Flight Delay Prediction

By

Aditya Nittala
Nishit Rao
Sagar Borkar
Vishal Manjunath

BIA 678-A – Big Data Technologies – Prof. David Belanger

ABSTRACT

Flight delays have created major issues in the current aviation system. Methods are needed to analyse the way delay propagates in the airport networks. Traditional methods are inadequate to the task. In this project, we have presented a new machine learning based flight delay prediction model that combines multi-label random forest classification and approximated delay propagation model. To improve the prediction performance, an optimal feature selection process is introduced and demonstrated to have better performance than directly using all the features of available datasets. Departure delay and late arrival delay are shown to be the most important features for delay prediction. To utilize these two features, a delay propagation model is proposed as a link to connect them to build a chained delay prediction model. Given the initial departure delay, the chained model is demonstrated to have the ability to predict the flight delay along the same aircraft's itinerary. By updating the actual departure delay with the iteration number along the itinerary, the model's accuracy can be further improved. Our application results clearly demonstrate the value of machine learning and delay propagation for analysing and predicting the air traffic delay in daily operation.

TABLE OF CONTENTS

1. Introduction

Air travel is one of the major modes of transportation for both personal travel as well as cargo transportation for businesses. With rapid growth of air traffic, increasing flight delays in have become a serious and prominent problem. Currently we are focussing on the flight delays in the United States. According to the Bureau of Transportation Statistics (BTS), nearly one in four airline flights arrived at its destination over 15 minutes late. It is reported that the annual total cost of air transportation delays was over $30 billion, which poses a significant challenge to the development of Next Generation Air Transportation System (NextGen). This fact motivates the need for accurate and practical prediction of flight delays, especially for individual flights. Beyond the delays directly created by limited airspace capacity, a third of the late arrivals were caused by an aircraft arriving late and thus having to depart late on its next flight, which is known as delay propagation. Given the fact that airlines fly their aircraft on daily scheduled itineraries that require visits to a sequence of airports, the late-arriving aircraft delay early in the day has a significant impact on the downstream delay performance. The delay propagation is inherent with the National Airspace System (NAS), which includes many connective resources, such as aircraft, crew, passengers, and gate space. Moreover, the increasing air traffic demand pushes the NextGen to reduce slack time between arrivals and departures, which will make the NAS further suffer from serious delay propagation through the network. Therefore, the modelling of delay propagation is a key factor for the success of accurate flight delay prediction. By providing the demand and capacity of the airports and flight itineraries, we can create a propagation model to analyse the delay propagation phenomenon. The goal is to successfully create a model which will be able to predict the flight delays.

2. Problem Statement

Nowadays, the aviation industry plays a crucial role in the world's transportation sector. Apart from personal travel, a lot of businesses rely on various airlines to connect them with other parts of the world. But several factors can directly affect the airline services by means of flight delays. To solve this issue, accurately predicting these flight delays allows passengers to be well prepared for the deterrent caused to their journey and enables airlines to respond to the potential causes of the flight delays in advance to diminish the negative impact. We intend to use exploratory data analysis to carefully extract the best features from these factors and successfully build a machine learning model that predicts flight delays.

3. Factors Causing Flight Delays

Every flight has many variables, which give detailed information about the specific flight. We will be looking into some of the variables that play a vital role in determining the flight delay. Each variable has a different weight that is used while training the model. Prior knowledge of these variables should be present to understand the dataset. These variables are: Delay Propagation, Air Traffic Control, Weather Conditions, Connecting Passengers, Security Clearance, Baggage Loading, Mechanical Failures, Catering, Bird Strikes, Weight Restrictions, Cargo, Refuelling etc.

4. Big Data Analysis

The dataset we have used contains 5 million records. In order to select the right data for our model we need to understand the data better. Data cleaning and formatting can help us do so, thus it is considered as the most critical part of this whole project.

4.1. Source of the data

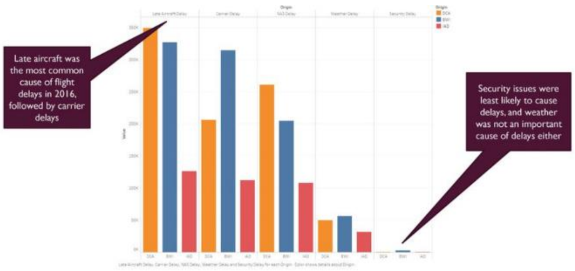
We used a total of three databases from Kaggle, all of which are open source and freely available online. They were chosen based on the features that could best describe the process of airline operations. The databases are: Airline on-time performance database of the Bureau of Transportation Statistics (BTS), Local Climatological Data (LCD) at the National Oceanic and Atmospheric Administration (NOAA) and Aviation System Performance Metrics (ASPM). The BTS provides the air traffic on-time performance data reported by certified U.S. Air carriers. To explain the arrival delay of the flights, causes of the delay are reported in five categories: air carrier, extreme weather, national aviation system, late arriving aircraft, and security. LCD contains weather summaries for major airports that include a daily account of temperature extremes, degree-days, precipitation amounts, winds and special weather. ASPM data contains airport capacity and throughput data for main airports of US.

