**Group 6: Predictive Analysis on Flight Delays**

|  |  |
| --- | --- |
| First Name | Last Name |
| Vineel Lakshman | Patchipulusu |
| Akshay | Srinivasan |
| Jashkumar | Shah |

Table of Contents

1.Introduction…………………………………………………………………………………………………………………………………….2

[2. Data 2](#_Toc467438673)

[3. Plans 2](#_Toc467438674)

4. KDD…………………………………………………………………………………………………………………………………………………3

4.1DataProcessing……………………………………………………………………………………………………………………………….3

4.2Data MiningProcess……………………………………………………………………………………………………………………….4

5. Evaluation Results…………………………………………………………………………………………………………………………..5

5.1 Evaluation Methods……………………………………………………………………………………………………………………….5

5.2 Results and Findings………………………………………………………………………………………………………………………6

6. Conclusions and Future work………………………………………………………………………………………………………….9

6.1 Conclusions…………………………………………………………………………………………………………………………………………9

6.2 Limitations……………………………………………………………………………………………………………………………………10

6.3 Potential Improvements or Future Work……………………………………………………………………………………..10

# **1. Introduction**

As a foundation inspiration to the project problem, consider that the yearly cost of domestic flight delays to the US economy was evaluated to be $30-40 billion per year as per the Joint Economic Committee, US Senate. Accurately anticipating flight delays permits travelers to be set up for the interruption of their voyage and permits aircraft to star effectively react to the potential reasons for the flight delays to relieve their effect.

For almost all the cities there are two airports like San Francisco International Airport (SFO) and Oakland International Airport (OAK) sit directly across the San Francisco Bay from each other and are separated by about 12 miles of water. Similarly, John F. Kennedy International Airport and Newark Liberty International Airport also sit directly across Hudson River and they are separated by about 34 Miles. Lastly, we have Midway International Airport and O’Hare International Airport separated by about 20 miles.

This model is designed for the business users who are very clear to travel on a certain date but are uncertain as to the probabilities on that date of a flight delay. The output from our model will be a simple recommendation of the airport from one of the three locations stated, from which the traveler should depart with his/her delay prediction. Accordingly, this will entail the development of a classification model.

# **2. Data**

To conduct this analysis and build a predictive model we obtained publicly-available data on flight arrivals and departures for major U.S. airports from the American Statistical Association website. The dataset entails 30 variables that describe each flight in terms of departure/arrival date and time, carrier, taxi time, time spent in the air, as well as departure and arrival delays and their causes. Although the entire dataset contains flight records from 2008 to 2015, we elected to only work with the data from 2013 through 2015. This ensured variable consistency across the entire dataset since data files for earlier years have variation in variable availability. This choice also helped us avoid El Nino/La Nina (Effect due to weather) years (2008, 2010, and 2011) which would introduce discrepancy into a model. The dataset is divided into individual files – one file per year – and contains around 7 million records each. Each record represents a unique flight record as a combination of a flight number and aircraft tail number.

# **3. Plans**

Here is our plan for this project.

**No. of rows of data per year:** 100000 rows out of which if we consider to predict for two airports it will be around 40,000 rows per year. However, it can be applied to other cities which have two airports to travel.

**No. of variables**: 30

**Variables used for classification**: (10) Year, month, Day of month, Day of week, Departure time, Airline carrier, Tail number, Origin, Destination, DepDelay.

Year – the year the flight departed

Month- specific month of the year

Day of month – the day of the month

Day of week – which day in a week

DepTime – The time of departure of the flight

Unique carrier – Which airlines the flight belongs to

Tail number – the number on the tail which identifies the flight of the airlines

Origin – from where the flight departs

Destination – the destination to which flight departs

DepDelay – how much time the flight delayed

b. Assumptions

To simplify model development process, the assumptions we made are as follows: -

1. Ignored the delay propagation between connecting flights: - Modelling the data with such factors into consideration would have been difficult. We had to consider advanced machine learning algorithms such as Markov chains and state space models

2. Ignore effect of cancelled flights

4. Travelers are agnostic to time sensitivity: We assumed that travelers only care about the date that they fly out, and are indifferent to the time of day of their flight. This assumption is mostly focused on model application than model development.

# **4. KDD**

## 4.1. Data Processing

**a. KNN**

* Data has been cleansed deleting the missing values and NA values.
* After performing the data cleansing, create a subset from the raw data consisting of necessary attributes required for analysis.
* Select only the data which has Origin SFO and OAK.
* Convert the categorical data variables Origin and UniqueCarrier to numeric to make data fit to the model.
* Create a delaylabel based on DelayTime attribute and then delete the column from the data.
* Normalize the data to predict accurately.

