IE7275: Data Mining in Engineering (Spring 2018)
Case Study Project Report

Predicting on-time performance of commercial airlines in US

Presented by:
(Group - 3)

Prithvi Yalamarti
Saurabh Deshpande

Under the guidance of:
Prof. Xuemin Jin

Table of Contents

Executive Summary

The goal of this project is to predict the arrival delay of 10 domestic passenger airlines in the USA. The data is collected from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics. The training data is the on time performance of 2016 Quarter 1 and the test data is the 2017 January month. The data is processed by normalizing followed by Principal Component Analysis (PCA). The various data mining classification techniques used in our project include Logistic Regression, Classification and Regression Trees (CART) and Linear Discriminant Analysis (LDA). The accuracy for logistic regression and linear discriminant analysis was found to be 1 and Accuracy for CART was found to be 0.9705599.

I. Background and Introduction

Nobody likes being stuck at an airport for hours, waiting on a delayed or canceled flight. While occasional delay is inevitable, on time performance of airlines give an insight into one of the most crucial factors over which overall reputation of an airline is based on - on time performance. Federal regulations require that every major air carrier publish its on-time percentages, considering any flight that doesn't reach the gate within 15 minutes of the projected arrival time being listed as "late."

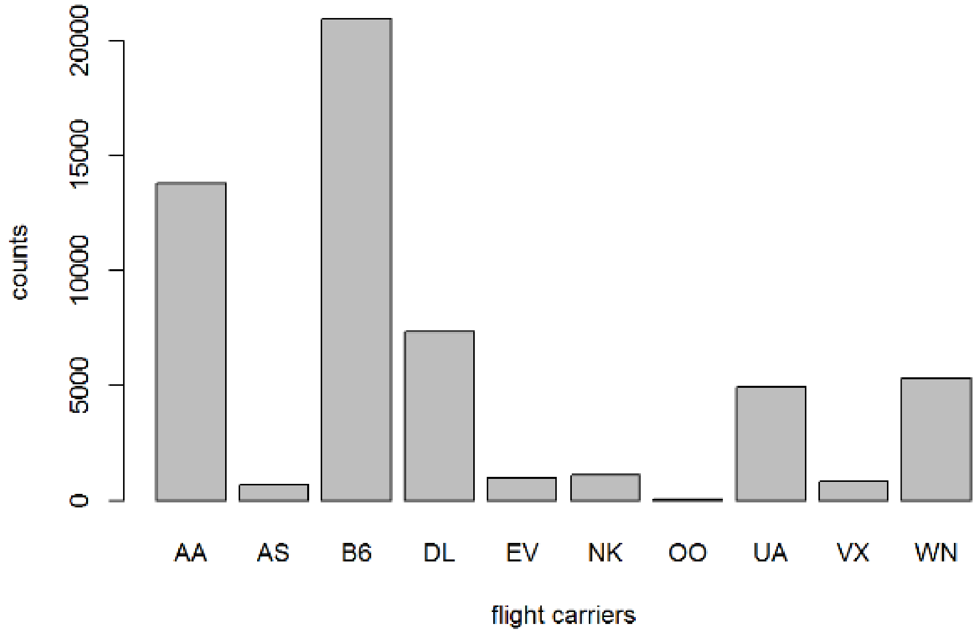
The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers in United States of America. This information can be accessed by customers while booking their trips. In this project, we intend to analyze the operating characteristics of US airlines viz. Departure and arrival times, carrier, departure and arrival cities, whether the flight was delayed on arrival or departure, type of delay (if flight was delayed). Data for top 10 airlines in USA, based on total revenues generated annually is analyzed. This study has been limited to flights flying from and into Boston in the first quarter of 2016.

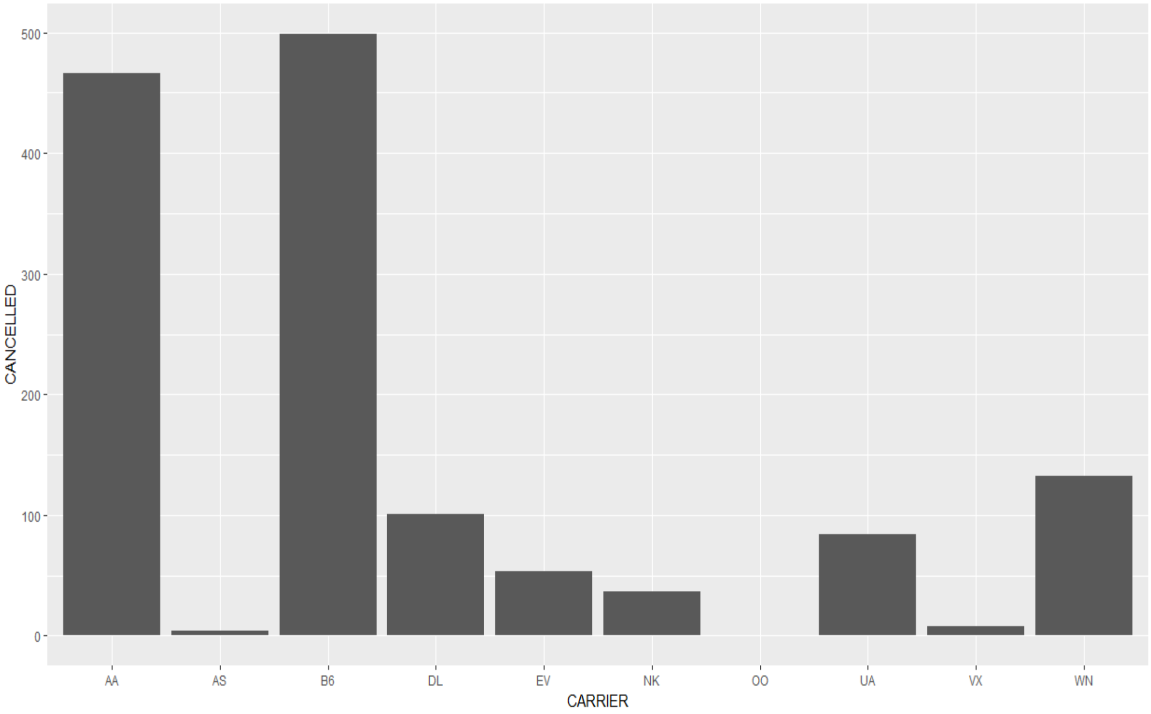
While the factors above do tell a story about historical performance of airlines, we intend on using the data for predicting the on time performance of future flights based on similar parameters using data mining and supervised machine learning techniques.

