# Flight Delay Prediction

View on GitHub

## About Us

Group #54

Group Members: Abdulaziz Albahar, Rishi Borra, Long Lam, My Phung, Joseph Thomas

Mentor: Zini Chakraborty

# Final Report

## Introduction and Background

### Literature Review

We will be looking at data from all past airplane departures to extract meaningful patterns about airline and flight delays. There has been a lot of past research done on this topic and many have proposed applying decision tree models such as AdaBoost and as well as combining both logistic regression and decision tree models to make predictions [3,4].

### Dataset Description

We will be using data from the Bureau of Transportation Statistics to ensure that we have the most accurate and up-to-date data (June 2024). We will be using all the data that is offered by the source, which is from January 2003 to the most latest entry, which is June 2024. This dataset has 20+ features and includes information about the airline, weather, airport and has over 500,000 data points per month.

### Dataset Link

View it here↗

## Problem Definition

### Problem

Flight delays are a significant problem in the aviation industry, impacting millions of passengers and resulting in economic losses and operational challenges. In 2007, the overall direct cost on airlines and passengers amounted to $28.9 billion, with an additional impact of $4 billion on the GDP [2].

### Motivation

There is a clear need for machine learning solutions that can learn from historical data and identify patterns to provide early warning of potential delays. A reliable prediction model could help airlines optimize their operations, allow passengers to plan their journeys more effectively, and ultimately reduce the economic impact of delays.

## Methods

### Data Preprocessing:

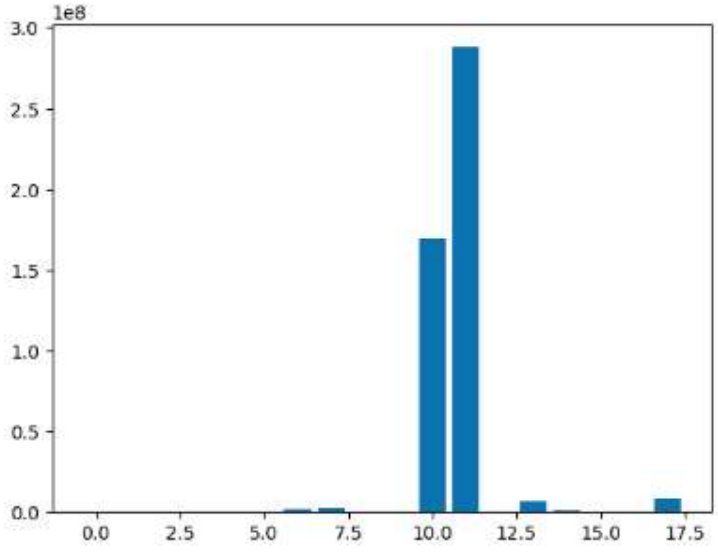
- Since our dataset was collected in batches, we started by merging these batches into a unified, complete dataset.
- We then removed duplicate entries from the complete dataset to eliminate any redundancy that may have occurred across batches during data collection.
- We designated our label column as the flight classifier, where on-time flights are represented as 0 and delayed flights as 1
- We removed any data points containing NaN values in the label column to ensure the dataset includes only flights labeled as either delayed or on-time.
- We excluded several features containing delay times (in minutes) as we determined that these could easily reveal the target label during classification
- Our dataset includes columns detailing delay times (in minutes) by types such as weather, late aircraft arrival, and security. We wanted to incorporate these delay types in our classification without allowing them to directly reveal the target label. Therefore, we modified these features by converting any delay greater than zero minutes to 1, indicating the presence of that delay type, and delays of exactly zero minutes to 0. In this way, a value of 1 signifies that a particular delay factor, such as bad weather, was involved for that flight.
- We splitted into training and testing datasets using scikit-learn.
- We then ordinal encode the X dataset and label encode the y label
- To address the class imbalance, where on-time flights significantly outnumber delayed flights, we downsampled the dataset to equalize the number of flights in each class. This helps to reduce overfitting and improves model performance.

## Feature Selection

We performed the Chi-Squared and Mutual Information test from scikit-learn. These methods help to prioritize features, improving model interpretability and potentially reducing dimensionality by focusing on the most significant variables.

### Chi Squared

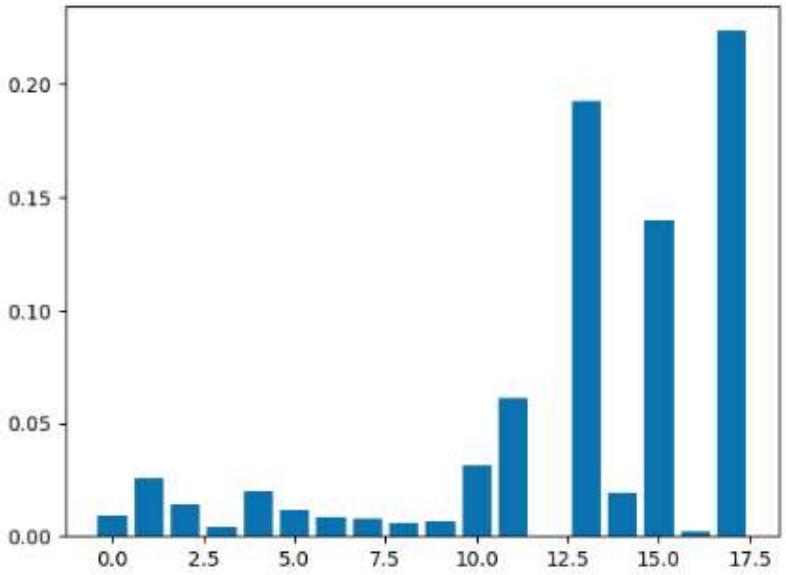
```
Feature (0) YEAR: 39109.483068
Feature (1) QUARTER: 2590.567289
Feature (2) MONTH: 3030.409596
Feature (3) DAY_OF_MONTH: 17084.767803
Feature (4) DAY_OF_WEEK: 2865.981582
Feature (5) OP_CARRIER_AIRLINE_ID: 99.571519
Feature (6) ORIGIN_AIRPORT_SEQ_ID: 1598843.984088
Feature (7) ORIGIN_CITY_MARKET_ID: 2833592.127641
Feature (8) DEST_AIRPORT_SEQ_ID: 565711.796257
Feature (9) DEST_CITY_MARKET_ID: 209385.954874
Feature (10) CRS_DEP_TIME: 169480273.448274
Feature (11) DEP_TIME: 287835731.789073
Feature (12) CANCELLED: 9617.146234
Feature (13) CARRIER_DELAY: 7125916.761822
Feature (14) WEATHER_DELAY: 966734.458429
Feature (15) NAS_DELAY: 32946.059007
Feature (16) SECURITY_DELAY: 55189.927454
Feature (17) LATE_AIRCRAFT_DELAY: 8003892.121785
```




The Chi-Squared test measures the association between each feature and the target variable and tells which features contribute most to classification. Notably, **DEP_TIME** (287,835,731) and **CRS_DEP_TIME** (169,480,273) have exceptionally high scores, suggesting a strong relationship with flight delays. Other features like **LATE_AIRCRAFT_DELAY** (8,003,892) and **CARRIER_DELAY** (7,125,917) can also be critical for the model. Features with lower scores, such as **OP_CARRIER_AIRLINE_ID** (99.57), may be less informative in the model.

### Mutual Information

```
Feature (0) YEAR: 0.009253
Feature (1) QUARTER: 0.025498
Feature (2) MONTH: 0.013996
Feature (3) DAY_OF_MONTH: 0.003725
Feature (4) DAY_OF_WEEK: 0.020009
Feature (5) OP_CARRIER_AIRLINE_ID: 0.011578
Feature (6) ORIGIN_AIRPORT_SEQ_ID: 0.008189
Feature (7) ORIGIN_CITY_MARKET_ID: 0.007756
Feature (8) DEST_AIRPORT_SEQ_ID: 0.006081
Feature (9) DEST_CITY_MARKET_ID: 0.006452
Feature (10) CRS_DEP_TIME: 0.031098
Feature (11) DEP_TIME: 0.061224
Feature (12) CANCELLED: 0.000000
Feature (13) CARRIER_DELAY: 0.192608
Feature (14) WEATHER_DELAY: 0.019430
Feature (15) NAS_DELAY: 0.140075
Feature (16) SECURITY_DELAY: 0.001796
Feature (17) LATE_AIRCRAFT_DELAY: 0.223624
```




Mutual Information assesses the dependence between each feature and the target variable, identifying the most relevant predictors for flight delays. The **LATE_AIRCRAFT_DELAY** (0.223624), **CARRIER_DELAY** (0.192608), and **NAS_DELAY** (0.140075) features exhibited the highest scores, highlighting their strong relationship with the target variable. Additionally, **DEP_TIME** (0.061224) and **CRS_DEP_TIME** (0.031098) also demonstrated their importance in predicting delays. On the other hand, features with low scores like **CANCELLED** (0.000000) suggests that they may not contribute to the model's predictive power.