**b. Naïve Bayes**

* Two different types of pre-processing have been done on the data to find out the best which fits to the algorithm.
* Cleansed data is taken and made a subset with the necessary attributes required for analysis.
* Filter the data where Origin is SFO and OAK.
* Further processing is done in two different ways

1. DepTime attribute is considered as it is in the raw data.
2. DepTime is changed into slots in a day like Earlymorning (2:AM – 7:00AM), Morning(7:00AM – 12:00PM), Afternoon(12:01 PM – 4:00PM), Evening(4:01PM – 8:00PM), Night(8:01PM – 1:59AM).

* Convert the numeric data to factor levels to fit data to the model.

**c. Logistic regression**

To construct the logistic regression model, we chose the following set of variables as the model inputs and prepared a dataset for modeling using Python Pandas:

* Like the Naïve Bayes model, day of week variable was chosen to represent its impact on flight delay.
* Flight time was additionally chosen to represent components, for example, busiest time and time-of-day particular climate conditions, for example, fog. To give an appropriate calculated model info, this variable was changed into straight out factor. In view of our exploration, we observed that haze related postpones have a tendency to happen before 1pm5. Hence, we discretized time of day into classifications from 07:00 to 12:59 (pinnacle haze time at SFO), 13:00 to 17:59 (light hours), and 18:00 to 06:59 (evening time).
* Origin and Destination variables were selected to account for any airport-related inefficiency that could be contributing to flight departure delays.
* Finally, Airline Carriers such as American Airlines, Delta, Northwestern, SkyWest, Alaskan etc. were selected as an input variable to account for possibility of some airlines having a higher propensity for delays.

In order to implement the logistic regression model, categorical variables (i.e., Origin, Destination, and Airline Carrier) had to be transformed into binary since categorical or non-ordinal values are not meaningful inputs for a logistic regression. This was accomplished through a data pivoting procedure. For example, a variable for each carrier (over 350 in the entire dataset) was created and value of ‘1’ was assigned to a single variable that represents a carrier for a specific flight record. All other carrier variables were filled with zero. Thus, each carrier was treated as an independent feature (since carriers are not correlated). Origin and Destination variables were transformed in the similar manner.

This data transformation was very computationally intensive and it resulted in a dataset of high dimensionality. We got multiple memory errors when we tried building our dataset using Pandas. Thus, we decided to exclude some variables that were used in Naïve Bayes classifier, although we thought that they might have some predictive power. Thus, we decided to drop month, day of month, and aircraft tail number as input variables. Otherwise, we would have to create dummy variables for every unique tail number (more than 7000 in 2013 dataset alone) which would have further increased computational complexity.

## 4.2. Data Mining Processes

Since the flight delay prediction is a problem on classification, we decided to implement KNN, Naïve Bayes and logistic regression. Some other models can be explored to find the best model.

As to address the project question, three different classification problems are explored. We choose to build a KNN model because it’s a standard classification model and simple to train the dataset. Function for KNN exists in R libraries. Similarly, Naïve Bayes is also best suited classification when categorical variables are included and has been demonstrated to yield best results competitive to other algorithms. Function for Naïve Bayes also exists in R libraries.

Logistic regression is implemented because the parameter estimates are fully efficient and is suitable for predictions with relatively few variables. Results are easy to interpret and can draw conclusion on the relative impact of the model based on coefficients. We supposed that these models provide a best output and tradeoff between difficult of implementation and accuracy of results.

# **5. Evaluations and Results**

## 5.1. Evaluation Methods

**a. KNN classifier**

* While building the KNN model, we considered the same attributes which are used in Naïve Bayes.
* In order to implement the classifier, the categorical data Origin, Uniquecarrier is converted to numeric to accommodate the requirements of the model.
* We applied KNN classification for different values of K to find out the best K value in predicting the delays.
* Data is divided in training and testing sets. Two years of data is considered as training and one year as testing.
* Store the training and testing delay labels in different variables.
* Apply the algorithm on testing data based on the training data to predict the testing delay labels.
* Apply CrossTable function to calculate the accuracy in the predictions using KNN classification.

**b. Naïve Bayes model**

To build Naïve Bayes model the inputs considered are

* Month, DayofMonth, DayOfWeek, DepTime, UniqueCarrier, Origin, DelayTime were selected to account for effect in airline delays weekly, monthly or seasonal fluctuations.
* Origin was selected in suggesting the customer to board from specific airport as the other has a probability of getting delayed.
* Uniquecarrier was selected because airline had a propensity for delays due to maintenance or scheduling.
* DelayTime was used to calculate the Delay label and an airline which is delayed for more than 10 min was considered as delay.
* This model was implemented with two different types of preprocessing technique and accuracy of prediction was calculated for the same.
* Data is divided in training and testing sets. Two years of data is considered as training and one year as testing.
* Store the training and testing delay labels in different variables.
* Fit Naïve Bayes model to the training data and then apply it to testing data to predict the testing delay labels.
* With the help of Confusion matrix calculate the accuracy in the predictions using Naïve Bayes classification algorithm.

