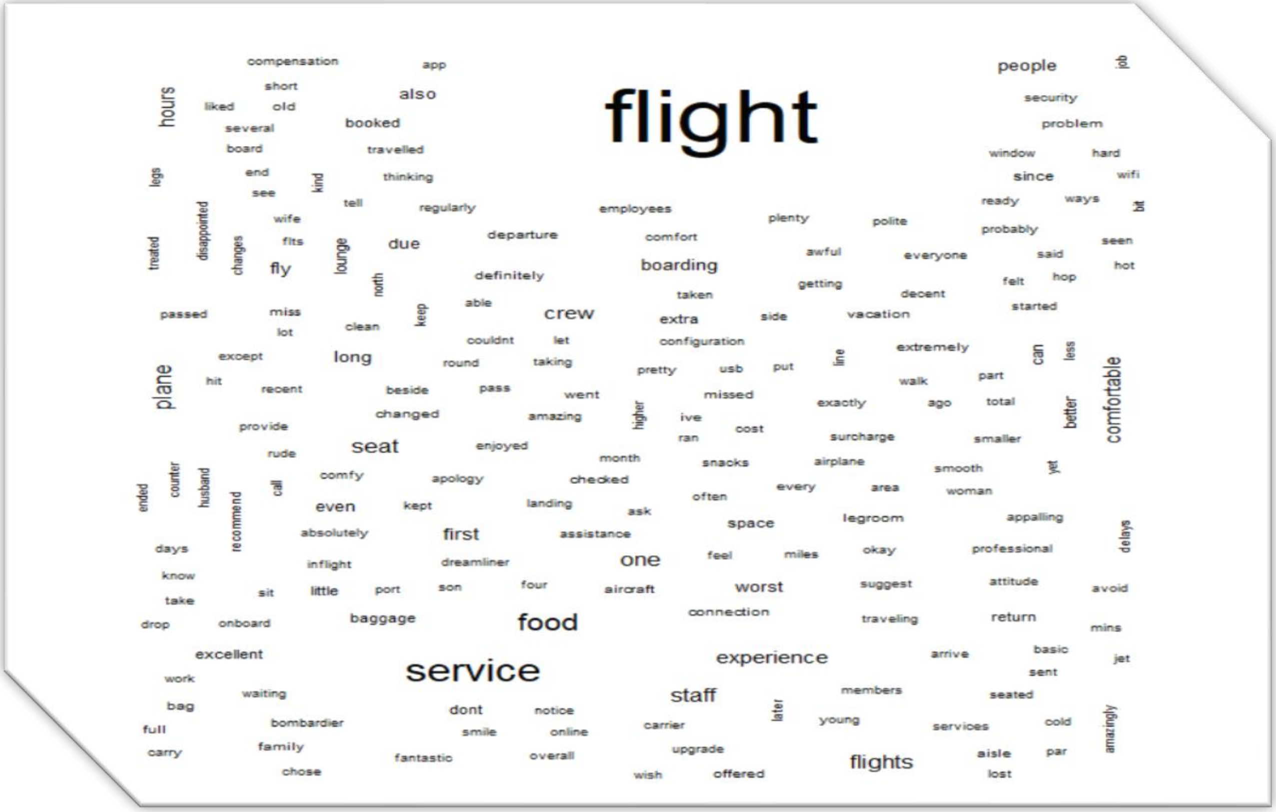
IST 687 PROJECT

SECTION M003: GROUP 1

AIRLINE DATABASE PROJECT



Aditya Sai Yerraguntla | Justin Lim | Nikita Dongare
Surjit Singh | Tejas Patil | Xiwei Shen

TABLE OF CONTENTS

INTRODUCTION

The airline industry is a mammoth with components of customer service and logistics playing a crucial role in customer satisfaction. The number of airlines operating has increased substantially with some major airlines and many smaller airlines. The smaller airlines usually work as partner airlines with the larger airlines to increase connectivity between airports. The different airlines have been operating on smaller profit margins due to stiff competition. This has led to customer retention as one of the primary ways to maintain or increase the airline profits.