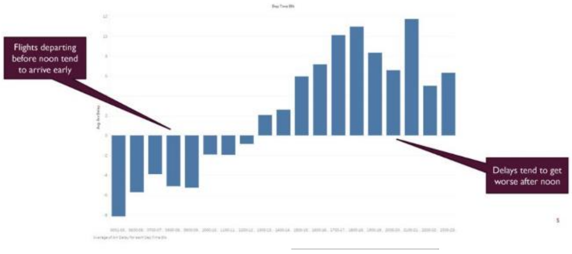
4.2. Pre-processing of the data

Data Cleaning and formatting includes techniques like pre-processing and oversampling. Due to numerous factors, a lot of data has been collected which may not necessarily add to the main cause of flight delays. Hence, we need to trim down the extra data so that we can easily process the useful data using the new machine learning model and find effective results with utmost accuracy. Since the dataset is so large, we must constrain it so that analysis on our system in a reasonable amount of time. To do so, the dataset is sampled randomly for 20k rows. Further, these observations have be randomly split into training and test sets, so that 10k observation is used for analysis.

4.3. Exploratory Data Analysis

Using the EDA approach, we analyse data sets to summarize the main characteristics using statistical graphics and data visualization methods. This analysis helps us determine the variables that are important for feature extraction.





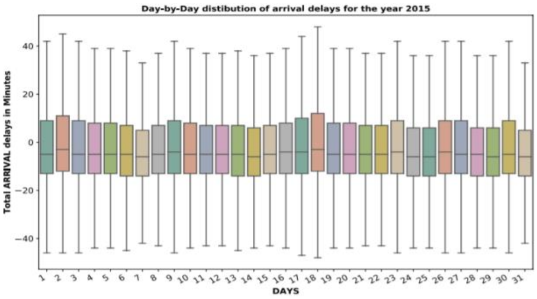
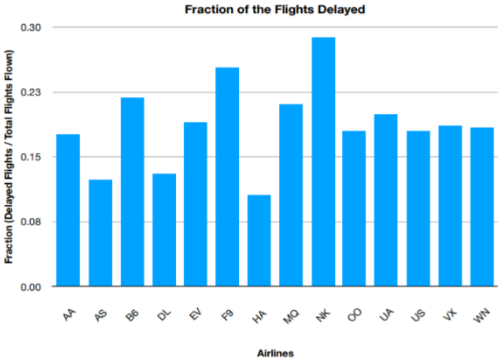
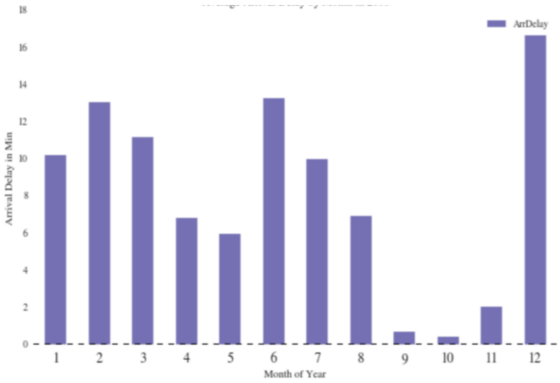
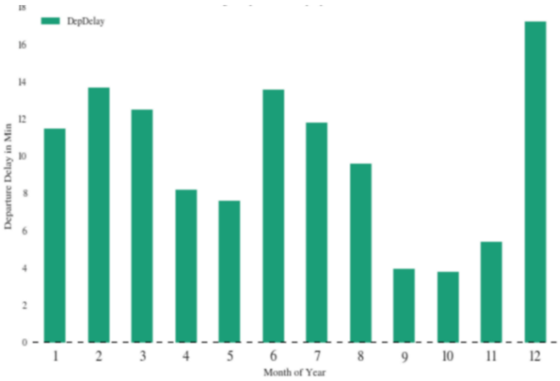


Fig. 2. Figure showing the day-by-day distribution of the arrival delays of flights for all the months in 2015







