School of Information Studies

SYRACUSE UNIVERSITY

IST 687 – Introduction to Data Science

M003 | Group 5

CUSTOMER CHURN IN THE AIRLINE INDUSTRY

Recommendations to improve NPS

Submitted by:

Bhavish Kumar | Joseph Sabel | Laxman Kumar | Sihan Yang | Vidushi Mishra

TABLE OF CONTENT

Contents

LIST OF FIGURES

1. INTRODUCTION

South East Airlines is currently suffering from customer churn, owing to poor customer experience and customer dissatisfaction. The customer satisfaction is identified through a likelihood to recommendation score filled in by the customer in their survey. This likelihood to recommend score is used to tag a customer as a Promoter, Detractor or Passive. The Net Promoter Score (NPS) is then measured as the difference between % of Promoters and % Detractors. Detractors are the customers who are most likely to churn and may cause more customers to churn by spreading negative reviews, whereas Promoters on the other hand may even bring in more customers by spreading positive reviews.

Currently, the airline has 14 partner airlines of which 2 of the airlines have a negative NPS which means that, they have more Detractors than Promoters. The goal of the project is to provide actionable insights and recommendations, that will help increase their NPS. Moreover, South East Airlines needs help in deciding which partners to keep, which partners to drop and which regional airlines should become their new partners.

The Dataset received contains 10,282 customer surveys that captures several customer and flight attributes along with the likelihood to recommend score provided by each customer.

2. BUSINESS QUESTIONS

Below are some key questions that have been identified and answered through the project

- How does the Likelihood to Recommend Score spread across different Traveler Class, Gender, Travel Type, Airline Status, Age Groups & Partner Airlines?
- What is the median Recommendation score for each Traveler Class, Gender, Travel Type, Airline Status, Age Groups & Partner Airlines?
- Does Flight Cancellation affect NPS score? i.e. Do cancelled flights have more Detractors than Promoters?
- How many survey observations do we have for each Partner Airline, Traveler Class, Travel Type, Airline Status and Age Groups across the 2 Genders?
- Based on NPS, which are the top performing partner airlines, and which are the poor performing partner airlines?
- Which are the partner airlines that Southeast needs to keep and which ones to drop?
- What are the customer & flight attributes that are most likely to make a customer a Promoter and a Detractor?
- For a new set of customers, how accurately can we predict whether they are going to be Promoters, Detractors or Passive?

3. DATA MUNGING

Data Munging is the initial process of refining raw data into content or formats better suited for consumption by downstream systems and users. It is a process of cleaning messy data into a form that is fit for a dedicated purpose.
Munging begins with data exploration & data transformation
The initial data was made available in the JSON format which was transformed into a tabular format and stored in a data frame. The following code was used to read data from JSON format and store it into a data frame.

```
dataset <- read_json("fall2019-survey-M03.json")
fall <- jsonlite::fromJSON("fall2019-survey-M03.json")
View(fall)   ### fall is the dataframe containing survey data with 10,282 rows and 32
columns
```

Below are the actions taken as part of data cleaning:

- Removed the comma and state code from destination and origin city columns
- Created hash map between state abbreviation and state name, as well as Partner Code and Partner Name and then got rid of the Partner Name, Origin state, destination state, free text & day of month columns from the clean data
- Treated 1 NA in Likelihood to Recommend score by obtaining mean at route and partner level, i.e. obtained the mean likelihood score at the corresponding origin state, destination state and partner airline level and replaced the NA with the mean.
    - Example: If the row containing NA was for Destination City ‘Chicago’, Origin City ‘Austin’ and for partner airline ‘FlyFast Airways’, then the mean of Likelihood to Recommend score was obtained on the subset for ‘Chicago’, ‘Austin’ and FlyFast Airways and then this mean was replaced with the NA.
- Treated 19 NAs for Flight Time in Minutes and Flight Arrival Delay column, for which the flight cancelled column had ‘No’, by obtaining the mean at route and partner level, like as explained above
- Replaced NAs with 0 for Flight Time in Minutes column where Flight Cancelled column had ‘Yes’
- Replaced NAs with NaN (Not a Number) for Flight Arrival Delay and Flight Departure Delay columns where Flight Cancelled column had ‘Yes’
- Renamed all the columns to get rid of the dot that was present in each column header

The following code was used to perform the above-mentioned cleaning process

```
    ####### - CLEANING DATA STARTS HERE - ######
    #Field with NAs are DepartureDelay, ArrivalDelay, FlightTimeinMin
    ## All the NAs in Departure Delay column are due to flight cancellation
    ## There are 19 Nas in Arrival Delay and Flight Time in Minutes even for flights which are not cancelled and those must be treated.

  #Hash mapping of partner code with the partner names
  partnerCodeToName <- hash(unique(fall$Partner.Code),unique(fall$Partner.Name))     ## Creating a hash map between partner code and partner name

  b <- data.frame(t(data.frame(strsplit(fall$Destination.City,","))))           ## Removing the comma and state code from destination and origin city
  rownames(b) <- NULL
  a <- data.frame(t(data.frame(strsplit(fall$Origin.City,","))))
  rownames(a) <- NULL

## Creating new Destination and origin state abbreviation columns
  fall$DestinationStateAbbr <- trimws(b$X2)
  fall$OriginStateAbbr <- trimws(a$X2)

## Creating hash map between state abbreviation and state name
  StateMap <- hash(fall$OriginStateAbbr,fall$Origin.State)
  df <- fall

## Reassigning cleaned values (after removing comma and state code) to dest & orig cities
  df$Destination.City <- b$X1
  df$Origin.City <- a$X1

  remove(a)
  remove(b)
 ## Removing the columns that are not necessary
  df$Day.of.Month <- NULL
  df$freeText <- NULL
  df$Origin.State <- NULL
  df$Destination.State <- NULL
  df$Partner.Name <- NULL
```

```
#Replacing 1 NA in likelihood with mean at route & partner level
  averageLikelihood <- df %>%
    group_by(Origin.City, Destination.City, Partner.Code)%>%
    summarise(mean(Likelihood.to.recommend, na.rm=TRUE))

  indexOfNA <- which(is.na(df$Likelihood.to.recommend))
  df$Likelihood.to.recommend[indexOfNA] <-

as.integer(averageLikelihood[averageLikelihood$Origin.City==df[indexOfNA,"Origin.City"] &
                averageLikelihood$Destination.City==df[indexOfNA,"Destination.City"] &
                averageLikelihood$Partner.Code==df[indexOfNA,"Partner.Code"],4])
  remove(indexOfNA)
  remove(averageLikelihood)

  #Code for filtering out the Na when flight cancelled is No

  df %>%
    select(Origin.City,Destination.City,Flight.cancelled,Flight.time.in.minutes) %>%
    filter(is.na(Flight.time.in.minutes) & Flight.cancelled=="No")

  #Replacing NA in Flight time in minutes when the flight was not cancelled, to mean at
route & partner airline level

  averageFlightTime <- df %>%                ## Creating dataframe with average flight
time in minutes at route and partner level
    group_by(Origin.City, Destination.City, Partner.Code)%>%
    summarise(mean(Flight.time.in.minutes, na.rm=TRUE))


  indexOfNAList <- which(is.na(df$Flight.time.in.minutes) & df$Flight.cancelled=="No")

  for(indexOfNA in indexOfNAList){
    df$Flight.time.in.minutes[indexOfNA] <-

as.integer(averageFlightTime[averageFlightTime$Origin.City==df[indexOfNA,"Origin.City"] &

averageFlightTime$Destination.City==df[indexOfNA,"Destination.City"] &
```

```
                              averageFlightTime$Partner.Code==df[indexOfNA,"Partner.Code"],4])
  }

  ## Setting Flight Time in minutes to 0 for all cancelled flights.
  indexOfNAList <- which(is.na(df$Flight.time.in.minutes) & df$Flight.cancelled=="Yes")
  for(indexOfNA in indexOfNAList){df$Flight.time.in.minutes[indexOfNA] <- 0}


  ### For 3 route airline combinations, the average flight time in minutes was not obtained
for which we manually found their flight time and assigned it
  indexOfNAList <- which(is.na(df$Flight.time.in.minutes) & df$Flight.cancelled=="No")
  df$Flight.time.in.minutes[indexOfNAList[1]] <- 90
  df$Flight.time.in.minutes[indexOfNAList[2]] <- 100
  df$Flight.time.in.minutes[indexOfNAList[3]] <- 135


  remove(indexOfNA)
  remove(averageFlightTime)
  indexOfNAList <- which(is.na(df$Flight.time.in.minutes))


  #Replacing Arrival and Departure delay Nas. Only Arrival Delay column has 19 NAs
wherever flight was not cancelled,
  ## which is treated in the same way as flight time in minutes

  averageArrivalDelay <- df %>%
    group_by(Origin.City, Destination.City, Partner.Code)%>%
    summarise(mean(Arrival.Delay.in.Minutes, na.rm=TRUE))

  ## Departure Delaay has Nas only when flight was cancelled

  indexArrivalDelay <- which(is.na(df$Arrival.Delay.in.Minutes) &
df$Flight.cancelled=="No")     ## Obtaining indexes where arrival delay is NA & flight was
not cancelled

  ### Treating Nas for arrival delay column where the flight was not cancelled
  for(i in indexArrivalDelay)
  {
    df$Arrival.Delay.in.Minutes[i] <-
     as.integer(averageArrivalDelay[averageArrivalDelay$Origin.City==df[i,"Origin.City"] &
                                averageArrivalDelay$Destination.City==df[i,"Destination.City"] &
                                averageArrivalDelay$Partner.Code==df[i,"Partner.Code"],4])
    if(is.na(df$Arrival.Delay.in.Minutes[i]))
     {
     df$Arrival.Delay.in.Minutes[i] <-df$Departure.Delay.in.Minutes[i]
     }

  }

  ### Replacing Nas where flight was cancelled with NaN (Not a number) for both Arrival
Delay and Departure Delay columns.

  df$Arrival.Delay.in.Minutes[which(is.na(df$Arrival.Delay.in.Minutes) &
df$Flight.cancelled=="Yes")] <- NaN
  df$Departure.Delay.in.Minutes[which(is.na(df$Departure.Delay.in.Minutes) &
df$Flight.cancelled=="Yes")] <- NaN

  df2 <- df

  ## Creating new column names to be renamed

  columnNames <-
c("DestinationCity","OriginCity","AirlineStatus","Age","Gender","PriceSensitivity",

"YearOfFirstFlight","FLightsPerYear","Loyalty","TypeOfTravel","TotalFreqFlyAccount",
             "ShoppingAmount","FoodExpenses","Class","FlightDate","PartnerCode",
             "ScheduleDepHour","DepartureDelayInMin","ArrivalDelayInMin",
             "FlightCancelled","FlightDuration","Distance","LikelihoodRecommendScore",
             "OriginLong","OriginLat","DestLong","DestLat","DestinationState","OriginState")

  colnames(df) <- columnNames

## Creating new Age Group column
  df$AgeGroup <- cut(df$Age, breaks = c(0,18,36,54, Inf), labels = c('0-18','18-36','36-
54','>54'), right = FALSE)
```

```
remove(i)

remove(indexArrivalDelay)
remove(averageArrivalDelay)

remove(indexDepartureDelay)
remove(averageDepartureDelay)

remove(columnNames)
remove(fall)
remove(df2)
remove(indexOfNAList)
saveRDS(df,file="CleanedData.Rda")
saveRDS(partnerCodeToName,file="PartnerName.Rda")
saveRDS(StateMap,file="StateName.Rda")

testdf <- readRDS(file = "CleanedData.Rda")
write.csv(df, file = "cleandata.csv")

#### DATA CLEANING ENDS HERE ####
```

4. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is the initial process of performing investigations on the data prior to model preparation, to discover patterns, to spot anomalies, to test hypothesis and other assumptions. EDA is usually performed using visualizations in the form of Box Plots, Bar Graphs, Histograms or Pie Charts etc. which help in explaining a lot of descriptive statistics. Descriptive Statistics such as the spread of the data, the range and the mean/median values were obtained across each of the categories.