Major airlines have loyalty programs to reward customers for frequently using their airline for travel. The loyalty programs help retain customers but also costs the airlines substantial amount of money in terms of lost revenue. There is no hard evidence on the extent to which the loyalty programs attract new customers or retain existing customers.

A recent survey on customer satisfaction by Southeast airlines has determined that there is a high attrition by the customers who use their airline and the partner airlines. The company is reassessing their loyalty program model and its ability to retain customers. The company is trying to use the customer satisfaction survey to determine the parameters which might help improve the customer satisfaction and retain customers.

The survey contains 10,282 records across 32 different attributes trying to capture the customer's information and satisfaction. It was determined that the customer attrition metric was a lagging indicator to determine the causes for a customer to switch airline preference. The Net promoter score (NPS) was indicated to be a good indicator to determine the customers overall satisfaction with the airlines. NPS asks a customer to rate on a scale of 1-10 (1 – Highly Unlikely; 10-Highly Likely) the likelihood to recommend the airlines to another person. The results of the NPS can help determine if the customer will be promoter, passive or detractor when it comes to "word of the mouth" promotion.

Our analysis of the data determines factors such as age, gender, delays in arrival etc. as key predictors for determining Net promoter Score. The analysis involves descriptive measures, visualizations and predictive models to build suggestions for Southeast airlines to improve customer satisfaction.

BUSINESS QUESTION

- Is the current airline customer reward system efficient for holding back customers?
- How age and gender of different customers affect the satisfaction score of the model?
- How does origin and destination location of a flight affect satisfaction score of the model?
- Will satisfaction score of customers will be impacted by different travel dates and month in which the journey take place?
- What impact travel type (business, personal etc.), status of the airline and class by which customer travel have on the satisfaction score?
- How is satisfaction affected by the time of the travel? Which can be obtained from delays of flights and actual flight time.
- Different customers spend differently while traveling. Using these spending habits can we get any insights about the satisfaction of a customer?
- What are the different factors that count for the NPS score? What makes a customer a promotor and what makes them a detractor?

DATA PREPROCESSING

The data contains 10282 rows and 32 attributes before the start of the data preprocessing. We check for the structures of the different attributes to get an understanding of the data that is available. The data contains several NA values which are spread across different attributes in the dataset. The NA's were narrowed down to 3 different attributes

