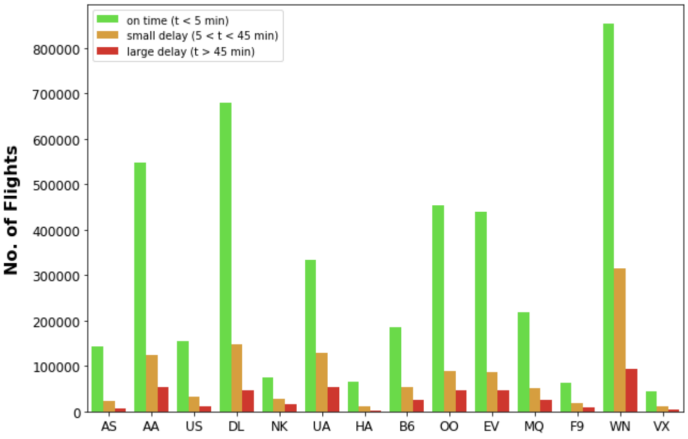
Flight Delay Prediction Using Machine Learning

Pranay Kumar Verma (39606-8464)

December 2019

1 Introduction

In present day scenario, Time is money. Flight delays end up hurting airports, passengers and airlines. Being able to predict how much delay a flight incurs will save passengers their precious time as well as hardships caused due to flight delays or in worse cases cancellations. The problem I am trying to solve is to accurately predict flight delays when we have certain features of the flight with us, like airlines who operate them, distance they have to cover, origin airport, target airport, departure times and so on. Being able to accurately predict flight delays can help the passengers know what delays they should be ready to face depending on where they fly from and the airlines they choose to fly. This can enable them to take a buffer, so they do not end up missing connecting flights or meetings. The goal of the project will be to do in depth analysis of the data and play with the input features to see how the prediction accuracy changes. The development of prediction models that perform accurately is difficult due to the complex nature of air transport. Below is a plot of different airline operators and the number of flights that incurred delays in comparison to those that were on time.

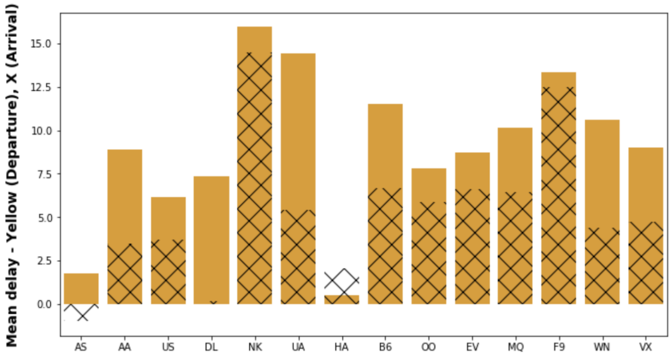


Delay Statistics

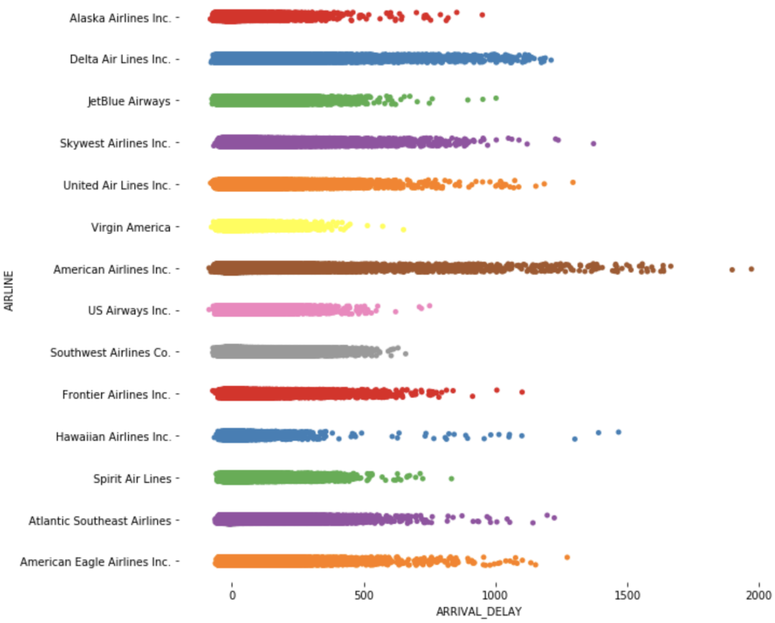
2 Prior Work

There has been a lot of research on how to deal with the problem of predicting flight delays using a slew of machine learning techniques, deep learning and even big data methodologies to predicting flight delays. But most try to focus only on one airline or just a few airports in their analysis, my goal is to create a model that can handle all flights and destinations. Below are some of the popular papers on this field -

