2019 Airport Delays with Weather and Airport Details

Spring 2024 Project: Team 4

By: Anurag Dwivedi, Evan Burch, Jonathan Hooper

Final Project Report

1. Problem Statement

The primary objective of this analysis is to develop a binary classification model that predicts whether a flight will experience a departure delay. This predictive model aims to identify potential delays based on a range of flight and airport-specific variables. The significance of this model lies in its potential to aid airlines, airport authorities, and passengers in anticipating delays, thereby enhancing the efficiency of flight operations and improving the overall travel experience.

The dataset for this analysis is sourced from the 2019 Airline Delays with Weather & Airport Details dataset, available on Kaggle (https://www.kaggle.com/datasets/threnjen/2019-airline-delays-and-cancellations/data)

This dataset is a comprehensive collection of flight-related information for the year 2019, encompassing a wide array of attributes that describe various aspects of flight operations. Key characteristics of this dataset include:

- Dimensionality: The dataset consists of 26 attributes in the full_data_flightdelay.csv file, offering a balanced blend of different data types. This includes 20 numeric attributes, 4 categorical attributes, and 2 text attributes. The diversity in data types provides a rich foundation for our predictive modeling. The shape and size of the dataset are (6489062, 26) and 168,715,612 respectively
- Sparsity: The dataset exhibits low sparsity, indicating that it has minimal missing entries. This characteristic suggests that the data is well-maintained and reliable for analytical purposes.

The effectiveness of the predictive model will be evaluated using several informal success measures. These include:

- K-Fold Cross Validation: A measure to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model
- F1 Score: As a measure that balances precision and recall, the F1 score will be crucial in assessing the model's ability to accurately identify delayed flights without excessive false positives or negatives.

- Confusion Matrix: This will provide a detailed breakdown of the model's performance in terms of true positives, true negatives, false positives, and false negatives, offering insights into the specific areas where the model excels or needs improvement.
- ROC AUC: This will help us measure the ability of our binary classifier to distinguish between positive and negative classes.
- Lift/Gain: measure of the effectiveness of a classification model calculated as the ratio between the results obtained with and without the model. Gain and lift charts are visual aids for evaluating performance of classification models. However, in contrast to the confusion matrix that evaluates models on the whole population, gain or lift charts evaluate model performance in a portion of the population.

2. Background and Related Work

Refer to Background and Related Work document on iLearn submission

3. Data and Exploratory Analysis

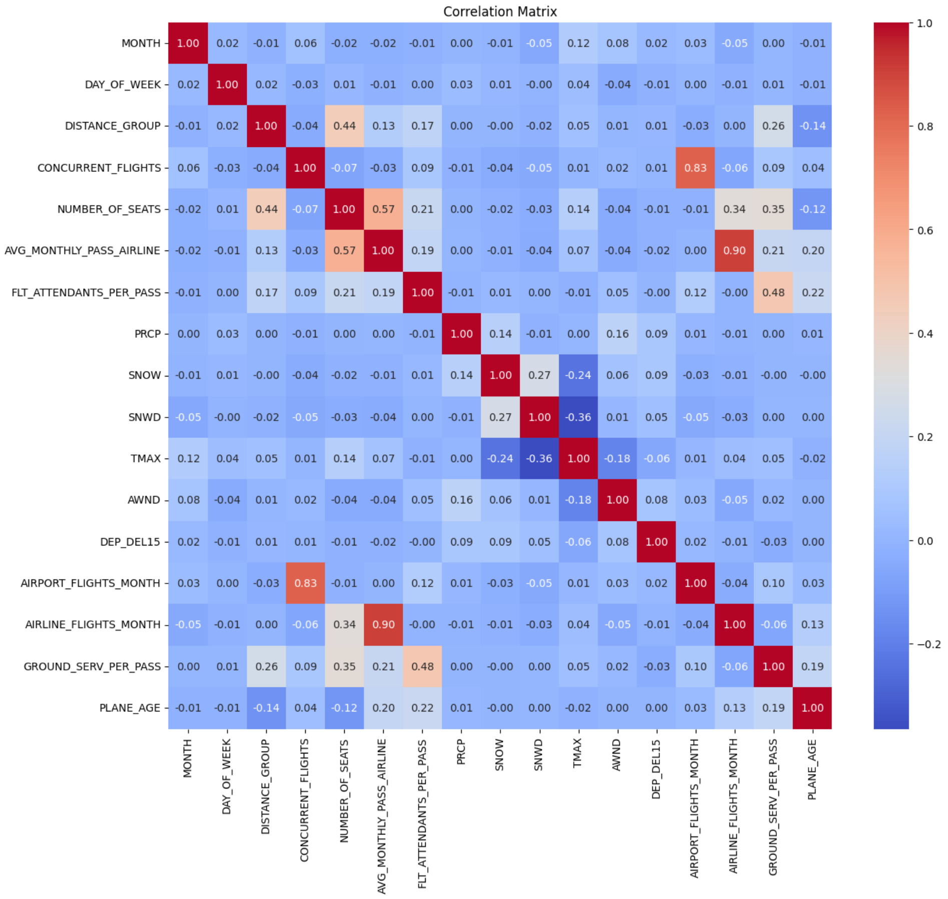
Below is a data dictionary describing the meaning of the attributes in the dataset.