4.1 Box plot – traveler class and likelihood to recommend score

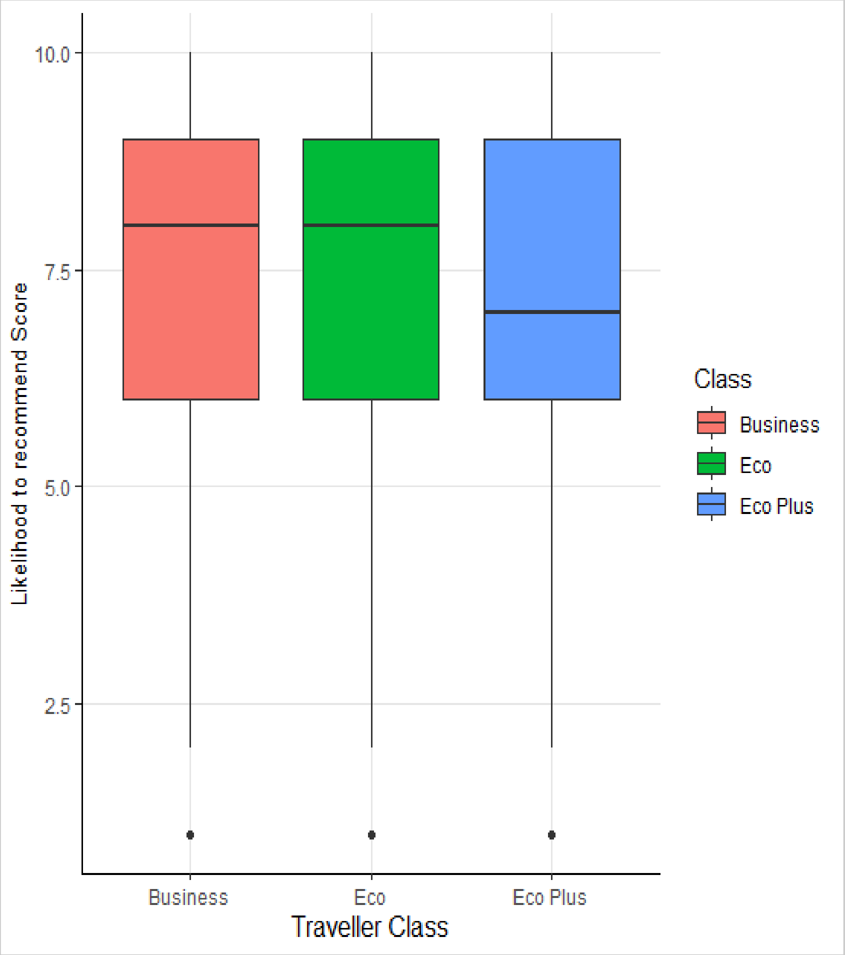


Figure 1 Box Plot of Likelihood to recommend score for each traveler class

From the above box plot we can observe that the median value of likelihood to recommend score is very low (< 7) for Eco Plus customers. Approximately, 50% of the

Eco Plus customers are not satisfied with the airline experience and hence they are either Detractors or are Passive.

The below written code was used to generate this box plot.

```
plotTheme <- theme_classic()+theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 10),plot.title = element_text(size = 15, hjust = 0.5),

panel.grid.major = element_line(color="#e6e6e6",linetype=1))

### *** BOX PLOTS *** ###

### Boxplot for each traveller classes

classPlot <- ggplot(df,aes(Class,LikelihoodRecommendScore))+geom_boxplot(aes(fill=Class))+ xlab("Traveller Class")+ ylab("Likelihood to recommend Score")+plotTheme

classPlot
```

4.2 Box plot – Gender and likelihood to recommend score

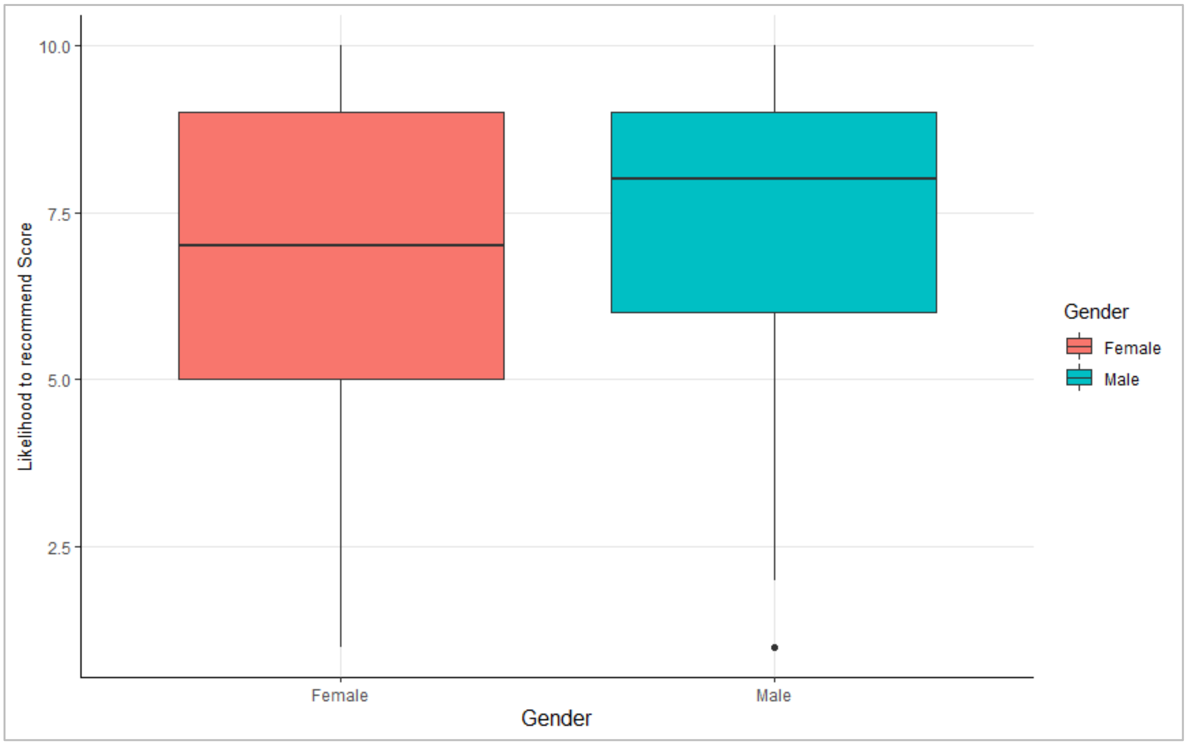


Figure 2 Box Plot of Likelihood to recommend score across the 2 genders

From the above boxplot we can observe that, female customers are less satisfied with the service, in comparison to male customers and hence, approximately 50% of female customers Detractors or Passive. On the other hand, 50% of the male customers are either Passive or Promoters.

The following code was written to obtain the above shown Box Plot

```
## Boxplot for the genders

genderPlot <-
ggplot(df,aes(Gender,LikelihoodRecommendScore))+geom_boxplot(aes(fill=Gender))+

  xlab("Gender")+ ylab("Likelihood to recommend Score")+plotTheme

genderPlot
```

4.3 Box plot – Traveler type and likelihood to recommend score

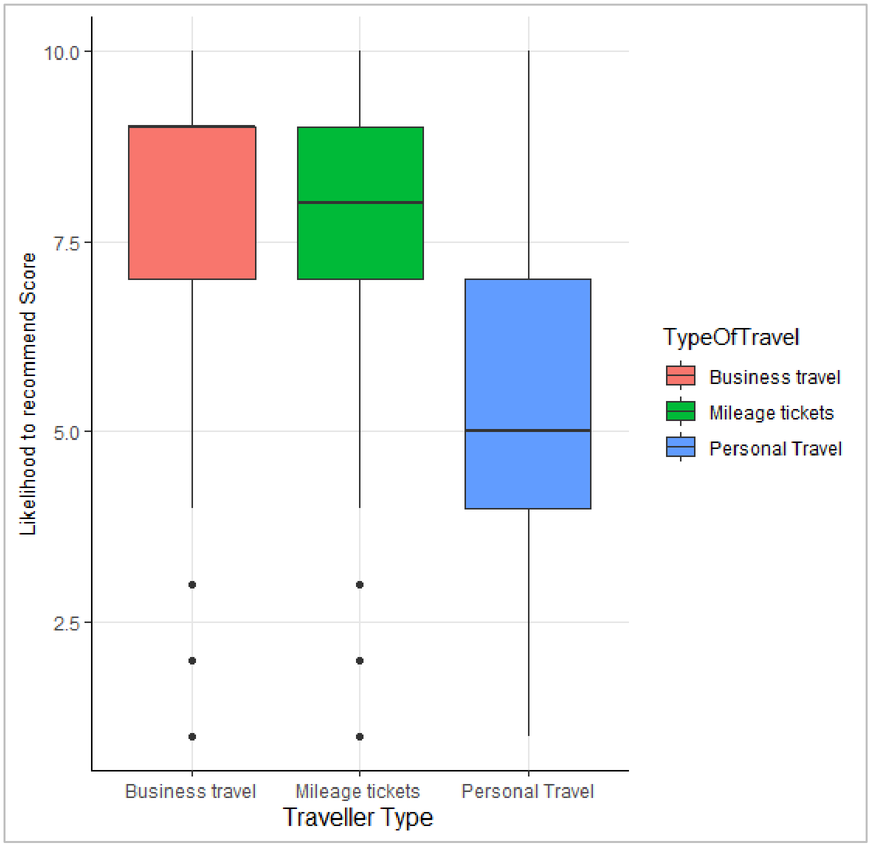


Figure 3 Box Plot of Likelihood to recommend score across the 3 traveler types

From the above figure, we can observe that customers on personal travel are very unhappy with the airline experience in comparison to the customers who are on business travel or on mileage tickets travel. It is very evident that, more than 50% of the personal travel customers are Detractors and moreover, 75% of the personal travel customers are either Passive or Detractors.

