**FLIGHT DELAY PREDICTION SYSTEM**

**BUAN 6341.501(FALL 2022) – APPLIED MACHINE LEARNING**

**Presented to Professor Zixuan Meng**

Graphical user interface

Description automatically generated

**Group7**

Sangram Tushar Mohite Patil (SXM210110)

Saurabh Dhananjay Jadhav (SDJ200000)

Ananya Reddy Aegumamidi (AXA210029)

Harshal Mahadik (HXM190046)

Ishika Bhargava (IXB200014)

**TABLE OF CONTENTS**

Table of Contents

[Introduction 2](#_Toc122030691)

[Objective & Approach 3](#_Toc122030692)

[Data cleaning & Preprocessing 3](#_Toc122030693)

[Exploratory Data Analysis 5](#_Toc122030694)

[Data Modelling 6](#_Toc122030695)

[Models 7](#_Toc122030696)

[**A. Random Forest Classifier** 7](#_Toc122030697)

[**B. KNN(K-Nearest Neighbours) Classifier** 8](#_Toc122030698)

[**C. Decision Tree Classifier** 9](#_Toc122030699)

[**D. Naive Bayes** 11](#_Toc122030700)

[Conclusion 12](#_Toc122030701)

[References 12](#_Toc122030702)

# **Introduction**

In the United States, the airlines reported 20.1% of the domestic flights were delayed by 15 minutes or more. The financial losses that the aviation industry is going through are primarily because of flight delays. They inconvenience airlines and passengers as the increased travel time increases the expenses associated with the passenger's accommodation and sometimes causes stress among passengers. The delays further affect the air traffic at the airport. These delays may arise due to air congestion, weather conditions, difficulties boarding passengers, or mechanical problems. So, what can be done as a passenger to avoid delayed flights? Is it possible to know if the flight gets delayed before it comes up on the departure boards? Or before the passenger is inside the plane? The answer to these questions is yes.

# **Objective & Approach**

This project aims to predict the flight delay using the full spectrum of available data. This project will: Apply and compare different ML models to get the best accuracy for the predictions, deliver flight delay predictions, i.e., for a specific flight on a particular day, giving the probability of the flight delay and Alert travelers that the flight might get delayed even before it arrives.

This project follows a standard approach which helps to predict the flight delay using efficient techniques. Methods that have been performed in a sequential manner to get the best possible results are Reading the data, Data cleaning & Pre-processing, Exploratory Data Analysis, Visualization, Model Building, Finalize and Validate the Data model, and Making Prediction.

# **Data cleaning & Preprocessing**

The first step included finding a good data set with all the initial requirements which could help us to get to the achievement of project objective. Our data is sourced from Kaggle. This dataset is of Monthly Air Travel Consumer Report from the Department of Transportation (DOT) from 2015 which consists of a summary of the number of on-time, delayed, cancelled, and diverted flights Following are three files (Airlines.csv, Airports.csv, Flights.csv) consume all the data and there are approximately 5819079 number of observations and 40 columns.

We must do some data cleaning before we can get to the actual analysis for our project. This  
dataset has a few issues that would make analysis difficult, so this section aims to fix those  
problems. While this section is mainly data cleaning, there is some exploratory data analysis for the purpose of examining how to best clean the data. The first thing we do is to read data set Flights.csv for predicting.

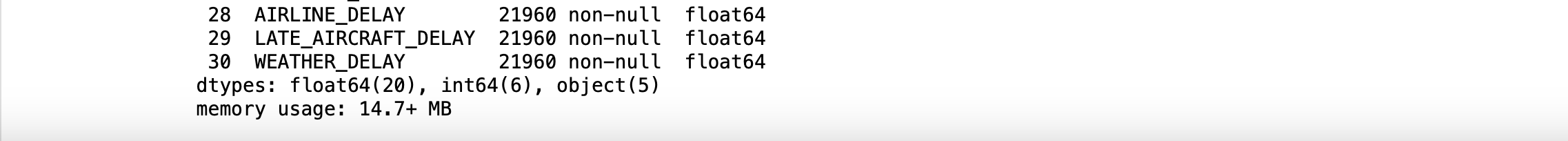
Graphical user interface, text

Description automatically generated

The next thing we do is we have taken an initial look at the dataset to get an understanding of it. Initially there are 62006 observations and 31 columns.