1. Arrival.Delay.in.Minutes
2. Departure.Delay.in.Minutes
3. Flight.Time.in.Minutes

The NA's in the arrival and departure delay were present because of the instances where the flights were cancelled. There were rows where the flight was not cancelled but NA's were still present. The Flight time in minutes has the same NA issue as the other two attributes. The instances where the flight was cancelled we replace the attributes with Median because it doesn't affect the skewness of the data (We replace with median even though the flight was cancelled because NA will not allow the model building to take place). When the flight was cancelled then we notice that there are NA's in departure delay but not in arrival delay so we replace the NA's in departure delay with the delay in arrival delay. To deal with the instances where the flight time instances had NA's we built a regression model between distance and flight time. The predicted values from the regression model were used to replace the missing values of Flight time.

```
#Running codes for clean test of Prep code

library(jsonlite)



#import the dataset

df <- jsonlite::fromJSON(dataset)

setwd("C:/Users/hp/Desktop/IST 687/PROJECT")



#save the data

write.csv(df, file = "AirplaneData.csv")

AD <- read.csv("AirplaneData.csv")
```

```
#Viewing and understanding the data

View(AD)

dim(AD) #Variables = 33, Observations = 10282

str(AD)

summary(AD)

#Checking the quartiles and mean/median of numerical variables

summary(AD[,sapply(AD, class) == "integer"])


#Number of variables of diffrent variables

sum(sapply(AD, class) == "factor")

sum(sapply(AD, class) == "integer")

sum(sapply(AD, class) == "numeric")
```

```
######################### Treating NA's #########################

#check for the Na's in every variable

apply(is.na(AD),2,sum)

a <- which(is.na(AD$Arrival.Delay.in.Minutes))

length(a) #NA's in Arrival

b <- which(is.na(AD$Departure.Delay.in.Minutes))

length(b) #NA's in Departure

c <- which(is.na(AD$Flight.time.in.minutes))

length(c) #NA's in Flight Time

d <- which(AD$Flight.cancelled == "Yes")

length(d) #Cancelled Flight

p <- a[(a %in% d)] #NA's in Arrival delay where flight.cancelled = YES
```

```
length(p) #223/242 NA values where flight was cancelled
q <- b[(b %in% d)] #NA's in Departure delay where flight.cancelled = YES
length(q) #218/218 NA values where flight was cancelled
r <- c[(c %in% d)] #NA's in flight.time where flight.cancelled = YES
length(r) #223/242 NA values where flight was cancelled


#For flight.cancelled = YES, replacing the NA's with median
#Median because, best estimate for missing value when data is skewed
AD$Departure.Delay.in.Minutes[q] <- round(median(AD$Departure.Delay.in.Minutes, na.rm = TRUE))
AD$Arrival.Delay.in.Minutes[p] <- round(median(AD$Arrival.Delay.in.Minutes, na.rm = TRUE))
AD$Flight.time.in.minutes[r] <- round(median(AD$Flight.time.in.minutes, na.rm = TRUE))
```

```
#NA's where flight.cancelled = NO
a1 <- which(is.na(AD$Arrival.Delay.in.Minutes))
length(a1)
b1 <- which(is.na(AD$Departure.Delay.in.Minutes))
length(b1)
c1 <- which(is.na(AD$Flight.time.in.minutes))
length(c1)


#Replace arrival delay NA's by corresponding departure delay values
#Reason: Flight departs late by x min, arrives late by x min
AD$Arrival.Delay.in.Minutes[a1] <- AD$Departure.Delay.in.Minutes[a1]
```

```
#Regresing Flight distance to predict NA's in flight time
#Time variable without NA's
time <- AD$Flight.time.in.minutes[-c1]

#Corresponding Flight distances
dist <- AD$Flight.Distance[-c1]

#Distance Values for which time prediction needs to be done
test <- AD$Flight.Distance[c1]

#Linear model: X = Distance, Y = Time
trendline <- lm(time ~ dist)
```

```
#summary: R-Squared 0.94, pvalue < 0.0001
summary(trendline)

#Predicting the values
pred <- round(predict(trendline, data.frame(dist = test)))
#replace flight time NA with pred
AD$Flight.time.in.minutes[c1] <- pred

#NA for likely hood to recommend
View(AD[which(is.na(AD$Likelihood.to.recommend)),])

#See all the flights running from this origin to destination
```

```
View(AD[AD$Destination.City == "Salt Lake City, UT" & AD$Origin.City == "Rock Springs, WY", ])



#we found that all the airline are operated by partner code "OO", Northwestern Airlines

#So, we replace the NA with the median of those flights

Salt_Rock  <- AD[AD$Destination.City == "Salt Lake City, UT" & AD$Origin.City == "Rock Springs, WY", ]

AD$Likelihood.to.recommend[2498] <- median(Salt_Rock$Likelihood.to.recommend, na.rm = TRUE)

########################## Cleaning other data ##########################

#Removing the state code from cities

AD$Destination.City <- gsub(",.*", "", AD$Destination.City)

AD$Origin.City <- gsub(",.*", "", AD$Origin.City)
```

```
#Deleting FreeText - Not needed in EDA or Models

#But saving this dataset for Textmining

  AD_Cleaned <- AD[, -which(colnames(AD) == 'freeText')]



#checking new data

  dim(AD_Cleaned)

  View(AD_Cleaned)

############################## Saving the cleaned data ###########################

#save the data

write.csv(AD_Cleaned, file = "Cleaned_Data.csv")

write.csv(AD, file = "FreeText.csv")
```

