Machine Learning Report

Siddharth Bahekar
Rachita Shah
College of Engineering
Northeastern University
Toronto,ON
bahekar.si@northeastern.edu
shah.rachit@northeastern.edu

Abstract

In this report, we have applied and explored the performance of four different supervised learning algorithms: Logistic Regression, Decision Tree, Gradient Boosting, and Random Forest, on the Flight Delay 2023 dataset. This report outlines the exploratory data analysis (EDA) and modeling process undertaken to train and evaluate these algorithms. The learning curve, model complexity, and model training have been performed and analyzed on the Flight Delay 2023 dataset to assess the effectiveness of each algorithm.

1. Dataset

The Flight Delay 2023 dataset provides a comprehensive set of attributes for predicting flight delays, allowing for an in-depth analysis of various factors contributing to flight performance. The dataset contains 12,373 rows and 21 columns. The problem at hand is to classify whether a flight will be delayed based on several operational attributes such as the number of flights, delay reasons (e.g., weather, carrier, security), and other relevant flight information. This dataset provides a comprehensive view of various factors, allowing for in-depth exploratory data analysis and predictive modeling.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12373 entries, 0 to 12372
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   year                 12373 non-null  int64
 1   month                12373 non-null  int64
 2   carrier              12373 non-null  object
 3   carrier_name         12373 non-null  object
 4   airport              12373 non-null  object
 5   airport_name         12373 non-null  object
 6   arr_flights          12355 non-null  float64
 7   arr_del15            12355 non-null  float64
 8   carrier_ct           12355 non-null  float64
 9   weather_ct           12355 non-null  float64
 10  nas_ct               12355 non-null  float64
 11  security_ct          12355 non-null  float64
 12  late_aircraft_ct     12355 non-null  float64
 13  arr_cancelled        12355 non-null  float64
 14  arr_diverted         12355 non-null  float64
 15  arr_delay            12355 non-null  float64
 16  carrier_delay        12355 non-null  float64
 17  weather_delay        12355 non-null  float64
 18  nas_delay            12355 non-null  float64
 19  security_delay       12355 non-null  float64
 20  late_aircraft_delay  12355 non-null  float64
dtypes: float64(15), int64(2), object(4)