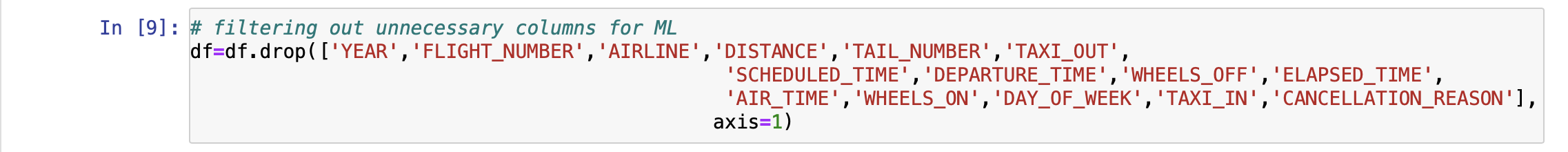
Table

Description automatically generated with medium confidence



For Modelling purposes, we have taken first 100k column records in the flight data and printing out to see all the columns available. The next thing we do is see if there are any NAs in the dataset, and if there are, remove them.

Next, we will drop the useless columns that will not be helpful in the analysis. We have dropped the group of columns because they don’t add value to machine learning values and also by doing this it improves the speed and efficiency of the model.



Now, we are filling Nan and 0 values with mean



Now removing some more columns that will not be used in analysis and printing out the columns to reverify again.

Table

Description automatically generated

Table

Description automatically generated

This concludes the data cleaning section. From here, we will move on to exploratory data analysis.

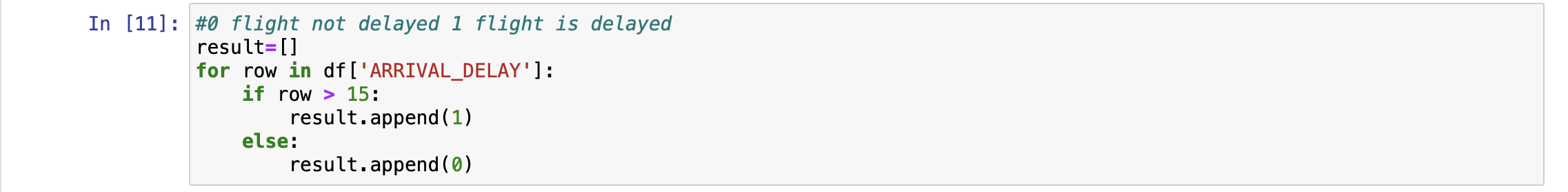
# **Exploratory Data Analysis**

First and foremost, we counted the number of flights that got diverted and the below is the output for the same.



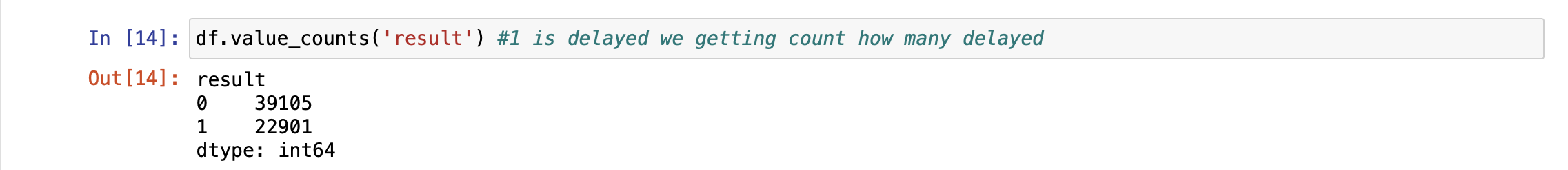
Next, finding number of delayed flights and the count of number of delayed flights are taken from the first 100k records.

Finding the flights delayed we have created a list that is result, where we find: if the value is greater than 15, then true is appended to the list and if it is false 0 is appended to the list.





Now counting number of times, the values of 0 and 1 occurred in the result. Here 1 shows number of flights delayed. Therefore, the number of flights delayed are 22901 i.e., 22.9% taken from first 100k records.