Month	Month
DAY_OF_WEEK	Day of Week
DEP_DEL15	TARGET Binary of a departure delay over 15 minutes (1 means there was a delay)
DEP_TIME_BLK	Departure time block
DISTANCE_GROUP	Distance group to be flown by departing aircraft
SEGMENT_NUMBER	The segment that this tail number is on for the day
CONCURRENT_FLIGHTS	Concurrent flights leaving from the airport in the same departure block
NUMBER_OF_SEATS	Number of seats on the aircraft
CARRIER_NAME	Carrier
AIRPORT_FLIGHTS_MON	Avg Airport Flights per Month
AIRLINE_FLIGHTS_MON	Avg Airline Flights per Month
AIRLINE_AIRPORT_FLIGHTS_MONTH	Avg Flights per month for Airline AND Airport
AVG_MONTHLY_PASS_AIRPORT	Avg Passengers for the departing airport for the month
AVG_MONTHLY_PASS_AIRLINE	Avg Passengers for airline for month
FLT_ATTENDANTS_PER_PASS	Flight attendants per passenger for airline
GROUND_SERV_PER_PASS	Ground service employees (service desk) per passenger for airline
PLANE_AGE	Age of departing aircraft
DEPARTING_AIRPORT	Departing airport

LATITUDE	Latitude of departing airport
LONGITUDE	Longitude of departing airport
PREVIOUS_AIRPORT	Previous airport that aircraft departed from
PRCP	Inches of precipitation for day
SNOW	Inches of snowfall for day
SNWD	Inches of snow on ground for day
TMAX	Max temperature for day
AWND	Max wind speed for day

Initially, the dataset was imported and manipulated using Pandas, a powerful data manipulation library in Python. This step involved reading the raw data and transforming it into a dataframe suitable for analysis. Subsequently, we employed Matplotlib and Seaborn, two widely used Python libraries, for preliminary data visualization. These visualizations aided in understanding the distribution and relationship of variables within the dataset. A critical aspect of our data preparation involved identifying and handling missing values. To this end, we applied *isna().sum()* to quantify the extent of missing data across different attributes. Following this, rows with missing values were excluded from the dataset using the *.dropna()* function. Even though the dataset contained little to none missing values, this decision was made for simplicity reasons

To further understand our dataset, we generated a comprehensive data profiling report using the *ydata_profiling* package. This report was saved as an HTML file and can be found on our GitHub repository for easy access and reference. It provides a detailed overview of each attribute, including data types, missing values, and descriptive statistics, and data distribution which was instrumental in identifying any irregularities or patterns in the data. Another significant step in our data preparation involved analyzing correlations between different attributes. For this analysis, we used a correlation matrix (see below), which necessitated the exclusion of non-numerical attributes. This process enabled us to understand the interdependencies between variables, which is crucial for identifying factors that significantly influence flight delays.