**c. Logistic Regression**

We built our own logistic regression model for this project instead of using any Python library. We modeled it as:

This model uses a gradient descent algorithm to estimate coefficients for each variable. The learning model is designed to iterate over the training data and converge to a global solution. Once the coefficient values are determined, the model is then tested against a different sample of the data.

**Model Training**

Instead of using the k-fold validation by year utilized for the Naïve Bayes classifier, we used a random sampling of data from the 2013-2015. First, SFO and OAK flights were separated out of the sample set. Then, a fixed number of delays from the SFO/OAK data were randomly selected. Next, a random sample of on-time flights from SFO and OAK was selected such that the number of on-time flights matched the number of delayed flights. Again, by ensuring an even mix of delayed and on-time flights, we avoided model over-fitting to only recognize on-time flights. The sample size was progressively increased, although the results showed little variation across sample sizes.

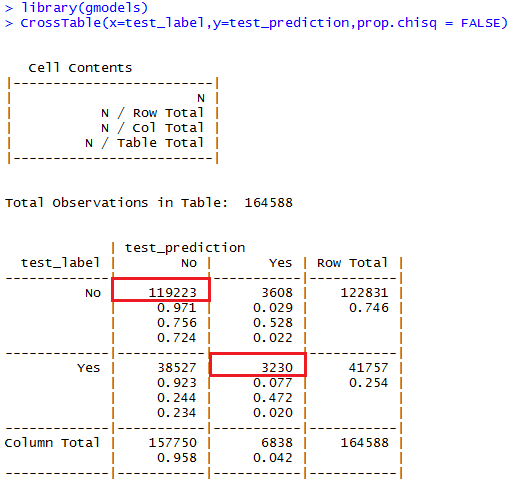
We encountered several problems during testing of this model. This suggests limitations of the model and our data. The number of delays in the sample was less than 25%. In order to maintain a 50%-50% mix in the training data, we were limited to using at most 60% of the data. However, a more significant challenge was that the training data size was limited by the data structure and memory of the hardware. Given the 350+ variables that we were attempting to train, a fairly large dataset is required to ensure there are sample cases of every variable and combination of variables in order to properly train the model. Because of the limitations on the size of the dataset that we could use to train the model, the model was not able to converge to produce true values for all coefficients. Retrospectively, it may have made sense for us to restrict the number of carriers to the major carriers (~30-50). This may have made convergence for the model more likely.

## 5.2. Results and Findings

**a. KNN**

* Test label predictions and test labels are placed in a CrossTable matrix to calculate the accurate prediction.
* Below is the result for taking **K=400** in KNN classification. From the below image True negative predictions are 119223 and True Positive predictions are 3230.
* Accuracy can be calculated as TP+TN/Total.