II. Data Exploration and Visualization

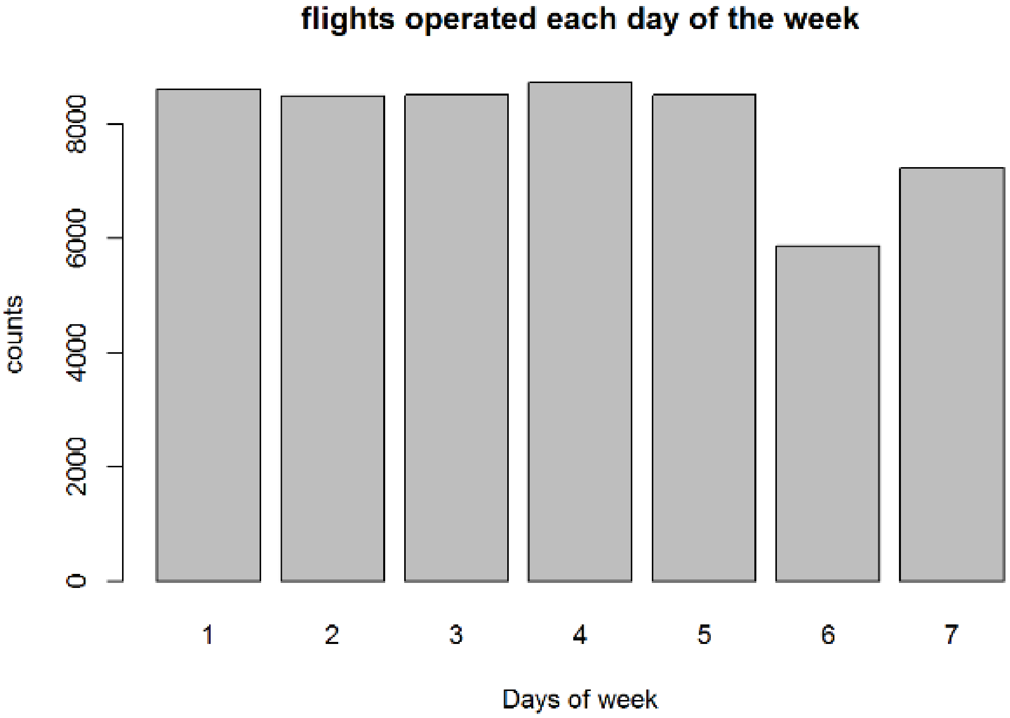
To be able to predict the performance of future flights, its important to understand the past performance of airlines.

Data for top 10 airlines in USA, based on total revenues generated annually is analyzed. Given below is a bar plot for comparing total number of flights flying from and into Boston in Q1 2016.





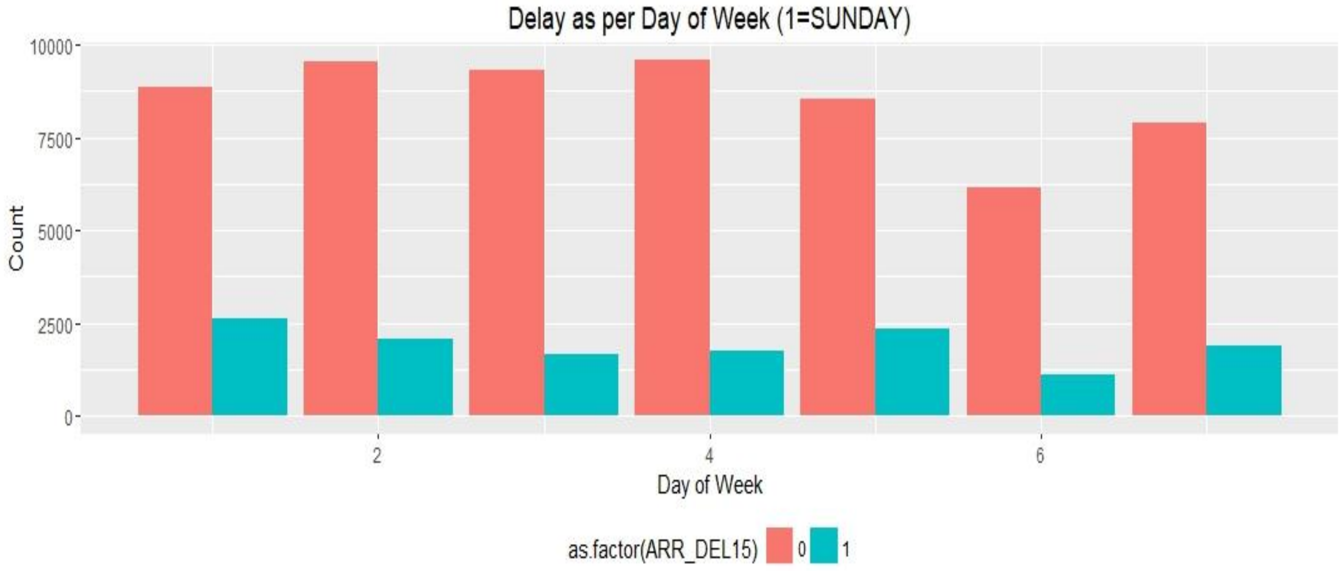
Our initial observations state that Jetblue has the highest number of flights. Also since it operates significantly more number of flights it had more cancellations too. Logically, it is more likely that total number of delays for Jetblue flights would be higher too.



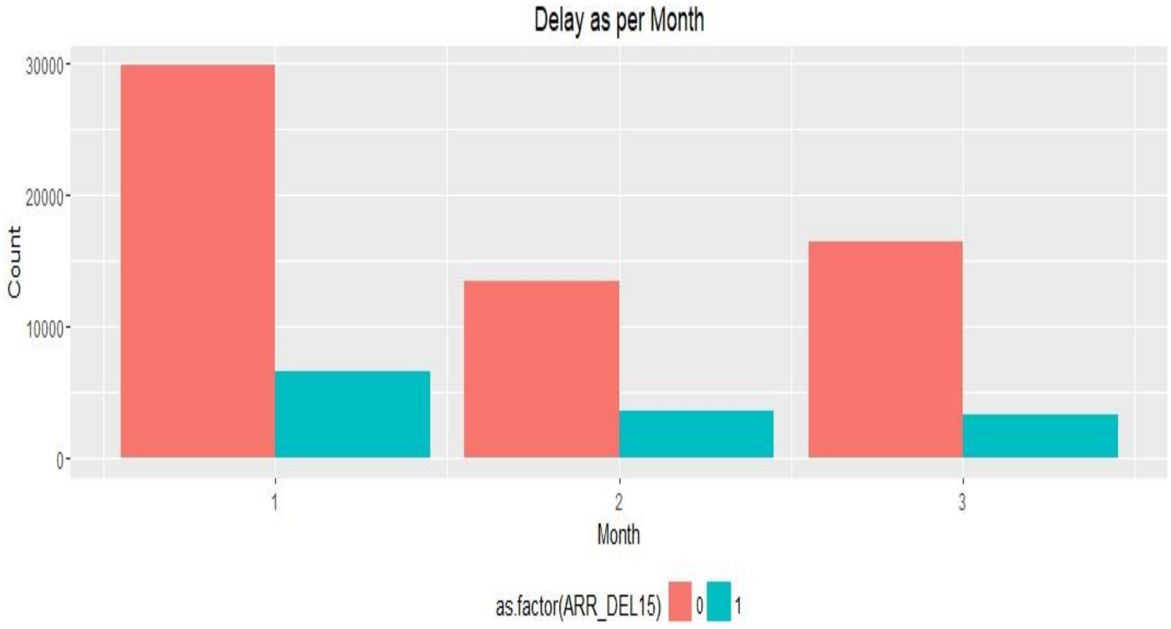
The above graph shows total number of flights operated on each day. While there is no significant variation in the total number of flights operated on each day of week apart on day 6, the total number of flights operating on day 6 - Friday is down by 26% when compared to the average number of flights flying over the entire week.