The following code was written to obtain the above box plot.

```
## Boxplot for traveler types

traveltypePlot <-
ggplot(df,aes(TypeOfTravel,LikelihoodRecommendScore))+geom_boxplot(aes(fill=TypeOfTravel))+

  xlab("Traveller Type")+ ylab("Likelihood to recommend Score")+plotTheme

traveltypePlot
```

4.4 Box plot – Airline status and likelihood to recommend score

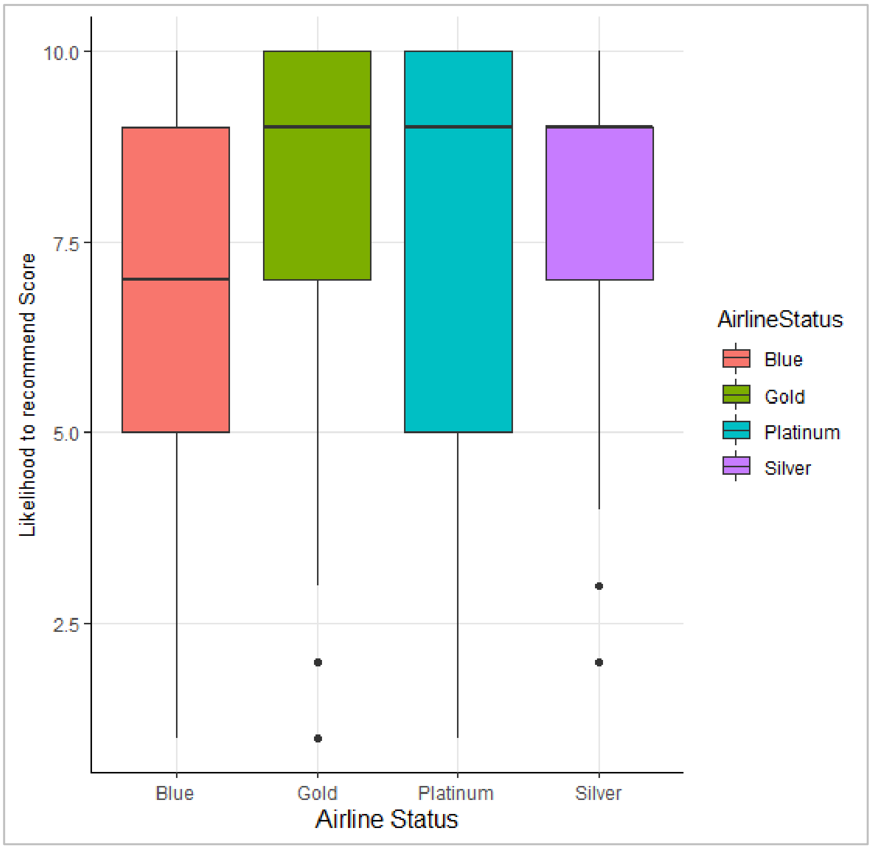


Figure 4 Box Plot of Likelihood to recommend score across the 4 Airline Status groups

From the above figure, we can observe that silver customers are most satisfied, and the silver group has the highest number of promoters with very less spread. Whereas, blue customers are less satisfied and hence 50% of the blue customers are either detractors or passive.

The below mentioned code was written to obtain the box plot generated in figure 4.

```
## Boxplot for airline status

airlineStatusPlot <-
ggplot(df,aes(AirlineStatus,LikelihoodRecommendScore))+geom_boxplot(aes(fill=Airline
Status))+ xlab("Airline Status")+ ylab("Likelihood to recommend Score")+plotTheme
```

The below mentioned code was used to obtain all 4 box plots in one screen in a single image.

```
p1 <- genderPlot+theme(legend.position = "none")

p2 <- traveltypePlot+theme(legend.position = "none")

p3 <- airlineStatusPlot+theme(legend.position = "none")

p4 <- classPlot+theme(legend.position = "none")+ggtitle("Box plot of Likelihood score
w.r.t gender, class, travel type and airline status")

subplot(p1,p2,p3,p4,nrows=2,margin = 0.05)
```

4.5 Box plot – Age group and likelihood to recommend score

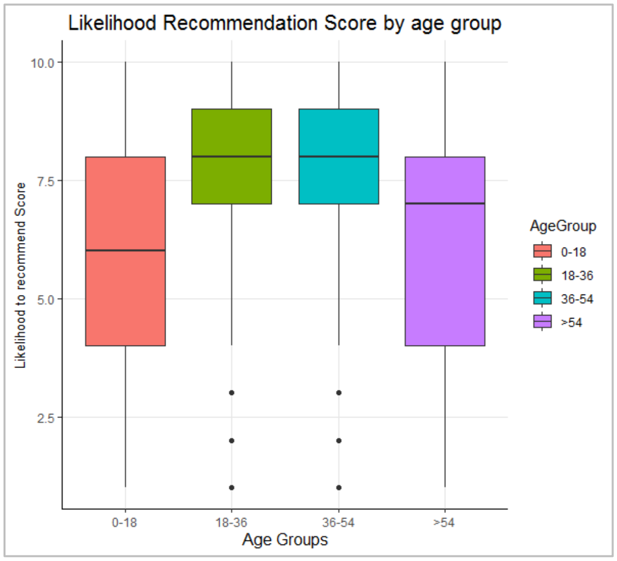


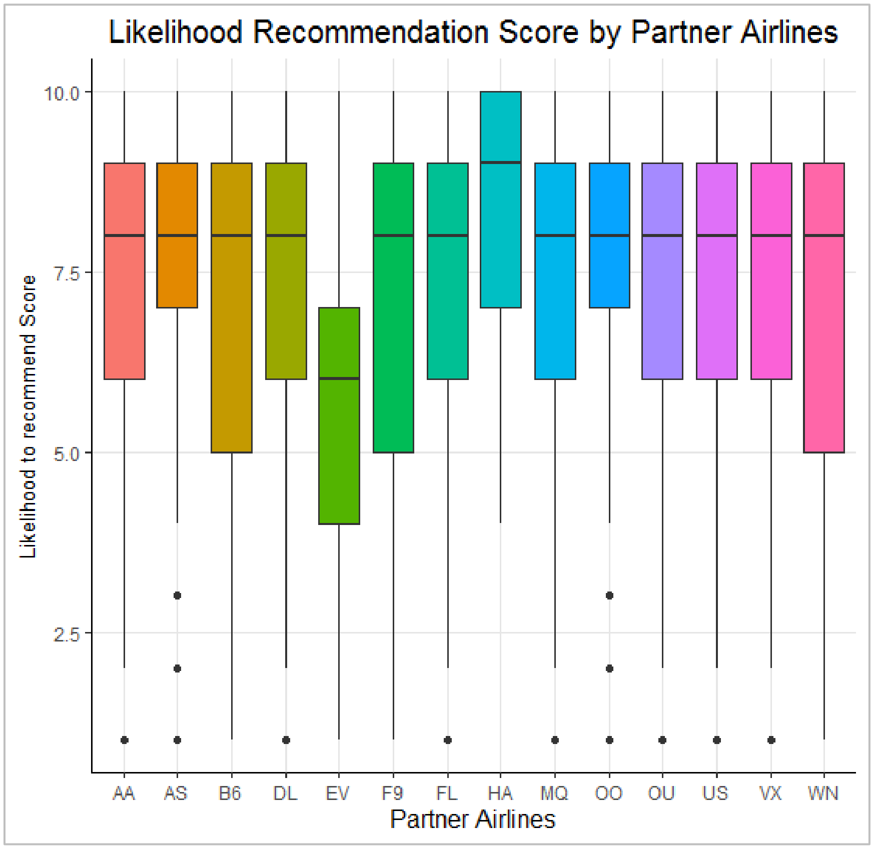
Figure 5 Box Plot of Likelihood to recommend score across the 4 Age Groups

From the above figure, we can observe that customers falling in the age group 0-18 & >54 are mostly unhappy with the experience, because of which more than 50% of the customers in the 0-18 age group are detractors. Also, more than 50% of the customers in the >54 age group are either passive or detractors. On the other hand, 50% of the customers in the 18-36 & 36-54 age groups are either Promoters or Passive.

The following code was written to generate the above shown box plot.

```
### Boxplot for Age Groups

ageGroupPlot <-ggplot(df,aes(AgeGroup,LikelihoodRecommendScore))+

 xlab("Age Groups")+ ylab("Likelihood to recommend Score")+

 ggtitle("Likelihood Recommendation Score by age group")+plotTheme

p1 <- ageGroupPlot+geom_boxplot(aes(fill=AgeGroup))
```

4.6 Box plot – Partner airlines and likelihood to recommend score



AA: Paul Smith Airlines Inc.

AS: FlyToSun Airlines Inc.

B6: OnlyJets Airlines Inc.

DL: Sigma Airlines Inc.

EV: FlyFast Airways Inc.

F9: GoingNorth Airlines Inc.

FL: FlyHere Airways

HA: West Airways Inc.

MQ: EnjoyFlying Air Services

OO: Northwest Business Airlines Inc.

OU: Oursin Airlines Inc.

US: Southeast Airlines Co.

VX: Cool&Young Airlines Inc.

WN: Cheapseats Airlines Inc.

Figure 6 Box Plot of Likelihood to recommend score across each partner airline

From the above figure we can observe that, in comparison to other Partner Airlines, ‘FlyFast Airways’ is performing very poorly with almost 75% of their customers being Detractors. On the other hand, ‘West Airways’ is performing very well with more than 50% of their customers being promoters.

The following code was written to generate the above boxplot

```
### Boxplot for Partner Airlines

scoreByPartner <-
ggplot(df,aes(PartnerCode,LikelihoodRecommendScore,partnerNameHash))+

 xlab("Partner Airlines")+ ylab("Likelihood to recommend Score")+

 ggtitle("Likelihood Recommendation Score by Partner Airlines")+

 plotTheme+theme(legend.position = "none")

p2 <- scoreByPartner+geom_boxplot(aes(fill=PartnerCode))

p2
```