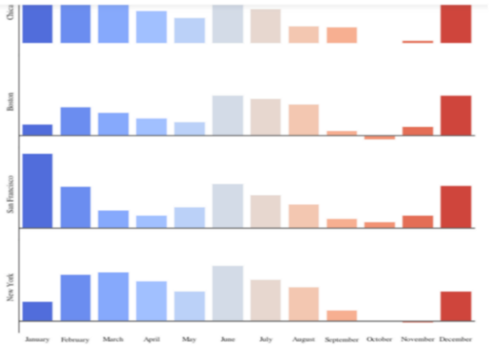


Figure 8: Arrival Delays at Airport

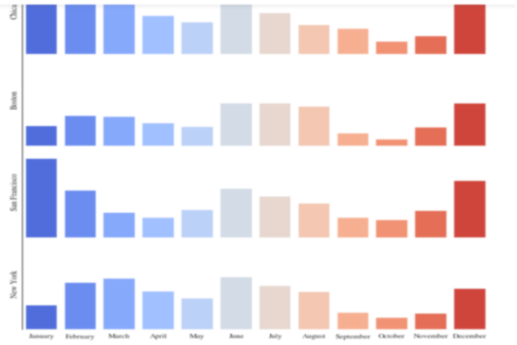


Figure 7: Departure Delays at Airport

5. Feature Selection and Extraction

Since the effect of the curse of dimensionality, training all the features from BTS, NOAA, ASPM may not have the best performance. The classifiers performance may decrease if the dimension increases without enough training samples. Too many features may confuse the classifier and make it less likely to find the key elements for classification. Normally, feature selection is applied to a portion of samples from the training set to improve the performance. The following delay-related features are extracted from the BTS database:

1. Flight Number

2. Carrier {American, United}

4. Origin {Airport Code: LAX}

3. Destination {Airport Code: SFO}

5. Date {MM/DD/YY}

6. Day {Mon, Tue, Wed, Thu, Fri, Sat, Sun}

7. Month {Jan, Feb, March}

8. Arrival Time {HH:MM AM/PM}

9. Departure Time {HH:MM AM/PM}

10. Cancellation Reason

6. Predictive Model Description

The purpose of this project is to create a machine learning model based on the features that we extracted above. These algorithms are: Decision Tree Classifier, and Random Forest Classifier. All models were implemented using scikit-learn. Many models such as

Naïve Bayes, SVM and Logistic Regression were also performed on the dataset however they could not withstand the scale of the dataset.

6.1. Decision Tree Classifier

Decision Tree is a supervised learning algorithm. The main idea behind the decision tree algorithm is to build a tree-like model from root to leaf nodes. All nodes receive a list of inputs, and the root node receives all the examples in the training set. Each node asks a true or false question for one of the features and in response to this question the data is partitioned in to two subsets. The challenge to building such a tree is which question to ask at a node and when. To do this, decision tree algorithm uses well known indices like entropy or Gini-impurity to quantify an uncertainty or impurity associated with a certain node.

6.2. Random Forest Classifier

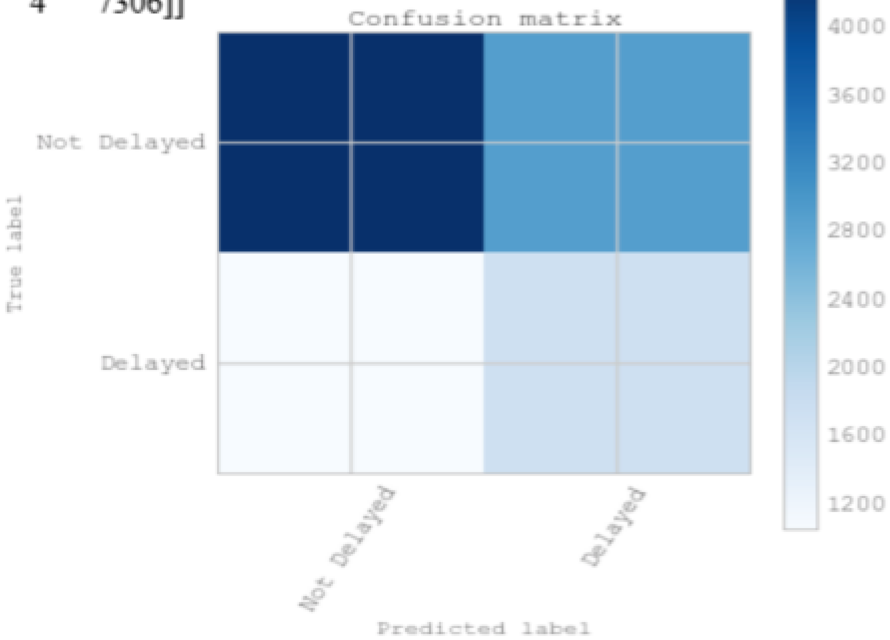
The Random Forest (RF) classifier is an ensemble method based on multiple decision trees. By combining the Bootstrap aggregating and random space, RF overcomes the drawbacks of individual decision tree. RF is widely used in industry because it can classify high dimensional data in short time with good performance and it has low sensitivity to outliers in the training data. Moreover, RF was chosen as the core for our prediction modules for two reasons. First, RF is tested to have superior performance than other classification models. Second, RF can output the importance of the features in its learning process.