memory usage: 2.0+ MB
None
```

2. Dataset Description

Flight Delay 2023

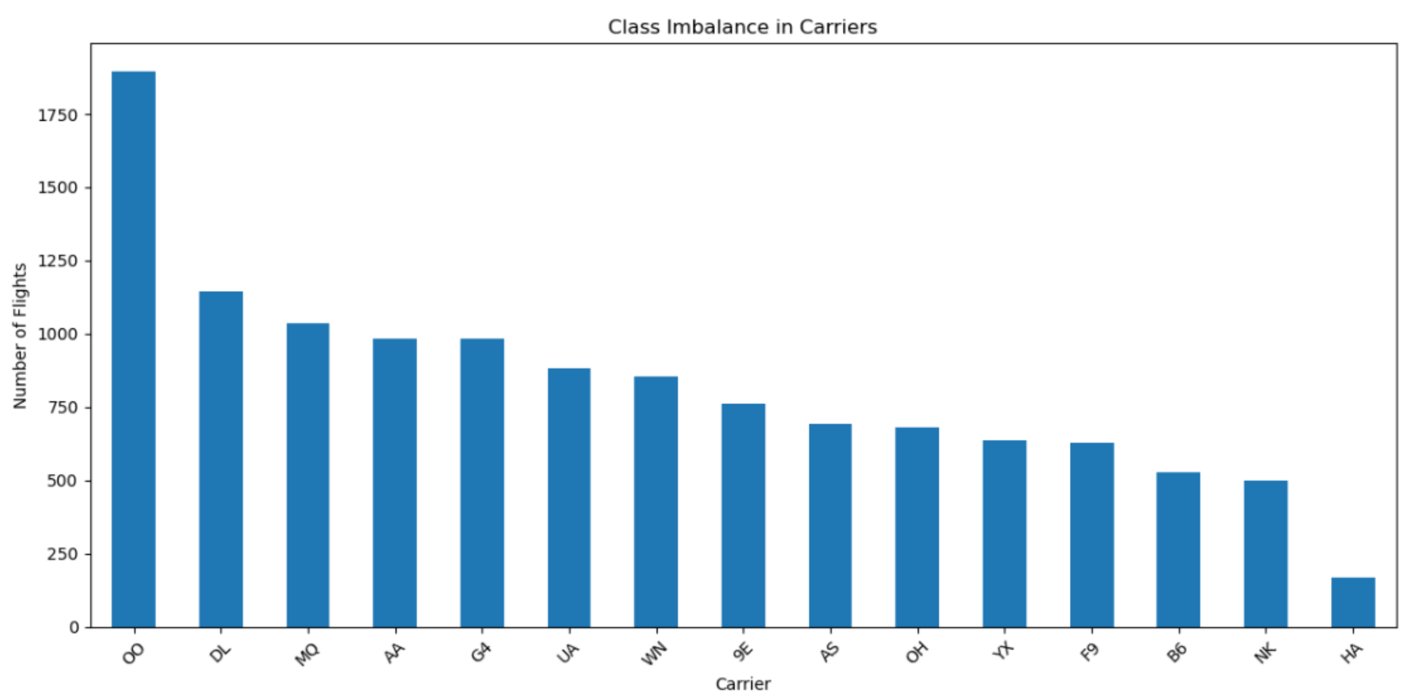
```
# Display summary statistics
data.describe()
```

	year	month	arr_flights	arr_del15	carrier_ct	weather_ct	nas_ct	security_ct	late_aircraft_ct
count	12373.0	12373.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000
mean	2023.0	4.512083	367.901416	82.842088	27.730462	2.996298	21.231170	0.250875	30.633336
std	0.0	2.291578	1016.136066	223.618477	65.802296	9.546003	65.005592	0.972349	94.869611
min	2023.0	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2023.0	3.000000	51.000000	8.000000	2.960000	0.000000	1.000000	0.000000	1.890000
50%	2023.0	5.000000	97.000000	21.000000	8.070000	0.880000	4.070000	0.000000	6.000000
75%	2023.0	7.000000	250.500000	59.000000	23.105000	2.380000	12.870000	0.000000	19.950000
max	2023.0	8.000000	20084.000000	4142.000000	1293.910000	266.420000	1068.200000	24.460000	2069.070000

	arr_cancelled	arr_diverted	arr_delay	carrier_delay	weather_delay	nas_delay	security_delay	late_aircraft_delay
count	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000	12355.000000
mean	6.120275	1.023553	5865.656819	2089.358478	315.852772	1076.464023	11.837394	2372.144152
std	22.909182	4.421002	18123.488293	6468.495635	1071.609355	3660.386878	51.388823	8050.544733
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	462.000000	153.000000	0.000000	34.000000	0.000000	99.000000
50%	1.000000	0.000000	1347.000000	486.000000	42.000000	168.000000	0.000000	411.000000
75%	4.000000	1.000000	3886.500000	1555.500000	219.000000	575.000000	0.000000	1510.500000
max	584.000000	125.000000	438783.000000	162563.000000	25206.000000	72056.000000	1477.000000	227959.000000

3. Dataset Imbalance

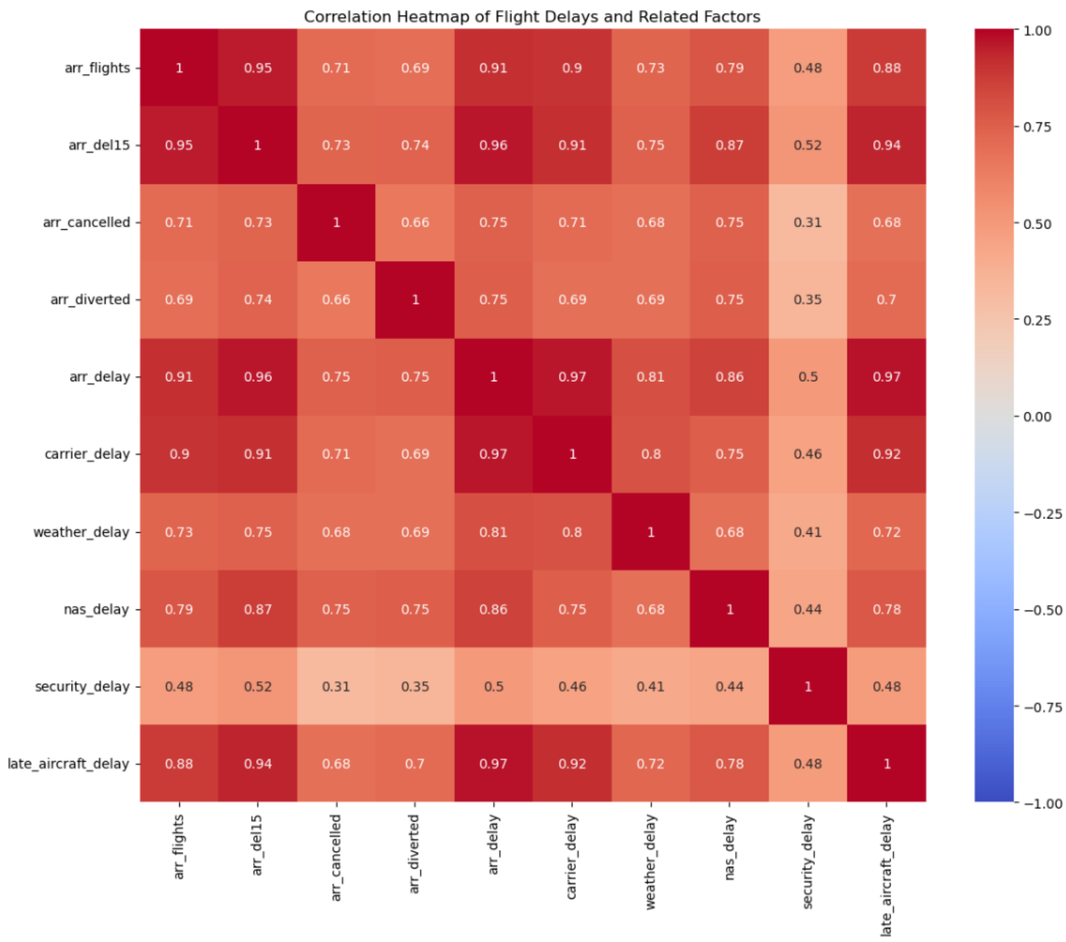
In this dataset we choose not to balance the data regarding the 'carrier' variable because it naturally reflects the distribution of flights among airlines, which is typical in the aviation industry. Our primary focus was to predict flight delays, not to classify the carriers. While there were differences in the number of flights for each airline, the imbalance was not severe; major carriers like Delta and American Airlines had many flights, while JetBlue had fewer, but the difference wasn’t extreme. Adjusting for this imbalance could introduce bias or distort relationships between variables. Moreover, our predictive models, such as Logistic Regression and Random Forest, can effectively handle moderate class imbalances, making balancing unnecessary for our analysis.