4.7 Bar plot – Flight cancellation status and recommender type

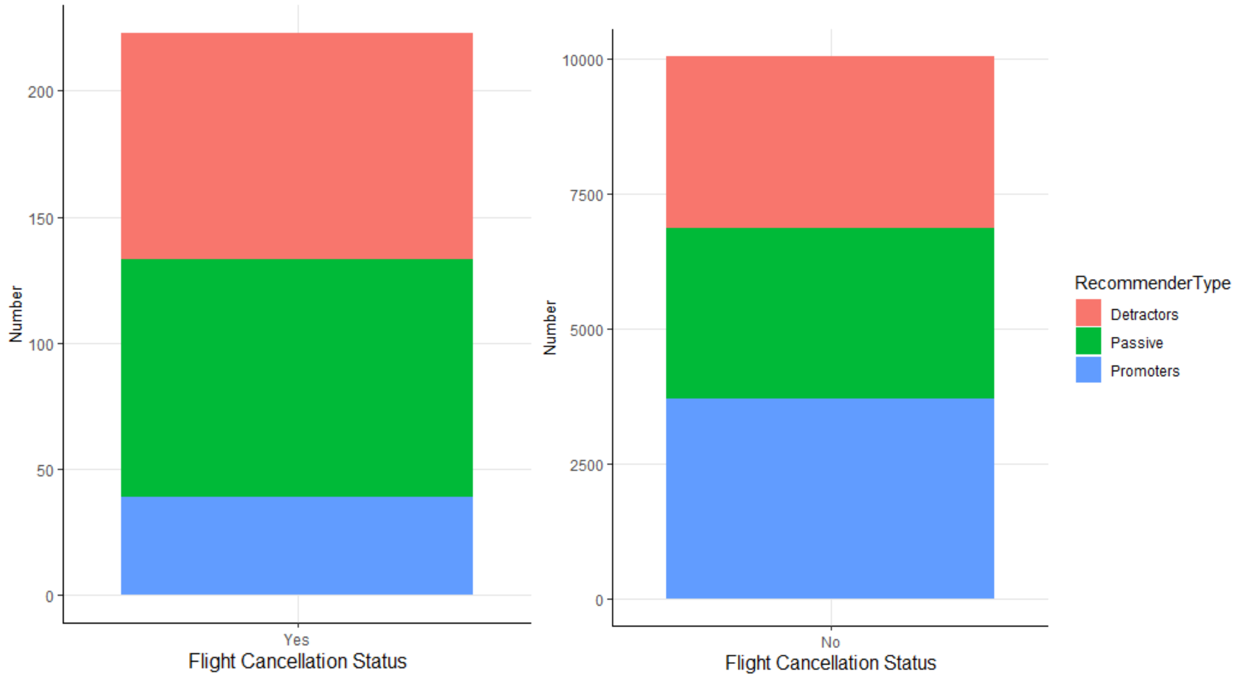


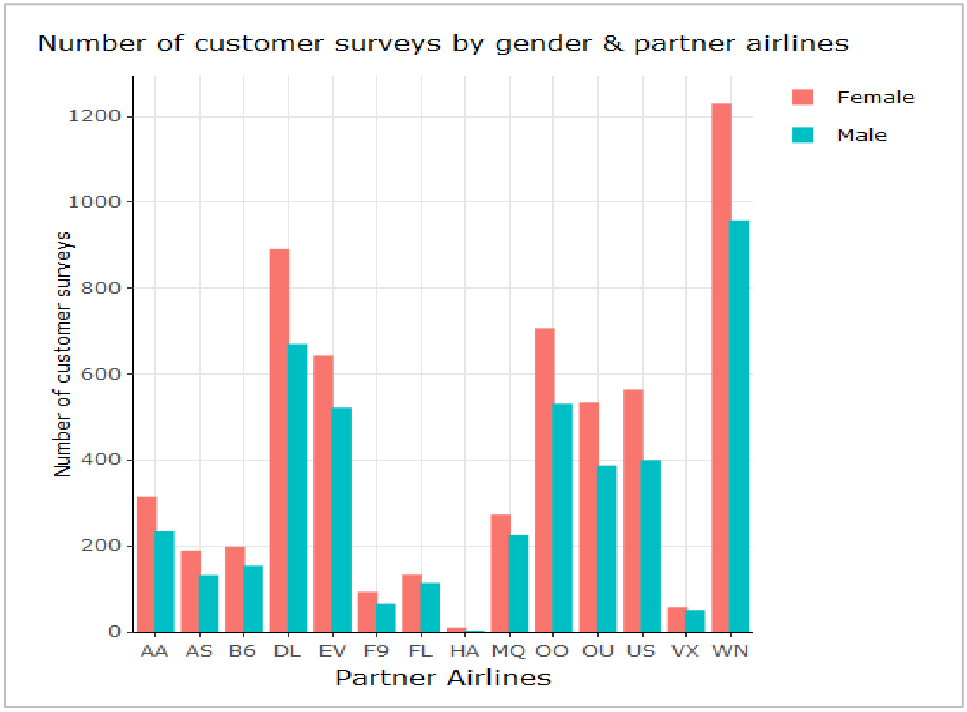
Figure 7 Overall NPS of Southeast airlines for the two flight cancellation status

From the above bar graph, we can observe that the number of promoters is higher for non-cancelled flights whereas cancelled flights have almost three times more number of detractors than promoters.

The following code was written to generate the above bar graph

```
### Number of PROMOTERS, DETRACTORS and PASSIVE for southeast airines, across
cancelled and non-cancelled flights

df$recommender_type <- cut(df$LikelihoodRecommendScore, breaks = c(0,7,9, Inf),
labels = c('Detractors','Passive','Promoters'), right = FALSE)   ## Creating Recommender
Type categorical variable

partner_cancellation_nps <-
data.frame(table(df$FlightCancelled,df$recommender_type))

## Number of Detractors, Promoters & Passives for each partner & cancellation status

colnames(partner_cancellation_nps) <-
c('FlightCancellationStatus','RecommenderType','Number')

partner_flight_status_no <-
ggplot(partner_cancellation_nps[partner_cancellation_nps$FlightCancellationStatus ==
'No',],aes(x=FlightCancellationStatus,y=Number,group=RecommenderType))+

  geom_col(aes(fill=RecommenderType))+

  xlab("Flight Cancellation Status")+ ylab("Number")+plotTheme

partner_flight_status_yes <-
ggplot(partner_cancellation_nps[partner_cancellation_nps$FlightCancellationStatus ==
'Yes',],aes(x=FlightCancellationStatus,y=Number,group=RecommenderType))+

  geom_col(aes(fill=RecommenderType))+

  xlab("Flight Cancellation Status")+ ylab("Number")+plotTheme
```

4.8 Relative frequency bar plot – Partner airlines and number of customer surveys



AA: Paul Smith Airlines Inc.

AS: FlyToSun Airlines Inc.

B6: OnlyJets Airlines Inc.

DL: Sigma Airlines Inc.

EV: FlyFast Airways Inc.

F9: GoingNorth Airlines Inc.

FL: FlyHere Airways

HA: West Airways Inc.

MQ: EnjoyFlying Air Services

OO: Northwest Business Airlines Inc.

OU: Oursin Airlines Inc.

US: Southeast Airlines Co.

VX: Cool&Young Airlines Inc.

WN: Cheapseats Airlines Inc.

Figure 8 Number of customer survey observations by gender & partner airlines

From the above bar graph, we can observe that, all the partner airlines have more female customers than male customers. Moreover, 'West Airways Inc.', 'Cool&Young Airlines' & also 'Going North Airlines' have very few customer survey observations in comparison to the other partners.

The following code was written to obtain the above bar graph

```
### NUMBER OF REVIEWS (customers) BY PARTNER & GENDER

dfT <- data.frame(table(df$PartnerCode,df$Gender))

colnames(dfT) <- c("PartnerCode","Gender","Count")

genderPlot <- ggplot(dfT,aes(x=PartnerCode,y=Count,group=Gender))+

 geom_col(aes(fill=Gender),show.legend=FALSE,position = "dodge")+

 xlab("Partner Airlines")+ ylab("Number of customer surveys")+

 ggtitle("Number of customer surveys by gender & partner airlines")+

 plotTheme+theme(plot.title = element_text(size = 12, hjust = 0.5))
```

```
ggplotly(genderPlot,tooltip=c("text","x","y"),dynamicTicks = TRUE)

#ggplotly(genderPlot,tooltip=c("text","x","y"),dynamicTicks = TRUE)
```

4.9 Bar plot – NPS score by partner airlines

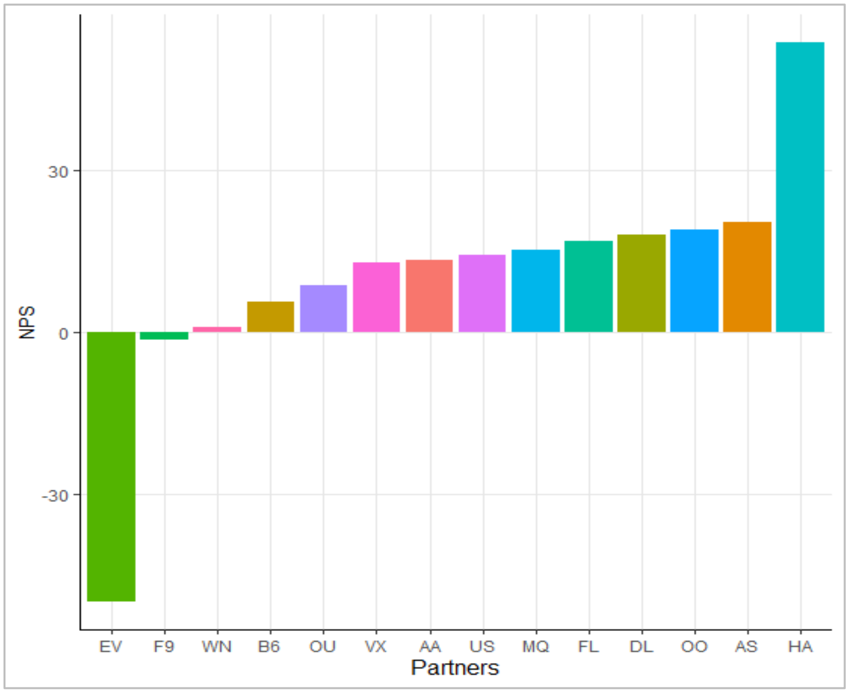


Figure 9 NPS % of each partner airline

AA: Paul Smith Airlines Inc.

AS: FlyToSun Airlines Inc.

B6: OnlyJets Airlines Inc.

DL: Sigma Airlines Inc.

EV: FlyFast Airways Inc.

F9: GoingNorth Airlines Inc.

FL: FlyHere Airways

HA: West Airways Inc.

MQ: EnjoyFlying Air Services

OO: Northwest Business Airlines Inc.

OU: Oursin Airlines Inc.

US: Southeast Airlines Co.

VX: Cool&Young Airlines Inc.

WN: Cheapseats Airlines Inc.

From the above bar graph, we can observe that ‘FlyFast Airways’ is performing extremely poor in comparison to the other partner airlines and they have a high negative NPS, i.e. they have many detractors and a small number of promoters. In addition to ‘FlyFast Airways’, ‘Going North Airlines’ also has a negative NPS. On the other hand, ‘FlyToSun Airlines’, ‘West Airways’ & ‘Northwest Business Airlines’ are performing well, with a positive NPS, i.e. they have more Promoters than Detractors.

4.10 Word Cloud

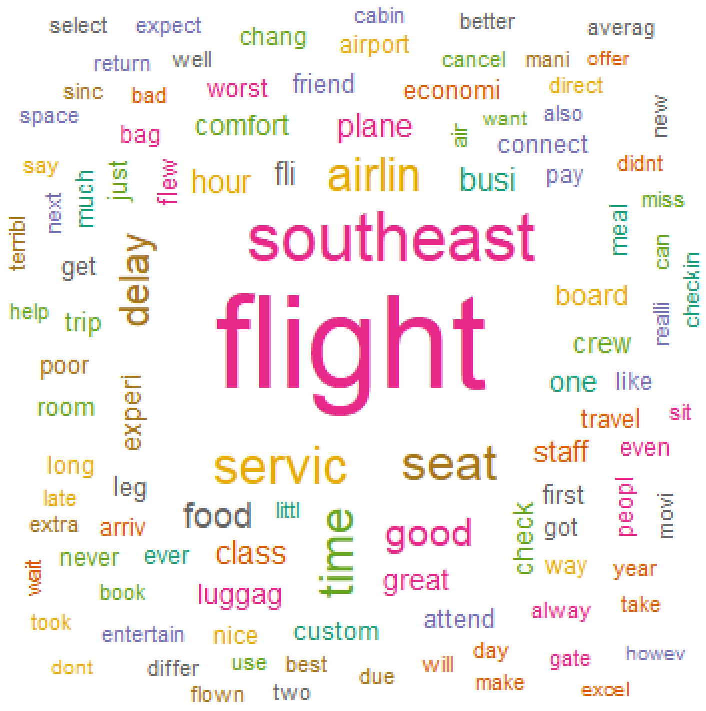


Figure 10 Word Cloud

5. MODEL PREPARATION

In order to help Southeast Airlines, we have decided to implement the following:

- Use Association Rules Mining to:

Identify Rules, i.e. set of customer & flight attribute values that are most likely to make a customer a Promoter

Similarly, identify set of customer & flight attribute values that are most likely to make a customer a Detractor

- Help decide which partners to keep & drop by predicting NPS of each partner airline for future flights by:

Predicting the Likelihood to Recommend score of customers for future flights using a linear model

For every customer predict whether he/she is going to be a Promoter, a Detractor or Passive using a Support Vector Machines classification model

5.1 LINEAR MODEL

Linear Modeling is a way to essentially predict a dependent variable based on the values of one or many independent variables. The linear model gives an output with information based on data that is already known and says how impactful certain variables are at predicting the dependent variable.

Linear models have three main important outputs that are important to analyze when running a model:

- Estimate: The Estimate tells you what coefficient each specific independent variable must be multiplied by to end up with the dependent variable when added together.
- P-Value: The P-Value tells you how "significant" each variable is, meaning how impactful it was in being related to the final output. If a P-Value is below .05, that often means that the variable is significant enough to be important when modeling.
- Adjusted R-Squared: Adjusted R-Square is a value which represents how well the model predicts the dependent variable, i.e. how much of the variation in the dependent variable can be explained by the independent variable. A "good" value changes depending on the case, but it is often seen as a goal for a model to explain 80% of the variation in the dependent variable, by the independent variables.