Exploratory Data Analysis

In this section, we have plotted scatter plots using GGPLOT2 to compare the relationship between different attributes and likelihood to recommend. We also used GGMAP to plot flight routes for some of the partner airlines to check if the routes are concentrated at one particular destination or origin point.

```
#installing ggplot2 packages
install.packages("ggplot2")
library(ggplot2)

#Plotting the United States map using ggplot
usMap <- borders("state", colour="grey", fill="white")
ggplot() + usMap

#Plotting all the flight routes available for all the partner airlines using
olat,olong,dlat,dlong attributes on to the US map
allUSA <-ggplot()+usMap +
  geom_curve(data=AD_Cleaned,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             col="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=AD_Cleaned,
             aes(x=olong, y=olat),
             colour="red",
             size=1.5) +
  geom_point(data=AD_Cleaned,
             aes(x=dlong, y=dlat),
             colour="blue") +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.ticks=element_blank(),
        plot.title=element_text(hjust=0.5, size=12))
allUSA

summary(AD_Cleaned$Partner.Name)
#Plot1: FlyFast Airways
#Creating a dataframe which contains observations related to FlyFast Airways
FlyFast <- subset(AD_Cleaned, Partner.Name=="FlyFast Airways Inc.")

#Plotting the flight routes for the FlyFast airlines
```

```
FlyF<-ggplot()+usMap +
  geom_curve(data=FlyFast,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             col="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=FlyFast,
             aes(x=olong, y=olat),
             colour="red",
             size=1.5) +
  geom_point(data=FlyFast,
             aes(x=dlong, y=dlat),
             colour="blue") +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.ticks=element_blank(),
        plot.title=element_text(hjust=0.5, size=12)) + ggtitle("Plot of flight routes for FlyFast
airlines")
FlyF
```

```
#Plot2: Going North Airlines
#Creating a dataframe which contains observations related to GoingNorth Airlines
Goingnorth <- subset(AD_Cleaned, Partner.Name=="GoingNorth Airlines Inc.")

#Plotting the flight routes for the GoingNorth airlines
GoingN<-ggplot()+usMap+
  geom_curve(data=Goingnorth,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             color="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=Goingnorth,
             aes(x=olong, y=olat),
             colour="blue",
             size=1.5) +
  geom_point(data=Goingnorth,
             aes(x=dlong, y=dlat),
             colour="red") +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
      axis.ticks=element_blank(),
      plot.title=element_text(hjust=0.5, size=12))+ ggtitle("Plot of flight routes for Going
North airlines")
GoingN
```

```
#Plot3: Cheapseats Airlines

#Creating a dataframe which contains observations related to Cheapseats Airlines
Cheapseats <- subset(AD_Cleaned, Partner.Name=="Cheapseats Airlines Inc.")

#Plotting the flight routes for the Cheapseats airlines
cheapseatsmap<-ggplot()+usMap+
  geom_curve(data=Cheapseats,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             color="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=Cheapseats,
             aes(x=olong, y=olat),