```
Imbalance ratio (max/min): 11.29
```

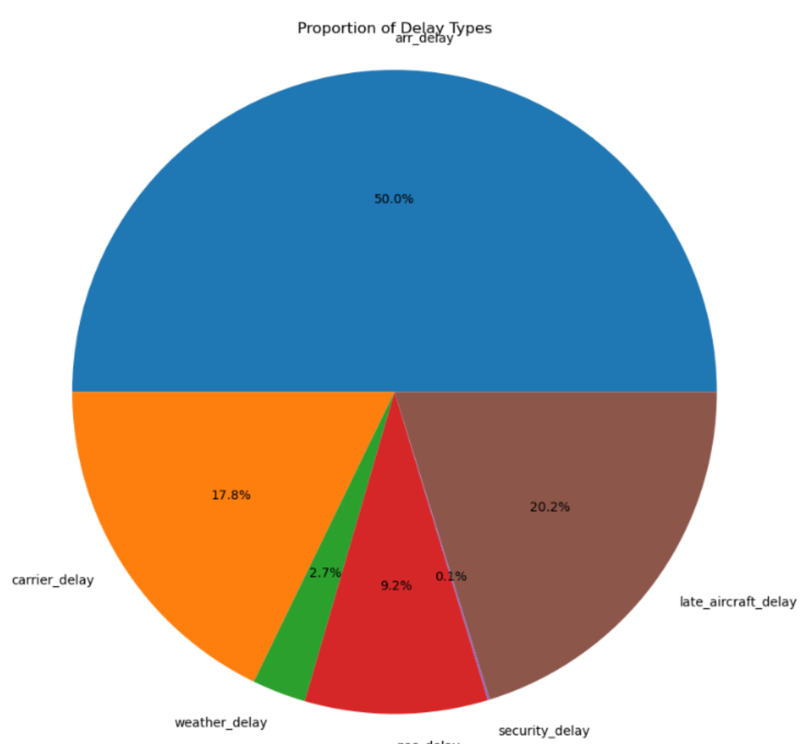
4. Correlation Matrices

We generated a correlation heatmap to visualize the relationships between different features in the Flight Delay 2023 dataset. A strong positive correlation was observed between 'carrier delay' and 'arrival delay,' indicating that longer delays attributed to carriers are closely associated with overall flight delays. The analysis revealed a weak correlation between 'security canceled' and 'arrival canceled,' indicating that security-related delays do not significantly affect the number of canceled flights upon arrival. These correlations provided valuable insights into which features might be most predictive of flight delays, guiding our feature selection process and helping us understand the underlying patterns in the data.



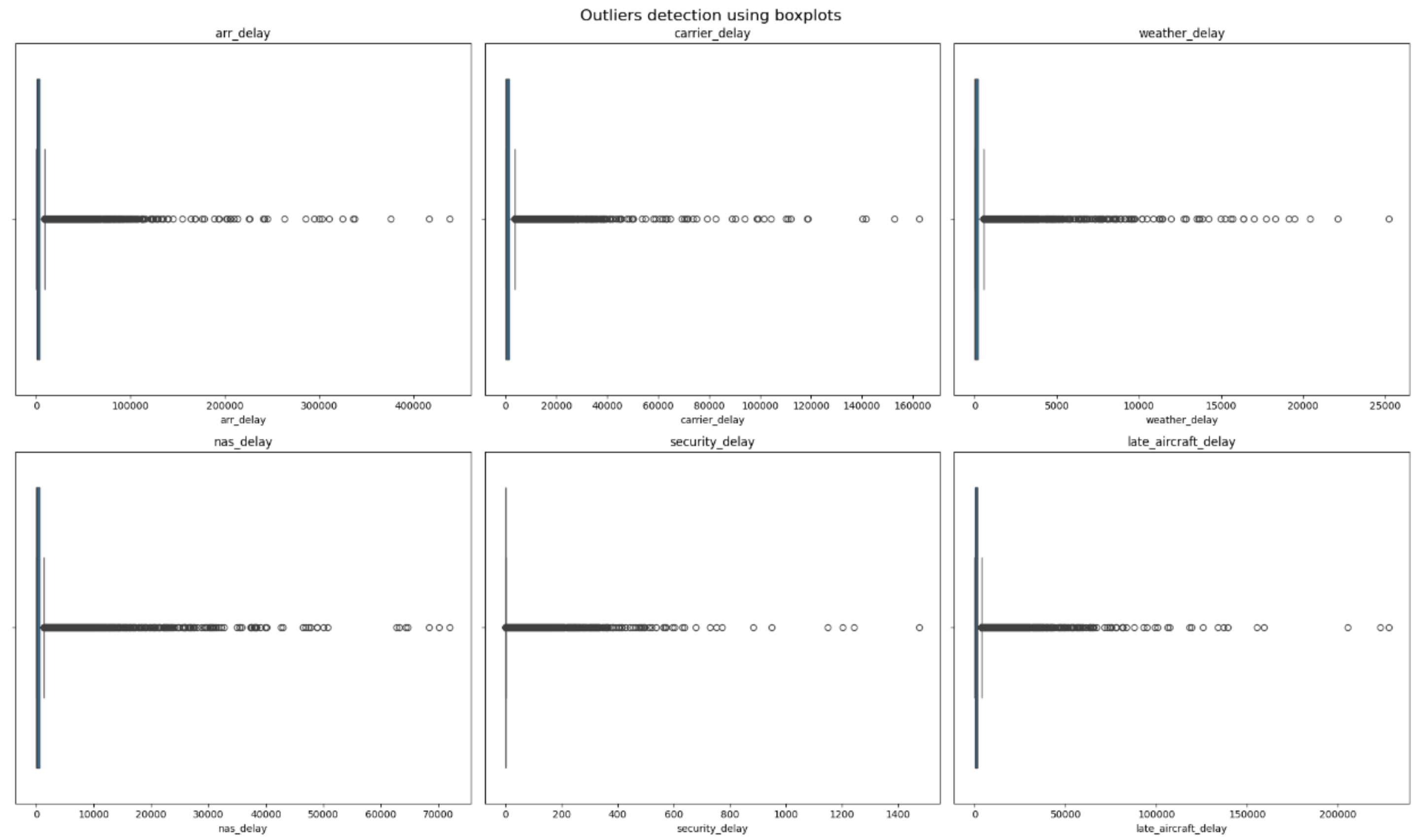
Proportion of Delay Types:

We created a pie chart to show the proportions of different delay types in the Flight Delay 2023 dataset. Late aircraft delays were the largest, making up 50% of all delays, followed by carrier delays (17.8%) and NAS delays (9.2%). Weather delays (2.7%) and security delays (0.1%) had a smaller impact. This chart highlights the major contributors to flight delays.



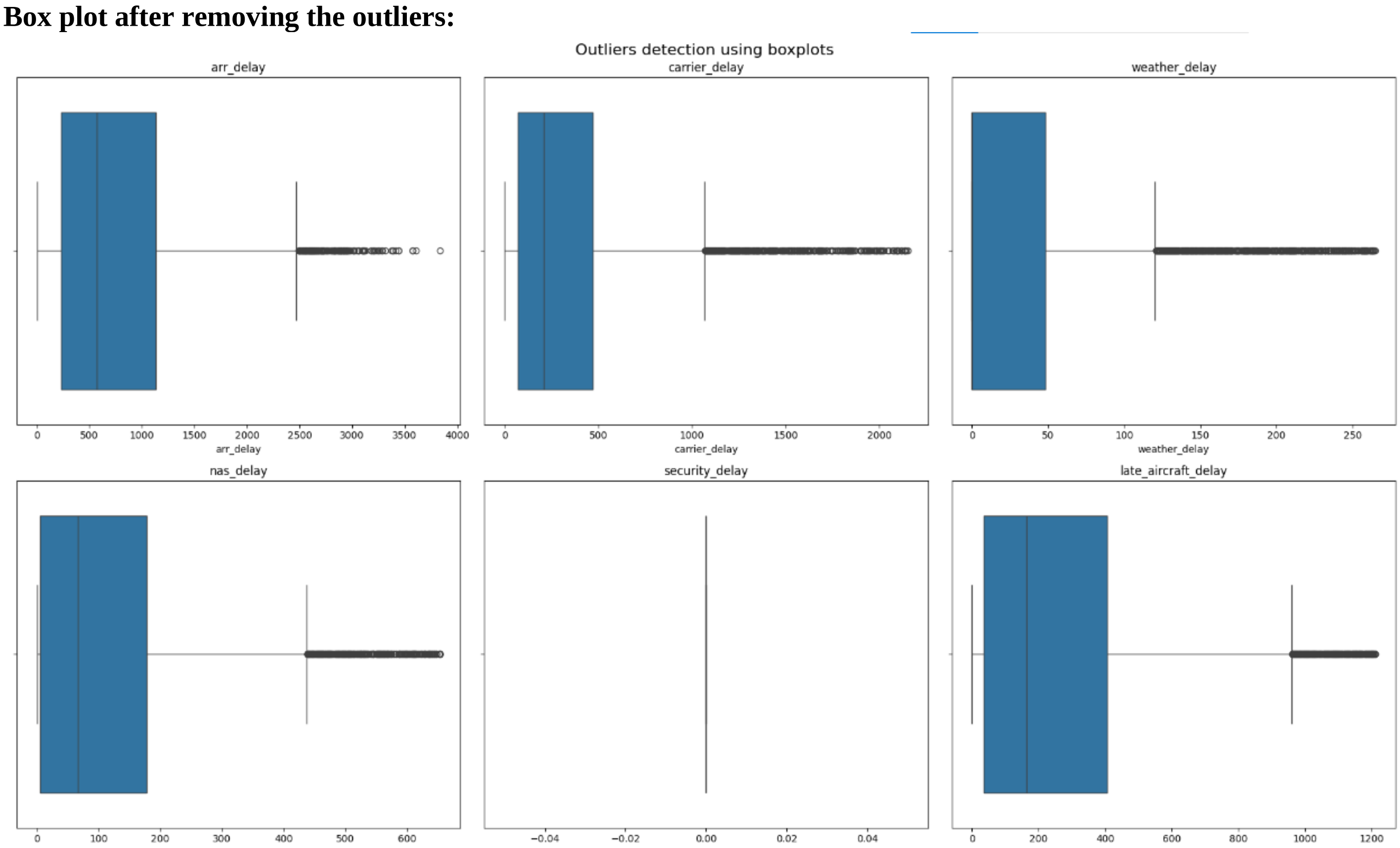
5. Outliers