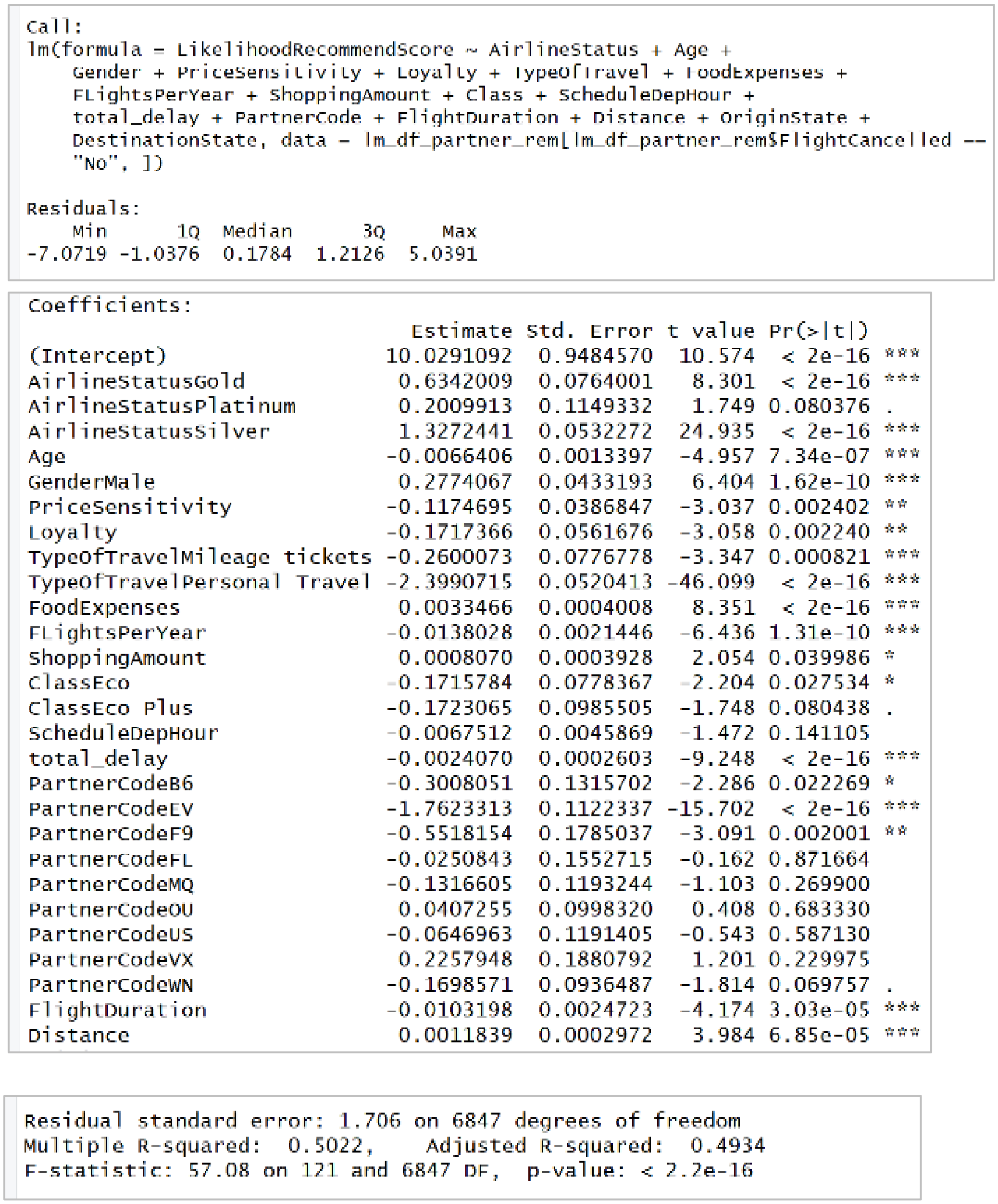
We used linear modeling techniques to predict which variables were most impactful on the "Likelihood to recommend" score for each passenger. Each independent variable was either a number or a string, and for the purposes of the linear model, we ran two separate models to see how people who had their flight cancelled felt versus people who did not have this issue. For us, this was appropriate because if the flights which were not cancelled experienced consistent negative reviews, then there must be a very big problem for the airline company.

We included many variables in our models, but we felt that each one was essential for making the best prediction as possible. The variables we used were: Airline Status, Age, Gender, Price Sensitivity, Loyalty, Type of Travel, Food Expenses, Flights per Year, Shopping Amount, Class, Schedule Departure Hour, Total Delay, Partner Code, Flight Duration, Distance, Origin State and Destination State. This was not the exhaustive list of variables provided, because we felt some such as Year of First Flight were not too important when figuring out whether a customer would recommend the airline's services. We also combined some of the variables, such as Arrival Delay and Departure Delay, giving us Total Delay. This just made our model a little smaller so there would be less noise.

Some restrictions were put on our models, as we took out a few partner airlines with the best "Likelihood to recommend" score, as well as the partner airline with the fewest reviews. These partner airlines were: West Airways, FlyToSun Airlines, Northwest Business Airlines, and Sigma Airlines, with West Airways having the best average "Likelihood to recommend" score and only 13 reviews. All the other partner airlines had over 100 reviews, so this small number had the possibility of throwing off our data.

The following code was written to generate a linear model that predicts the Likelihood to Recommend score for Non Cancelled Flights.

```
lm_df <- readRDS(file = "CleanedData.Rda")

lm_df$total_delay <- lm_df$ArrivalDelayInMin + lm_df$DepartureDelayInMin

## CREATING DF by removing partner HA that has very less observations and other partners with high NPS scores

## HA has >50%, AS, OO & DL have around 19%. Hence these 4 partners are removed from our linear model. Also VX is removed as it has very few rows.

lm_df_partner_rem <- lm_df[(lm_df$PartnerCode != 'HA') & (lm_df$PartnerCode != 'AS') & (lm_df$PartnerCode !='OO') & (lm_df$PartnerCode !='DL'),]

##Linear Model for non-cancelled flights after excluding high performing partners and partners with low number of observations. ADJ R square = 49.3 %

linearmodel_non_cancel <- lm(LikelihoodRecommendScore ~ AirlineStatus + Age + Gender + PriceSensitivity + Loyalty + TypeOfTravel + FoodExpenses + FLightsPerYear +ShoppingAmount + Class + ScheduleDepHour+ total_delay + PartnerCode + FlightDuration + Distance + OriginState + DestinationState, data = lm_df_partner_rem[lm_df_partner_rem$FlightCancelled == 'No',])

summary(linearmodel_non_cancel)
```



```
Call:
lm(formula = LikelihoodRecommendScore ~ AirlineStatus + Age +
    Gender + PriceSensitivity + Loyalty + TypeOfTravel + FoodExpenses +
    FlightsPerYear + ShoppingAmount + Class + ScheduleDepHour +
    total_delay + PartnerCode + FlightDuration + Distance + OriginState +
    DestinationState, data = lm_df_partner_rem[lm_df_partner_rem$FlightCancelled ==
    "No", ])

Residuals:
    Min      1Q  Median      3Q     Max
-7.0719 -1.0376  0.1784  1.2126  5.0391
```

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 10.0291092  0.9484570  10.574  < 2e-16 ***
AirlineStatusGold            0.6342009  0.0764001   8.301  < 2e-16 ***
AirlineStatusPlatinum        0.2009913  0.1149332   1.749 0.080376 .
AirlineStatusSilver          1.3272441  0.0532272  24.935  < 2e-16 ***
Age                         -0.0066406  0.0013397  -4.957 7.34e-07 ***
GenderMale                   0.2774067  0.0433193   6.404 1.62e-10 ***
PriceSensitivity            -0.1174695  0.0386847  -3.037 0.002402 **
Loyalty                     -0.1717366  0.0561676  -3.058 0.002240 **
TypeOfTravelMileage tickets -0.2600073  0.0776778  -3.347 0.000821 ***
TypeOfTravelPersonal Travel -2.3990715  0.0520413 -46.099  < 2e-16 ***
FoodExpenses                 0.0033466  0.0004008   8.351  < 2e-16 ***
FlightsPerYear              -0.0138028  0.0021446  -6.436 1.31e-10 ***
ShoppingAmount               0.0008070  0.0003928   2.054 0.039986 *
ClassEco                    -0.1715784  0.0778367  -2.204 0.027534 *
ClassEco Plus               -0.1723065  0.0985505  -1.748 0.080438 .
ScheduleDepHour             -0.0067512  0.0045869  -1.472 0.141105
total_delay                 -0.0024070  0.0002603  -9.248  < 2e-16 ***
PartnerCodeB6               -0.3008051  0.1315702  -2.286 0.022269 *
PartnerCodeEV               -1.7623313  0.1122337 -15.702  < 2e-16 ***
PartnerCodeF9               -0.5518154  0.1785037  -3.091 0.002001 **
PartnerCodeFL               -0.0250843  0.1552715  -0.162 0.871664
PartnerCodeMQ               -0.1316605  0.1193244  -1.103 0.269900
PartnerCodeOU                0.0407255  0.0998320   0.408 0.683330
PartnerCodeUS               -0.0646963  0.1191405  -0.543 0.587130
PartnerCodeVX                0.2257948  0.1880792   1.201 0.229975
PartnerCodeWN               -0.1698571  0.0936487  -1.814 0.069757 .
FlightDuration              -0.0103198  0.0024723  -4.174 3.03e-05 ***
Distance                     0.0011839  0.0002972   3.984 6.85e-05 ***
```

```
Residual standard error: 1.706 on 6847 degrees of freedom
Multiple R-squared:  0.5022,    Adjusted R-squared:  0.4934
F-statistic: 57.08 on 121 and 6847 DF,  p-value: < 2.2e-16
```

Figure 11 Output of Linear Model for Non-Cancelled Flights (state variables omitted due to space)

The above obtained results after running the linear models show that 49.3% of variation in the “Likelihood to recommend” score can be explained by variation in predictor attributes for when the flight was not cancelled. We can also see that many of the numeric and non-location-based variables were impactful when predicting the “Likelihood to recommend” score, with variables such as Total Delay and Airline Status being among the most significant.

The accuracy of the above generated linear model was predicted by creating a training dataset and a test dataset, wherein the same model was built on the training dataset and then it was used to predict the likelihood to recommend score on the test dataset. The accuracy was then measured as the % of observations where the actual value is same as the predicted value. For the likelihood to recommend score prediction, we obtained an accuracy of 22%, whereas we obtained a 54% accuracy for the categorization of the predicted likelihood to recommend score, i.e. the predicted likelihood to recommend score was categorized as Promoter/Detractor/Passive and was checked against the actual recommender type to obtain a 54% accuracy.

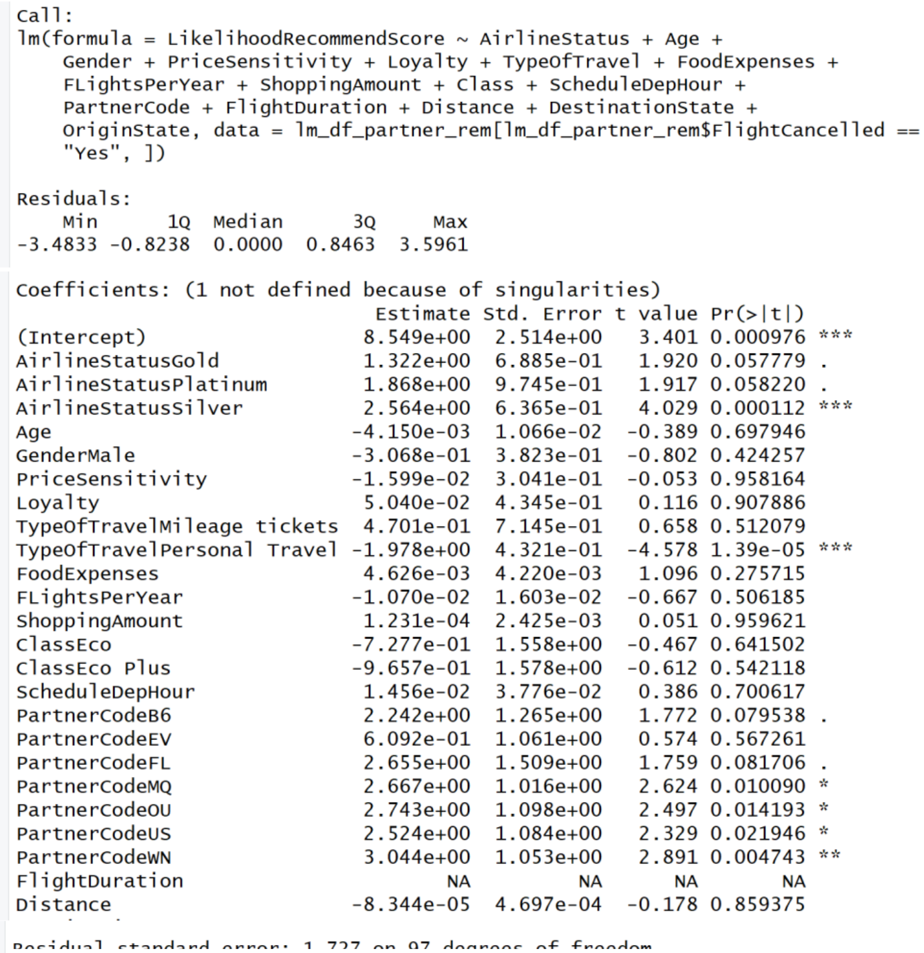
The following code was used to measure the accuracy.