- Chakrabarty [5] proposed a Model which made use of Gradient Boosting Classifier to predict the arrival delay for AA(American Airlines) among the top 5 busiest US airports. This paper was used to understand the basic underlying principles of how gradient boosting can be used to enhance machine learning models for classification.
- Manna [8] created a model using Gradient Boosting Regressor after analysing the raw flight data. The model aimed to predict flight arrival and departure delays. This paper was referred to understand the research on Machine Learning Algorithm Gradient Boosted Decision Tree and how it was applied to flight delay prediction.



(a) Airline Mean Delays



(b) Airlines arrival delay distribution

- Ding [7] used Naïve Bayes, C4.5 and also a multiple linear regression model, and compared their outcomes. This paper was utilized to understand Naïve Bayes, and how to compared the results of different models.
- Ni [10] This paper illustrated the relationships between the problems of flight delay prediction and machine learning algorithms like SVM and Logistic Regression.

The goal is to first predict the arrival delay the flight will incur including the departure delay as one of the features. Then departure delay feature will be dropped and try to predict it using the remaining features and gauge how it performs. In addition the plan is to also run some classification models to see how classification does on our data set. Above are some visualizations of the arrival and delay times (Figure a) as well as the average delay per airline (Figure b).

3 Models and Methodologies used

3.1 The DataSet

The dataset was located from Kaggle [1], this dataset was collected from U.S department of transportation. The dataset tracks the performance of domestic flights within the united states. This dataset has information about flights from 2015, and it contained the flowing information about the flights Year, Month, Day, Day Of Week, Airline, Flight Number , Tail Number, Origin Airport, Destination Airport, Scheduled Departure, Departure Time, Departure Delay, Taxi Out, Wheels Off, Scheduled Time, Elapsed Time, Air Time, Distance,Wheels On, Taxi In, Scheduled Arrival, Arrival Time, Arrival Delay, Diverted, Cancelled, Cancellation Reason, Air System Delay, Security Delay, Airline Delay, Late Aircraft Delay, Weather Delay.

3.2 PreProcessing

Before we begin the process of training the models, it is essential to perform some preprocessing steps. The techniques and methodologies used for preprocessing are mentioned below :

1. *Handling missing values* – The dataset contains small percentage of missing values for certain columns like Departure delay, taxi out and so on. These rows containing missing values are dropped as they make up a very small portion of the dataset.
2. *Formatting times* – Initially the times in the dataset are in the form of 4 digit numbers which are not of much use, so these are transformed into HH:MM format. New columns which have the formatted time are created for Departure time, Scheduled arrival, Scheduled departure and arrival time.

3. *Feature selection* - Some of the features listed above are not really needed for the prediction of delays, so the following were dropped only the following features were kept for the prediction purposes Airline operator, Origin Airport, Destination Airport, Distance, Actual Departure, Date, Day, Scheduled Departure, Departure Delay, Actual Arrival, Scheduled Arrival, Arrival Delay, Scheduled Time, Elapsed Time, Air Time, Taxi In, Taxi Out, Diverted.

4. *Label Encoding* – Some of the features are in the form of a string these were converted to number values using Label encoder and were assigned number beginning with zero, this is done so that the dataset is more machine learning friendly as models tend to not perform well with strings as features.

5. *Normalize the values and scale* – Using pythons inbuilt library called standard scalar the dataset was scaled to have mean of zero and a standard deviation of 1. This is done to get all features to the same scale relatively as the features vary in magnitudes and the distance metrics that the algorithms use internally are sensitive to a large variation in magnitudes.