7. Performance Metrics

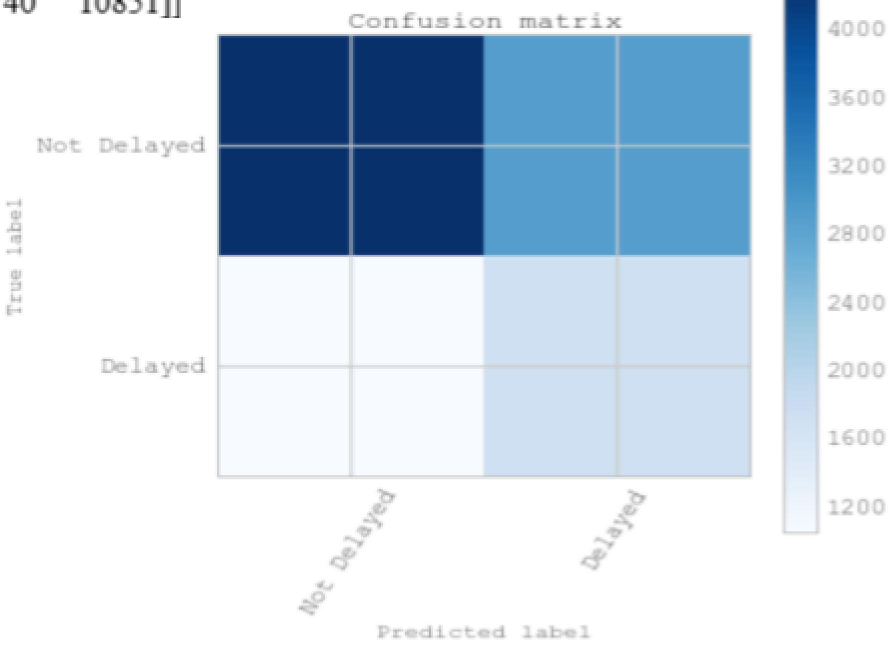
To evaluate the performance of algorithms, we used different metrics which are based on the confusion matrix. Confusion matrix is a tabular representation of a classification model performance on the test set, which consists of four parameters: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Decision Tree Confusion Matrix

```
[[12633     57]
 [    4   7306]]
```



```
[[19075     34]
 [   40  10851]]
```



7.1. Accuracy

The most used metric is accuracy, which represents the percentage of properly anticipated observations. **Accuracy = TP + TN/ TP +TN + FP + FN**

A high accuracy number indicates a good model, but a false positive or false negative can jeopardize trust. As a result, we consider three additional metrics: precision, recall, and AUC-score.

7.2. Recall

The total number of positive classifications out of true class is referred to as recall. It reflects the proportion of articles anticipated to be true out of the total number of true articles in our dataset. **Recall = TP/ TP+FN**

7.3. Precision

Precision score, on the other hand, is the ratio of true positives to all real occurrences predicted. Precision refers to the number of articles tagged as true out of all the positively predicted (true) articles. **Precision = TP/ TP + FP**

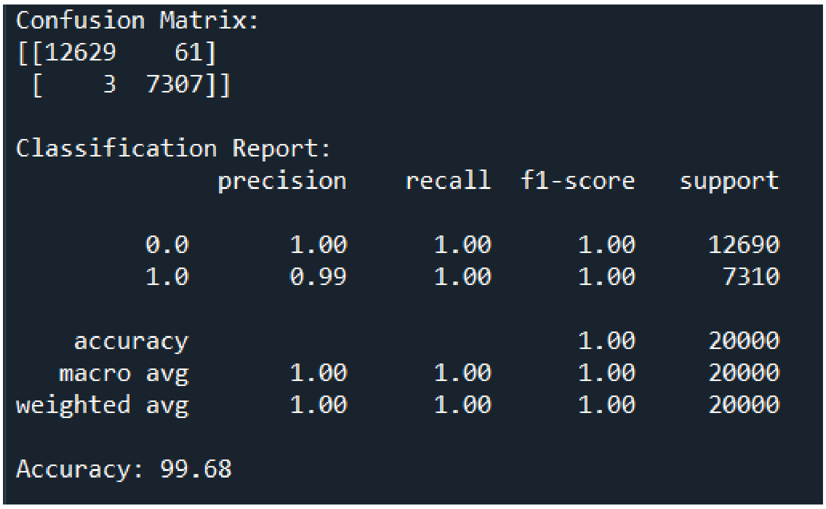
7.4. AUC Score

AUC is the two-dimensional Area under the ROC curve. It is a probability curve that plots the True Positive Rate against the False Positive Rate at various thresholds and essentially separates the signal from the noise. **AUC score = (1 – Specificity)**