In this section, we focus on identifying and removing outliers from important columns in the dataset using the Interquartile Range (IQR) method. This method calculates lower and upper bounds based on the IQR, which allows us to identify and remove data points that fall outside these limits. The limits are defined as follows:-
Lower Bound: (Q1 - 1.5 times IQR)
Upper Bound: (Q3 + 1.5 times IQR)
where Q1 and Q3 are the first and third quartiles, respectively.
We analyze six numerical columns related to flight delays, including `arr_delay`, `carrier_delay`, `weather_delay`, and others. Boxplots are used to visually detect these outliers. After removing the outliers, the dataset is cleaned, which helps enhance the accuracy of our models. We also compare the original dataset size to the size after outlier removal to demonstrate the effect of this process.




```
Original dataset shape: (12373, 21)
Dataset shape after removing outliers: (7054, 21)
```

Box plot after removing the outliers:



6. Null Values

In our analysis of the flight delay dataset, we decided not to modify the null values for several reasons. First, the number of null values was small compared to the overall dataset, meaning they would have a minimal effect on our findings. Keeping these null values helped us preserve the original data's integrity and avoid potential bias that could come from methods like filling in or removing these values. In flight data, null values can have important meanings; for example, a null value in a delay category might indicate that there was no delay of that type for a specific flight. Our main goal was to identify overall patterns and trends in flight delays, which we could achieve with the existing data. By not altering the null values, we also allowed for more flexibility in our analysis, making it easier to use different techniques to handle missing data when needed. This decision balanced the need for accurate data representation with the practical considerations of conducting our analysis efficiently.

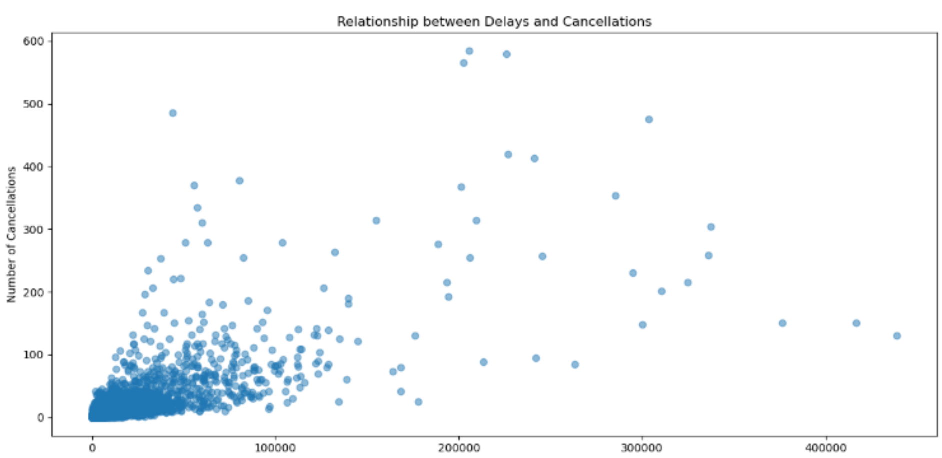
```
print(df.isnull().sum())
```

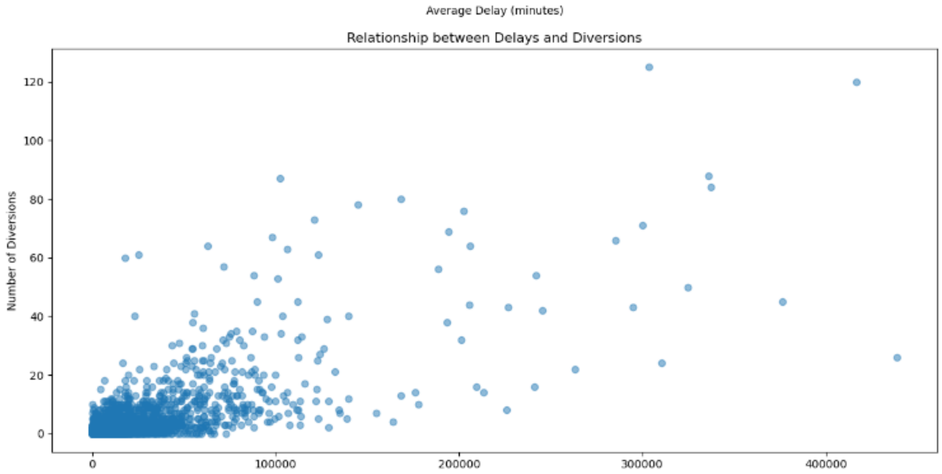
```
year                   0
month                  0
carrier                0
carrier_name           0
airport                0
airport_name           0
arr_flights           18
arr_del15             18
carrier_ct            18
weather_ct            18
nas_ct                18
security_ct           18
late_aircraft_ct      18
arr_cancelled         18
arr_diverted          18
arr_delay             18
carrier_delay         18
weather_delay         18
nas_delay             18
security_delay        18
late_aircraft_delay   18
dtype: int64
```

7. Determining Relationships

In the "Relationship between Delays and Cancellations," we plotted the average arrival delay (arr_delay) against the number of cancellations (arr_cancelled). Each point on the plot represented a specific airline and