing datasets by visualizing them through histograms. This comparison helped identify potential discrepancies, feature distributions, and data alignment across the two datasets.

Analytical and Predictive Task Phase:

During the analytical and predictive task phase, we were assigned to perform binary classification, multi-class classification, and regression analysis.

Binary Classification

For binary classification, a new column delay_binary was created. This column is a binary indicator:

- **1** if the flight is delayed
- **0** if the flight is not delayed

Here, delay_binary was set as the target variable (**y**), and all other features were considered as predictors (**X**). The dataset was split into training and test sets. The model was trained on the training data and evaluated on the test data. Performance was assessed using the **confusion matrix** and **classification report**.

Multi-Class Classification

For multi-class classification, a new column delay_category was introduced with four categories to represent the degree of delay:

- **0**: Flight is not delayed
- **1**: Slightly delayed
- **2**: Moderate delay
- **3**: Very late

In this case, delay_category was treated as the target variable (**y**), and the other features were treated as predictors (**X**). Similar to the binary classification, the data was split into training and test sets, with the model trained on the training data and evaluated on the test data. Accuracy was visualized using the **confusion matrix** and **classification report**.

Regression Analysis

For regression analysis, the **delay time** was selected as the target variable (**y**), with all other features being predictors (**X**). The dataset was split into training and test sets. The model was trained on the training data and evaluated on the test data. Model performance was analyzed using the **confusion matrix** and **classification report**, providing insights into the predictive accuracy of the delay time estimates

Model Optimization and Evaluation Phase

Hyperparameter Tuning

To enhance the performance of the predictive models, hyperparameter tuning was performed using advanced techniques such as:

- **Grid Search:** Exhaustively searches over specified hyperparameter values to find the optimal combination.
- **Random Search:** Randomly samples hyperparameter combinations to identify the best-performing configuration more quickly than grid search.

These techniques were applied to optimize the models' learning parameters and improve accuracy by fine-tuning key aspects such as learning rates, number of estimators, depth of trees, and other relevant hyperparameters.

Validation

To ensure the trained models perform well on unseen data, **k-fold cross-validation** was applied.

- **k-fold Cross-Validation:** Divides the dataset into *k* subsets. The model is trained on *k-1* subsets and validated on the remaining subset. This process is repeated *k* times, with each subset used as a validation set once.

This technique reduces the risk of overfitting and provides a better generalization of the model's performance.

Model Comparison

Once the models were trained and validated, a comparison was conducted across all models to determine which performed optimally. Metrics such as **accuracy, confusion matrix, and classification reports** were analyzed for classification models, while **mean squared error (MSE)** and **R^2** scores were considered for regression analysis.

Model Testing Phase

Making Predictions

The trained models were used to make predictions on the test dataset. This involved:
Applying the predictive model to the test dataset to generate outputs.
Ensuring outputs align with the problem's requirements (binary classification, multi-class classification, or regression).

Saving Predictions for Kaggle Submission

The predictions were formatted to meet Kaggle's submission standards. This process varied based on the task type:

- Regression Predictions
- Predict delay times in minutes.
- Binary Classification Predictions
- Predict whether the flight is "on-time" or "delayed."
- Multi-Class Classification Predictions
  Predict categories based on the delay status (e.g., "slightly delayed," "moderate delay," "very late"