## ML Algorithms and Models

### Random Forest Algorithm

The **random forest algorithm** is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting. We use RandomForestClassifier from the scikit-learn library. A random forest model is well-suited for predicting flight delays due to its ability to handle intricate, non-linear relationships in the data and provide robust predictions with high accuracy. Flight delay prediction involves numerous variables such as weather conditions and departure times with complex interdependencies. Random forests excel here by creating an ensemble of decision trees that capture various aspects of the data, helping to uncover nuanced patterns that individual trees might miss. Furthermore, random forests are relatively resistant to overfitting, especially when dealing with a large number of features, making them an effective choice since generalizing from historical data is essential. Lastly, random forests are interpretable because they offer feature importance ranking, revealing which features most significantly impact flight delay predictions. By visualizing and analyzing individual decision trees, airlines can also gain insights into specific scenarios that lead to delays, providing actionable information for operational improvements. [5]

**Logistic Regression**

The **logistic regression algorithm** is a learning method that is used for binary classification. We use a LogisticRegression classifier from the scikit-learn library to predict if there will either be a delay or not be a delay. The logistic regression model is a simple and effective model for predicting flight delays and is highly interpretable. This makes the model great for understanding how specific features affect the final prediction as well as makes the model extremely efficient when it comes to working with large datasets. While logistic regression can't see complex non-linear patterns, we chose to use this model to see what kind of results we can extract and set up a baseline for our dataset.

**Neural Network**

**Neural Networks** are a gradient descent type of algorithm where datasets are forward propagated then back propagated through the network to train the model. We used the Keras library to design sequential hidden layers that our data will go through. Our neural network contains 3 fully connected hidden layers. It also used ReLU as our activation function to capture non-linear patterns in our dataset, and finally the sigmoid function to convert the output into soft assignment of a classification.

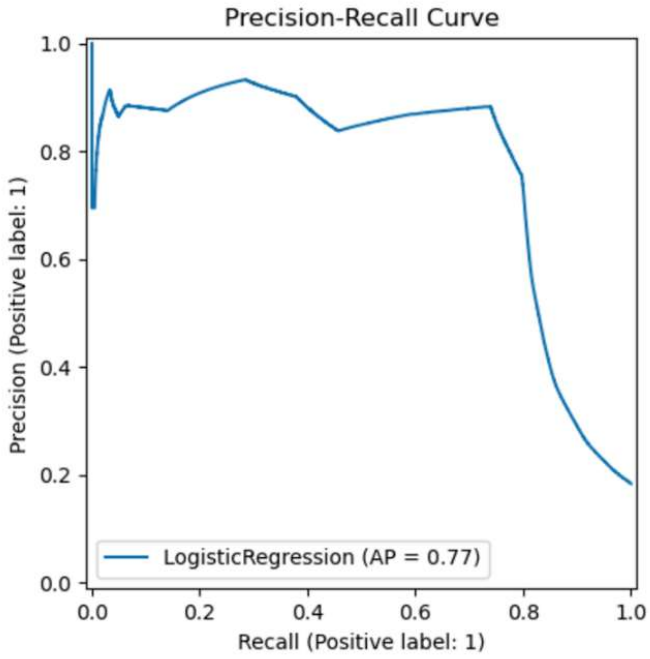
Before feeding the data for training, we normalize it to speed up training and prevent vanishing gradients.

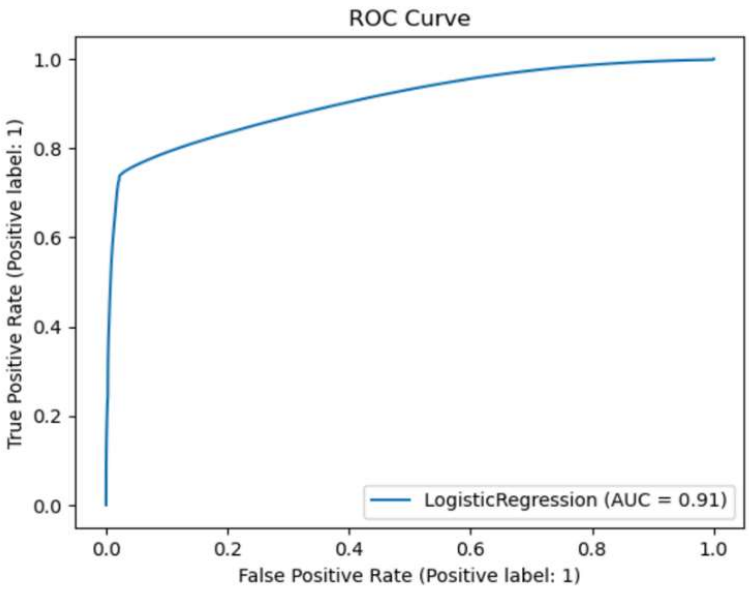
# Results and Discussion

## 1. Logistic Regression

### Model Trained on Raw Data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 1.00 | 0.91 | 33,525,954 |
| 1 | 0.88 | 0.14 | 0.24 | 7,516,590 |

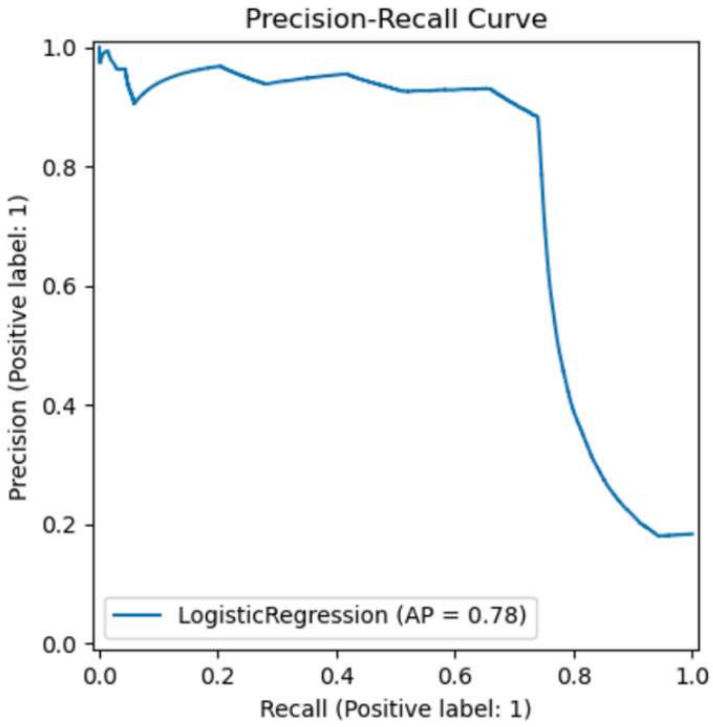


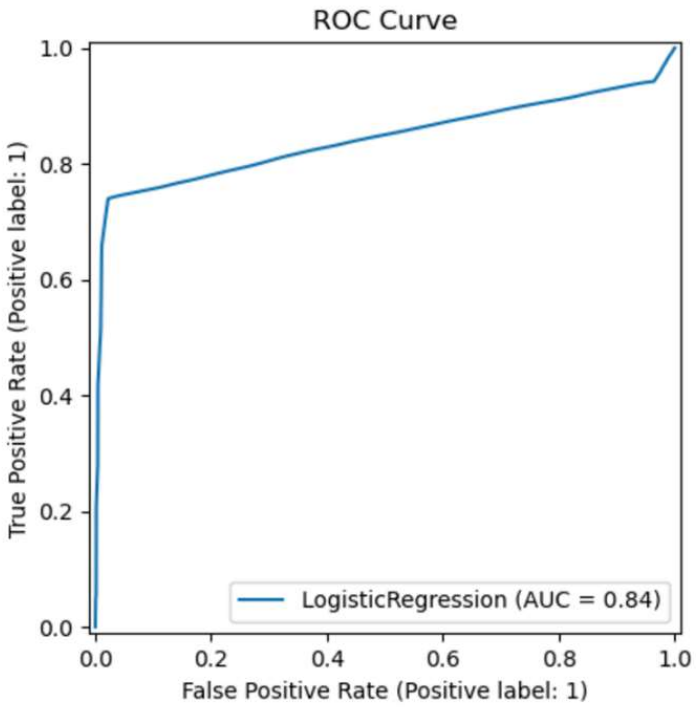




First, we trained the Logistic Regression model on the raw data and it achieved an accuracy of **83.89%**. The Precision-Recall Curve and the classification report indicate that the model achieves high precision (88%) for the delayed class (label 1), but recall is alarmingly low (14%). In contrast, the on-time class (label 0) shows excellent precision (84%) and recall (100%). The area under the ROC curve (0.91) suggests the model performs well overall. However, the disparity between the classes indicates that the model is biased toward the on-time class. This issue arises because the raw data lacks preprocessing so addressing this imbalance through preprocessing is critical to improving the model's ability to generalize and perform well across both classes.