6. *Feature creation for classification* - A new feature with binary value 0 and 1 was created to run the classification model. This feature was created on the basis of delay time, if it was greater than 0 the value was set to 1 else it was set to 0.

3.3 Models Used

Multiple models were used for both regression and classification. The models used for regression and classification are as follows :

3.3.1 Regression

1. *Simple linear regression* - This approach tries to find a linear relationship between the value to be predicted and the attributes that are being used to predict the same. This is one of the simplest machine learning algorithms which works by trying to obtain a formula for the prediction of one variable using others provided there exists a causal relationship in between them. The basic intuition of liner regression can be expressed by the below formula where f are the features we use for the prediction, [11] Δ corresponds to the weights of each of them and ϵ is an arbitrary constant.

$$\mathrm{y}_{\mathrm{pred}} = \Delta_1 f_1 + \Delta_2 f_2 + \Delta_3 f_3 + .. + \Delta_n f_n + \epsilon$$

2. *Random forest regression* - This is an ensemble technique which can be used for both regression as well as classification tasks. It creates multiple decision tress using a technique called *bagging* [2]. Bagging involves training all the decision trees on different data samples. The final prediction is made by combining the results of all the decision trees rather than just relying on one of them.

3. *Boosted linear regression* - This is an ensemble machine learning method which combines multiple weak models into one. It builds the model stage wise with the intuition that the next best model when combined with all the previous models will minimize the overall prediction errors [2].

3.3.2 Classification

1. *K neighbors classifier* - This algorithm works by first computing the k closest neighbors of the value that needs to be predicted and based on those neighbors it assigns a class value. The K-nearest neighbour algorithm [3] is as follows -

   - Find k value, which is number of nearest neighbours
   - Compute distances between data to be predicted and the training data and sort them.
   - Check labels of k nearest neighbours and assign the majority class value as the prediction.

2. *Logistic regression* - This algorithm uses a logistic model which has an 'S' shaped curve to predict the values. This model is used when there are 2 output classes like true or false, it works by calculating probabilities and converts it into a function [6]. It uses the hypothesis equation:

$$\mathrm{h}_\theta(\mathrm{x}) = \mathrm{g}(\theta^{\mathrm{T}}\mathrm{x}) = \frac{1}{1 + e^{\text{-}\theta^{\mathrm{T}}\mathrm{x}}} \qquad \text{where } \theta^{\mathrm{T}}\mathrm{x} = \theta_0 + \textstyle\sum_{j=1}^{n} \theta_j - x_j$$

We can find the parameter θ using gradient ascent and max likelihood estimation as per the below equations:

$$\theta \colon = \theta + \alpha \nabla_{\theta} \mathrm{l}(\theta)$$

$$\mathrm{l}(\theta) = \textstyle\sum_{i=1}^{m} y^{(i)}\, log(h(x^{(i)})) \;+\; (1 - y^{(i)})\, log((1 - hx^{(i)})$$

3. *Decision trees* - The main idea of this algorithm is to create a tree structure with each node containing a choice to either go towards the left or right branch. At each level the node puts forward a simple question to which the answer is either true or false, and based on it the data is partitioned into 2 subsets. The goal of this algorithm is to continue to ask questions and keep building the tree until it can get get the purest possible splits. To judge the impurity at each level decision trees use metrics like Entropy or Gini value to quantify impurities. Usually the induction process is very slow but the deduction is very fast as it just needs to traverse the created tree and reach the leaf.

3.4 Metrics Used

3.4.1 Regression Analysis

The following metrics are used to evaluate the performance of the models used for regression.