8. Result

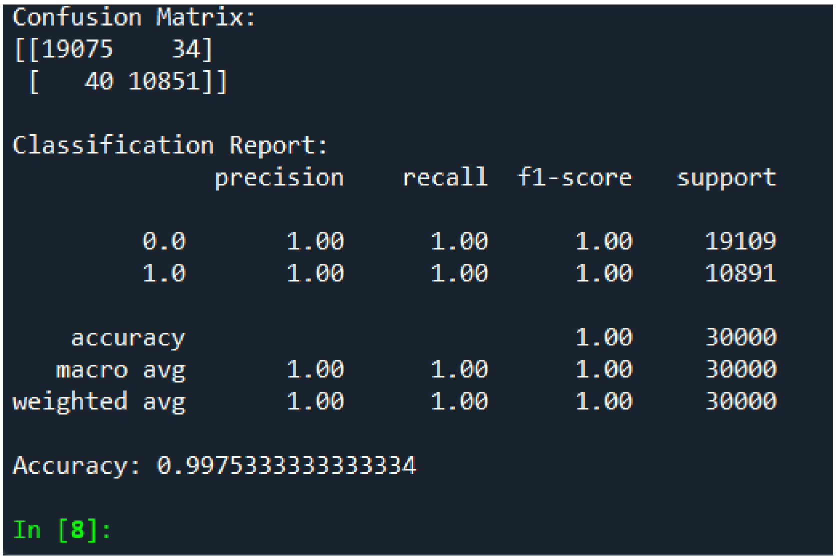
This is the comparison of performance metrics of all the datasets.

Decision Tree Classifier



```
Confusion Matrix:
[[12629    61]
 [    3  7307]]

Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     12690
         1.0       0.99      1.00      1.00      7310

    accuracy                           1.00     20000
   macro avg       1.00      1.00      1.00     20000
weighted avg       1.00      1.00      1.00     20000

Accuracy: 99.68
```

Random Forest Classifier



```
Confusion Matrix:
[[19075    34]
 [   40 10851]]

Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     19109
         1.0       1.00      1.00      1.00     10891

    accuracy                           1.00     30000
   macro avg       1.00      1.00      1.00     30000
weighted avg       1.00      1.00      1.00     30000

Accuracy: 0.9975333333333334

In [8]:
```

9. CONCLUSION

Based on the comparison of the results of performance metrics of various classifiers, we can conclude that Decision Tree Classifier is the best model to predict flight delays. The other models we used were random forest classification and logistic regression. To improve the prediction performance, an optimal feature selection process is introduced and demonstrated to have better performance than directly using all the features. Departure delay and late arriving aircraft delay are shown to be the most important features for delay prediction. However, a chained delay propagation model does not have the ability to solely predict the flight delay along the same aircraft's itinerary. This decision tree machine learning based method has been shown to be more accurate and practical for delay prediction in daily air traffic operation.

The future work of this project includes incorporating a larger dataset. There are many ways to pre-process a larger dataset like running a Spark cluster over a server or using a cloud-based services like AWS and Azure to process the data. With the new advancement in the field of deep learning, we can use Neural Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology. It is divided into three basic parts for data modelling that includes feed forward networks, feedback networks, and self-organization network. Feed-forward and feedback networks are generally used in the areas of prediction, pattern recognition, associative memory, and optimization calculation, whereas self-organization networks are generally used in cluster analysis. Neural Network offers distributed computer architecture with important learning abilities to represent nonlinear relationships. Also, the scope of this project is very much confined to flight and weather data of United States, but we can expand the scope to include international flights.

10. References

[1] Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A. A., and Zou, B., "Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the United States, NEXTOR," 2010.

[2] Chen, J. and Sun, D., "Stochastic Ground-Delay-Program Planning in a Metroplex," Journal of Guidance, Control, and Dynamics, Vol. 41, No. 1, 2017, pp. 231–239. 15 of 16 American Institute of Aeronautics and Astronautics

[3] Ferguson, J., Kara, A. Q., Hoffman, K., and Sherry, L., "Estimating domestic US airline cost of delay based on European model," Transportation Research Part C: Emerging Technologies, Vol. 33, 2013, pp. 311–323.

[4] Chen, J., Chen, L., and Sun, D., "Air traffic flow management under uncertainty using chance-constrained optimization," Transportation Research Part B: Methodological, Vol. 102, 2017, pp. 124–141.

