# STAT652 PROJECT - Predicting the flight Departure Delay

## 1.INTRODUCTION

Flight Delays are major inconvenience to the passengers. It is a serious and widespread problem in United States. Flight delays directly hit the country's economy as they limit the ability of the air transport system to serve the needs of country's economy. Statisticians over the years have deduced that increased delays directly correlate with the increased costs. In this project, my primary focus is on predicting the flight delays that depart from three major airports in New York City in year 2013. The project involves analyzing the data, selecting the essential data and various methodologies implemented to predict the flight delay. Explanation regarding the dataset, the criterion of selecting the data and prediction models implemented, has been done section wise. In the later sections results are explained. The project report also contains an appendix which includes '.rmd' code used for prediction in the project.

## 2. DATA

The original Data is maintained by Bureau of Transportation Statistics, from where the data has been sourced and grouped in four datasets in nycflights13 package. Data were combined from four datasets from this package.

- Flights
- Weather
- Airports
- Planes

The center of interest in this project will be predicting the flight delays and analyze the various features that affect the flight delay.

### 2.1 Data Wrangling

Observing the data in the train dataset 'fltrain', it was observed that many of the columns had NA (Not Available) values. Handling of missing data is one of the important and challenging things. It can be dealt in two ways like removing observations for any missing data or by imputing missing values. However on the basis of omission or insertion of data in missing place, our predictions/ inference could be biased. For the sake of simplicity, I have removed the observations with missing data. As discussed during the class hours, to ensure that many of the data points are not omitted, we choose the criterion of retaining the columns that have no more than 5% missing dataset, which are 10000 for the dataset provided. On the basis of that all the data from 'planes' dataset as well as the columns - 'wind_gust' & 'pressure' were discarded. In addition to that, the rows having NA values were also discarded. The dimension of the dataset after removing the NA values was 184316 rows and 33 columns.

Further the columns 'year.x', 'month' and 'day' were combined to date object under column 'dep_date'. Since we are predicting departure delay, we can further drop the columns of 'dep_time', 'arr_time' and 'arr_delay' as we are least concerned with these values. They might be associated with the departure delay ('dep_delay') but if so, the casual effect is likely in reverse. Other columns like 'sched_arr_time', \`tailnum\`, \`flight\`, \`name\`, \`air_time\`, \`hour\`, \`minute\`, \`time_hour\`, \`tz\`, \`dst\` are also discarded since we are using other variables in the dataset that represent these variables. For ex. 'air_time' is highly correlated with the 'distance' that we have kept in the dataset. Similarily 'tz', 'dst', are related to the (tzone) time zone of destination, so no need of including them for the sake of simplicity. I have considered 'tzone' instead of 'dest'(destination) since for some models like the Decision Trees and random Forest Model, won't run on large number of categorical variables, ('dest' has 104 categorical variables) also 'tzone' (time zone of destination) is related to 'dest'. Since I am already considering the 'distance' variable and as the primary focus is on predicting the departure delay, I have omitted the 'dest' column. After all the Data Wrangling process, the final dimension of our dataset is 184316 rows and 17 columns. Of these 17 columns, ‘dep_delay’ is our target variable and rest is the features.

Correlation matrix of all the variables after data cleaning and arranging:

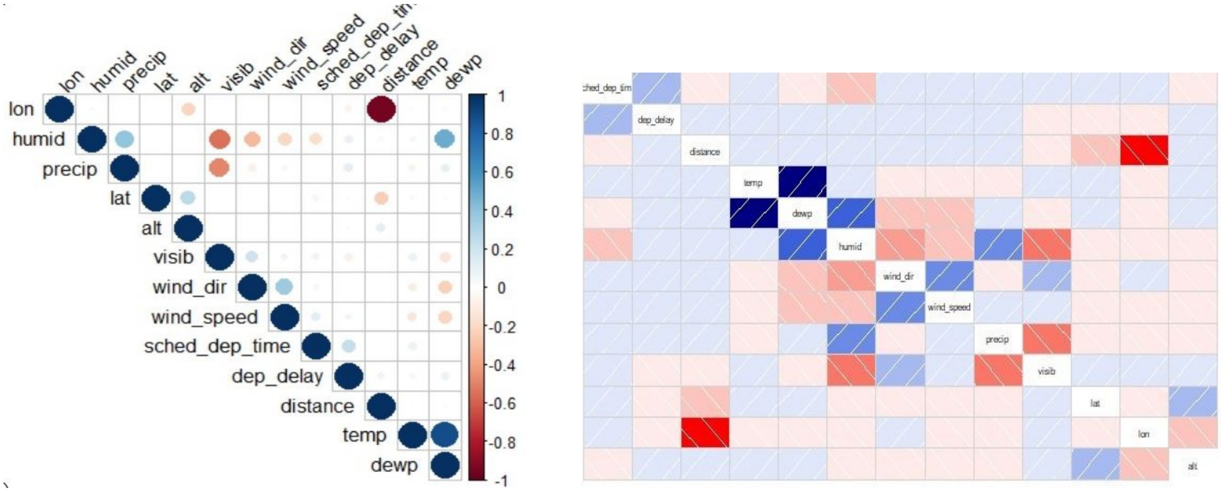

*Fig.1. Correlation Matrix of the variables in the final train dataset.*

Other plots showing the relationship between response variable (dep_delay) and predicted variable

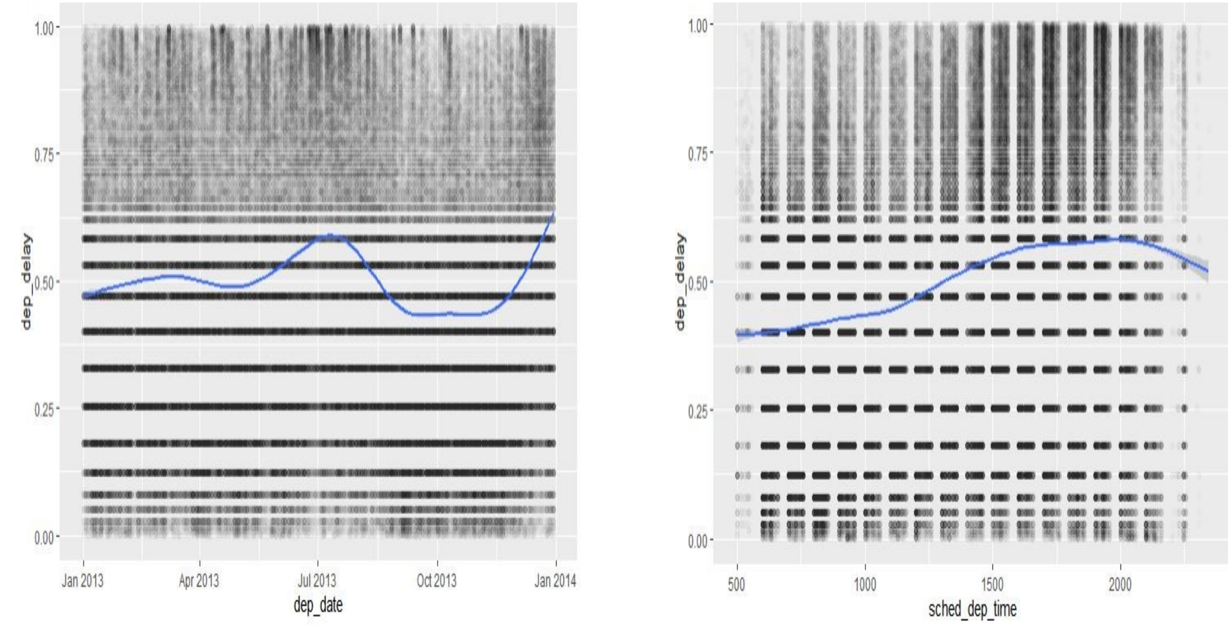

*Fig.2. Relation between departure delay vs Scheduled departure time and log of Distance*

From the above plots and plots in the following page, we see that the relationship between the response variable and the predictors is nonlinear in nature which gives us an intuition to include non-linear terms in the quantitative predictors in our models.