The number of on time flights is 44661 which is approximately 82% of the total flights flying and the flights delayed is around 18%

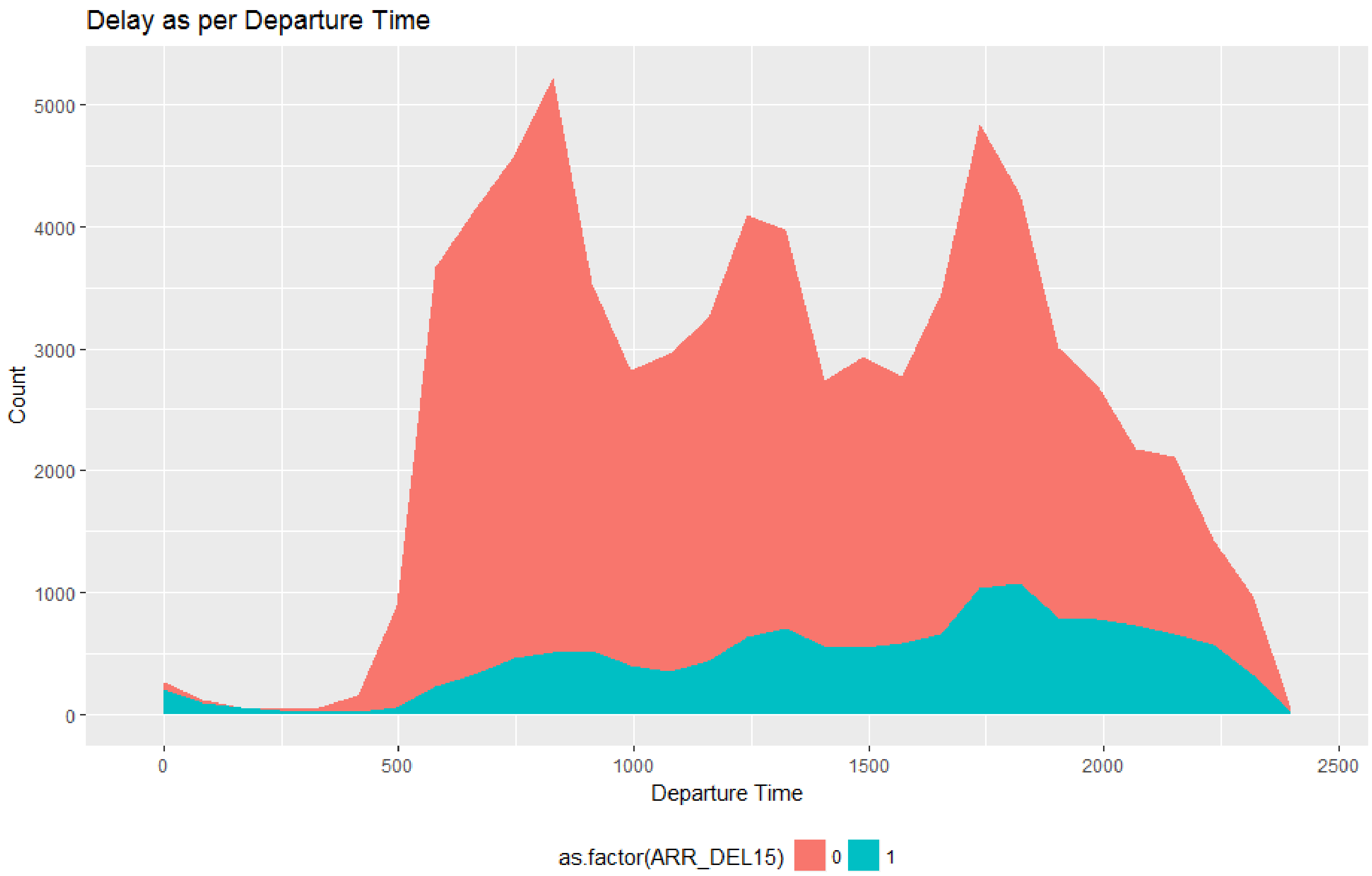
	frequency	percentage
0	44661	81.95
1	9835	18.05



Delays when compared with respect to days of the week, are more on Sundays. Least number of delays happen on Friday. As seen earlier, there is a trend of flight delays taking place when more number of flights are scheduled to fly.



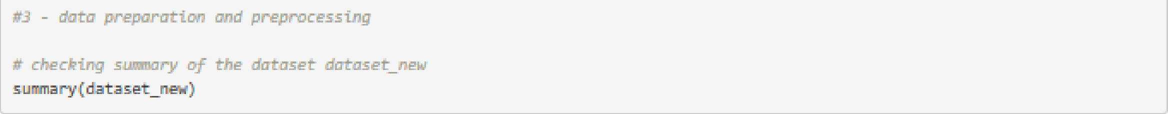
As per above graphs, maximum flights are delayed in the month of January. One of the reasons for those delays would be the fact that Boston has snow storms in the month of January and February. More such incidences, more is the likelihood of the flight getting delayed or even getting cancelled.



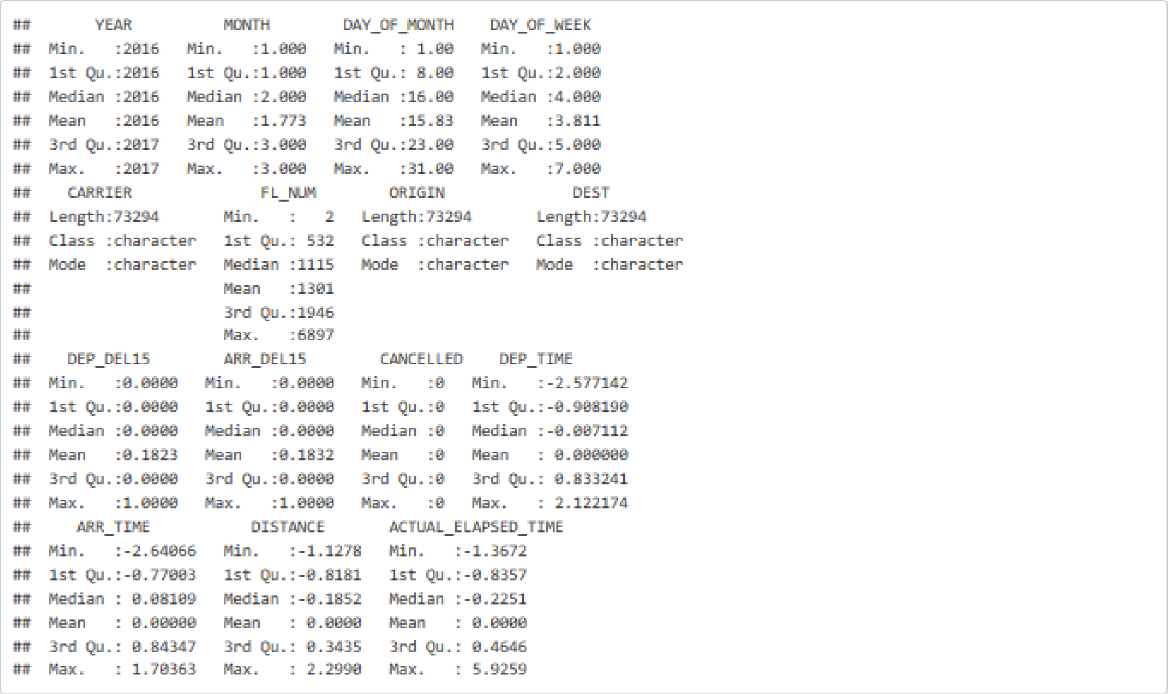
At a day to day level of aggregation, the number of flights are more in the morning at around 7 AM and around at 5.30 PM. Also what is seen is, the number of flights delayed is more in the evening, when compared to morning.