- *Mean Absolute Error (MAE)* - This tells us the variations in between the expected and actual values of the predictions.

  $$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_i - x_i|$$

  Where $\mathrm{y_i}$ is the predicted value and $\mathrm{x_i}$ is the actual value for the $\mathrm{i^{th}}$ instance

- *Mean Squared Error (MSE)* - This measures the average of the sum of the squared errors.

  $$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n}(y_i - x_i)^2$$

  Where $\mathrm{y_i}$ is the predicted value and $\mathrm{x_i}$ is the actual value for the $\mathrm{i^{th}}$ instance

- *Root Mean Squared Error (RMSE)* - This is just the squared root of MSE, this frequently used as a measure of difference over MSE as the units end up being the same as original

  $$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_i - x_i)^2}$$

  Where $\mathrm{y_i}$ is the predicted value and $\mathrm{x_i}$ is the actual value for the $\mathrm{i^{th}}$ instance

- *R2 Score (Coefficient of Determination)* - This is a statistical measure of how close the actual values are to the fitted regression line.

  $$\mathrm{r}^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

  Where $\mathrm{y_i}$ is the actual value , $\hat{y}_\mathrm{i}$ is the predicted value and $\bar{y}$ for the $\mathrm{i^{th}}$ instance

3.4.2 Classification

The following metrics are used to evaluate the performance of the models used for classification.

- *Score* - This is the default evaluation method that is in-built in each of the classifiers

- *Precision* - This tells us that the values the classifier predicted as true, how many of them were actually true.

  $$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- *Recall* - Recall tells us that how many true values was the classifier able to correctly recall from what it has learnt

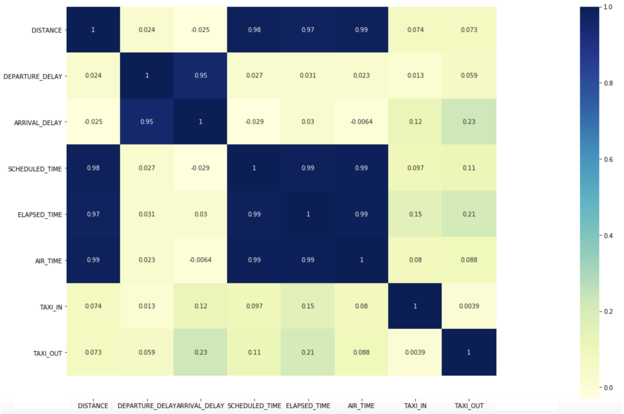
  $$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- *F1 Score* - This is the harmonic mean of precision and recall

  $$\text{F1 Score} = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

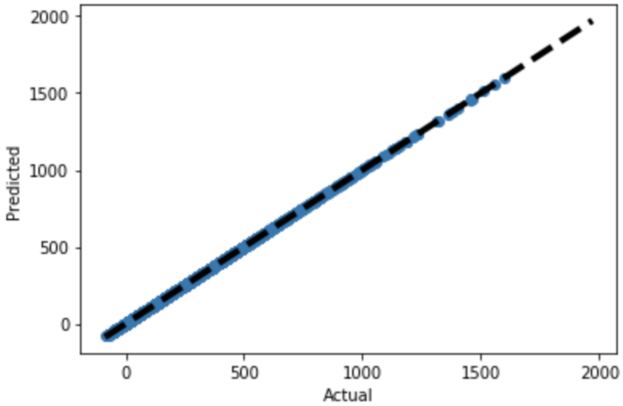
4 Results and Findings

The first thing I do is to plot a correlation matrix to show us the relation between the different variables in the dataset. The correlation matrix gives us a great insight as to which variables are related to each other, the one we are most interested about is arrival delay. We can clearly see there is a high correlation in between the departure delay and arrival delay. Even though they have a high correlation some flights do actually arrive on time even after they have a departure delay. Starting off we keep the highly correlated variable in the training set of attributes to see how it performs. But, the really interesting part would be when we remove departure delay how does this affect the performance of our models.



4.1 Regression Analysis

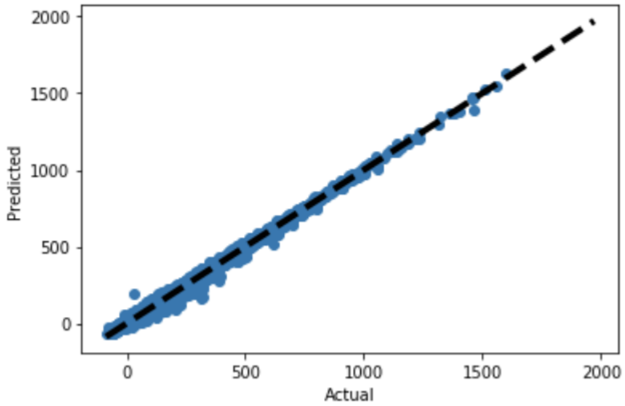
1. *Linear Regression* - Starting off with one of the simplest regression models, which is trained using the cleaned data set that has been split into train and test parts.