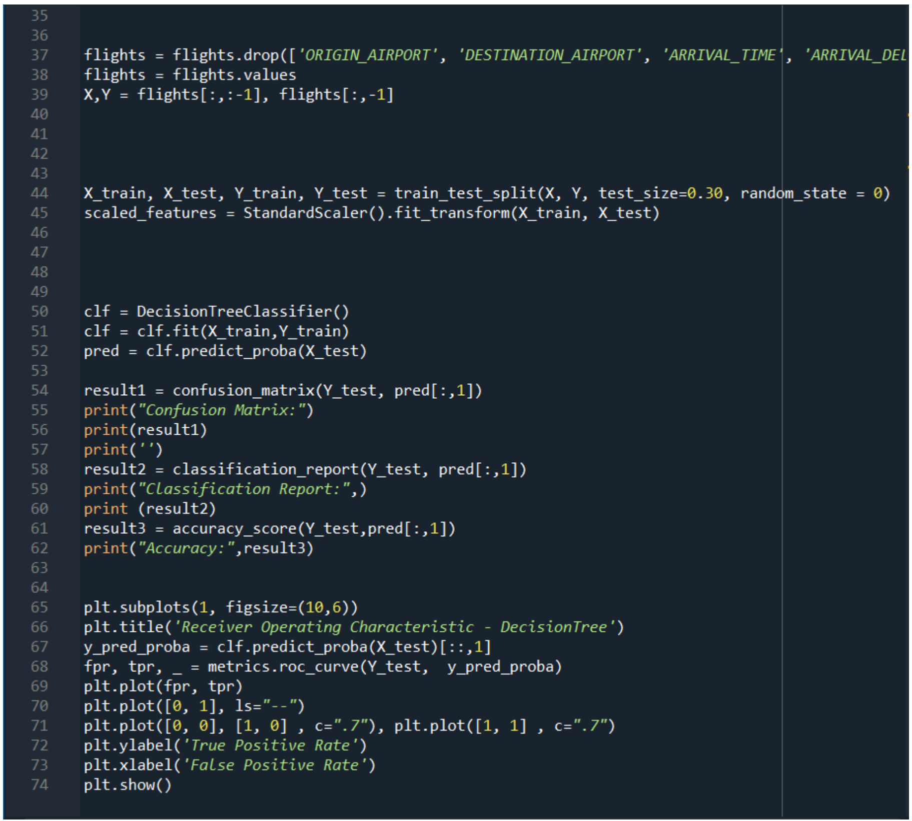
[5] Mueller, E. R. and Chatterji, G. B., "Analysis of aircraft arrival and departure delay characteristics," AIAA aircraft technology, integration and operations (ATIO) conference, 2002.

[6] Klein, A., "Airport delay prediction using weather-impacted traffic index (WITI) model," Digital Avionics Systems Conference (DASC), 2010 IEEE/AIAA 29th, IEEE, 2010, pp. 2–B.

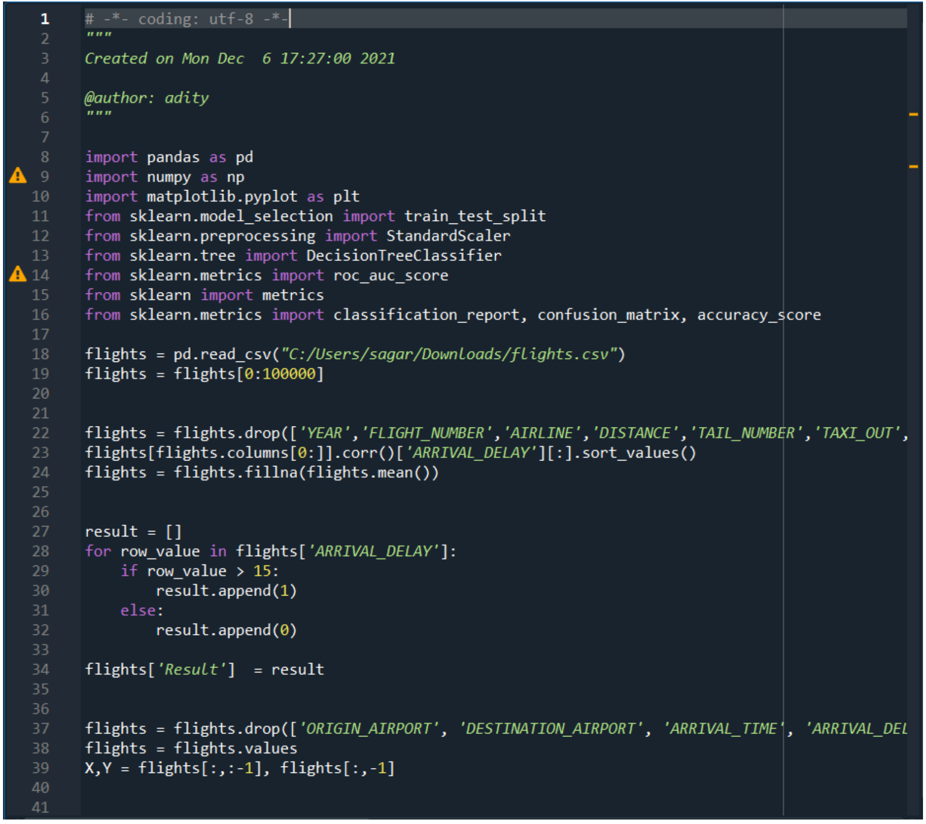
[7] Rebollo, J. J. and Balakrishnan, H., "Characterization and prediction of air traffic delays," Transportation research part C: Emerging technologies, Vol. 44, 2014, pp. 231–241.