             colour="blue",
             size=1.5) +
  geom_point(data=Cheapseats,
             aes(x=dlong, y=dlat),
             colour="red") +
  theme(axis.line=element_blank(),
      axis.text.x=element_blank(),
      axis.text.y=element_blank(),
      axis.title.x=element_blank(),
      axis.title.y=element_blank(),
      axis.ticks=element_blank(),
      plot.title=element_text(hjust=0.5, size=12))+ ggtitle("Plot of flight routes for Cheapseats
airlines")
cheapseatsmap

#Plot4: Cool&Young Airlines
#Creating a dataframe which contains observations related to Cool&Yound Airlines
coolyoung <- subset(AD_Cleaned, Partner.Name=="Cool&Young Airlines Inc.")

#Plotting the flight routes for cool&yound airlines
coolnyoungmap<-ggplot()+usMap+
  geom_curve(data=coolyoung,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             color="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=coolyoung,
          aes(x=olong, y=olat),
          colour="blue",
          size=1.5) +
  geom_point(data=coolyoung,
          aes(x=dlong, y=dlat),
          colour="red") +
  theme(axis.line=element_blank(),
      axis.text.x=element_blank(),
      axis.text.y=element_blank(),
      axis.title.x=element_blank(),
      axis.title.y=element_blank(),
      axis.ticks=element_blank(),
      plot.title=element_text(hjust=0.5, size=12))+ ggtitle("Plot of flight routes for
Cool&Young airlines")
coolnyoungmap
```

```
#Plot5: EnjoyFlying Air services
#Creating a dataframe which contains observations related to EnjoyFlying Air services
enjoyflying <- subset(AD_Cleaned, Partner.Name=="EnjoyFlying Air Services")

#Plotting the flight routes for EnjoyFlying Air services
enjoyflyingmap<-ggplot()+usMap+
  geom_curve(data=enjoyflying,
          aes(x=olong, y=olat, xend=dlong, yend=dlat),
          color="black",
          size=.5,
          curvature=0.2) +
  geom_point(data=enjoyflying,
          aes(x=olong, y=olat),
          colour="blue",
          size=1.5) +
  geom_point(data=enjoyflying,
          aes(x=dlong, y=dlat),
          colour="red") +
  theme(axis.line=element_blank(),
      axis.text.x=element_blank(),
      axis.text.y=element_blank(),
      axis.title.x=element_blank(),
      axis.title.y=element_blank(),
      axis.ticks=element_blank(),
      plot.title=element_text(hjust=0.5, size=12))+ ggtitle("Plot of flight routes for
EnjoyFlying airlines")
enjoyflyingmap
```

```
#Plot 6: Northwest Business Airlines
#Creating a dataframe which contains observations related to Northwest Airlines
```

```
Northwest <- subset(AD_Cleaned, Partner.Name=="Northwest Business Airlines Inc.")
#Plotting the flight routes for Northwest Airlines
Northwestmap<-ggplot()+usMap+
  geom_curve(data=Northwest,
             aes(x=olong, y=olat, xend=dlong, yend=dlat),
             color="black",
             size=.5,
             curvature=0.2) +
  geom_point(data=Northwest,
             aes(x=olong, y=olat),
             colour="blue",
             size=1.5) +
  geom_point(data=Northwest,
             aes(x=dlong, y=dlat),
             colour="red") +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.ticks=element_blank(),
        plot.title=element_text(hjust=0.5, size=12))+ ggtitle("Plot of flight routes for Northwest
Business airlines")
Northwestmap
```