- MAE = 1.5327891237063537e-06
- MSE = 3.0655780798908132e-06
- RMSE = 0.0017508792305269982
- R2 Score = 0.9999999980588673

As observed, we are able to predict most arrival delays accurately with a mean error of less than 1. With such a low MAE as well as RMSE we can confirm the algorithm worked admirably for this particular dataset. Let us have a look at some more models were used.

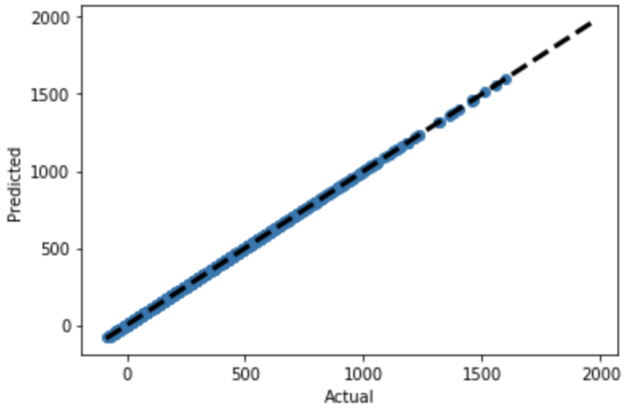
2. *Random Forest Regression* - Applying this advanced machine learning algorithm to see how it does compared to linear regression.



- MAE = 0.6128663078407527
- MSE = 4.1269854484672495
- RMSE = 2.031498325981897
- R2 Score = 0.9973867810876257

From the numbers as well as from the figures it is obvious that linear regression performed better than random forest. Let us try out another more complex variant of linear regression as it seems to be performing well.

3. *Boosted Linear Regression* - Boosted linear regression is a variant of linear regression but much more complex. It internally uses a variety of methods to fit the model and come up with results. Lets have a look at the results.

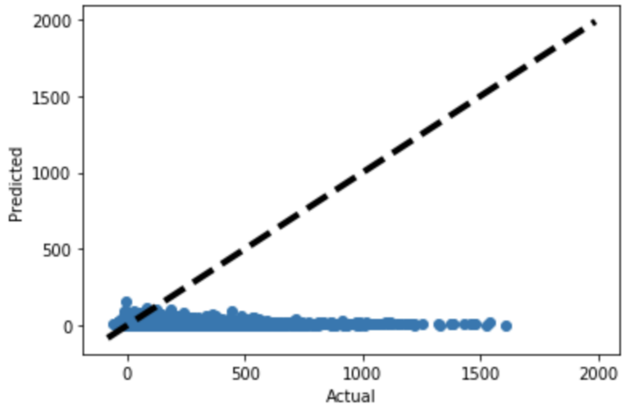


- MAE = 1.5327891007407566e-06
- MSE = 3.06557807989065e-06
- RMSE = 0.0017508792305269516
- R2 Score = 0.9999999980588673

On comparing with simple linear regression, this algorithm does not do much better although it is much more complex. For a the complexity it has, it does not show better results than an algorithm that is fairly elementary. We can see the scores are nearly identical and so is the plot.

4.2 Regression Analysis (After Removing the highly correlated variable)

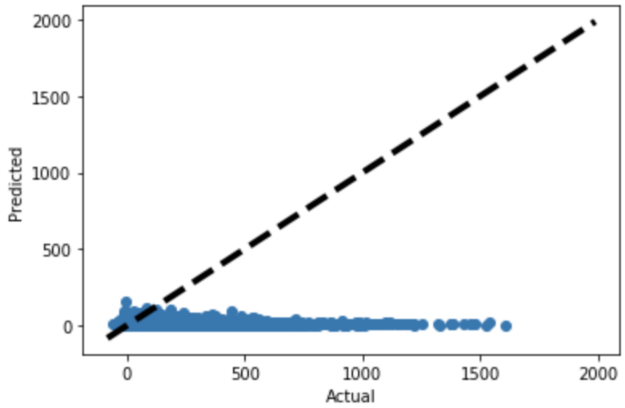
1. *MLP Regressor* - Thinking that now the prediction task is much more difficult, starting of with a neural network to predict the delays.