# **Data Modelling**

In this section, we will go over the models we tested, their performance, and which model we  
ultimately chose for the flight delay prediction system. After validating all the classification models, the models that were chosen based on their performance are Random Forest Classifier, KNN Classifier, Decision Tree Classifier, and Naive Bayes (Gaussian).

This is the data which we are working with after data cleaning. Before using the models we must split the data into two parts they are as training set and test set in the ratio of 70:30. Here the training data set is used for fitting the model, that is, to estimate the unknown parameters in the model and the test data set is used for evaluating its accuracy.The training set should be larger than the test set so that there is more data with which to train the models. The training set consists of 70% of the data.

Graphical user interface, text

Description automatically generated

## **Models**

### **A. Random Forest Classifier**

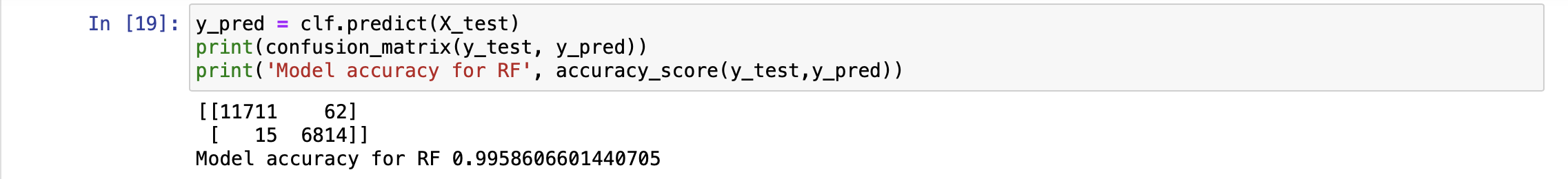
Random forest algorithm builds decision trees on different samples and takes their majority vote for classification.

Firstly, we are importing the package of random forest classifier for building the random forest model. Then importing standard scaler package because the variables need to be scaled for the future analyses to be more accurate. We use the standard scaler to normalize the data.

In the below code, we are finding the testing and training accuracy of the random forest model.

Testing Score: 0.99586, Training Score: 1.0

Next, obtaining the confusion matrix and then the Model accuracy for RF: 0.995860Graphical user interface, text, application, email

Description automatically generated

Then Plotting the confusion matrix of Random Forest Classifier

Graphical user interface, application

Description automatically generated

After this we also found the F1, Precision and Recall score for the random forest classifier. Here, The macro average is the arithmetic mean of the individual class related to precision, memory, and f1 score. We use macro average scores when we need to treat all classes equally to evaluate the overall performance of the classifier against the most common class labels. The obtained F1, Precision, Recall score are 0.9955 , 0.99485, 0.99626

Text, letter

Description automatically generated

### **B. KNN(K-Nearest Neighbours) Classifier**

KNN identifies the K nearest neighbours to a given observation point. It then uses K points to evaluate the proportions of each type of target variable and predicts the target variable that has the highest ratio.

Here as random forest classifier , KNN classifier, and GridSearchCV packages are imported. Here we are taking the nearest neighbours as 3 and then finding the mean validation accuracy and test accuracy for the best K. The output obtained are: Mean Validation Accuracy for chosen K: 0.95212, Test Accuracy for the best K: 0.956079

Graphical user interface, text, application, email

Description automatically generated

Next, made the confusion matrix and plotted. Also obtained the Model Accuracy for KNNC is 0.987

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Now, we obtained the F1, Precision and Recall score 0.98684, 0.98569, 0.98804 as follows

Text, letter

Description automatically generated

### **C. Decision Tree Classifier**

Decision Tree is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

First we are importing the package to use the model and then using standard scaler to normalize the data. We are optimizing first before building the tree to avoid overfitting(pre-pruning). Then we obtained the training and test data set scores as 0.99182 and 0.99205

Graphical user interface, text, application, email

Description automatically generated

As a next step, the confusion matrix is done and plotted and therefore printing Model Accuracy for DCT as 0.99182.