```
## Generate Training and Test Data for the abve built model to measure accuracy.

#install.packages('splitstackshape')
library(splitstackshape)
randindex1 <- sample(1:dim(lm_df_partner_rem))
cut_point2_3 <- floor(2*dim(lm_df_partner_rem)[1]/3)
train_non_cancel_rem <- stratified(lm_df_partner_rem,
c("OriginState","DestinationState"), 0.66)      ### 66.6% of data is used as Training Data



test_non_cancel_rem <-
lm_df_partner_rem[randindex[(cut_point2_3+1):dim(lm_df_partner_rem)[1]],]

linearmodel_train_non_cancel_rem <- lm(LikelihoodRecommendScore ~ AirlineStatus + Age
+ Gender + PriceSensitivity + Loyalty + TypeOfTravel +
                        FoodExpenses + FLightsPerYear + ShoppingAmount + Class + total_delay +
PartnerCode + FlightDuration + DestinationState + OriginState,
                        data = train_non_cancel_rem[train_non_cancel_rem$FlightCancelled ==
'No',])
summary(linearmodel_train_non_cancel_rem)

test_predictions <-predict(linearmodel_train_non_cancel_rem,test_non_cancel_rem)
comparison_table <-
data.frame(test_non_cancel_rem$LikelihoodRecommendScore,test_non_cancel_rem$reco
mmender_type,round(test_predictions,0))
colnames(comparison_table) <- c('actual','actual_recommender_type','predicted')
comparison_table$predicted_recommender_type <- cut(comparison_table$predicted,
breaks = c(0,7,9, Inf), labels = c('Detractors','Passive','Promoters'), right = FALSE)

length(which(comparison_table$actual==comparison_table$predicted))/length(compariso
n_table$actual)    ## 22.1% Accuracy based on actual Likelihood to recommend score
predicted values.
```

```
length(which(comparison_table$actual_recommender_type==comparison_table$predicte
d_recommender_type))/length(comparison_table$actual)      ## 54% Accuracy based on
classifying predicted likelihood to recommend score
```

The following code was written to generate a linear model that predicts the Likelihood to Recommend score for Cancelled Flights.

```
## Linear model for cancelled flights after excluding high performing partners and partners
with low number of observations. ADJ R square = 38.07 %

linearmodel_cancel <- lm(LikelihoodRecommendScore ~ AirlineStatus + Age + Gender +
PriceSensitivity + Loyalty + TypeOfTravel +
            FoodExpenses + FLightsPerYear +ShoppingAmount + Class + ScheduleDepHour+
PartnerCode + FlightDuration + Distance + DestinationState + OriginState,
           data = lm_df_partner_rem[lm_df_partner_rem$FlightCancelled == 'Yes',])
summary(linearmodel_cancel)
```



```
Call:
lm(formula = LikelihoodRecommendScore ~ AirlineStatus + Age + 
    Gender + PriceSensitivity + Loyalty + TypeOfTravel + FoodExpenses + 
    FLightsPerYear + ShoppingAmount + Class + ScheduleDepHour + 
    PartnerCode + FlightDuration + Distance + DestinationState + 
    OriginState, data = lm_df_partner_rem[lm_df_partner_rem$FlightCancelled == 
    "Yes", ])

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4833 -0.8238  0.0000  0.8463  3.5961 

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 8.549e+00  2.514e+00   3.401 0.000976 ***
AirlineStatusGold           1.322e+00  6.885e-01   1.920 0.057779 .  
AirlineStatusPlatinum       1.868e+00  9.745e-01   1.917 0.058220 .  
AirlineStatusSilver         2.564e+00  6.365e-01   4.029 0.000112 ***
Age                        -4.150e-03  1.066e-02  -0.389 0.697946    
GenderMale                 -3.068e-01  3.823e-01  -0.802 0.424257    
PriceSensitivity           -1.599e-02  3.041e-01  -0.053 0.958164    
Loyalty                     5.040e-02  4.345e-01   0.116 0.907886    
TypeOfTravelMileage tickets  4.701e-01  7.145e-01   0.658 0.512079    
TypeOfTravelPersonal Travel -1.978e+00  4.321e-01  -4.578 1.39e-05 ***
FoodExpenses                4.626e-03  4.220e-03   1.096 0.275715    
FLightsPerYear             -1.070e-02  1.603e-02  -0.667 0.506185    
ShoppingAmount              1.231e-04  2.425e-03   0.051 0.959621    
ClassEco                   -7.277e-01  1.558e+00  -0.467 0.641502    
ClassEco Plus              -9.657e-01  1.578e+00  -0.612 0.542118    
ScheduleDepHour             1.456e-02  3.776e-02   0.386 0.700617    
PartnerCodeB6               2.242e+00  1.265e+00   1.772 0.079538 .  
PartnerCodeEV               6.092e-01  1.061e+00   0.574 0.567261    
PartnerCodeFL               2.655e+00  1.509e+00   1.759 0.081706 .  
PartnerCodeMQ               2.667e+00  1.016e+00   2.624 0.010090 *  
PartnerCodeOU               2.743e+00  1.098e+00   2.497 0.014193 *  
PartnerCodeUS               2.524e+00  1.084e+00   2.329 0.021946 *  
PartnerCodeWN               3.044e+00  1.053e+00   2.891 0.004743 ** 
FlightDuration                     NA         NA      NA       NA    
Distance                   -8.344e-05  4.697e-04  -0.178 0.859375    

Residual standard error: 1.727 on 97 degrees of freedom
```

Figure 12 Output of Linear Model for Cancelled Flights (state variables omitted due to space)

The results for the linear models for the cancelled flights show that only 38.1% of variation in the “Likelihood to recommend” score can be explained by variation in predictor attributes. This is fewer than for the model that was for the non-cancelled flights and this means that the “Likelihood to recommend” score has more at play than just the variables we had access to. There were far fewer significant variables, with Airline Status – Silver and Type of Travel – Personal Travel being the most significant non-state related variable.

The Accuracy of the linear model for cancelled flights was measured using the following code

```
lm_df_partner_rem_cancel <- lm_df_partner_rem[lm_df_partner_rem$FlightCancelled ==
'Yes',]
randindex2 <- sample(1:dim(lm_df_partner_rem_cancel))
cut_point2_3 <- floor(2*dim(lm_df_partner_rem_cancel)[1]/3)
train_cancel_rem <- stratified(lm_df_partner_rem_cancel,
c("OriginState","DestinationState"), 0.66)      ### 66.6% of data is used as Training Data

test_cancel_rem <-
lm_df_partner_rem_cancel[randindex2[(cut_point2_3+1):dim(lm_df_partner_rem_cancel)[
1]],]

linearmodel_train_cancel_rem <- lm(LikelihoodRecommendScore ~ AirlineStatus + Age +
Gender + PriceSensitivity + Loyalty + TypeOfTravel +
                          FoodExpenses + FLightsPerYear + ShoppingAmount + Class +
PartnerCode + FlightDuration + DestinationState + OriginState,
                         data = lm_df_partner_rem_cancel)
summary(linearmodel_train_cancel_rem)

test_predictions2 <-predict(linearmodel_train_cancel_rem,test_cancel_rem)
comparison_table2 <-
data.frame(test_cancel_rem$LikelihoodRecommendScore,test_cancel_rem$recommender
_type,round(test_predictions2,0))
colnames(comparison_table2) <- c('actual','actual_recommender_type','predicted')
comparison_table2$predicted_recommender_type <- cut(comparison_table2$predicted,
breaks = c(0,7,9, Inf), labels = c('Detractors','Passive','Promoters'), right = FALSE)

length(which(comparison_table2$actual==comparison_table2$predicted))/length(comparis
on_table2$actual)    ## 35% Accuracy based on actual Likelihood to recommend score
predicted values.
```