11. Appendix

Decision Tree Code:

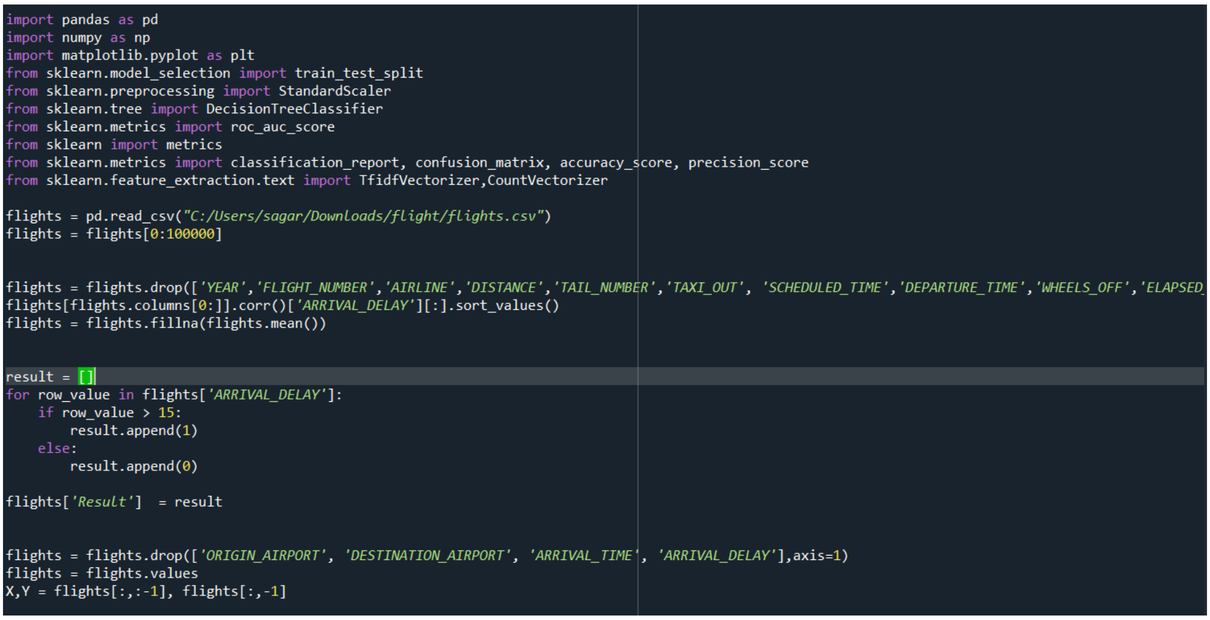


```


flights = flights.drop(['ORIGIN_AIRPORT', 'DESTINATION_AIRPORT', 'ARRIVAL_TIME', 'ARRIVAL_DEL
flights = flights.values
X,Y = flights[:,:-1], flights[:,-1]




X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.30, random_state = 0)
scaled_features = StandardScaler().fit_transform(X_train, X_test)




clf = DecisionTreeClassifier()
clf = clf.fit(X_train,Y_train)
pred = clf.predict_proba(X_test)

result1 = confusion_matrix(Y_test, pred[:,1])
print("Confusion Matrix:")
print(result1)
print('')
result2 = classification_report(Y_test, pred[:,1])
print("Classification Report:",)
print (result2)
result3 = accuracy_score(Y_test,pred[:,1])
print("Accuracy:",result3)


plt.subplots(1, figsize=(10,6))
plt.title('Receiver Operating Characteristic - DecisionTree')
y_pred_proba = clf.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(Y_test,  y_pred_proba)
plt.plot(fpr, tpr)
plt.plot([0, 1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

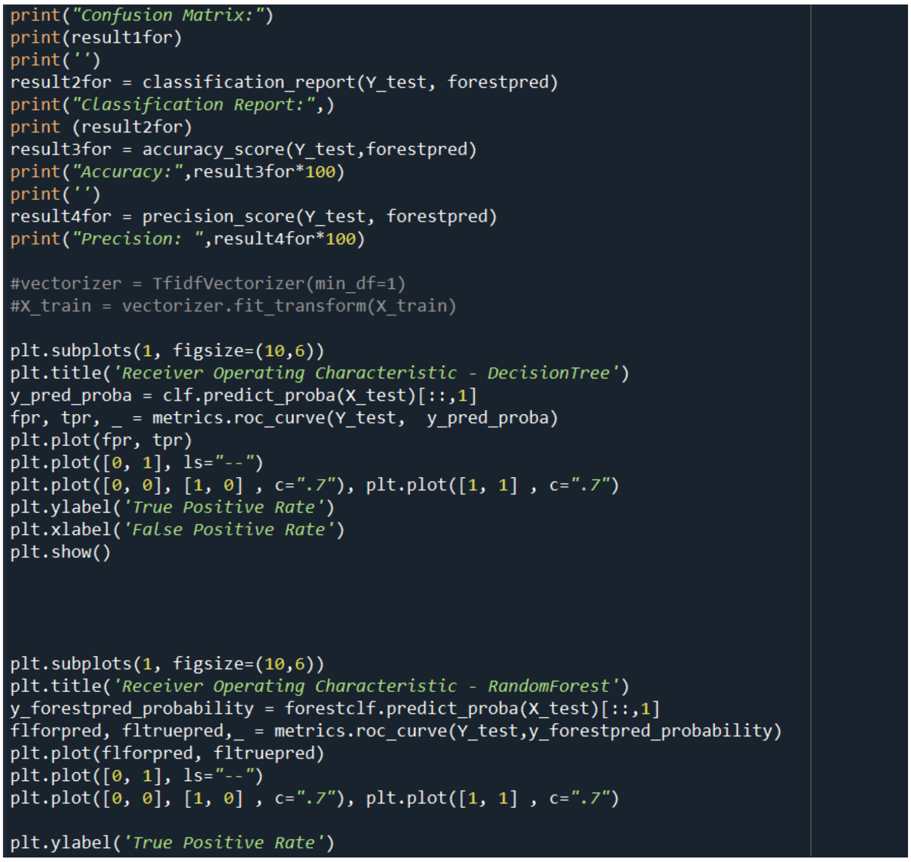