III. Data Preparation and Preprocessing

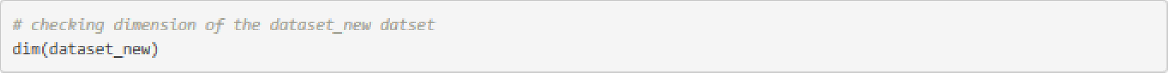
In the data preparation and preprocessing section, we performed tasks including data summary, dimension reduction, correlation analysis, PCA analysis. We started off by finding the summary of the normalized training dataset. Next, we check the dimension of the the normalised dataset. We see that there are 73294 rows and 15 columns.



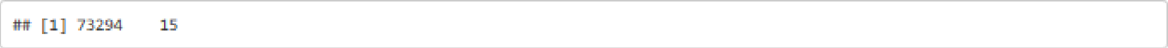
```
#3 - data preparation and preprocessing

# checking summary of the dataset dataset_new
summary(dataset_new)
```



```
##       YEAR          MONTH        DAY_OF_MONTH    DAY_OF_WEEK
##  Min.   :2016   Min.   :1.000   Min.   : 1.00   Min.   :1.000
##  1st Qu.:2016   1st Qu.:1.000   1st Qu.: 8.00   1st Qu.:2.000
##  Median :2016   Median :2.000   Median :16.00   Median :4.000
##  Mean   :2016   Mean   :1.773   Mean   :15.83   Mean   :3.811
##  3rd Qu.:2017   3rd Qu.:3.000   3rd Qu.:23.00   3rd Qu.:5.000
##  Max.   :2017   Max.   :3.000   Max.   :31.00   Max.   :7.000
##    CARRIER              FL_NUM        ORIGIN              DEST
##  Length:73294       Min.   :   2   Length:73294       Length:73294
##  Class :character   1st Qu.: 532   Class :character   Class :character
##  Mode  :character   Median :1115   Mode  :character   Mode  :character
##                     Mean   :1301
##                     3rd Qu.:1946
##                     Max.   :6897
##    DEP_DEL15        ARR_DEL15        CANCELLED    DEP_TIME
##  Min.   :0.0000   Min.   :0.0000   Min.   :0   Min.   :-2.577142
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0   1st Qu.:-0.908190
##  Median :0.0000   Median :0.0000   Median :0   Median :-0.007112
##  Mean   :0.1823   Mean   :0.1832   Mean   :0   Mean   : 0.000000
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0   3rd Qu.: 0.833241
##  Max.   :1.0000   Max.   :1.0000   Max.   :0   Max.   : 2.122174
##     ARR_TIME           DISTANCE       ACTUAL_ELAPSED_TIME
##  Min.   :-2.64066   Min.   :-1.1278   Min.   :-1.3672
##  1st Qu.:-0.77003   1st Qu.:-0.8181   1st Qu.:-0.8357
##  Median : 0.08109   Median :-0.1852   Median :-0.2251
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84347   3rd Qu.: 0.3435   3rd Qu.: 0.4646
##  Max.   : 1.70363   Max.   : 2.2990   Max.   : 5.9259
```

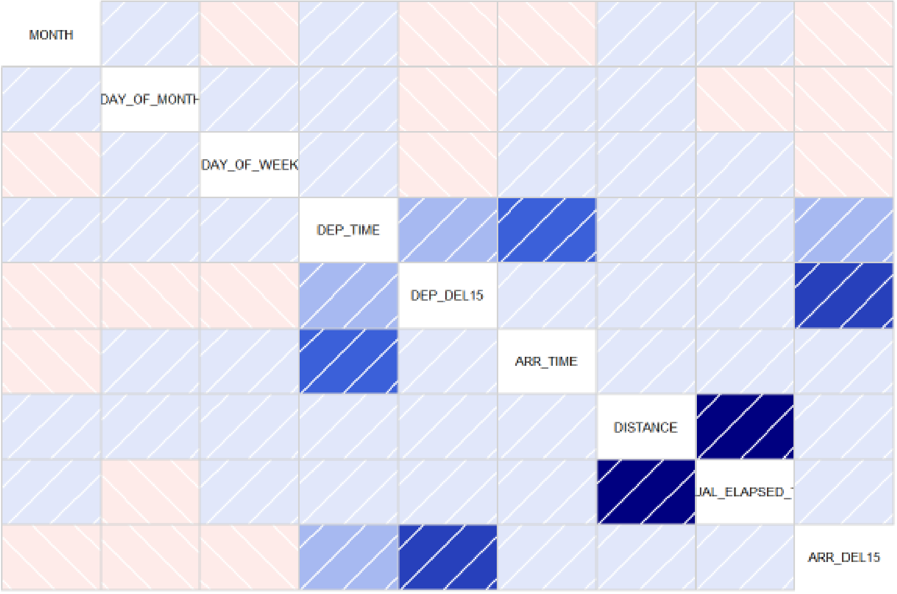


```
# checking dimension of the dataset_new datset
dim(dataset_new)
```

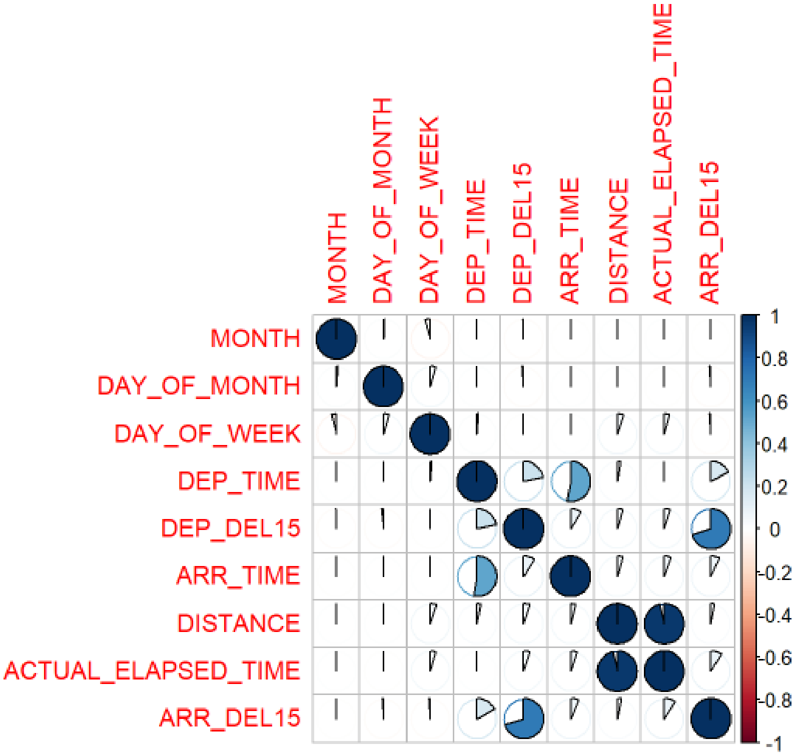


```
## [1] 73294    15
```

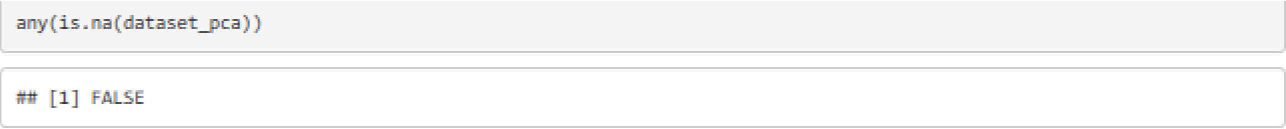
Next, using the corrgram function in corrgram library, we plot correlation matrix. We see that the attributes distance and actual elapsed time have a strong correlation.