Upon examination of the dataset, we did not identify any anomalies or outliers that warranted further action. Such anomalies could potentially indicate data entry errors or exceptional circumstances impacting flight schedules. Their absence suggests a dataset that is representative of typical flight operations. In terms of data imputation, we have not found it necessary to employ such techniques in our current data analysis. Imputation techniques are often employed to preserve the integrity of the dataset particularly when confronted with significant amounts of missing data. However, in our specific case, the dataset exhibited minimal to no missing values, rendering the need for imputation unnecessary.

In conclusion, the data preparation and exploratory data analysis phase have been foundational in setting the stage for the subsequent development and evaluation of predictive models for flight delay prediction. We plan to continue our exploratory data analysis to gain any further insights and may need to come back to this stage for reference or changes during the process of the project.

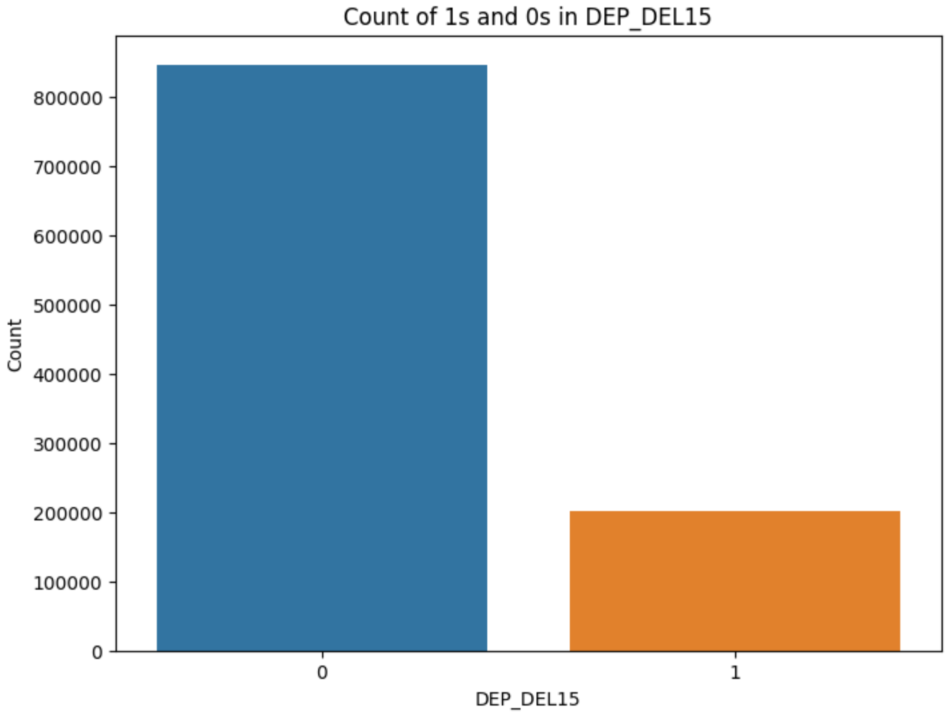
Questions we want to answer from the data:

- The concentration of delayed and non-delayed flights?
- Are delays due to day_of_week and day_of_month?
- The concentration of delay's by 'DEP_TIME_BLK'?

- Which airport in Origin stands out in delays? Which airport in Destination stands out in delays?
- How does weather affect the departure time for any travel situation? Which kind of weather?
- Which carriers are least and most reliable for on-time departure?

4. Methods

In our data analysis process, we primarily focused on data visualization, cleaning, and wrangling methods to understand and prepare the dataset for further exploration. For data visualization, we utilized box plots and histograms to gain insights into the distribution and tendencies of the data, providing a visual representation of summary statistics. Below is the distribution of our target attribute, DEP_DEL15, which is either 1 or 0. This indicates whether the flight in question was delayed by more than 15 minutes. We can see that there is a disparity here in our data. There are many more non-delayed flights than delayed flights.