```
length(which(comparison_table2$actual_recommender_type==comparison_table2$predicted_recommender_type))/length(comparison_table2$actual)    ## 73% Accuracy based on classifying predicted likelihood to recommend score
```

For the likelihood to recommend score prediction, we obtained an accuracy of 35%, whereas we obtained a 73% accuracy for the categorization of the predicted likelihood to recommend score, i.e. the predicted likelihood to recommend score was categorized as Promoter/Detractor/Passive and was checked against the actual recommender type to obtain a 73% accuracy.

5.2 ASSOCIATIVE RULES MINING

Associative Rule Mining is a rule-based machine learning that can be used for finding patterns in data. It finds features that occur together as well as that are correlated to each other.

The three main Parameters of the Associative Rule Mining are as follows:

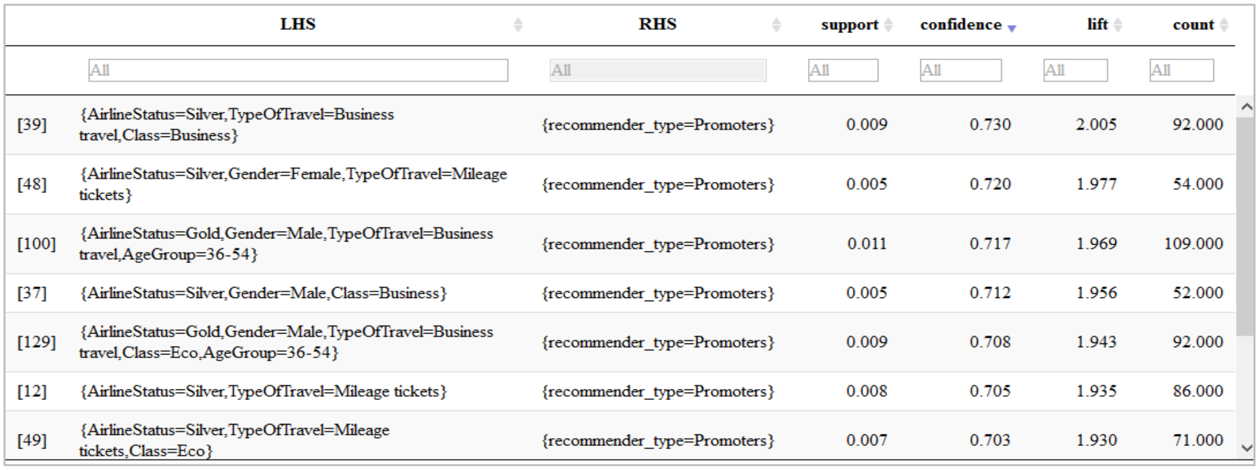
- Support: It tells us how much historical data supports the rule. It is the joint probability of two events A and B.
- Confidence: It tells us how confident are we that the rule holds. It is the conditional probability of an event, B given the event, A.
- Lift: It is the ratio of confidence to support.

We have used Associative Rules Mining in our project to identify the set of customer attributes that make a customer a Promoter & Detractor by placing recommender type on the right-hand side.
We have used the following attributes to detect whether the customer recommends this airline or not: Airline Status, Age Groups, Gender, Type of Travel & Class
Recommender Type is the attribute we have taken on the right-hand side.

The following code was used to perform Associative Rule Mining

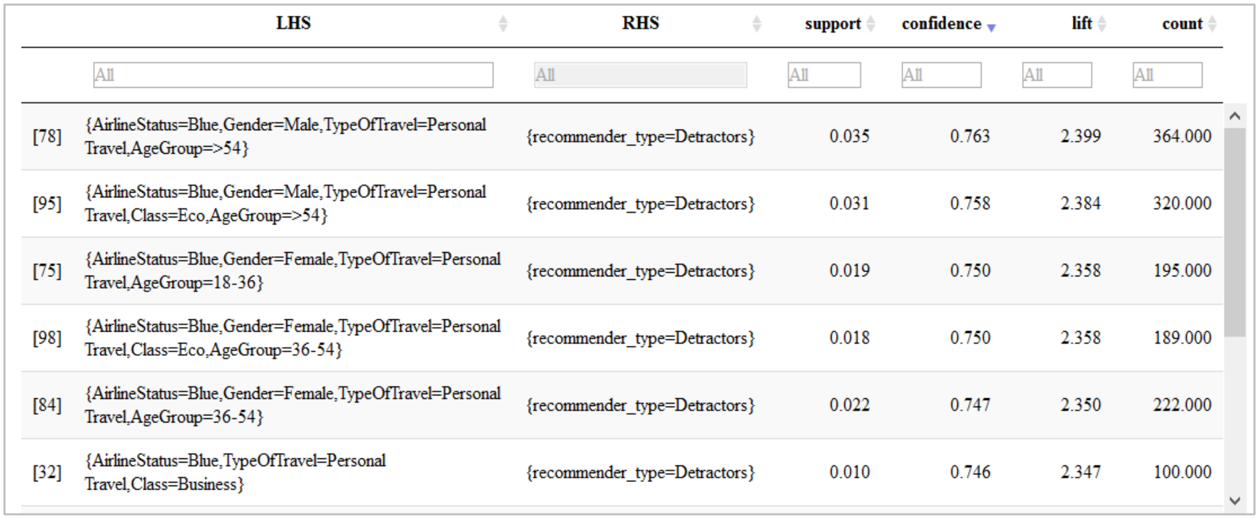
```
df <- readRDS(file = "CleanedData.Rda")
library(arules)
library(arulesViz)
library(tidyverse)
View(df)
table(df$recommender_type)
df$recommender_type <- cut(df$LikelihoodRecommendScore, breaks = c(0,7,9, Inf),
                labels = c('Detractors','Passive','Promoters'), right = FALSE)
## Creating Recommender Type categorical variable
```

```
Associative_Df <- df %>% select (AirlineStatus,Gender,
                        TypeOfTravel,Class,AgeGroup,
                        recommender_type)

Associative_Df$AirlineStatus <- as.factor(Associative_Df$AirlineStatus)
Associative_Df$AgeGroup <- as.factor(Associative_Df$AgeGroup)
Associative_Df$Gender <- as.factor(Associative_Df$Gender)
Associative_Df$TypeOfTravel <- as.factor(Associative_Df$TypeOfTravel)
Associative_Df$Class <- as.factor(Associative_Df$Class)
Associative_Df$recommender_type <- as.factor(Associative_Df$recommender_type)

Associative_DfX <- as(Associative_Df,"transactions")

ruleset <- apriori(Associative_DfX,     ### RULES FOR DETRACTORS
              parameter=list(support=0.005,confidence=0.5), # Setting support as 0.5% and
confidence as 50%
              appearance = list(default="lhs", rhs=("recommender_type=Detractors")))

inspect(ruleset)
inspectDT(ruleset)
ruleset_p <- apriori(Associative_DfX,     ### RULES FOR PROMOTERS
              parameter=list(support=0.005,confidence=0.5), # Setting support as 0.5% and
confidence as 50%
              appearance = list(default="lhs", rhs=("recommender_type=Promoters")))