- MAE = 18.92521792240611
- MSE = 1376.49885439204
- RMSE = 37.10119747921945
- R2 Score = 0.013987506217613577

As now no variables have a high correlation with the arrival delay which we are trying to predict, it becomes very difficult even for a neural network to predict arrival delay with low margins of error. Just to confirm that the prediction task is going to be substantially more difficult after the removal of the only variable which has a high correlation, trying out a different method to follow.

2. *Random Forest Regression* - Trying out a less advanced technique to see how it fairs in comparison

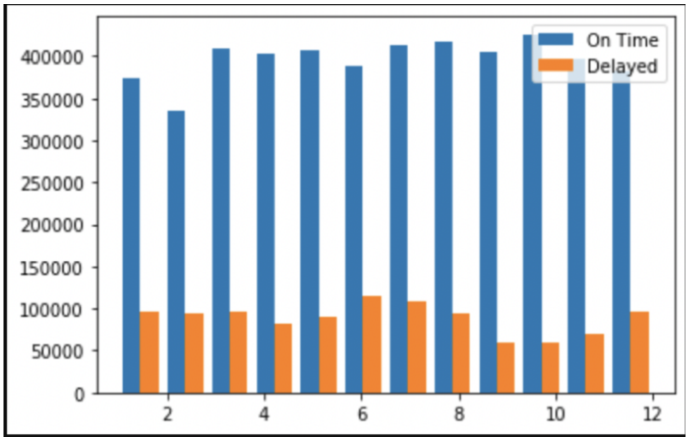


- MAE = 21.2618474705449
- MSE = 1619.6116813587962
- RMSE = 40.2443993787806
- R2 Score = -0.16015886813142388

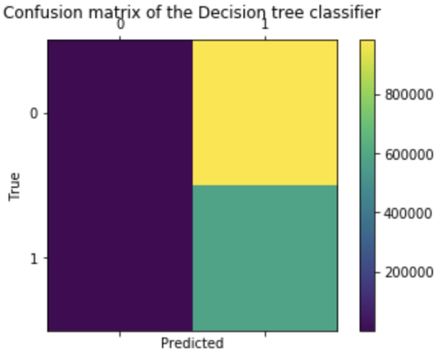
As we can see it performs even worse, the R2 score comes out to be negative which means the algorithm has not learned anything. Thus, removing the only variable with a high correlation with the value we are trying to predict is not such a good idea. A few more models were tried out and every time the results just got worse thus they have omitted them from this report, but they can be found in the code file.

4.3 Classification Analysis

The dataset we have is imbalanced i.e it contains about 80 percent of flights that had no delay and only 20 percent which have a delay. We needed to handle that before we begin to proceed with any of the algorithms. A technique called Synthetic Minority Over-sampling Technique (SMOTE) was used, what this does is that it over samples from the class in minority in other words it generates 'fake' samples to resemble those of the class in minority to balance the class. This is very important otherwise our algorithm can just predict 'no' all of the times and still get 80 percent accuracy. Shown in the Figure is a representation of the class distribution as seen in the original dataset.



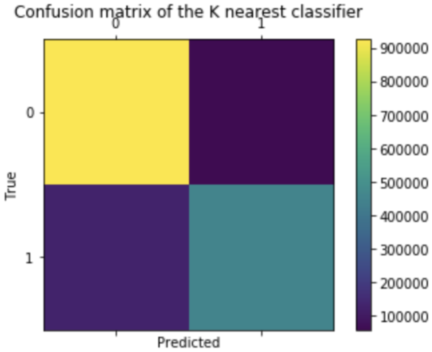
1. *Decision Tree Classifier* - First, decision tree is used to predict the flight delay. Below were the stats after the algorithm was trained and used to predict.