### Model Trained on Preprocessed Data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.94 | 0.98 | 0.96 | 33,525,954 |
| 1 | 0.88 | 0.73 | 0.80 | 7,516,590 |

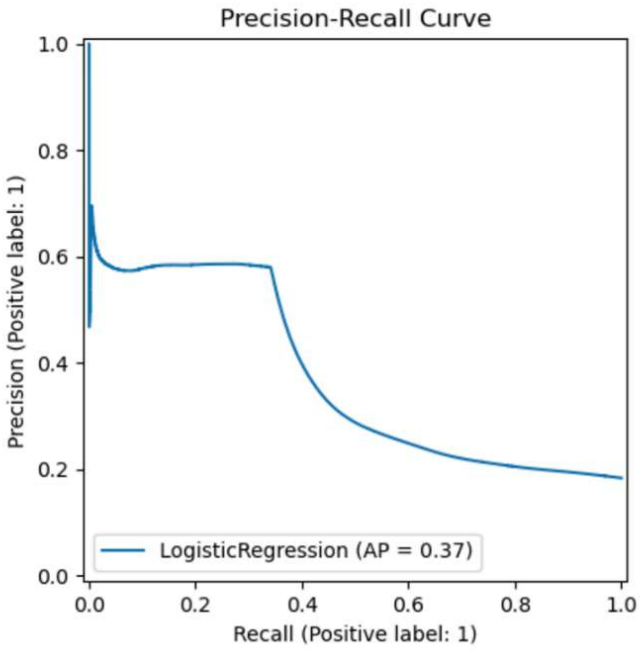


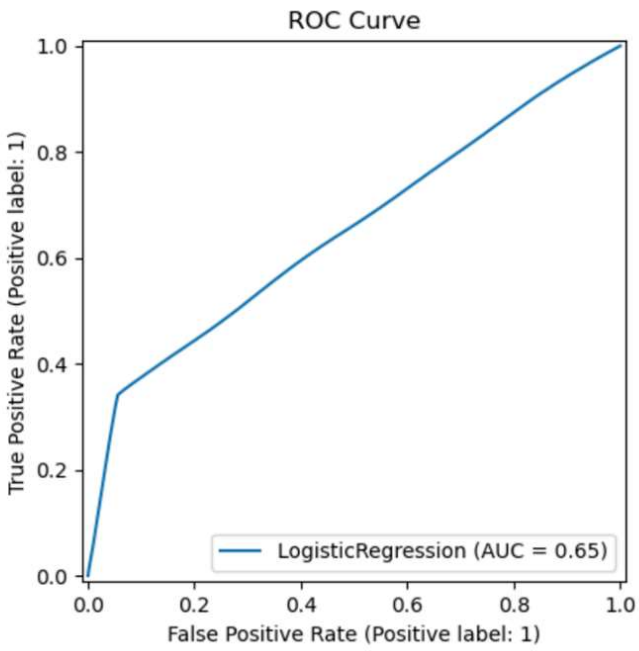




After training on preprocessed data, the logistic regression model achieved an accuracy of **93.43%**, which shows significant improvement compared to the raw data results. The classification report reflects balanced performance. These results suggest that preprocessing steps were effective in improving the model's ability to generalize across both classes. However, the Precision-Recall Curve only demonstrates a slightly higher AP (0.78) and the ROC curve had a lower AUC (0.84). To improve the prediction further, feature selection methods could be employed to identify and retain the most relevant predictors while eliminating redundant or irrelevant ones.

### Model Trained on Data without Top Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0** | 0.82 | 1.00 | 0.90 | 33,525,954 |
| **1** | 0.69 | 0.00 | 0.01 | 7,516,590 |



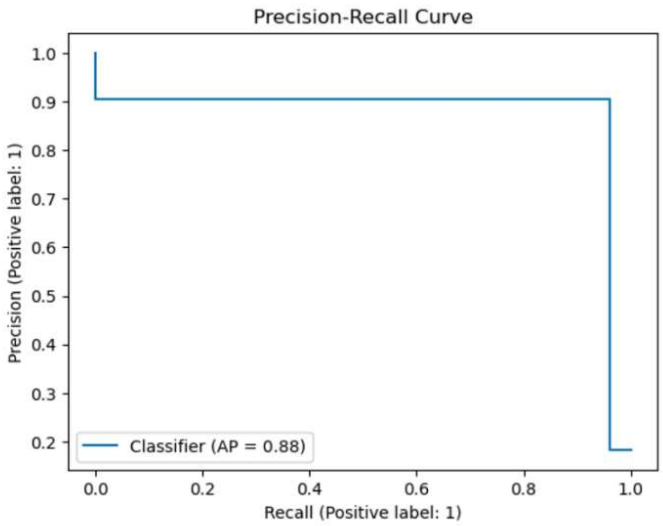

First, we trained the logistic regression model on data without top Chi-Square features, and it achieved an accuracy of **81.74%**. The classification results for the logistic regression model, trained on a dataset without the top 9 Chi-Square features, show notable performance discrepancies between the two classes. The precision for class 0 is high at 0.82, with perfect recall (1.00), indicating that the model is highly successful at identifying on time cases. However, the precision for class 1 is much lower at 0.69, and the recall is 0.00, suggesting that the model fails to identify any instances of delay. The precision-recall curve, with an AP of 0.37, reflects the model's struggle with imbalanced class distribution. The ROC curve indicates a relatively poor ability to distinguish between the two classes overall,and it is somewhat better than random guessing. The exclusion of the top 9 Chi-Square features likely contributed to this performance degradation, as those features could have been crucial in enhancing the model's ability to distinguish between classes.

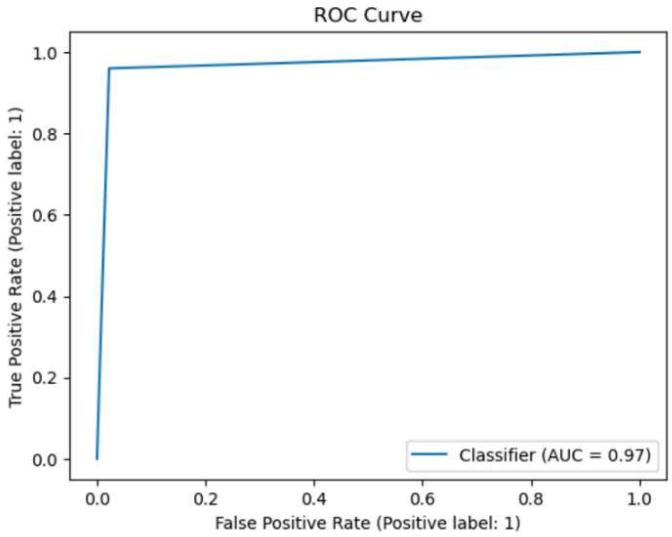
Next, we trained two models - Random Forest and Neural Network - using three different feature sets: All features, Chi-Squared-selected features, and Mutual Information-selected features. This approach allows us to analyze the impact of feature selection methods on model performance and compare the strengths and weaknesses of each algorithm.