inspect(ruleset_p)
inspectDT(ruleset_p)
```



	LHS	RHS	support	confidence	lift	count
	All	All	All	All	All	All
[39]	{AirlineStatus=Silver,TypeOfTravel=Business travel,Class=Business}	{recommender_type=Promoters}	0.009	0.730	2.005	92.000
[48]	{AirlineStatus=Silver,Gender=Female,TypeOfTravel=Mileage tickets}	{recommender_type=Promoters}	0.005	0.720	1.977	54.000
[100]	{AirlineStatus=Gold,Gender=Male,TypeOfTravel=Business travel,AgeGroup=36-54}	{recommender_type=Promoters}	0.011	0.717	1.969	109.000
[37]	{AirlineStatus=Silver,Gender=Male,Class=Business}	{recommender_type=Promoters}	0.005	0.712	1.956	52.000
[129]	{AirlineStatus=Gold,Gender=Male,TypeOfTravel=Business travel,Class=Eco,AgeGroup=36-54}	{recommender_type=Promoters}	0.009	0.708	1.943	92.000
[12]	{AirlineStatus=Silver,TypeOfTravel=Mileage tickets}	{recommender_type=Promoters}	0.008	0.705	1.935	86.000
[49]	{AirlineStatus=Silver,TypeOfTravel=Mileage tickets,Class=Eco}	{recommender_type=Promoters}	0.007	0.703	1.930	71.000

Figure 13 Output of Associative Rules Mining for Promoters



	LHS	RHS	support	confidence	lift	count
	All	All	All	All	All	All
[78]	{AirlineStatus=Blue,Gender=Male,TypeOfTravel=Personal Travel,AgeGroup=>54}	{recommender_type=Detractors}	0.035	0.763	2.399	364.000
[95]	{AirlineStatus=Blue,Gender=Male,TypeOfTravel=Personal Travel,Class=Eco,AgeGroup=>54}	{recommender_type=Detractors}	0.031	0.758	2.384	320.000
[75]	{AirlineStatus=Blue,Gender=Female,TypeOfTravel=Personal Travel,AgeGroup=18-36}	{recommender_type=Detractors}	0.019	0.750	2.358	195.000
[98]	{AirlineStatus=Blue,Gender=Female,TypeOfTravel=Personal Travel,Class=Eco,AgeGroup=36-54}	{recommender_type=Detractors}	0.018	0.750	2.358	189.000
[84]	{AirlineStatus=Blue,Gender=Female,TypeOfTravel=Personal Travel,AgeGroup=36-54}	{recommender_type=Detractors}	0.022	0.747	2.350	222.000
[32]	{AirlineStatus=Blue,TypeOfTravel=Personal Travel,Class=Business}	{recommender_type=Detractors}	0.010	0.746	2.347	100.000

Figure 14 Output of Associative Rules Mining for Detractors

From the results obtained after running the Associative Rule Mining model, we observe that customers with airline status ‘Silver’ travelling by business class for business purpose are highly likely to become Promoters. On the other hand, male customers over 54 years of age with airline status ‘Blue’ travelling for personal reasons are highly likely to become Detractors.

5.3 Support Vector Machines

SVM or Support Vector Machine is a popular supervised machine learning algorithm which can be used for regression and classification problems. We had come across three keywords supervised, regression and classification.

Supervised learning or predictive approach maps input x to output y, with the given set of input-output pairs $D = \{(x_i\ y_i\ )\}^{N}_{i=1.}$ Here D is called the training set and N is the number of training examples.

Regression is like classification except the response variable Y is continuous. We try to fit the input $x_i \in \mathbf{R}$ in a straight line or a quadratic function. Figure two figure plots the same data using linear regression (degree 1) and with polynomial regression (degree 2)

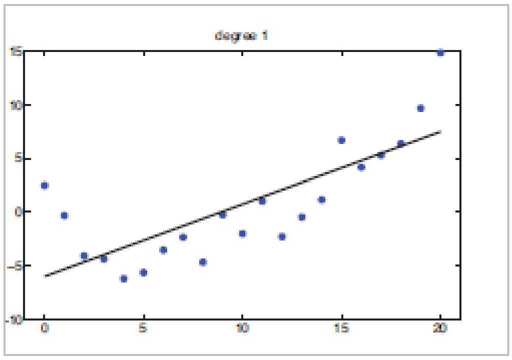


Figure 15 Linear Regression

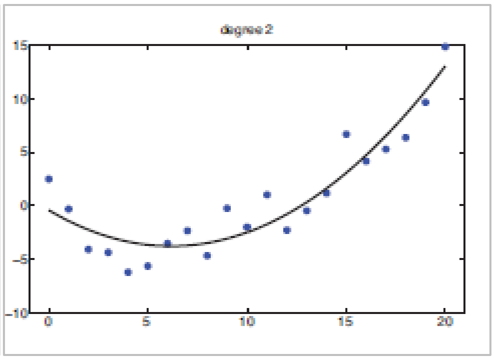


Figure 16 Polynomial Regression with degree 2

The goal of the classification is to learn a mapping from inputs x to output y, where y ∈ *{1,2,…..,C}* with C being the number of classes. If C=2 this is called binary classification. If C>2 is called multiclass classification. Classification tries to group the input label w.r.t output classes.

We have used SVM multiclass classification model for this section. It is a nonlinear mapping for transforming the original training data into higher dimension. It searches for the optimal hyperplane which separates two or more classes. SVM uses the support vectors (training set) and margins to find this hyperplane.

Let's take an example with two classes where the classes are linearly separable. The data set *D* be given as (***X***1, *y*1), (***X***2, *y*2), : : : , (***X***j*D*j, *y*j*D*j), where ***X****i* is the set of training tuples with associated class labels, *yi* . Each *yi* can take one of two values either when the buy_computer=yes and buy_computer=no. From the below figure we can deduce that the data is linearly separable with infinite number of hyperplane (straight line) which can be drawn to separate both the classes.

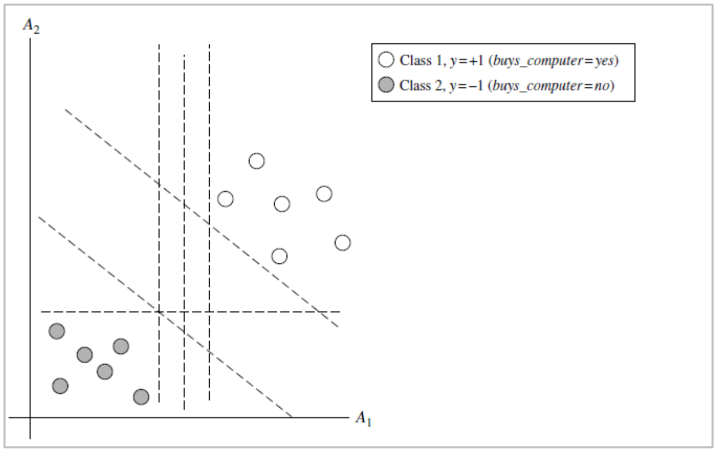

A_2
○ Class 1, y=+1 (*buys_computer=yes*)
● Class 2, y=−1 (*buys_computer=no*)
A_1

Figure 17 2D figure of linearly separable data with infinite number of possible hyperplane

There are thousands of combinations possible which can separate both the data. But we need to find the best one, we expect that the hyperplane with large margin can classify data more accurately that the hyperplane with less margin. SVM solves this problem by searching for **maximum marginal hyperplane MMH**.

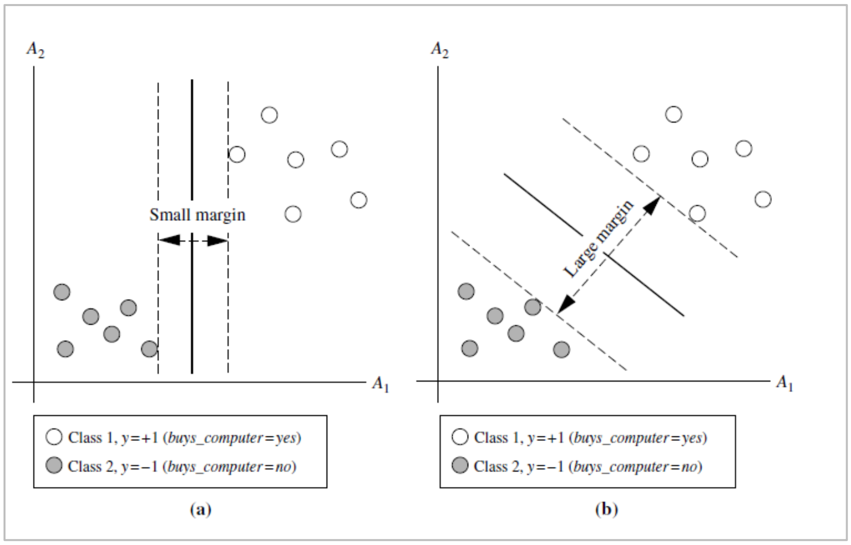

A_2
Small margin
A_1
○ Class 1, y=+1 (*buys_computer=yes*)
● Class 2, y=−1 (*buys_computer=no*)
(a)
A_2
Large margin
A_1
○ Class 1, y=+1 (*buys_computer=yes*)
● Class 2, y=−1 (*buys_computer=no*)
(b)

Figure 18 Two possible hyperplane

There are many cases possible and number of classes can be more than 2. There are even cases when the data is linearly inseparable. So the decision boundary constructed will be nonlinear. Sometimes we may overfit the model and hyperplane build is so perfect that it perfectly fit the training set by building a very complex nonlinear hyperplane which exactly separates the training data. But same may not apply on data other than training set.

Airline data has three classes and all the may or may not be linarly separable for each other. Before using model, we had split the dataset into two set training and test with split ratio 0.75.

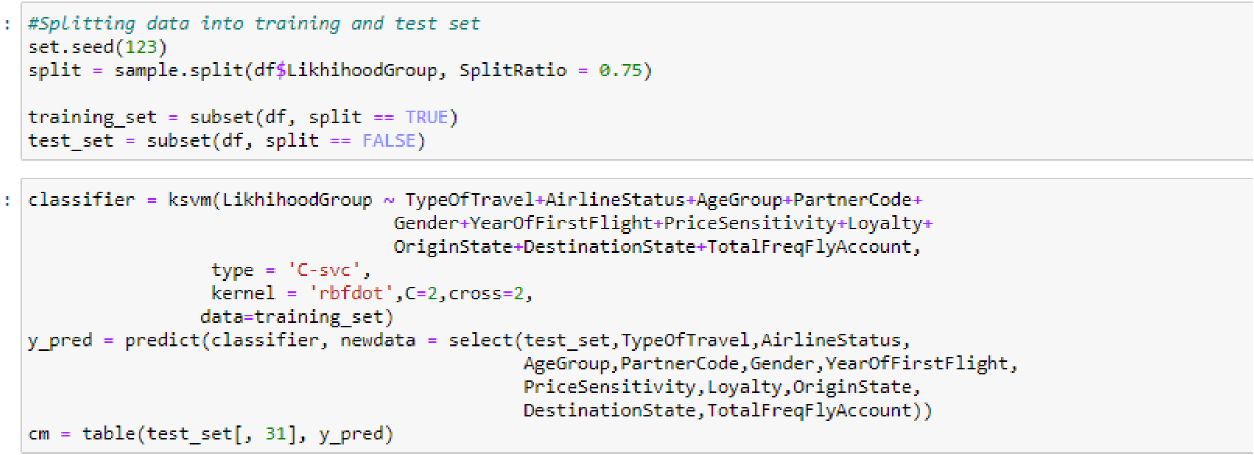
Kernlab library of the R is sed for modeling. Dependent variable selected is likelihood to recommend score while a set of independent variables is used for model building. Set includes travel type, airline status, age group, partner code, gender, year of first flight, price sensitivity, loyalty, origin state, destination set and total frequent flyer account.

We got **61.29 percent accuracy** with 52.79 no information rate

Various fine-tuning possibilities is used with the best possible fit as below.

- 'C-svc' is used for type parameter of the ksvm which means the C classification is used.

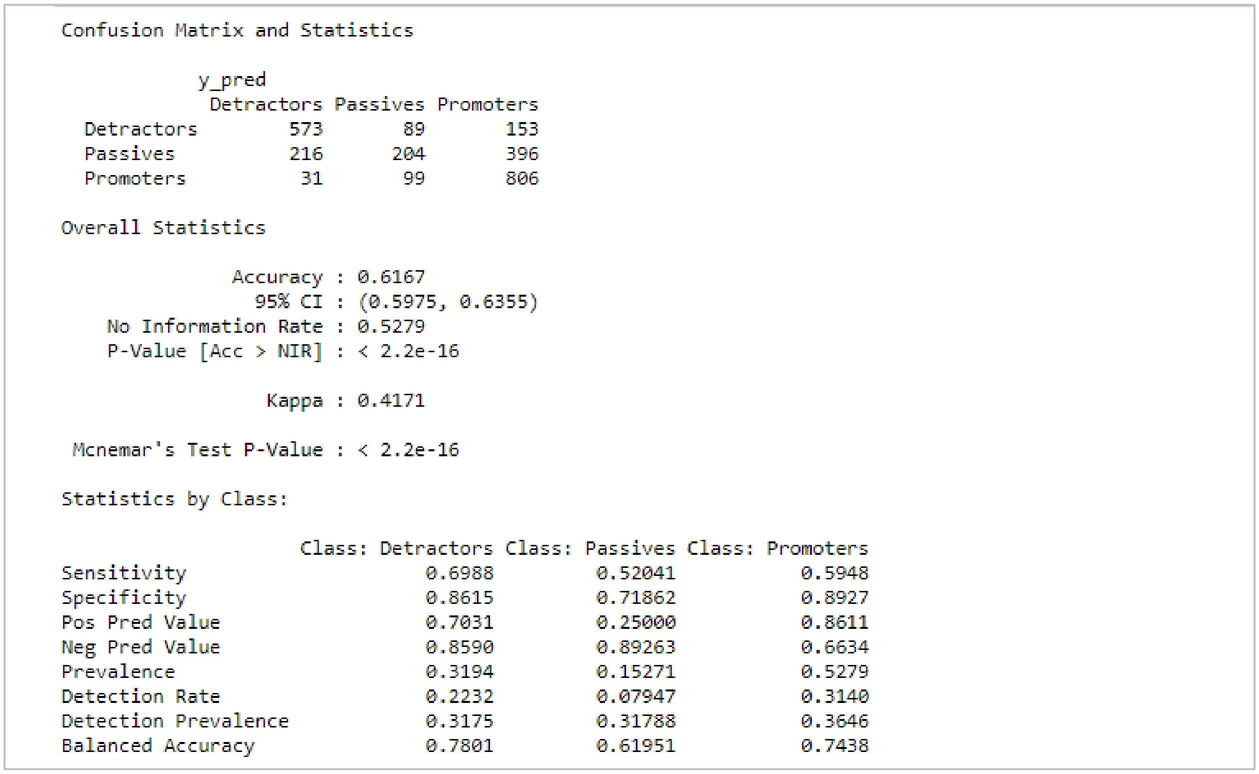
- ‘rbfdot’ kernel is used for the kernel parameter. Gaussian kernel or radial basis kernel is used.
- Cost of constraint violation is 2 i.e. C=2
- K-fold cross validation with k=2 is used. k-fold is performed to assess the quality of the model.



```
: #Splitting data into training and test set
  set.seed(123)
  split = sample.split(df$LikhihoodGroup, SplitRatio = 0.75)

  training_set = subset(df, split == TRUE)
  test_set = subset(df, split == FALSE)
```

```
: classifier = ksvm(LikhihoodGroup ~ TypeOfTravel+AirlineStatus+AgeGroup+PartnerCode+
                                  Gender+YearOfFirstFlight+PriceSensitivity+Loyalty+
                                  OriginState+DestinationState+TotalFreqFlyAccount,
                   type = 'C-svc',
                   kernel = 'rbfdot',C=2,cross=2,
                  data=training_set)
  y_pred = predict(classifier, newdata = select(test_set,TypeOfTravel,AirlineStatus,
                                                AgeGroup,PartnerCode,Gender,YearOfFirstFlight,
                                                PriceSensitivity,Loyalty,OriginState,
                                                DestinationState,TotalFreqFlyAccount))
  cm = table(test_set[, 31], y_pred)
```

Figure 19 SVM Code



```
Confusion Matrix and Statistics

            y_pred
             Detractors Passives Promoters
  Detractors        573       89       153
  Passives          216      204       396
  Promoters          31       99       806

Overall Statistics

               Accuracy : 0.6167
                 95% CI : (0.5975, 0.6355)
    No Information Rate : 0.5279
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4171

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Detractors Class: Passives Class: Promoters
Sensitivity                     0.6988         0.52041           0.5948
Specificity                     0.8615         0.71862           0.8927
Pos Pred Value                  0.7031         0.25000           0.8611
Neg Pred Value                  0.8590         0.89263           0.6634
Prevalence                      0.3194         0.15271           0.5279
Detection Rate                  0.2232         0.07947           0.3140
Detection Prevalence            0.3175         0.31788           0.3646
Balanced Accuracy               0.7801         0.61951           0.7438
```

Figure 20 Confusion Matrix SVM

6. RECOMMENDATIONS

- Economy Plus travel class customers need to be targeted specially to reduce the churn, because 50% of the economy plus customers are either Passive or Detractors
- Female customer churn is high because 50% of the female customers are either passive or detractors, less than 25% are promoters and hence they require special attention
- Large number of Business Travelers are Promoters and around 75% of Personal Travelers are Detractors, because of which Personal Travelers & Business Travelers need to be treated alike to increase NPS for Personal travelers
- Southeast should focus on improving the experience of Customers with Blue Airline status to improve NPS
- Flight cancellations result in decrease in NPS and southeast should focus on reducing flight cancellations, as 41% of customers who experienced flight cancellations are detractors and only 18% are promoters
- ‘FlyFast Airways’ & ‘Going North Airlines’ are the worst performing partners with a negative NPS. Hence Southeast should focus on improving the experience of customers with these 2 partners
- Around 50% of the customers < 18 years old & >54 years old are detractors and hence customers of these age groups require special targeting
- Customers with flight status silver and on business travel should continue to receive the same experience to maintain them as promoters

References

1. Plots - https://ggplot2.tidyverse.org/reference/index.html
2. SVM - https://en.wikipedia.org/wiki/Support-vector_machine
3. SVM Model - https://www.rdocumentation.org/packages/kernlab/versions/0.9-29/topics/ksvm
4. SVM Explanation - Machine Learning- A Probabilistic Perspective by kevin murphy