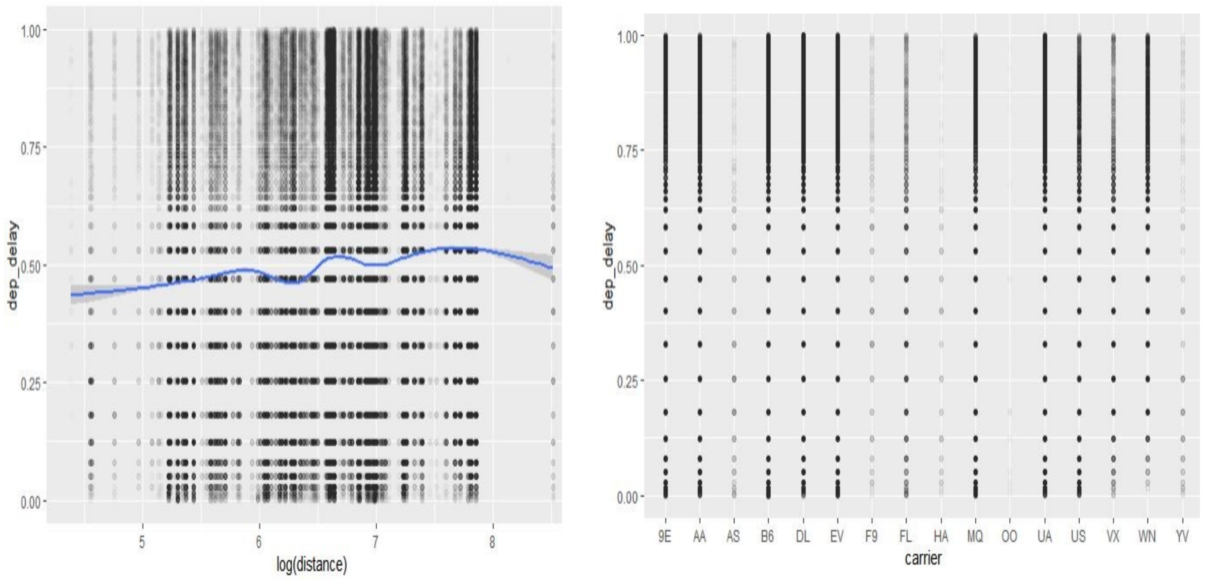

*Fig.3. Plot showing relationship between departure delay and log of distance. Plot of Carriers corresponding to the departure delay.*

# 3. METHODS

### 3.1 Splitting of data into Training & Validation dataset

As we have a large dataset comprising 184,316 rows, we split two-third of the data into training set and one-third into Validation respectively and then used various statistical models to predict the departure delay ('dep_delay') and measured the accuracy of the model by calculating the mean squared error (MSE) on validation set. The best model was then chosen to run on test dataset and the MSE was observed. A low error on the test dataset indicates that the selected trained model has predicted well.

### 3.2 Statistical Modeling

Statistical Model can be explained as the mathematical representation of the observed data. This entails predicting the distribution and testing for statistical validation of the said prediction.

In order to predict departure in delay ('dep_delay'), following statistical models have been used.

1. **Generalized Additive Model (GAM)**: GAM is a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and the primary focus is on inference about these smooth functions.
2. **Decision Trees**: It is a type of supervised learning algorithm that can be used in both regression and classification problems. In this project Decision trees were used for prediction purpose. The decision trees are the most fundamental component of the Random Forest. Decision tree output is easy to understand and interpret.
3. **Random Forest**: It is a supervised learning algorithm that builds multiple decision trees and merges them together to get a more accurate and stable prediction. For this project limited subset was used of the train dataset since I was unable to run the Random Forest method on the complete train set("fl_tr").

4. **Xgboost** (method considered but not focus of project): It is an optimized distributed gradient boosting library which is highly efficient, flexible and portable. It implements gradient boosting decision tree algorithm.
5. **Gradient Boosting(GBM)**: It is a supervised learning technique used for both regression and classification problems. It is one of the most powerful techniques for building the predictive models.

## 4. Results

| MODEL | VALIDATION ERROR | TEST ERROR |
|---|---|---|
| GAM | 0.07066494 | |
| BOOSTING | 0.06383549 | 0.06353717 |
| DECISION TREES | 0.07628095 | 0.07586896 |
| RANDOM FOREST*(limited dataset)* | 0.06731114 | 0.06707085 |
| XGBOOST | 0.06584833 | 0.06541933 |

### BOOSTING (gbm)

Here the package **gbm** is used and within it the gbm() function is used to fit the boosted regression trees to the train dataset. gbm() is being used with the option distribution=”**gaussian**” since the problem dealt with is regression. **n.trees** is the argument specifying the total number of trees to fit. **shrinkage** is the parameter that is applied to each tree in the expansion. By default it is 0.1. I have also used **interaction.depth** parameter which limits the depth of each tree.