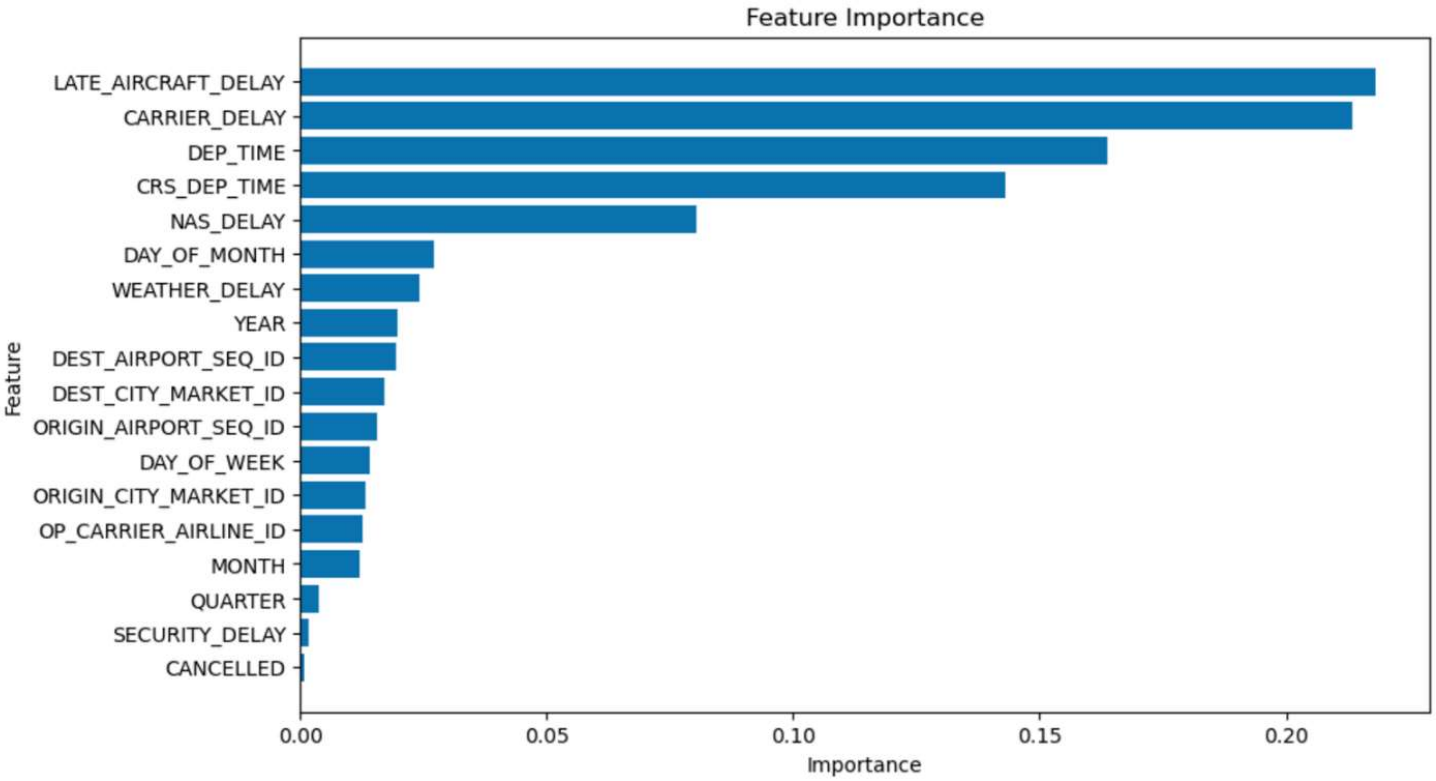
## 2. Random Forest

### All Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.98 | 0.98 | 33,525,955 |
| 1 | 0.91 | 0.96 | 0.93 | 7,516,586 |



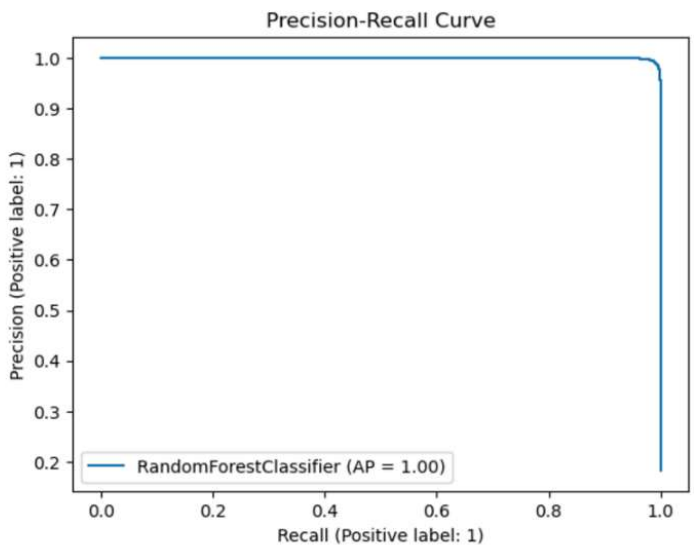



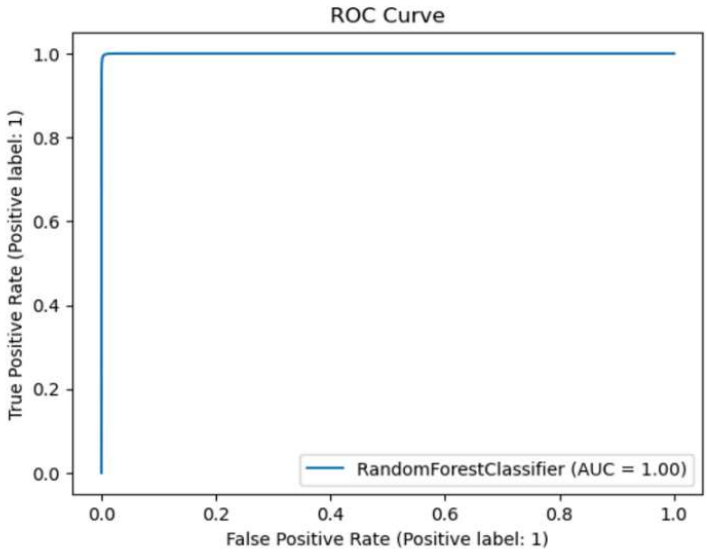



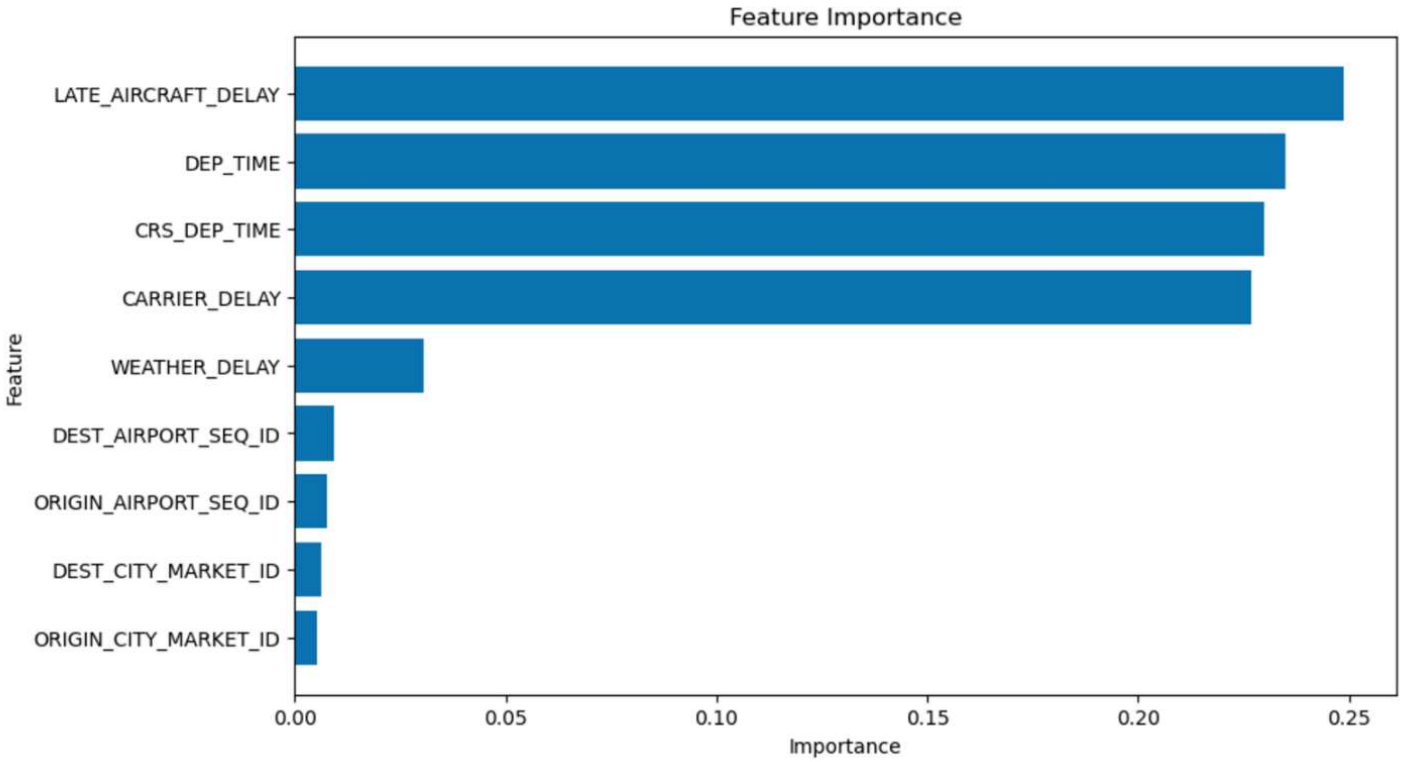

After training on all features, the random forest model shows high overall accuracy (**97.44%**). The Precision-Recall curve above shows that the classifier performs reasonably well at distinguishing the delay class but struggles as recall increases. The ROC curve shows that the model can separate positive and negative instances well. The confusion matrix also suggests that the model can struggle with Class 1 (Delay) precision due to the significant class imbalance.