To represent the corrplot in a better way, we used corrplot function from the corrplot package and used the type full and method pie. The output conveys the same message that the variables Actual Elapsed Time and Distance are strongly correlated.



Now we do the Principal Component Analysis. We start of by creating dummy for binary features in the dataset. A new dataset is formed. When we check for Null values in this Dataset, the output is False.

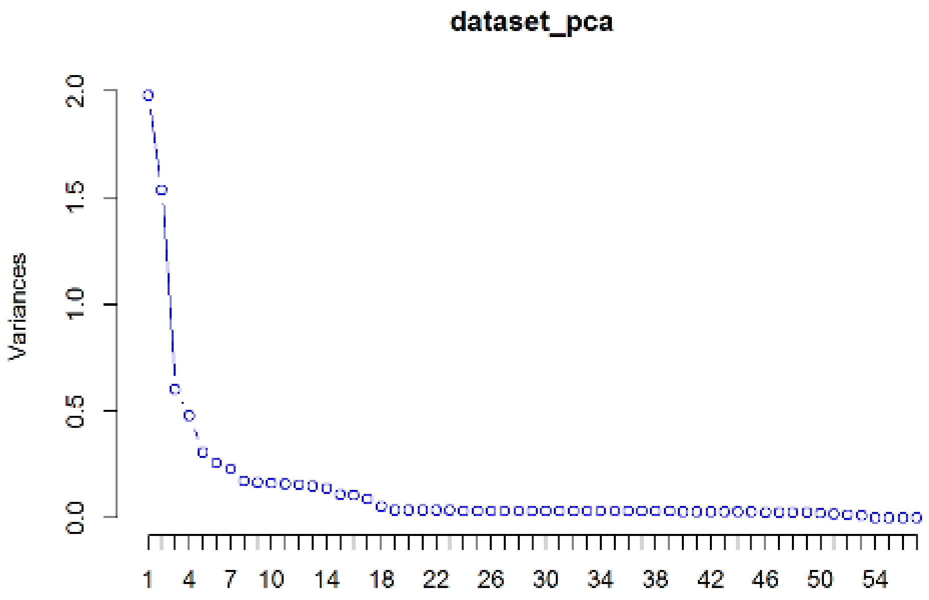


```
any(is.na(dataset_pca))
```

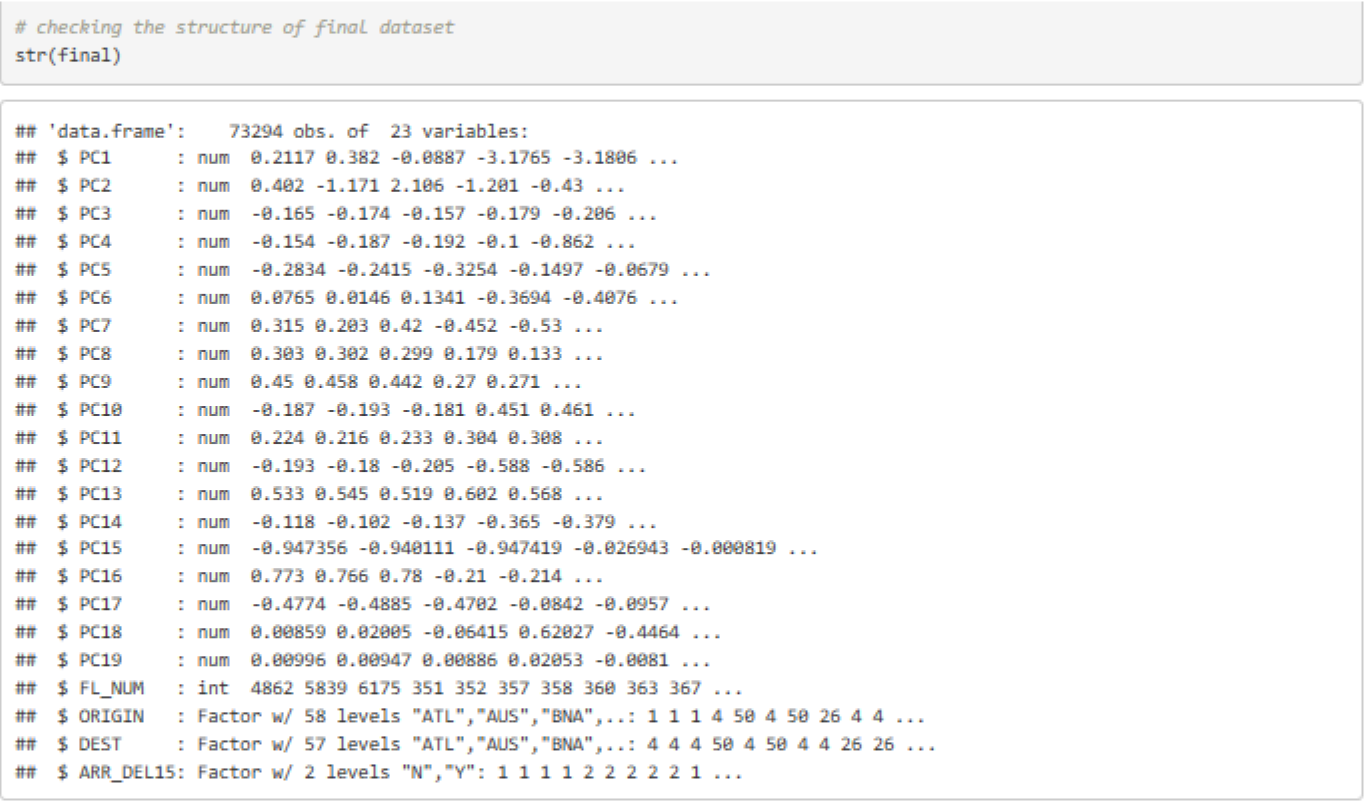
```
## [1] FALSE
```

The next step is check the summary of the PCA dataset. The output is as follows.

We plot the PCA using the screenplot() function. This is done as it helps us to visualise and select the feature that covers almost 95% of the variability of the data. The plot is as follows.