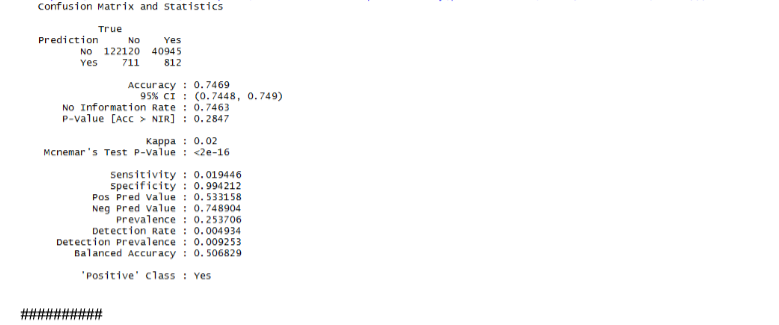
Accuracy = 119223+3238/164588 =74.399%



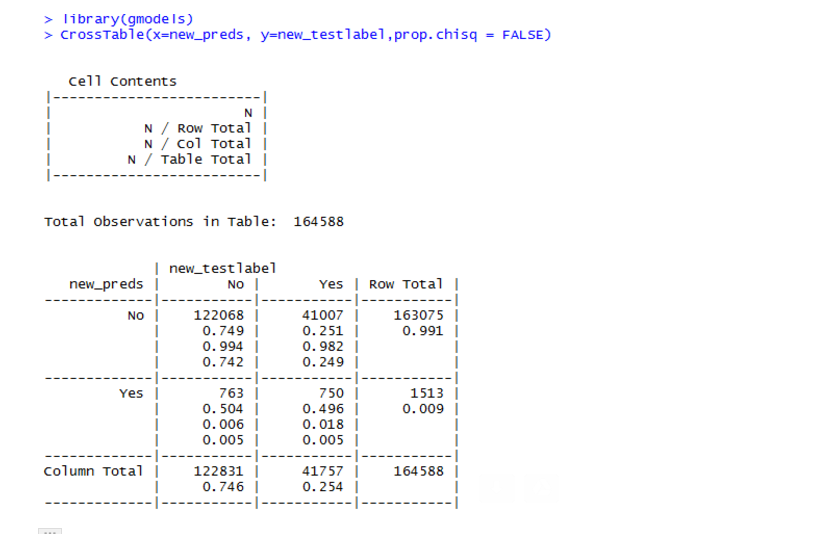
* Accuracy is calculated for different k values and is plotted below.

**b. Naïve Bayes**

* Naïve Bayes is implemented with two different pre-processing techniques and the results are shown below.
* Confusion matrix function was used to calculate the accuracy for 1st pre-processing technique and CrossTable was used for another technique.

**1.** 

**2.**

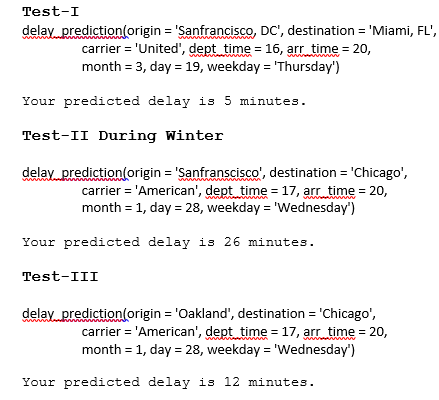


* 2nd technique shows the accuracy of the predictions after converting the day into slots.
* Accuracy = 122068 + 750/164588 = 74.62%

**c. Logistic regression**

Logistic regression can be a very powerful model for predicting binary outcomes, but there is a high cost to computing this model. The model must iteratively converge to a solution – a process that depends on the quality of the training data. A best practice for implementing logistic regression models is being thoughtful about the quality and structure of the data, as well as the predictive power of variables before implementing the model. We couldn’t use all the attributes available in the data because of the variable conversion. As mentioned earlier, converting Tail Number alone would have resulted in over 7000 additional dummy variables.

**Prediction: -**

****

**Model Selection**

In choosing our final model, it is difficult to make an objective comparison given the differences in training/validation methods used by models and model results. Given the limitations of the other two models encountered during its implementation and unreliability of its results, we elected to move forward with recommending Logistic regression for prediction and application.

# **6. Conclusions and Future Work**

## 6.1. Conclusions

We explored three models in attempt to predict departure delays at SFO and OAK. Our Naïve Bayes classification model performed relatively well and we believe it is a robust enough to be applied in predicting delays at all major U.S. airports in addition to SFO and OAK. We encountered multiple problems when developing the logistic regression model, mostly around complex data transformations necessary to ensure meaningful inputs and computational complexity encountered during training. Despite multiple attempts to improve the model via variation in input variables and sample sizes, the model did not converge to a global solution. We only got a Mean absolute error of close to 19% which is very large. Thus, it cannot be used for departure delay prediction with a given dataset.

As data is considered for only 5 airlines carrier’s accuracy is low for Naïve Bayes, if all the carriers are considered then the prediction accuracy will be high for both Naïve Bayes and KNN. As categorical data is considered Naïve Bayes fits best to the data rather than KNN and logistic.

The key takeaways and learnings from this project include:

(1) The importance of conducting thorough explanatory data analysis, asking the right question from the dataset, and foreseeing data limitations,

(2) Data structure and computational complexity have to be taken into consideration when selecting predictor variables

(3) The process of modeling is an iterative process requiring multiple adjustments such as ensuring that training data is well balanced to produce non-biased outcomes.

## 6.2. Limitations

Our successful Logistic regression model could be further improved by augmenting the flights dataset with additional data sources such weather data since weather patterns are likely important factors influencing departure delays. In addition, other classification models like Decision Trees and SVM can be explored for this dataset. More complex models that can incorporate delay propagation (e.g., Markov chain) can also be considered.

## 6.3. Potential Improvements or Future Work

We can improve the efficiency of the Project by following methods: -

* Allow integration of weather forecast data into the models only if user is few days away from departure.
* Try creating separate models for each flight path to see if that improves the accuracy of our prediction
* The Holidays variable will need to be updated for each year used, as some holidays are not on the same date every year. In the program's current form, it only looks at holidays for the date range we originally downloaded the data from.