Additionally, for data cleaning and wrangling, we relied on Pandas, to efficiently handle missing values and ensure the dataset was in a suitable state for deeper exploration. While considering other methods such as imputation techniques for handling missing values, we decided against them due to the simplicity of our dataset and the effectiveness of directly dropping missing values. This approach prioritized simplicity, efficiency, and effectiveness in data analysis and preparation, aligning well with the problem statement and dataset characteristics.

Furthermore, to handle categorical variables, we utilized the LabelEncoder from the sklearn.preprocessing module. This allowed us to convert categorical labels into numerical values, making them compatible with machine learning algorithms such as logistic regression.

The process involved iterating through a list of categorical labels, applying the LabelEncoder to each, and transforming the corresponding columns in the dataset.

Following data preparation, we trained a logistic regression model on the preprocessed dataset. The accuracy achieved by the logistic regression model was approximately 80.83%. Additionally, alongside logistic regression, we explored the performance of other classification models on our preprocessed dataset. After encoding categorical variables and preparing the data, we evaluated the dataset using two additional models: K-Nearest Neighbors (KNN) and Decision Tree Classifier. For KNN, we set the value of k to 5 and observed an accuracy of approximately 80.06%. KNN is a non-parametric, instance-based learning algorithm that classifies new instances based on their similarity to training instances. Similarly, employing Decision Tree Classifier yielded an accuracy of about 76.80%. Decision trees are a powerful supervised learning method used for classification tasks, which partitions the data into subsets based on certain features.

Here's a summary of the algorithms/models we used:

1. Logistic Regression: This is a fundamental algorithm for binary classification problems. It works by modeling the probability of a binary outcome based on input features, using a logistic function. It's particularly effective due to its simplicity, efficiency, and the interpretability of its results. In our context, logistic regression helps in estimating the probability of a flight being delayed, based on various predictors like weather conditions and flight details.
2. K-Nearest Neighbors (KNN) Classifier: KNN is a non-parametric, instance-based learning algorithm. It classifies a new data point based on how its neighbors are classified. 'Neighbors' are determined based on distance metrics like Euclidean or Minkowski distance. This algorithm assumes that similar data points can be found near one another. Its effectiveness in our project stems from its ability to capture complex patterns in the data without making assumptions about the underlying distribution.
3. Decision Tree Classifier: This algorithm uses a tree-like model of decisions and their possible consequences. It splits the dataset into branches based on decision nodes, where each node represents a feature in the dataset. Decision trees are easy to interpret and can handle both numerical and categorical data. They are particularly useful in our project for handling non-linear relationships and interactions between different flight-related factors.

Each of these algorithms has been chosen for their complementary strengths: logistic regression for its straightforward approach and interpretability, KNN for its effectiveness in capturing the intricacies in data, and decision trees for their flexibility and ease of interpretation, especially in dealing with complex, non-linear data scenarios.

These results provide insights into the performance of different classification algorithms on our dataset, allowing us to select the most suitable model for our specific problem statement and requirements.

5. Tools

In our analysis, we employed a diverse array of tools to thoroughly understand our dataset.

Initially, we analyzed summary statistics through box plots and histograms, providing us with an overview of the data and its distribution and tendencies. We then utilized Pandas for data cleaning and wrangling, ensuring our dataset was aptly prepared for deeper exploration. Further, Matplotlib and Seaborn were used for advanced visualizations like correlation matrices and count plots, aiding in identifying relationships between variables and the distribution of categorical data. We also reviewed the dataset's data types for additional insight. A significant aspect of our analysis was the use of a report function, generating and saving an HTML file to our GitHub that details key dataset characteristics, including variable counts, observations, and missing data. This comprehensive view was crucial in guiding our analytical approach for data analysis as well as future use when we deploy our prediction models.

Based on the tools we used for building our initial models, we started with some further data preprocessing. We recognized the need to convert categorical variables into a numerical format, a necessary step for most machine learning algorithms which typically require numerical input. This was adeptly handled through the use of the LabelEncoder from scikit-learn, a standard and efficient tool for such transformations. By applying this to key categorical features like 'CARRIER_NAME', 'DEP_TIME_BLK', and 'DEPARTING_AIRPORT', we effectively transformed them into a machine-readable format. Additionally, the implementation of StandardScaler for normalizing our dataset was a crucial step. This normalization is vital, especially when employing algorithms like KNN and Logistic Regression, as these models are sensitive to the scale of the data. The scaler ensures each feature contributes proportionately, eliminating any undue influence of features with larger magnitudes.