## Chi-Squared Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0** | 1.00 | 0.99 | 1.00 | 33,525,955 |
| **1** | 0.97 | 1.00 | 0.98 | 7,516,586 |

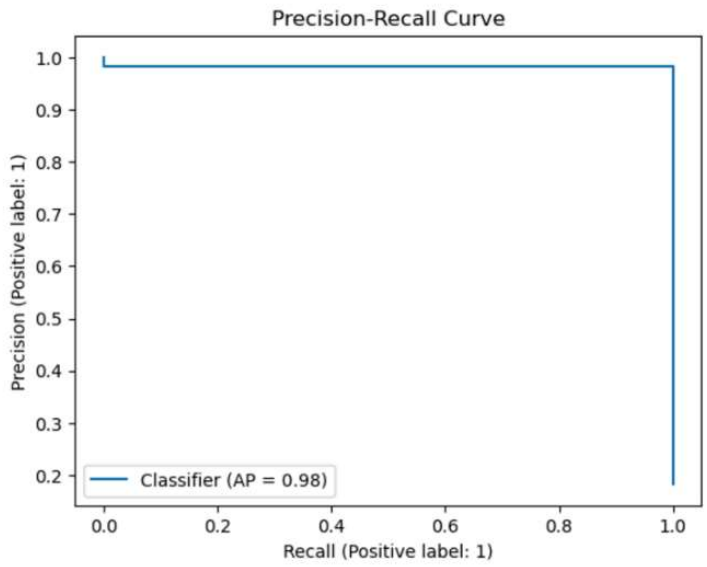


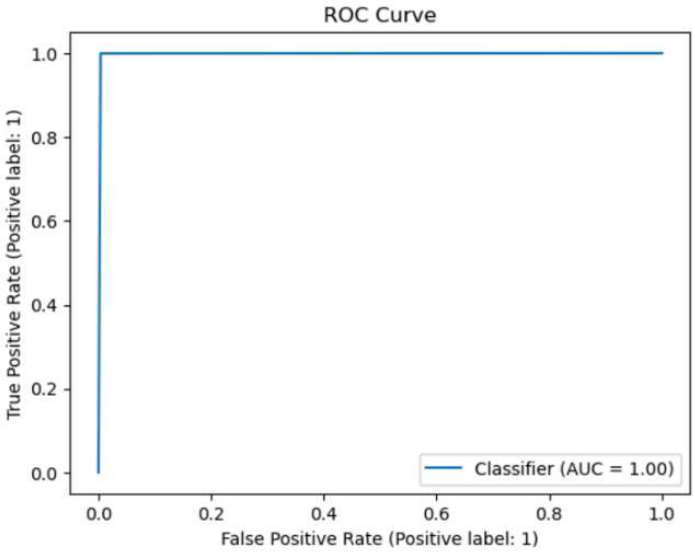



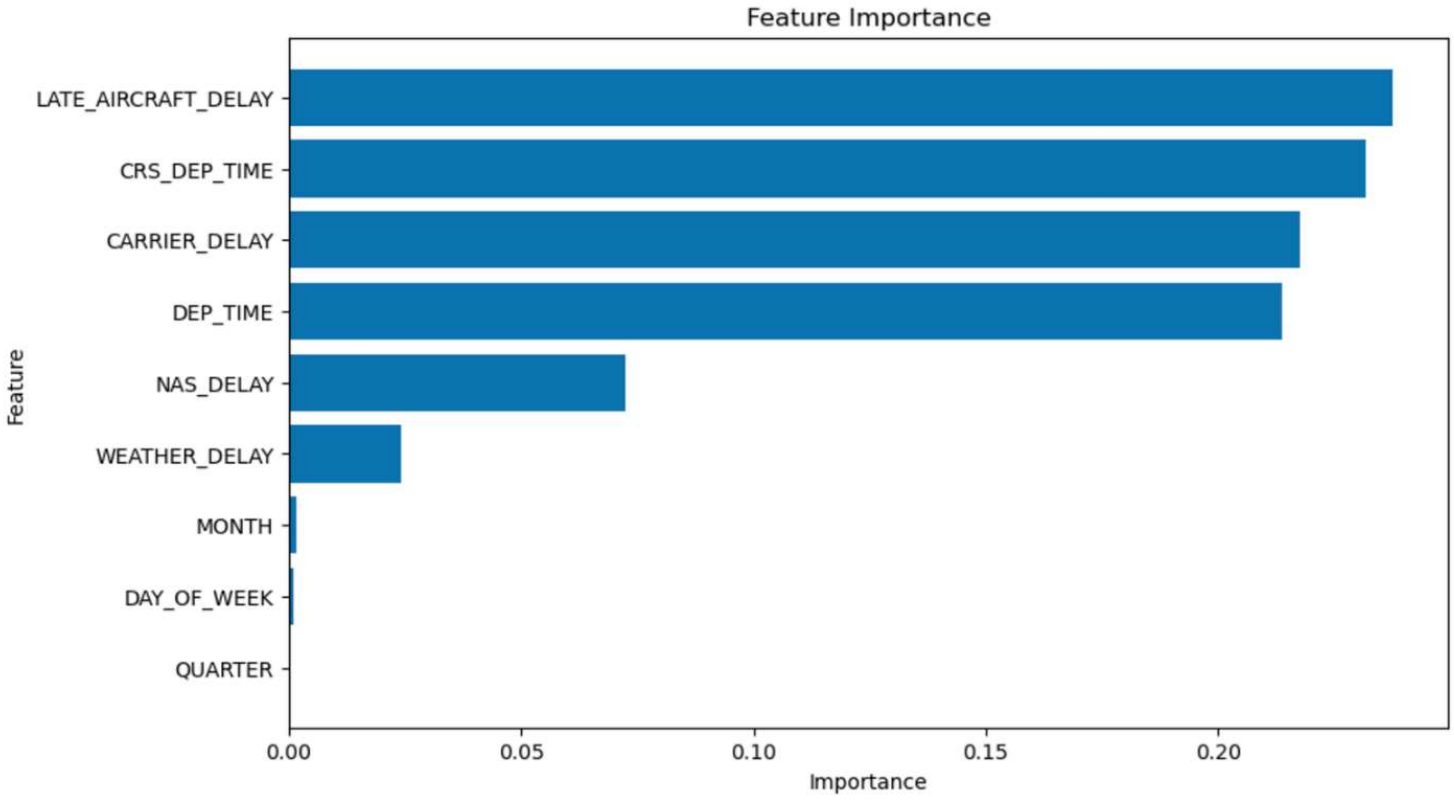




## Mutual Information Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 33,525,954 |
| 1 | 0.98 | 1.00 | 0.99 | 7,516,590 |



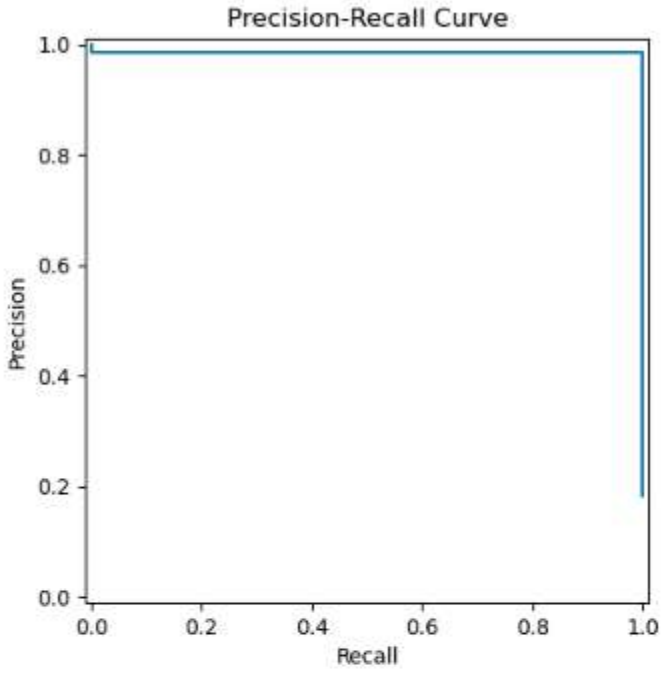

After training on Chi-Squared features and Mutual Information features, the Random Forest model shows a high accuracy of **99.33%** and **99.68%**, respectively. This is supported by high precision, recall, and F1-scores. The Precision-Recall curve and the ROC curve above also suggest this almost perfect performance. **LATE_AIRCRAFT_DELAY**, **DEP_TIME**, **CRS_DEP_TIME**, and **CARRIER_DELAY** were the most significant predictors of flight delays. **LATE_AIRCRAFT_DELAY** and **CARRIER_DELAY**

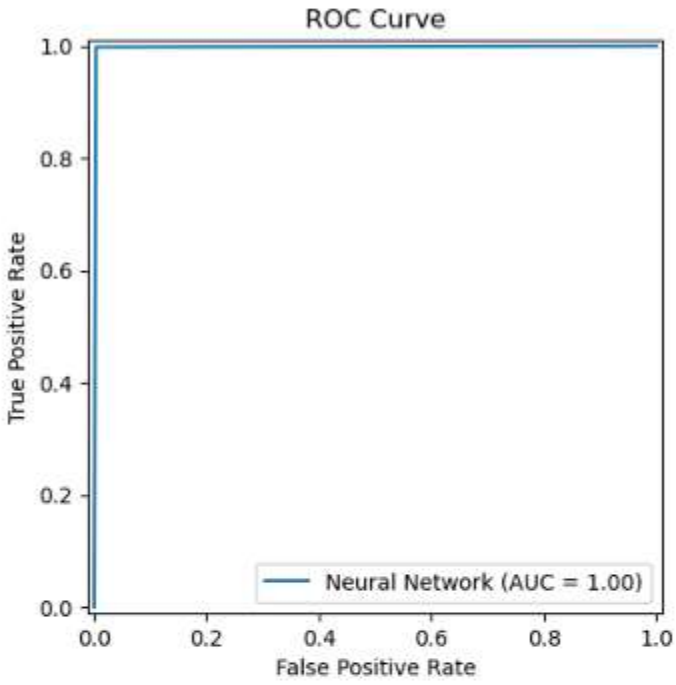
capture specific delay sources, while **DEP_TIME** and **CRS_DEP_TIME** provide critical scheduling information.