Graphical user interface, text

Description automatically generated

Graphical user interface, application, email

Description automatically generated

Then, obtained values.F1 score: 0.99124, Precision score: 0.98931, and Recall score: 0.99329

Text, letter

Description automatically generated

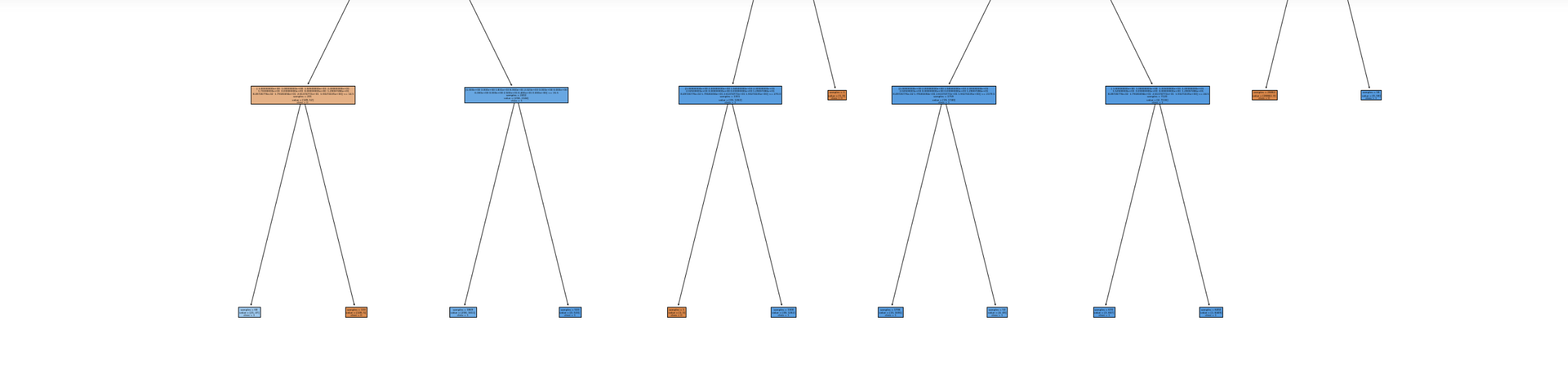
Now, plotting the decision tree,

Text

Description automatically generated

Below is the decision tree plot.

A picture containing sky

Description automatically generated

### **D. Naive Bayes**

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. In this present naïve bayes algorithm, we are using gaussian naïve bayes model.

As a part of first step, we are importing the package for building the gaussian naïve bayes model. and then found Testing Accuracy: 0.98940 and Training Accuracy: 0.98914

Then, as a part of second step made the confusion matrix and found the Model Accuracy for GNB as 0.98940

Finally, obtained the values as.F1 Score:0.98867, Precision Score: 0.98598 , and Recall Score: 0.99163

Graphical user interface, text, application, email

Description automatically generated

**Results**

we obtained the output of performance scores of all the four models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Model Accuracy** | **F1 Score** | **Precision Score** | **Recall Score** |
| Random Forest | 99.58% | 0.99555 | 0.99485 | 0.99626 |
| KNN | 98.77% | 0.98684 | 0.98569 | 0.98804 |
| Decision Tree | 99.18% | 0.9893 | 0.98931 | 0.99329 |
| Naive Bayes(Gaussian) | 98.94% | 0.98867 | 0.98598 | 0.99163 |

We can see from the above outputs the model accuracy that random forest performed the best, followed by decision tree, and then followed by Naïve Bayes (Gaussian) much lower is the KNN classifier. The training score should not be the one with which we decide the best model, however, since it is prone to overfitting. We use the test data for an unbiased score since it is data that has not been shown to the model during training. For the test scores, the random forest model performed better than the Decision tree model and better than the Naïve Bayes (Gaussian) and much better than the KNN classifier in F1, Precision and Recall score. Based on these scores, the best choice is the random forest model, and this is the one that  
we will select for predicting the flight delay.

# **Conclusion**

The purpose of this project was to find which model is best at predicting the flight delay of  
flights using the data from Kaggle. We performed data cleaning to make the data usable and more robust for the models, we did exploratory data analysis to gain insights about the data, and we created four models and compared their test performances. In the end, the random forest model had the best test scores, so we selected this model to perform flight delay predictions in the future.

# **References**