After plotting the chart, we create factors for the target features and check the structure of the final dataset.



```
# checking the structure of final dataset
str(final)
```

```
## 'data.frame':    73294 obs. of  23 variables:
##  $ PC1      : num  0.2117 0.382 -0.0887 -3.1765 -3.1806 ...
##  $ PC2      : num  0.402 -1.171 2.106 -1.201 -0.43 ...
##  $ PC3      : num  -0.165 -0.174 -0.157 -0.179 -0.206 ...
##  $ PC4      : num  -0.154 -0.187 -0.192 -0.1 -0.862 ...
##  $ PC5      : num  -0.2834 -0.2415 -0.3254 -0.1497 -0.0679 ...
##  $ PC6      : num  0.0765 0.0146 0.1341 -0.3694 -0.4076 ...
##  $ PC7      : num  0.315 0.203 0.42 -0.452 -0.53 ...
##  $ PC8      : num  0.303 0.302 0.299 0.179 0.133 ...
##  $ PC9      : num  0.45 0.458 0.442 0.27 0.271 ...
##  $ PC10     : num  -0.187 -0.193 -0.181 0.451 0.461 ...
##  $ PC11     : num  0.224 0.216 0.233 0.304 0.308 ...
##  $ PC12     : num  -0.193 -0.18 -0.205 -0.588 -0.586 ...
##  $ PC13     : num  0.533 0.545 0.519 0.602 0.568 ...
##  $ PC14     : num  -0.118 -0.102 -0.137 -0.365 -0.379 ...
##  $ PC15     : num  -0.947356 -0.940111 -0.947419 -0.026943 -0.000819 ...
##  $ PC16     : num  0.773 0.766 0.78 -0.21 -0.214 ...
##  $ PC17     : num  -0.4774 -0.4885 -0.4702 -0.0842 -0.0957 ...
##  $ PC18     : num  0.00859 0.02005 -0.06415 0.62027 -0.4464 ...
##  $ PC19     : num  0.00996 0.00947 0.00886 0.02053 -0.0081 ...
##  $ FL_NUM   : int  4862 5839 6175 351 352 357 358 360 363 367 ...
##  $ ORIGIN   : Factor w/ 58 levels "ATL","AUS","BNA",..: 1 1 1 4 50 4 50 26 4 4 ...
##  $ DEST     : Factor w/ 57 levels "ATL","AUS","BNA",..: 4 4 4 50 4 50 4 4 26 26 ...
##  $ ARR_DEL15: Factor w/ 2 levels "N","Y": 1 1 1 1 2 2 2 2 2 1 ...
```

After this we use the Holdout method for train and test dataset. This method is used to randomly split the dataset

IV. Data Mining Techniques and Implementation

After careful analysis of data, we decided to implement three classification methods namely, Logistic Regression, Classification and Regression Trees (CART) and Linear Discriminant Analysis (LDA) for the prediction of Arrival Delay in flights. The predictor variables that we chose are Flight number, Origin, Destination, year, month, day of the month, Departure time, Actual Elapsed time, Distance and Arrival time. The response variable we choose is Arrival Delay. If a flight is 15 or more than 15 minutes late during arrival, it is considered in the Arrival Delay.

The definition of the data mining techniques are as follows:

- Logistic Regression - A method in Machine Learning that measures the relationship between the categorical dependent variable and one or more independent variable by estimating probabilities using a logistic function.
- CART - Classification and Regression Tree helps in recursively partitioning response variables into subsets based on their relationship to one or more (usually many) predictor variables.
- Linear Discriminant Analysis - A method in Machine Learning which is used to find a linear combination of features that characterizes or separates two or more classes of objects or events.

In our project, since we have performed the Principal Component Analysis, we use the principal components for the model generation.

V. Performance Evaluation

Three types of modelling done in this study are: 1) Logistic Regression 2) Classification and Regression Trees (CART) 3) Linear Discriminant Analysis (LDA)

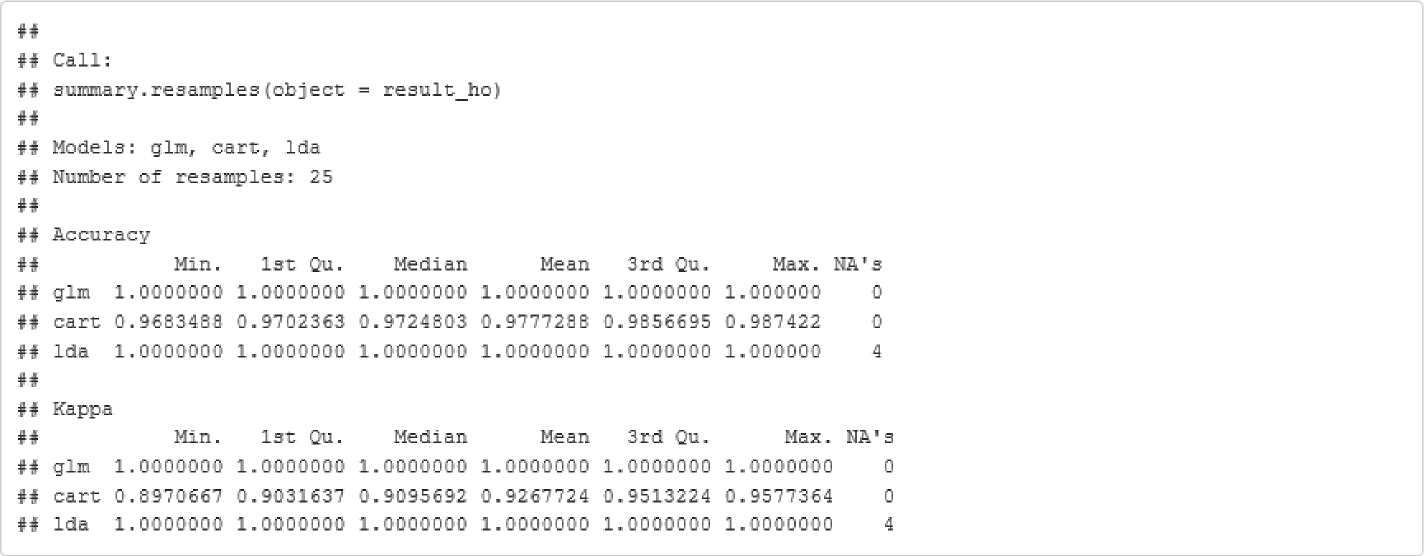
1) Logistic Regression: There are 54612 samples, 21 predictor variables and 2 classes: 'N', 'Y'
   Resampling results for Logistic Regression is accuracy of 1. The Kappa index of agreement (KIA) will tell you how much better, or worse, classifier is than what would be expected by random chance. If we were to randomly assign cases to classes (i.e. a kind of terribly uninformed classifier), we'd get some correct simply by chance. Therefore, we will always find that the Kappa value is lower than the overall accuracy. The Kappa index is however considered to be a more conservative measure than the overall classification accuracy. We got an accuracy of 1 and Kappa index value of 1 which clearly states the model has been well implemented.
2) Classification and Regression Tree (CART): CART was implemented and accuracy was used to select the optimal model using the largest value. Final value used for the model was cp=0.08697813 with highest accuracy of 0.9777288
   Given below are the values of accuracy, cp and kappa values for the CART technique.

cp	Accuracy	Kappa
0.08697813	0.9777288	0.9267724
0.12246521	0.9610560	0.8698644
0.71580517	0.8738842	0.3601431