- Score = 0.9826194584914554
- F1 score : 0.2725298912293622
- Precision Score : 0.6644134619299129
- Recall Score : 0.5006117468892067

We can see that decision trees resulted in a classifier with a very high score but the F1 Score, precision and recall all were not upto the mark. The confusion matrix clearly tells us the model did not perform well. The True Positive value of the model is very low and thus it implies that the model is not able to predict most of the flight delays.

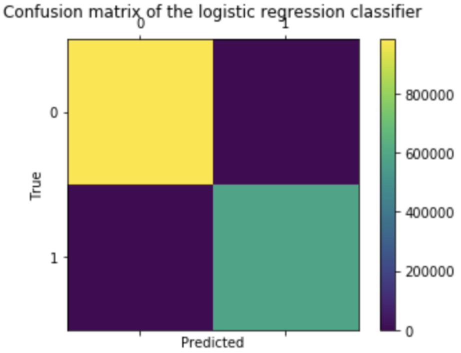
2. *K neighbors Classifier* - This technique works by finding K Nearest Neighbours of the row that needs to be predicted. But in this case, finding neighbours is not so easy as, firstly we have a lot of attributes and as we know in a high dimensional data space all points are closer to each other.



- Score = 0.3719942980276847
- F1 score : 0.8696222002024336
- Precision Score : 0.8825899500527832
- Recall Score : 0.8609813308550668

The model seems to be performing quite well in comparison to the previous algorithm. This algorithm has a high number of true positives as well as true negatives, but it still makes about a 15 percent error in its predictions. Let us try and see if it can be made even lower by using a different technique.

3. *Logistic Regression* - This method uses a logistic function to create a model for a binary function. It estimates probabilities using this function and assigns the class based on them.



- Score = 1.0
- F1 score : 1.0
- Precision Score : 1.0
- Recall Score : 1.0

This model results in perfect prediction!! All the scores are perfect and nothing is misclassified. This model perfectly fits this dataset after SMOTE and the preprocessing that was done. This is a significant improvement over the previous technique and based on this we can to some extent conclude that the variables have a linear relation in between them, as in both classification as well as regression the techniques that aim to find linear relations performed the best.

5 Conclusion and Further Work

After this project it is imminent that the choice of the methods that can be used with noteworthy results is highly dependent on the balance of the dataset, the type of problem (Regression or Classification). Many machine learning models like Logistic Regression, Decision Tree Classifier, Random Forest Regression and Linear Regression and its variant Boosted Linear Regression were applied to predict the arrival delay of flights. If given the right set of input parameters, even these simple algorithms were able to predict the delays with high accuracy. Although good results were obtained, there is a huge scope for future work. If weather and air traffic control information is made available we can then go on to predict arrival delay even without the inclusion of departure delay as an attribute. Also, we can progress into predicting if a flight will be delayed or cancelled based on weather factors like snow, rain, storms and so on.

References

[1] Dataset Acquired from: `https://www.kaggle.com/usdot/flight-delays`.

[2] Random Forest Regression and Boosting (Afroz Chakure): `https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f`.

[3] K Nearest Neighbors: `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#Algorithm`.

[4] `https://en.wikipedia.org/wiki/Flightcancellationanddelay`.

[5] Navoneel Chakrabarty, Tuhin Kundu, Sudipta Dandapat, Apurba Sarkar, and Dipak Kumar Kole. Flight arrival delay prediction using gradient boosting classifier. In *Emerging Technologies in Data Mining and Information Security*, pages 651–659. Springer, 2019.

[6] Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*. John Wiley & Sons, 2015.

[7] Yi Ding. Predicting flight delay based on multiple linear regression. In *IOP Conference Series: Earth and Environmental Science*, volume 81, page 012198. IOP Publishing, 2017.

[8] Suvojit Manna, Sanket Biswas, Riyanka Kundu, Somnath Rakshit, Priti Gupta, and Subhas Barman. A statistical approach to predict flight delay using gradient boosted decision tree. In *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5. IEEE, 2017.

[9] Brett Naul. Airline departure delay prediction.

[10] Jianmo Ni, Xinyuan Wang, and Ziliang Li. Flight delay prediction using temporal and geographical information.

[11] Astrid Schneider, Gerhard Hommel, and Maria Blettner. Linear regression analysis: part 14 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International*, 107(44):776, 2010. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2992018/`.