Moving to the selection and application of machine learning models, our code demonstrates a keen understanding of model applicability. The first choice, Logistic Regression, is a wise baseline model for a binary classification problem like predicting flight delays. Its simplicity and interpretability make it an excellent starting point for such tasks. Then, we shifted gears to K-Nearest Neighbors (KNN), a model that stands out for its simplicity and effectiveness in classification problems. By classifying samples based on the majority class of their nearest neighbors, KNN provides a way to make predictions without assuming a specific distribution of the data, an advantage in complex real-world scenarios. Finally, our use of Decision Trees demonstrates an inclination to capture potentially non-linear relationships in the dataset. Decision Trees are particularly adept at handling complex datasets by breaking down the data space into smaller regions.

Through these models, our project encapsulates a thorough and varied approach to predictive modeling, showcasing our ability to adapt and apply different methodologies to glean insights from the data at hand.

6. Results

Our work focused on exploring various classification models to address the problem statement using a dataset that required cleaning and preprocessing. We evaluated the performance of three classification models: Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree Classifier.

Performance Measures:

Accuracy: We used accuracy as the primary performance measure to assess the effectiveness of each model in correctly classifying instances within the dataset.

F1 score: We used F1 score because it focusses on the prediction of the positive class.

ROC AUC: We used ROC AUC because it performs well on imbalanced data.

Results:

Baseline Model: As a baseline model, we considered logistic regression due to its simplicity and widespread applicability in classification tasks. Logistic regression achieved an accuracy of approximately 80.83% on the preprocessed dataset.

Primary Models: For a primary model, we used both K-Nearest Neighbors as well as Decision Tree Classifier due to their more robust nature. Despite a more nuanced approach, the primary models performed no better than our baseline model on accuracy, with small improvements to F1 score and ROC AUC.

Primary Model Comparisons:

Model	**Test accuracy**	**Test F1 score**	**Test ROC AUC**
Logistic Regression	80.83%	0.03	50.60%
K-Nearest Neighbors (KNN) (k=5)	80.06%	0.31	58.60%
Decision Tree Classifier	76.8%	0.32	58.60%