```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec  6 17:27:00 2021

@author: adity
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_auc_score
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

flights = pd.read_csv("C:/Users/sagar/Downloads/flights.csv")
flights = flights[0:100000]


flights = flights.drop(['YEAR','FLIGHT_NUMBER','AIRLINE','DISTANCE','TAIL_NUMBER','TAXI_OUT',
flights[flights.columns[0:]].corr()['ARRIVAL_DELAY'][:].sort_values()
flights = flights.fillna(flights.mean())


result = []
for row_value in flights['ARRIVAL_DELAY']:
    if row_value > 15:
        result.append(1)
    else:
        result.append(0)

flights['Result']  = result


flights = flights.drop(['ORIGIN_AIRPORT', 'DESTINATION_AIRPORT', 'ARRIVAL_TIME', 'ARRIVAL_DEL
flights = flights.values
X,Y = flights[:,:-1], flights[:,-1]

```

Random Forest Classifier



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_auc_score
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer

flights = pd.read_csv("C:/Users/sagar/Downloads/flight/flights.csv")
flights = flights[0:100000]


flights = flights.drop(['YEAR','FLIGHT_NUMBER','AIRLINE','DISTANCE','TAIL_NUMBER','TAXI_OUT', 'SCHEDULED_TIME','DEPARTURE_TIME','WHEELS_OFF','ELAPSED_
flights[flights.columns[0:]].corr()['ARRIVAL_DELAY'][:].sort_values()
flights = flights.fillna(flights.mean())


result = []
for row_value in flights['ARRIVAL_DELAY']:
    if row_value > 15:
        result.append(1)
    else:
        result.append(0)

flights['Result']  = result


flights = flights.drop(['ORIGIN_AIRPORT', 'DESTINATION_AIRPORT', 'ARRIVAL_TIME', 'ARRIVAL_DELAY'],axis=1)
flights = flights.values
X,Y = flights[:,:-1], flights[:,-1]
```



```
print("Confusion Matrix:")
print(result1for)
print('')
result2for = classification_report(Y_test, forestpred)
print("Classification Report:",)
print (result2for)
result3for = accuracy_score(Y_test,forestpred)
print("Accuracy:",result3for*100)
print('')
result4for = precision_score(Y_test, forestpred)
print("Precision: ",result4for*100)

#vectorizer = TfidfVectorizer(min_df=1)
#X_train = vectorizer.fit_transform(X_train)

plt.subplots(1, figsize=(10,6))
plt.title('Receiver Operating Characteristic - DecisionTree')
y_pred_proba = clf.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(Y_test,  y_pred_proba)
plt.plot(fpr, tpr)
plt.plot([0, 1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()




plt.subplots(1, figsize=(10,6))
plt.title('Receiver Operating Characteristic - RandomForest')
y_forestpred_probability = forestclf.predict_proba(X_test)[::,1]
flforpred, fltruepred,_ = metrics.roc_curve(Y_test,y_forestpred_probability)
plt.plot(flforpred, fltruepred)
plt.plot([0, 1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")

plt.ylabel('True Positive Rate')
```