airport. The purpose of this visualization was to determine if airports or airlines with higher average delays also tend to have more cancellations. This could potentially indicate whether longer delays are associated with a higher likelihood of flight cancellations. Similarly, for "Relationship between Delays and Diversions" we used the same x-axis (average arrival delay) but plotted the number of diversions (arr_diverted) on the y-axis. This visualization aimed to explore whether there is any connection between the length of delays and the frequency of flight diversions.

The data indicates that if there is a strong connection between higher delays and increased cancellations, it suggests that addressing the factors leading to longer delays could also help reduce cancellation rates.





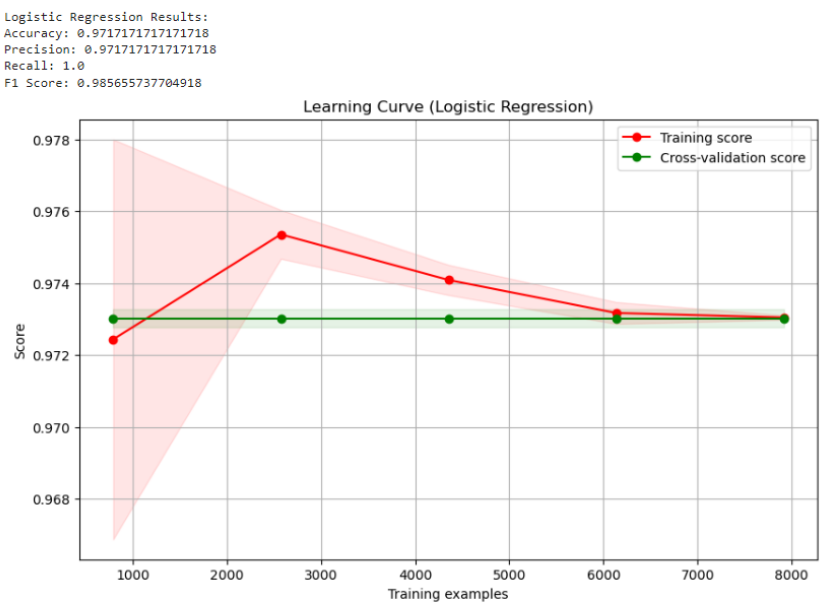
8. Why are these datasets interesting?

We find this dataset interesting because it provides important information about flight delays and performance in the U.S. aviation industry, which affects millions of travellers every year. By looking at this data, airline operators and airport authorities can better understand what causes flight disruptions, helping them make travel smoother for everyone. The dataset includes various details like arrival delays, cancellations, and reasons for these issues, allowing a clear look at how things like weather and airline performance impact our flights.