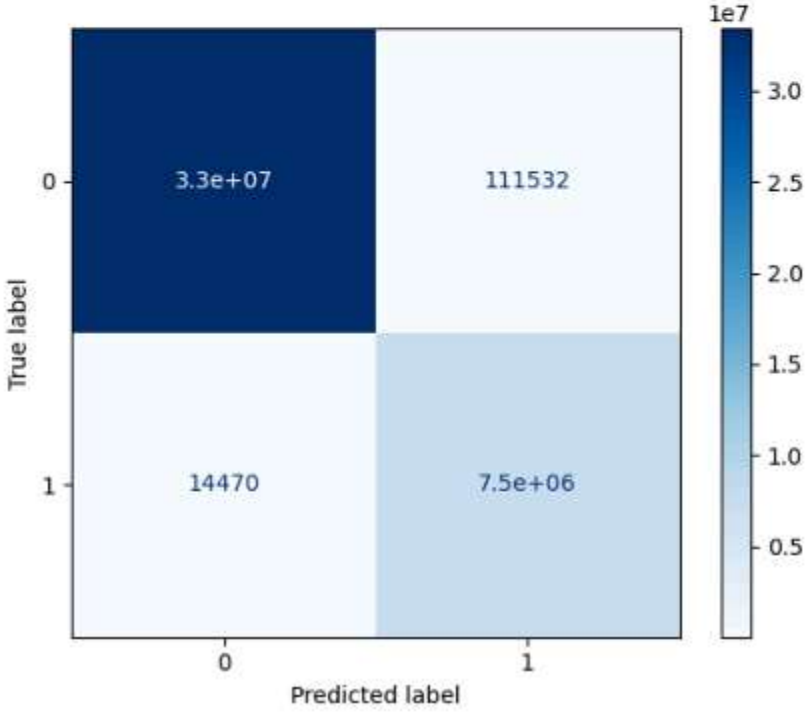
## 3. Neural Networks

### All Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 33,525,955 |
| 1 | 0.99 | 1.00 | 0.99 | 7,516,586 |



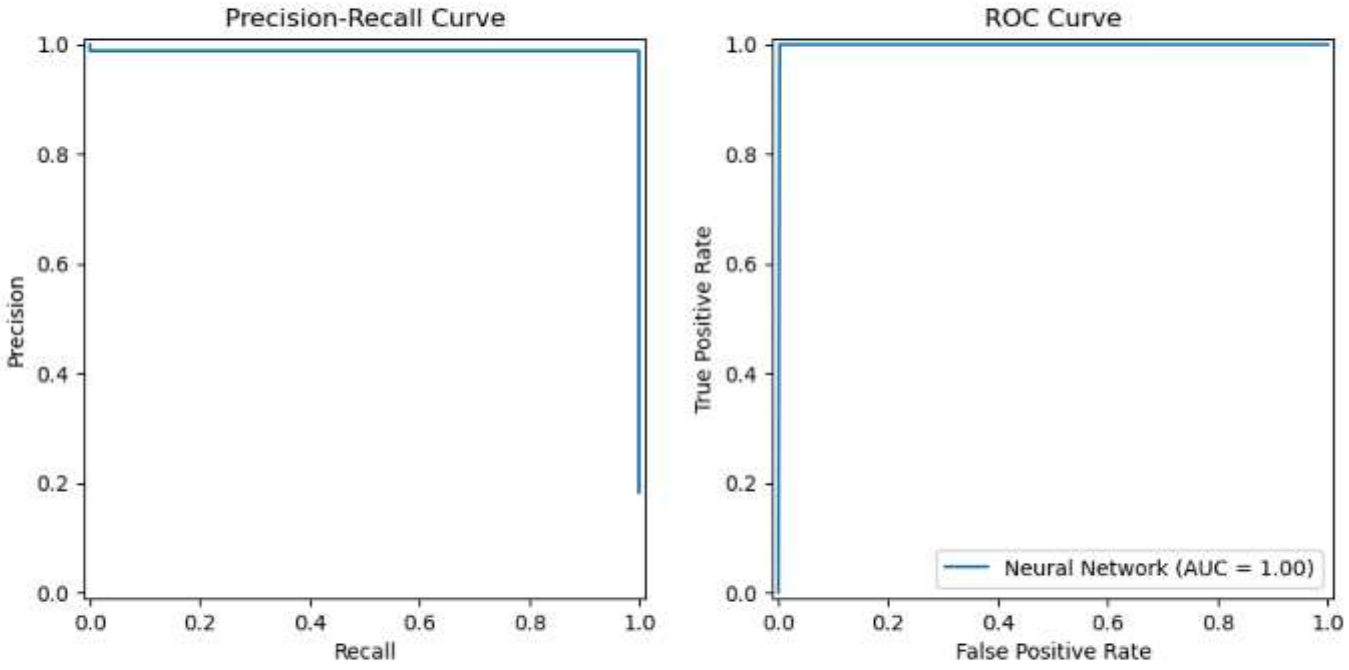



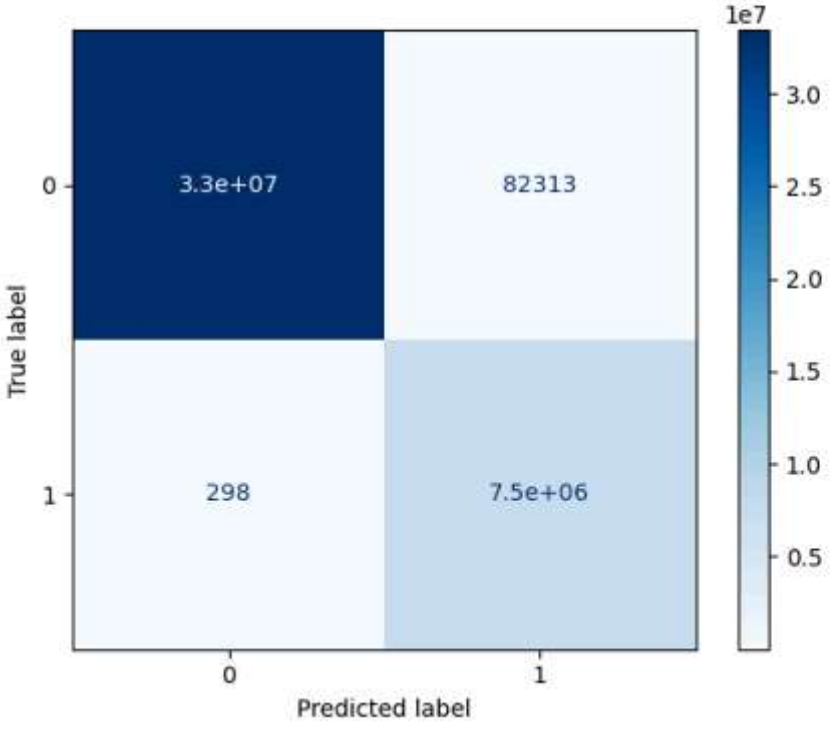

After training on all features, the Neural Network model achieved an impressive overall accuracy of **99.69%**. The Precision-Recall curve and ROC curve were flawless, with an AUC score of 1.00, indicating exceptional performance. It slightly outperformed the random forest model in this

comparison. However, 111,532 on-time cases were incorrectly predicted as delayed, and 14,470 delayed cases were misclassified as on-time.