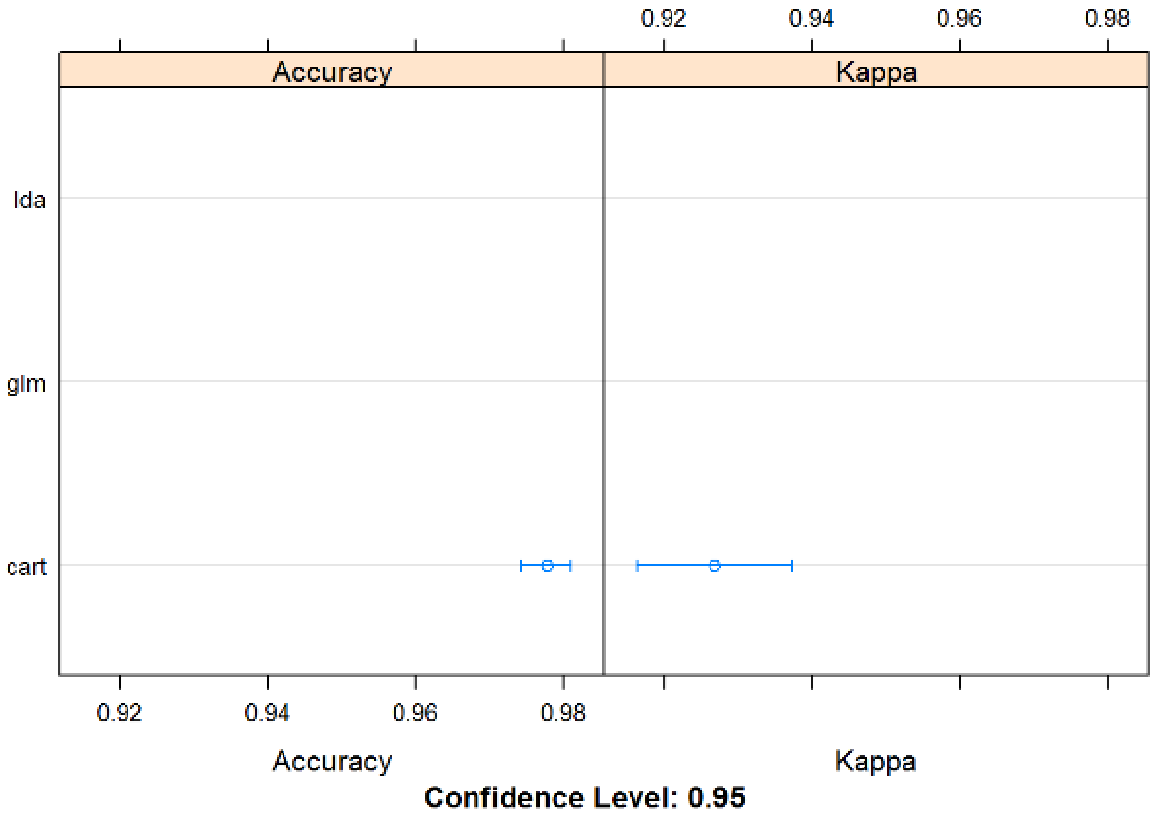
3) Linear Discriminant Analysis (LDA): Values for Linear Discriminant Analysis were similar to logistic regression. Accuracy and kappa value of 1.

The model was then evaluated using the values from test data frame. The accuracy for logistic regression and linear discriminant analysis was found to be 1. Accuracy for CART was found to be 0.9705599.

Performance of three techniques is as follows:



```
##
## Call:
## summary.resamples(object = result_ho)
##
## Models: glm, cart, lda
## Number of resamples: 25
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.     Max. NA's
## glm  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.000000    0
## cart 0.9683488 0.9702363 0.9724803 0.9777288 0.9856695 0.987422    0
## lda  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.000000    4
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glm  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000    0
## cart 0.8970667 0.9031637 0.9095692 0.9267724 0.9513224 0.9577364    0
## lda  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000    4
```



VI. Discussion and Recommendation

In the project explained above, main aim was to predict on time performance of flights. As mentioned above, top 10 airline in US with highest revenues were considered. The motivation behind targeting Q1-2016,2017 was to help customers booking flight tickets with critical flight delay information in the winter season. Since winter has the most number of flight cancellations and delays as compared with other seasons throughout the year, we felt it was something with which we could definitely contribute to. The logistic regression, LDA and CART were the three methods used in data mining. While KNN and SVM were tried by us in earlier version of our data mining techniques, we removed it from the later versions. In terms of KNN, it was tried earlier by many others in similar projects. The only but major disadvantage with SVM is that it takes time for model to be implemented. Its speed and size are two most important disadvantages in training and also in validation set.

While this project handles data mining techniques which deal with binary classification (logistic regression), LDA helps find the 'boundaries' around clusters of classes. It projects the data points on a line so that the clusters 'are as separated as possible', with each cluster having a relative (close) distance to a centroid.
CART helps in recursively partitioning response variables into subsets based on their relationship to one or more (usually many) predictor variables.

Going further, beyond binary classification a project can focus on determining by how much exactly the flight would be delayed. That would require further investigation into the type of data mining techniques to be used. There remains a lot to be done.

VII. Summary

The goal of this project is to predict the arrival delay of 10 domestic passenger airlines in the USA. The data is collected from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics. The training data is the on time performance of 2016 Quarter 1 and the test data is the 2017 January month. The data is processed by normalizing followed by Principal Component Analysis (PCA). The various data mining classification techniques used in our project include Logistic Regression, Classification and Regression Trees (CART) and Linear Discriminant Analysis (LDA). The accuracy for logistic regression and linear discriminant analysis was found to be 1 and accuracy for CART was found to be 0.9705599.