9. Machine Learning Model

Logistic Regression:

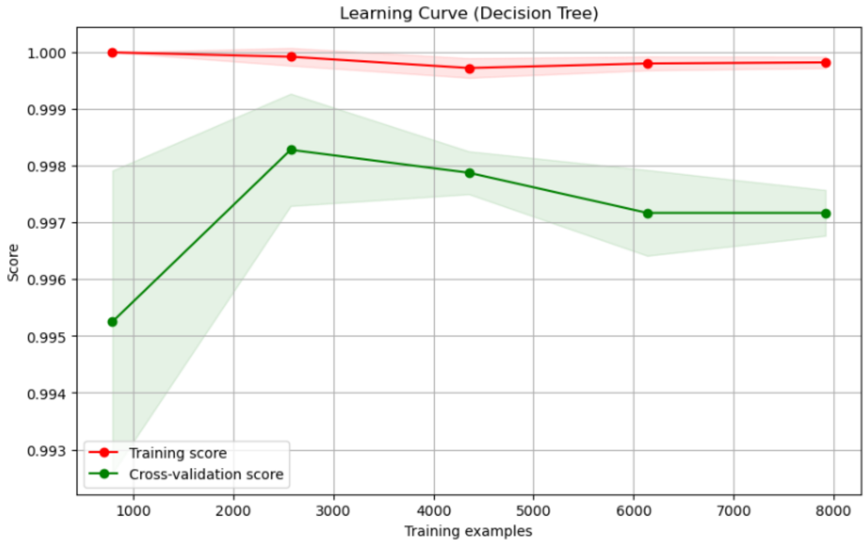
We employed Logistic Regression, a fundamental classification algorithm, to predict flight delays. We utilized the sklearn.linear_model.LogisticRegression class with default parameters, including the 'liblinear' solver and L2 regularization. The model was trained on scaled features to ensure all variables contributed equally to the prediction. Logistic Regression uses the sigmoid function as its activation function, transforming the linear combination of inputs into a probability between 0 and 1. We chose this algorithm for its interpretability and efficiency in handling binary classification problems. The performance was evaluated using accuracy, precision, recall, and F1 score metrics. A learning curve was plotted to visualize the model's performance as a function of training set size, helping to identify potential overfitting or underfitting issues.



Decision Tree Algorithm:

We implemented the sklearn.tree.DecisionTreeClassifier with default parameters, including the Gini impurity criterion for measuring the quality of splits. Decision Trees do not require an activation function as they make decisions based on feature thresholds. We selected this algorithm for its ability to capture non-linear relationships and provide easily interpretable results through its tree structure. The model's performance was assessed using the same metrics as Logistic Regression. Additionally, we generated a learning curve to understand how the model's performance changed with increasing training data, which is particularly useful for identifying overfitting in Decision Trees.

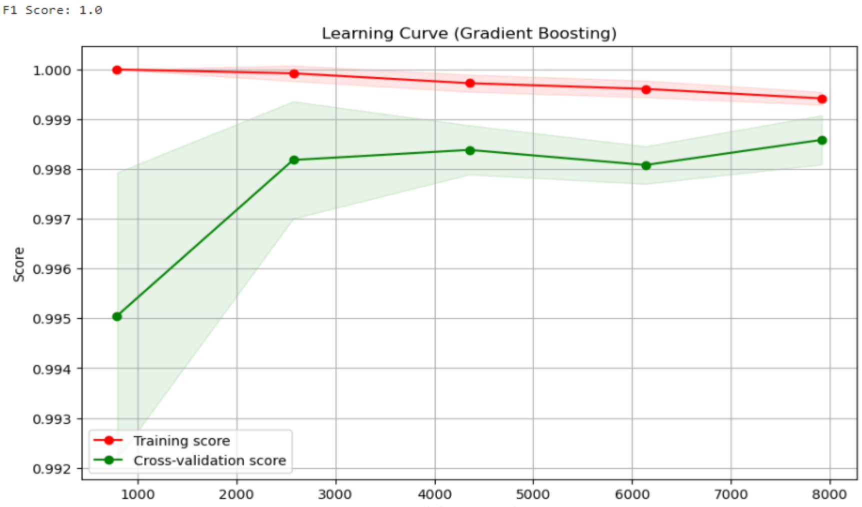
```
Decision Tree Results:
Accuracy: 0.997979797979798
Precision: 1.0
Recall: 0.997920997920998
F1 Score: 0.9989594172736732
```



Gradient Boosting:

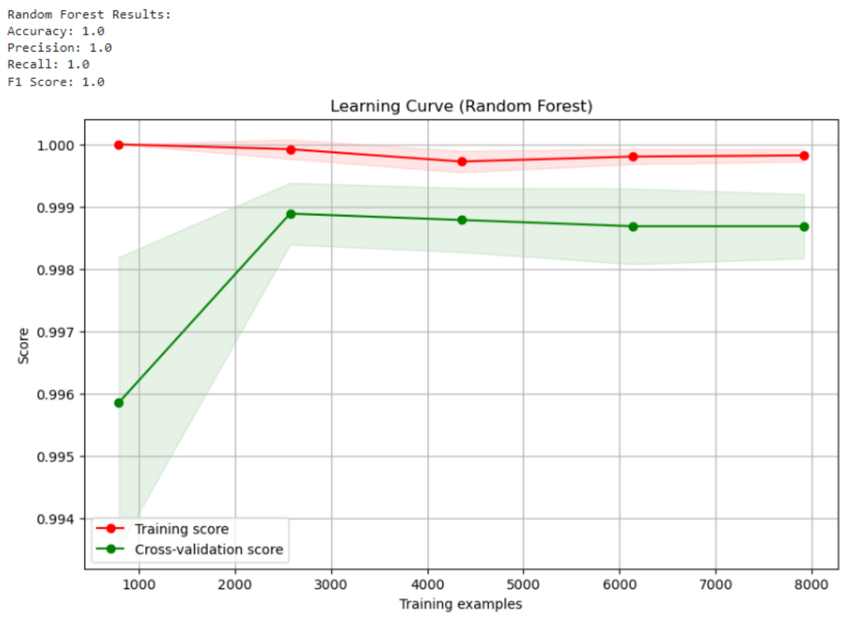
We implemented Gradient Boosting using sklearn.ensemble.GradientBoostingClassifier. This algorithm builds trees sequentially, with each tree correcting the errors of the previous ones. We used the default parameters, including 100 estimators and a learning rate of 0.1. Gradient Boosting typically uses shallow decision trees as weak learners, with the default maximum depth of 3. The algorithm's strength lies in its ability to create a strong predictive model through the combination of weak learners. We evaluated its performance using the same metrics as the other algorithms and plotted a learning curve to understand how the model's complexity and performance evolved with increasing training data.