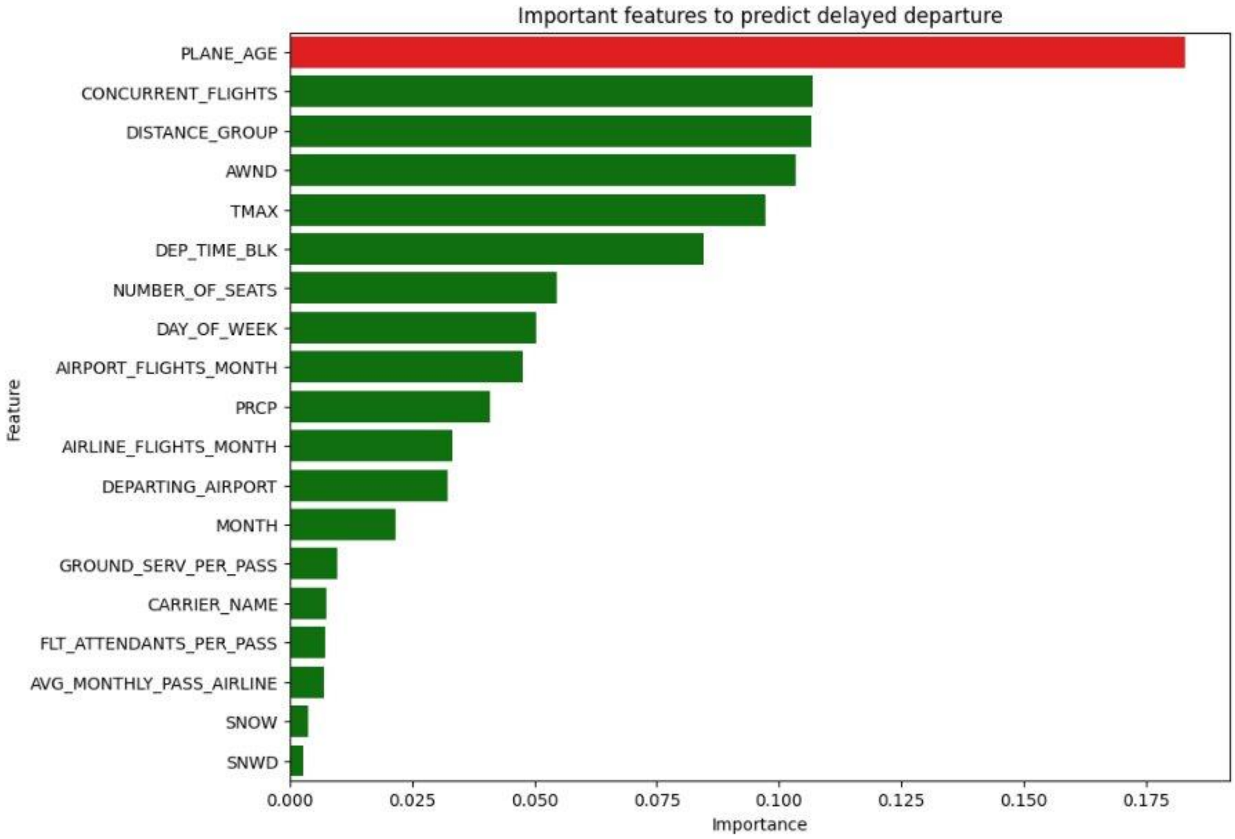
Explanation of Differences:

1. Logistic Regression vs. KNN: Both logistic regression and KNN achieved similar accuracy levels, with logistic regression slightly outperforming KNN. This could be attributed to the nature of the dataset and the effectiveness of logistic regression in handling linear relationships between features and target variables.
2. Logistic Regression vs. Decision Tree Classifier: Logistic regression outperformed the decision tree classifier in terms of accuracy. Decision trees tend to overfit the training data, which might have led to a lower accuracy compared to logistic regression, especially in this scenario where the dataset was relatively simple.

Discussion and Improvements

- The poor F1 score for Logistic Regression is likely due to the imbalance of the target variable. There are more normal flights than there are delayed flights. A possible solution to this problem would be to use R-SMOTE to balance out the distribution.
- Model Complexity and Overfitting: Decision Trees, in particular, are prone to overfitting. Pruning the tree or using ensemble methods like Random Forests could improve performance.

- Hyperparameter Tuning: For KNN, we chose k=5. Exploring different values of 'k' through cross-validation could yield better results.
- Alternative Models: Depending on the complexity of our data, other models like Support Vector Machines, Random Forests, or Neural Networks might offer improved accuracy.
- The visualization below shows the feature importance for the Decision Tree classifier. Plane_age stands out as the most important feature.



7. Conclusions and Future Work

It was surprising that Logistic Regression performed just as well as the more complex models we used in terms of accuracy. For Decision Tree classifier, the biggest takeaway was that plane_age was the most important feature used by the model.

Given more time, we would explore alternate models, such as Neural Networks or SVM. Improving the F1 score of Logistic Regression by using R-SMOTE would also be a priority.

In the end, we were able to build a few machine learning models to accurately predict whether a flight will be delayed or not. By leveraging our extensive dataset of flight schedules, weather, airport operations, and passenger trends, these models discern key delay factors and enhance forecasting accuracy, aiding stakeholders in preempting and addressing disruptions. Such advancements improve reliability for passengers, optimize airline operations, and enhance resource utilization for airports, fostering a more sustainable and economically beneficial airline industry.

8. Appendix

GitHub: https://github.com/Evan-Burch/2019-Airline-Delays

Google Colab:
https://colab.research.google.com/drive/1SVY3tGdtDEr2mk82kmbMPaCevmzpVC-T?usp=sharing