Appendix: R Code for use case study

```
#1 - data loading
# loading dataset in to R studio using read.csv()
# stringsAsFactors = F restricts R from making factors for character features
# na.strings to replace blank with NA values
train <- read.csv("Training Data.csv", header = T, stringsAsFactors = F, na.strings = c("","NA"))
test <- read.csv("Test Data.csv", header = T, stringsAsFactors = F, na.strings = c("","NA"))
dataset <- rbind(train,test)
# checking structure of dataset
str(dataset)

#2 - data exploration and visualization
any(is.na(dataset)) # checking for NA values in dataset
colnames(dataset)[colSums(is.na(dataset))>0] # columns whihc have NA values
colnames(dataset)[colSums(is.na(dataset))>0.5*nrow(dataset)] # columns which have NA values more than 50%
dataset <- dataset[,c(-14,-17,-18,-19,-20)] # sub setting dataset and removing columns which have NA values more than 50%
# checking frequency for the flight delay
frequency <- table(dataset$ARR_DEL15)
# checking perecentage of flight delay
percent <- round(prop.table(table(dataset$ARR_DEL15))*100,2)
# creating a tavble of frequency and percentage of flight delay
cbind(frequency = frequency, percentage = percent)
library(sqldf) # calling library sqldf
# creating dataset_new dataset by eliminating all the records where the flight has been cancelled
index <- sqldf('select * from dataset where CANCELLED = 0 AND YEAR = 2016')
dataset_new <- sqldf('select * from dataset where CANCELLED = 0')
# checking for NA values
colnames(dataset_new)[colSums(is.na(dataset_new))>0]
# imputing the missing values using median and mean
dataset_new$ARR_TIME[is.na(dataset_new$ARR_TIME)] <- mean((dataset_new$ARR_TIME), na.rm = T)
dataset_new$ARR_DEL15[is.na(dataset_new$ARR_DEL15)] <- median((dataset_new$ARR_DEL15), na.rm = T)
dataset_new$ACTUAL_ELAPSED_TIME[is.na(dataset_new$ACTUAL_ELAPSED_TIME)] <- mean((dataset_new$ACTUAL_ELAPSED_TIME),
na.rm = T)
# again checking for NA values
colnames(dataset_new)[colSums(is.na(dataset_new))>0]
ggplot(dataset_new, aes(x=CARRIER, fill = as.factor(ARR_DEL15))) +
  geom_bar(position = "dodge") +
  labs(title = "Delay as per Carrier", x = "Carriers", y = "Count") +
  theme(legend.position = "bottom")
ggplot(dataset_new, aes(x=DAY_OF_WEEK, fill = as.factor(ARR_DEL15))) +
  geom_bar(position = "dodge") +
  labs(title = "Delay as per Day of Week (1=SUNDAY)", x = "Day of Week", y = "Count") +
  theme(legend.position = "bottom")
ggplot(dataset_new, aes(x=MONTH, fill = as.factor(ARR_DEL15))) +
  geom_bar(position = "dodge") +
  labs(title = "Delay as per Month", x = "Month", y = "Count") +
  theme(legend.position = "bottom")
ggplot(dataset_new, aes(x=DEP_TIME, fill = as.factor(ARR_DEL15))) +
  geom_area(stat = "bin") +
  labs(title = "Delay as per Departure Time", x = "Departure Time", y = "Count") +
  theme(legend.position = "bottom")
# subsetting the numeric features from dataset
numeric_features <- dataset_new[c("MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "DEP_TIME", "DEP_DEL15", "ARR_TIME",
"DISTANCE", "ACTUAL_ELAPSED_TIME", "ARR_DEL15")]
# creating correlation matrix
correlation <- cor(numeric_features, method = "pearson")
correlation
library(caret) # calling library caret
# creating a function normalize to standardize the features
normalize <- function(x){
  (x-mean(x))/sd(x)
}
# applying the normalize function to the dataset using lapply()
norm <- data.frame(lapply(dataset_new[,c("DEP_TIME", "ARR_TIME", "DISTANCE", "ACTUAL_ELAPSED_TIME")], normalize))

# subsetting dataset_new by removing the numerical features
```

```
dataset_new <- dataset_new[,c(-9,-11,-14,-15)]
# revising dataset_new dataset by adding normalized dataset
dataset_new <- cbind(dataset_new,norm)


#3 - data preparation and preprocessing
# checking summary of the dataset dataset_new
summary(dataset_new)
# checking dimension of the dataset_new datset
dim(dataset_new)
library(corrgram) # calling library corrgram
# plotting the corelation matrix using corrgram()
corrgram(correlation)
library(corrplot) # calling library corrplot
# plotting the correlation matrix using corrplot()
corrplot(correlation, type="full", method = "pie")
# creating dummy for creating the binary features in the dataset
dummy <- dataset_new[c("YEAR", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "CARRIER")]
library(ade4) # library ade4 for using acm.disjonctif()
# created the binary variables for the categorical features using acm.disjonctif()
dummy <- acm.disjonctif(dummy)
dataset_new <- dataset_new[ ,-c(1:5)]
# Created the new data set
dataset_pca <- data.frame(c(dummy, dataset_new))
any(is.na(dataset_pca))
dataset_pca <- prcomp(dataset_pca[ ,c(-54,-55,-56,-59)])
# checking the summary of PCA
summary(dataset_pca)
# plotting the PCA using screeplot(). This help to visualize and select the feature that covers almost 95% of the varibaility of the data.
screeplot(dataset_pca, npcs = 57, type = "lines", col=20)
# setting the seed to 1
set.seed(1)
# creating the flight_pca data frame
dataset_pca <- data.frame(dataset_pca$x[ , ])
# Appending the response variable to the dataset
final <- data.frame(c(dataset_pca, dataset_new[c("FL_NUM","ORIGIN","DEST","ARR_DEL15")]))
final <- final[ , c(1:19, 60:63)]
# creating factor for the target features
final$ARR_DEL15 <- as.factor(final$ARR_DEL15)
final$ARR_DEL15 <- factor(final$ARR_DEL15, levels = c('0','1'), labels = c("N","Y"))
# checking the structure of final dataset
str(final)
# train dataset for holdout method
index <- sample(73294, 54612)
holdout_train <- final[index, ]
# test dataset for holdout method
holdout_test <- final[-index, ]
library(glm2)
library(MASS)
# backwards stepwise elimination for feature selection
model <- glm(ARR_DEL15 ~ ., family = binomial, data = holdout_train)
backwards <- step(model, direction = "backward")
#
#Step:  AIC=266
#ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 + PC16 + PC17 + PC18 +
PC19 + FL_NUM + ORIGIN + DEST
# Model generation for holdout CV
# logistic regression
glm_ho <- train(ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 + PC16 +
PC17 + PC18 + PC19 + FL_NUM + ORIGIN + DEST, data = holdout_train, method = "glm", metric = "Accuracy")
glm_ho
# CART
cart_ho <- train(ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 + PC16 +
PC17 + PC18 + PC19 + FL_NUM + ORIGIN + DEST, data = holdout_train, method = "rpart", metric = "Accuracy")
cart_ho
(cart_ho)
library(tree)
```

```
tree.model <- tree(dataset_new$ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 +
PC15 + PC16 + PC17 + PC18 + PC19 + FL_NUM + ORIGIN + DEST)
plot(tree.model)
text(tree.model)
# SVM
svm_ho <- train(ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 + PC16 +
PC17 + PC18 + PC19 + FL_NUM + ORIGIN + DEST, method = "svmLinear", metric = "Accuracy")
svm_ho
# LDA
lda_ho <- train(ARR_DEL15 ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 + PC16 +
PC17 + PC18 + PC19 + FL_NUM + ORIGIN + DEST, data = holdout_train, method = "lda", metric = "Accuracy")
lda_ho
# Evaluating model using test datafrme
# logistic regression
glm_pred <- predict(glm_ho, new = holdout_test)
# checking the accuracy of the logistic regression model
accuracy_glm_ho <- mean(glm_pred == holdout_test$ARR_DEL15)
accuracy_glm_ho
# CART
cart_pred <- predict(cart_ho, new = holdout_test)
# checking the accuracy of the CART model
accuracy_cart_ho <- mean(cart_pred == holdout_test$ARR_DEL15)
accuracy_cart_ho
# SVM
lda_pred <- predict(lda_ho, new = holdout_test)
# checking the accuracy of the SVM model
accuracy_lda_ho <- mean(lda_pred == holdout_test$ARR_DEL15)
accuracy_lda_ho
# Comparing the result of all three models
result_ho <- resamples(list(glm = glm_ho, cart = cart_ho, lda  = lda_ho))
summary(result_ho)
dotplot(result_ho)
```