### Chi-squared Features

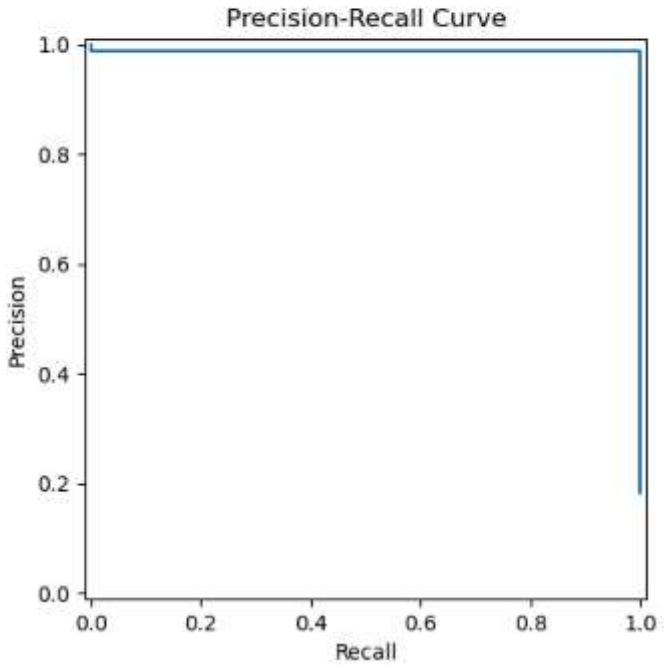
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 33,525,955 |
| 1 | 0.99 | 1.00 | 0.99 | 7,516,586 |

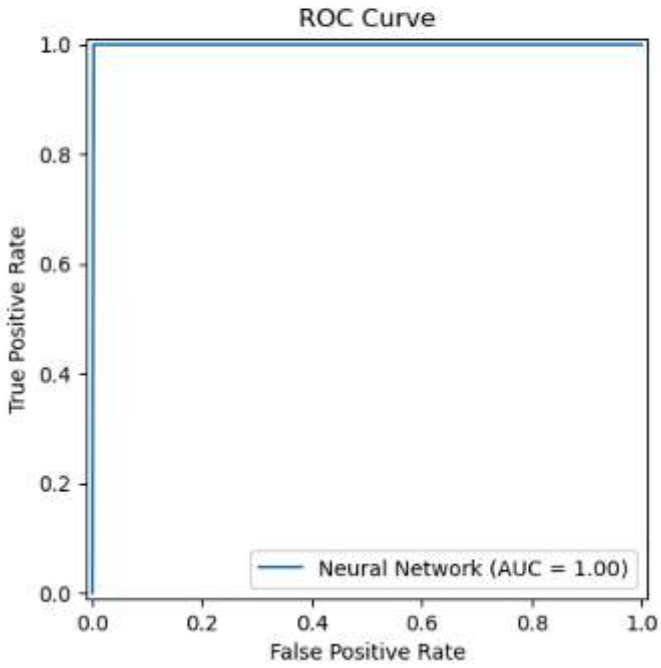


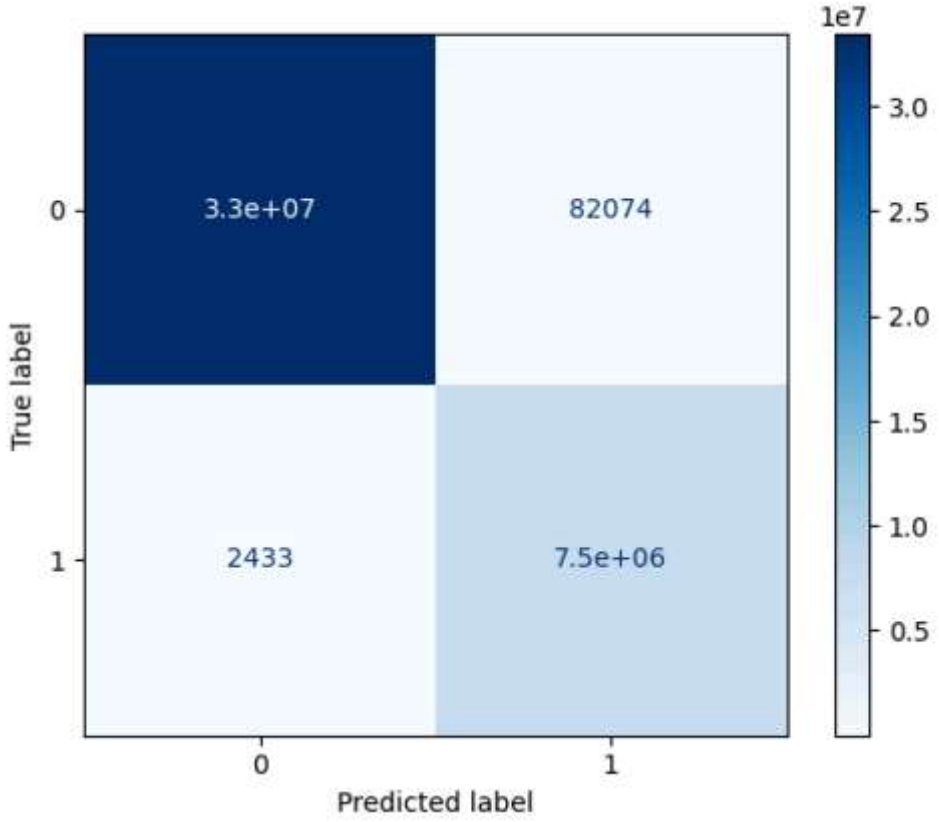



## Mutual Information Features

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0** | 1.00 | 1.00 | 1.00 | 33,525,955 |
| **1** | 0.99 | 1.00 | 0.99 | 7,516,586 |



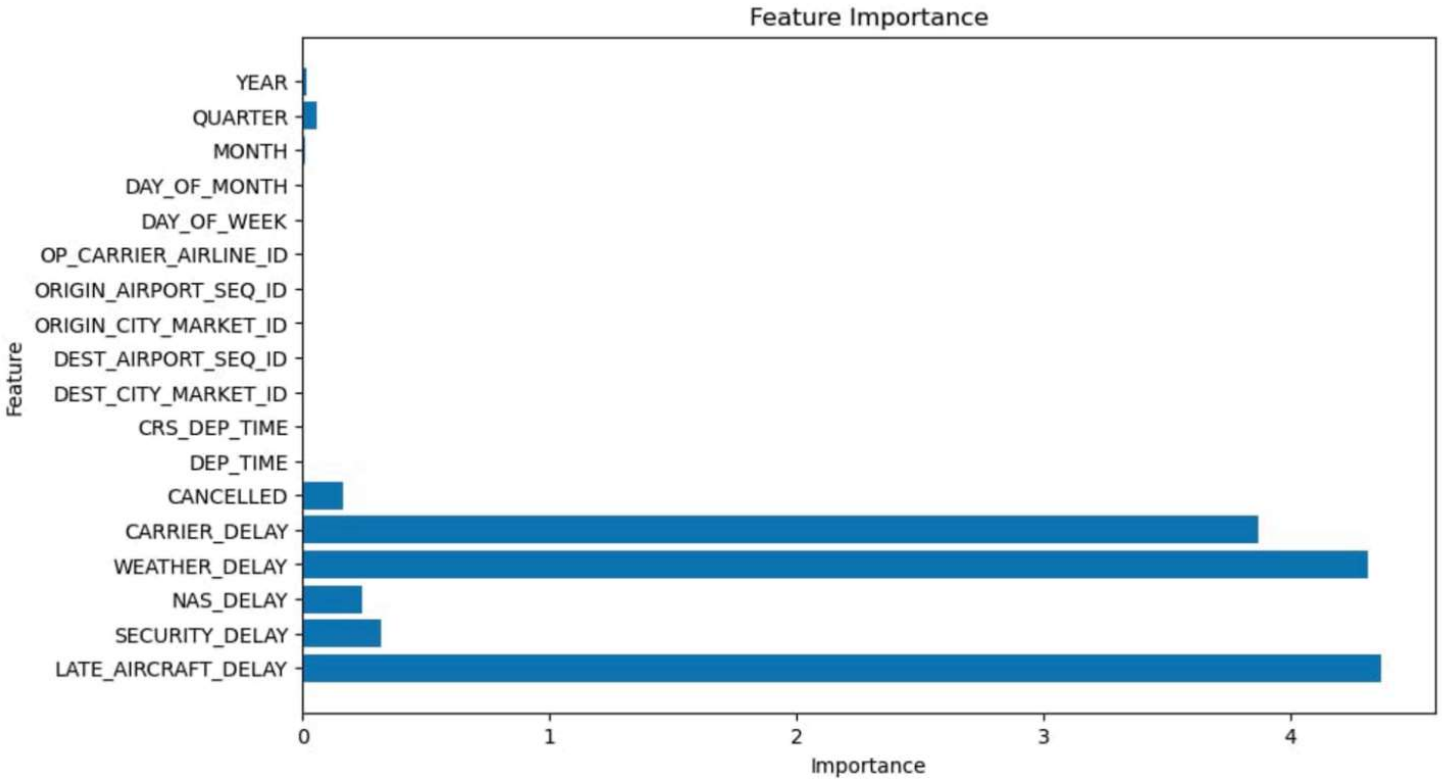
After training on Chi-Square features and Mutual Information features, the Neural Network model performs even better with an accuracy of **99.68%**. Again, a virtually flawless performance is also suggested by the Precision-Recall curve and the ROC curve above. Evidently, the confusion matrices also show a decrease in the number of misclassification cases compared to training on all features.