```
Gradient Boosting Results:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0
```



Random Forest:

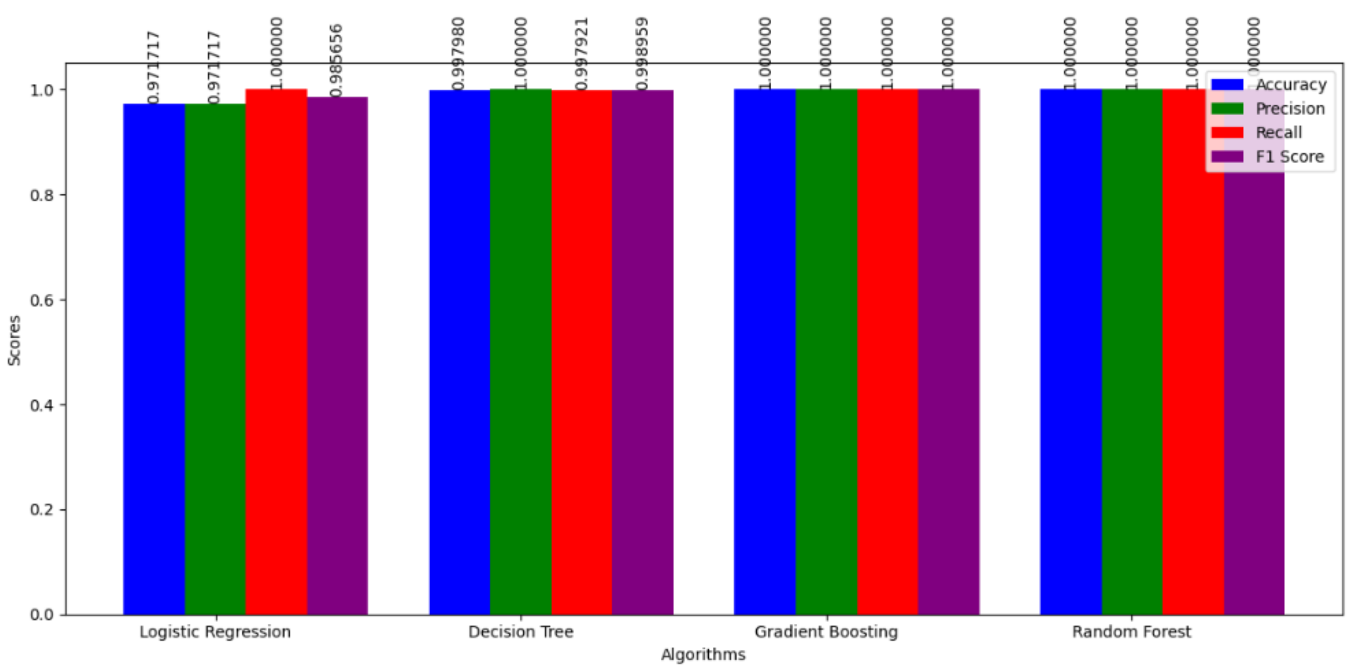
The Random Forest algorithm, implemented using sklearn.ensemble.RandomForestClassifier, was chosen to leverage the power of ensemble learning. This method creates multiple decision trees and aggregates their predictions, typically resulting in improved generalization compared to a single decision tree. We used the default parameters, including 100 trees and the Gini impurity criterion. Random Forests inherently perform feature selection and are less prone to overfitting. The model's performance was evaluated using the standard classification metrics, and a learning curve was plotted to visualize how the ensemble's performance improved with more training data.



10. Conclusion

Among all the algorithms tested, Gradient Boosting and Random Forest delivered the best performance, achieving perfect scores in accuracy, precision, recall, and F1 score. While Logistic Regression and Decision Tree also showed strong results, their slight differences in metrics indicate they may not generalize as effectively as the ensemble methods.

Between Gradient Boosting and Random Forest, both are excellent choices for handling complex patterns in flight delay predictions. However, Random Forest stands out as the more suitable option for this dataset. It offers greater stability, faster training times, and a lower risk of overfitting, particularly when working with large or varied data. Although Gradient Boosting is powerful, it is more complex and requires careful tuning. Therefore, Random Forest is the more practical and reliable choice for predicting flight delays in this analysis.



11. References

- https://www.kaggle.com/datasets/sriharshaeedala/airline-delay
- https://archive.ics.uci.edu/
- https://github.com/xzachx/Flight-Delays/blob/master/flight_delays.ipynb
- https://scikit-learn.org/1.5/index.html
- https://chat.openai.com/