Initially I ran the gbm model with n.trees=1000 and shrinkage = 0.01, the mean squared error (mse) that I got on the validation test set was 0.072920. Then I tried bit of hyperparameter tuning on the gbm model. I used n.trees=4000 for training the model on train dataset and used n.trees=2000 on the validation data. I got the mse as 0.0707484. Finally I tried with adding parameter-> interaction.depth = 3 and shrinkage used was 0.3. For this particular parameters I trained the model on n.trees=2000 on the train set and n.trees= 2000 on validation and test data set. MSE on test data obtained was **0.06353717**.

Feature Importance :

```
                           var    rel.inf

sched_dep_time sched_dep_time 21.6004582
dep_date             dep_date 19.3249105
carrier               carrier 17.0260676
humid                   humid 11.1298546
dewp                     dewp  6.6016445
logdistance       logdistance  4.0649670
temp                     temp  3.9770804
wind_speed         wind_speed  2.8929207
wind_dir             wind_dir  2.8153492
origin                 origin  2.7746767
lat                       lat  1.9565159
lon                       lon  1.6201913
logalt                 logalt  1.4433430
precip                 precip  1.3172472
visib                   visib  1.0224936
tzone                   tzone  0.4322795
```

The important features for predicting the departure delay through boosting was found to be scheduled departure, departure date, carrier , humidity and dew.

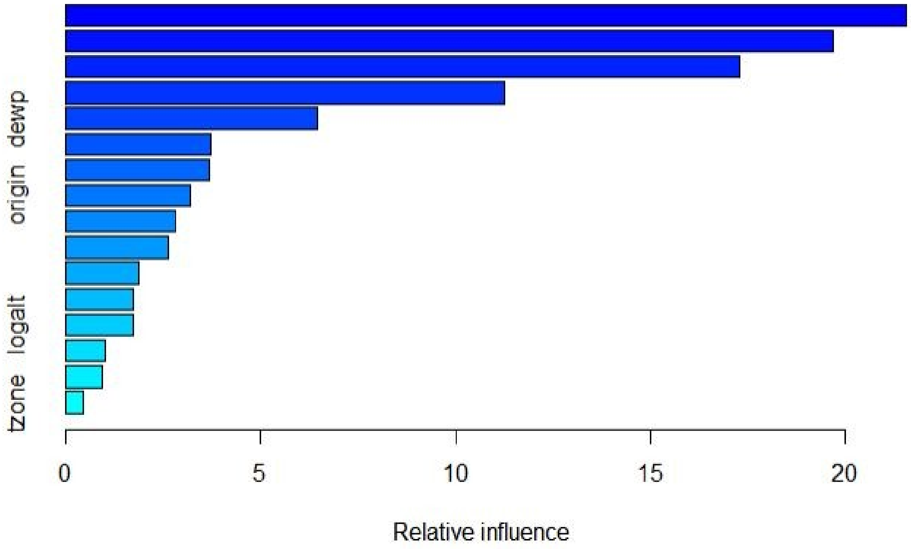

*Fig.4. Relative influence of features*

# 5. **Conclusion and Discussion**

## 5.1 **Summary of Result**

I started with gaining insight on the data exploration and distribution of data. Various Statistical models were considered for predicting the departure delay (“dep_delay”) of flights. Instead of choosing the classification method for predicting whether the flight was delayed or not, I chose to use the regression method as I actually was very much interested in modeling relationship between one outcome variable and several input variables. Among various models used Boosting method gave the least mean squared error. This indicates that Xgboost method with parameter tuning would also probably give better results. Highest mean squared error on the validation and test data set were observed for Decision regression trees. randomForest statistical method was run on only 35,000 rows of the train dataset because on the full training data set the model was not running. It was taking too much time for the execution.

## 5.2 **Future Areas of Work**

randomForest method could not be run on the full dataset. However with required computation power and proper tuning of parameters I would have liked to predict the departure delay. So it stays as one of the key focus area for the future work.