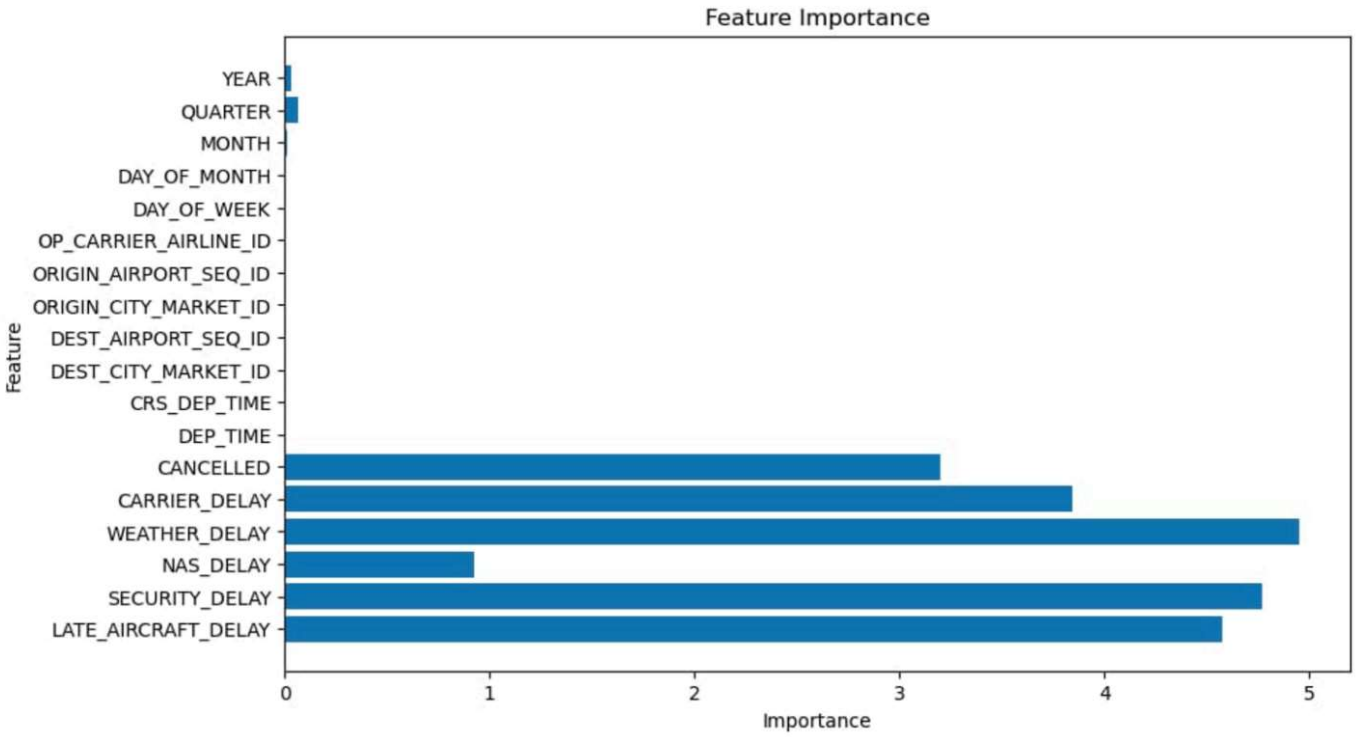
## Model Comparison

The performance evaluation of three models - logistic regression, random forest, and neural network - highlights the impact of preprocessing and feature selection on classification outcomes. Training on raw and preprocessed data using logistic regression reveals that preprocessing is essential, as the model performs poorly without it, suggesting the raw data lacks the structure needed for effective learning. Additionally, removing the top Chi-Square features results in poor performance, indicating that the excluded features could play a critical role in model performance. When trained on preprocessed data with all features, logistic regression shows the weakest performance, while the neural network outperforms other models, showcasing its superior ability to model complex patterns. Across all scenarios, random forest and neural network models consistently deliver strong results, with the neural network performing slightly better overall. The large dataset likely contributes to the models' robust performance but some features may be overly revealing. This underscores the importance of careful feature selection and preprocessing in maximizing predictive performance.

## Next Steps

We learned that preprocessing and feature selection play a significant role in boosting our model classification performance. Our overall results across models post preprocessing and feature selection were consistently strong. However, we also need to understand why we achieved such results. We suspect that several features in the dataset may be overly revealing as seen in the Feature Importance graphs retrieved from the Linear Regression models training on raw dataset (first graph) and preprocessed dataset (second graph) below.




In the graphs, **LATE_AIRCRAFT_DELAY**, **WEATHER_DELAY** and **CARRIER_DELAY** features have consistently high importance both before and after preprocessing, outweighing other features. We would like to explore alternatives to these features that also capture weather, security, and late aircraft conditions without revealing the delay time caused by such conditions. Such alternatives may include passenger count in the security lines at a given time, weather report around departure time, or aircraft conditions around departure time. By using these alternatives, our input will be

more objective with respect to predicting flight delay and more practical since we will not know the delay time caused by the mentioned conditions until the delay has already happened. After collecting them, we will retrain all models and compare their feature importance and prediction results with their existing counterparts.

We are also considering implementing an LSTM model to capture long-term trends in our dataset. LSTMs excel at identifying long-term dependencies through their memory cells, making them well-suited for uncovering flight delay patterns over time. Additionally, LSTMs can process sequential data in both forward and backward directions, enabling a more comprehensive analysis of hidden trends across the flight history. This bidirectional capability could provide deeper insights into the factors influencing delays throughout the dataset.

Another tree-based model worth exploring is XGBoost. Unlike Random Forest, which constructs trees independently, XGBoost builds decision trees sequentially, with each tree learning from the errors of its predecessor. This boosting mechanism often leads to higher accuracy, particularly when outliers are carefully addressed and hyperparameters are finely tuned. By leveraging XGBoost's capabilities, we anticipate achieving more precise results in our analysis.

### Project Goals

Our goal is to maximize the recall value in our model evaluation (>90%). Incorrectly identifying a flight to have no delay could pose planning issues for customers. However, incorrectly identifying flights to have a delay also poses economic problems for the airline. We'll also optimize for a high precision value, but not at the expense of our customers.

## Gantt Chart

View it here↗

## Contribution Table

| Name | Proposal Contributions |
|---|---|
| Rishi Borra | Introduction, Logistic Regression Model |
| My Phung | Problem Definition, Metrics/Visualization |
| Long Lam | Data Preprocessing, Feature Selection, Next Step |
| Joseph Thomas | Random Forest Model |

| Name | Proposal Contributions |
| --- | --- |
| Aziz Albahar | Neural Network Model |

## References


[1] “Twenty Years Later, How Does Post-9/11 Air Travel Compare to the Disruptions of COVID-19? | Bureau of Transportation Statistics,” www.bts.gov, Sep. 10, 2021. https://www.bts.gov/data-spotlight/twenty-years-later-how-does-post-911-air-travel-compare-disruptions-covid-19

[2] M. O. Ball, C. Barnhart, M. Dresner, and Augusto Voltes, “Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the...,” ResearchGate, Oct. 2010. https://www.researchgate.net/publication/272202358_Total_Delay_Impact_Study_A_Comprehensive_Assessment_of_the_Costs_and_Impacts_of_Flight_Delay_in_the_United_States (accessed Oct. 04, 2024).

[3] S. Choi, Y. J. Kim, S. Briceno and D. Mavris, “Prediction of weather-induced airline delays based on machine learning algorithms,” 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 2016, pp. 1-6, doi: 10.1109/DASC.2016.7777956. keywords: {Meteorology;Delays;Atmospheric modeling;Data models;Predictive models;Training;Schedules}

[4] V. Natarajan, S. Meenakshisundaram, G. Balasubramanian and S. Sinha, “A Novel Approach: Airline Delay Prediction Using Machine Learning,” 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 1081-1086, doi: 10.1109/CSCI46756.2018.00210. keywords: {Delays;Airports;Atmospheric modeling;Logistics;Predictive models;Meteorology;Mathematical model;Flight delay prediction;Logistic regression;Decision tree algorithm;Analytical modeling;Delay evaluation}

[5] G. Biau and E. Scornet, “A random forest guided tour,” TEST, vol. 25, no. 2, pp. 197–227, Apr. 2016, doi: https://doi.org/10.1007/s11749-016-0481-7.


---