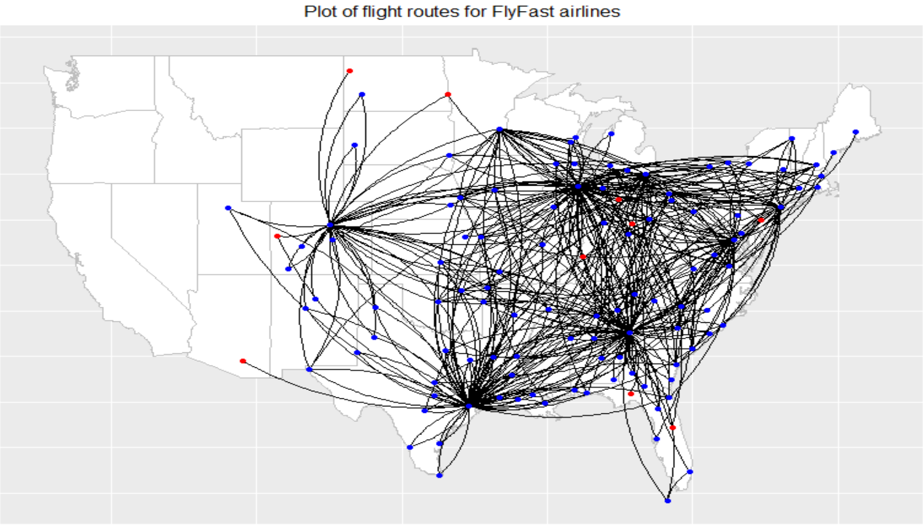


Figure 1: Plot of flight routes for FlyFast airlines

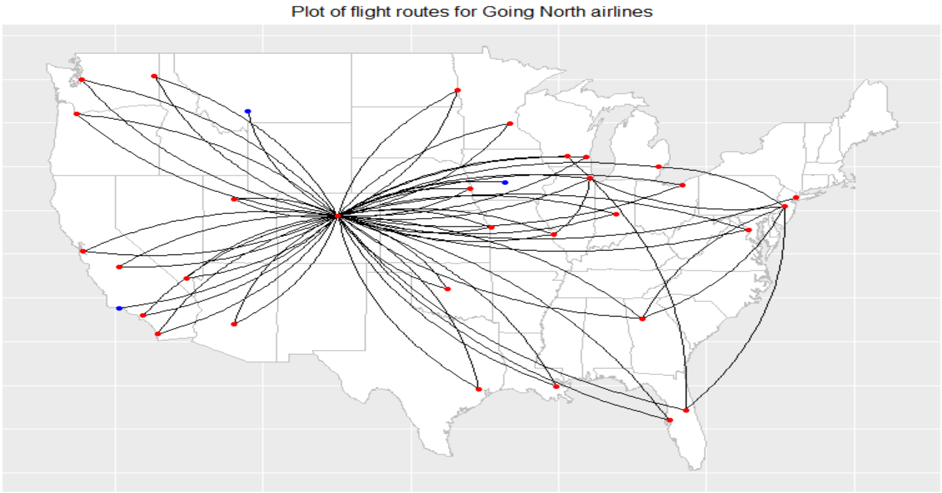


Figure 2: Plot of flight routes for GoingNorth airlines

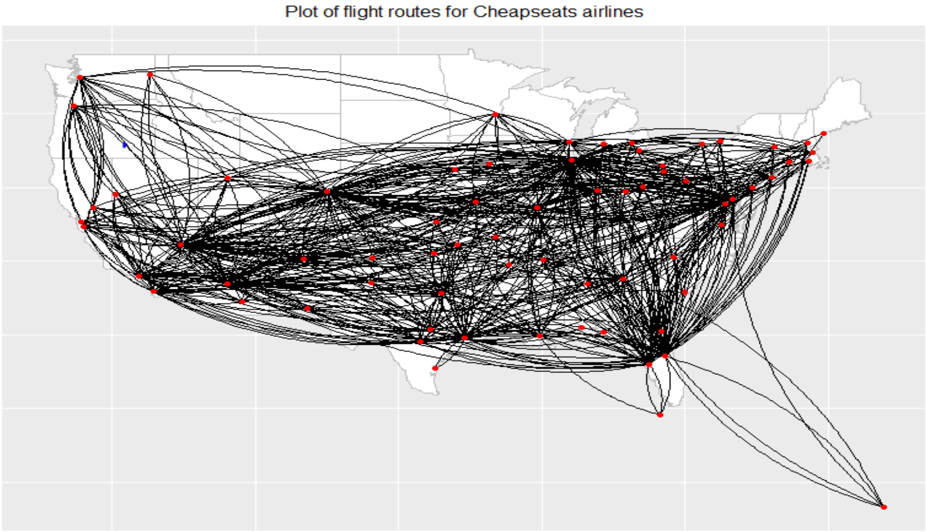


Figure 3: Plot of flight routes for Cheapseats airlines

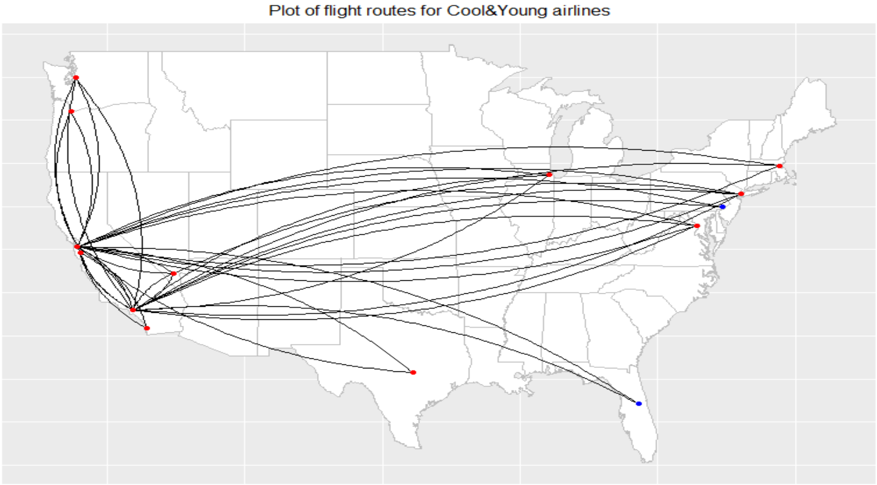


Figure 4: Plot of flight routes for Cool&Young airlines

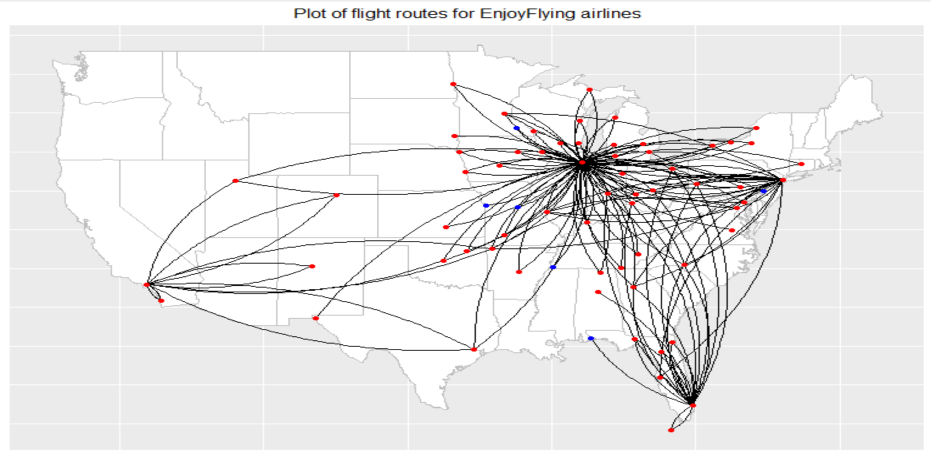


Figure 5: Plot of flight routes for EnjoyFlying Airlines

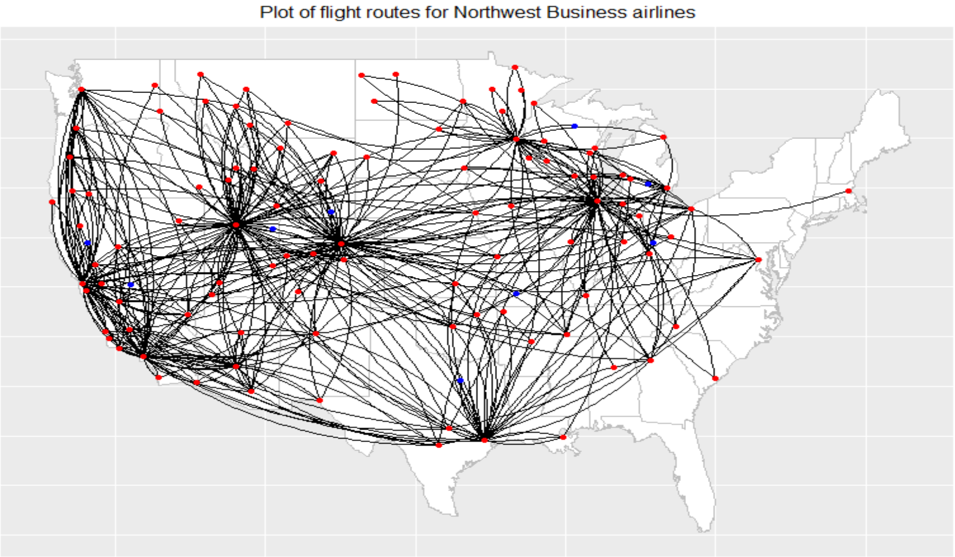


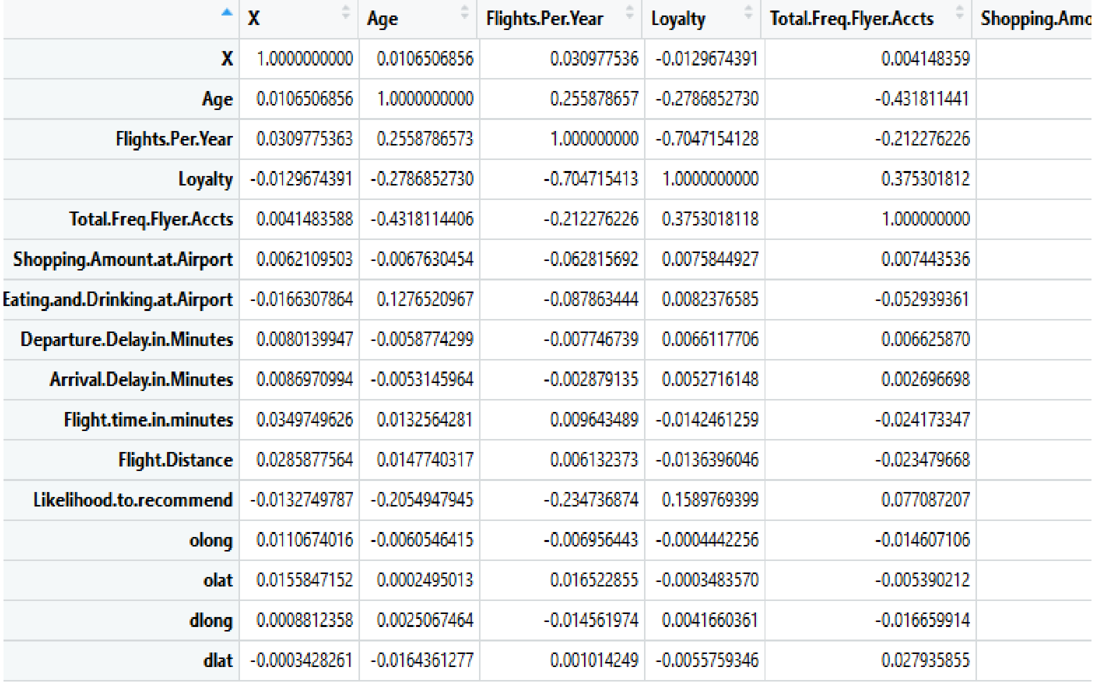
Figure 6: Plot of flight routes for Northwest Business Airlines

From the maps, we can observe that airline routes for some airlines are concentrated at a particular point whereas airline routes for some airlines are spread out across the United States map. The first plot depicts the airline routes for FlyFast airlines which indicates that FlyFast airline routes are mostly spread out towards the east coast. The second plot shows the airline routes for Going North airlines which are concentrated in the mid-west region. Similarly, the plot 4 and 5 shows the flight routes for Cool&Young airlines and Enjoy Flying airlines which have routes concentrated in the east and west coast respectively. The flighroutes for Cheapseats and Northwest airlines are concentrated but spread out across the United States.

Exploratory Data Analysis (EDA) for data science is something that gives detailed, zoomed and classified information about the dataset. Without EDA, we only get the superficial knowledge of dataset. We will use this EDA for selecting the appropriate variables in our dataset.

Let’s start with correlation test for all the numerical variables in our model.

```
############# Correlation Plot ################
#Checking the correlation between the numeric data
  Numeric_AD <- AD_Cleaned[sapply(AD_Cleaned, is.numeric)]
  corrTable <- cor(Numeric_AD)
  View(corrTable)
  corrplot(corrTable, type = "upper", tl.cex=0.5)

#MULTICOLLINEARITY TEST:
#Arrival.delay is highly correlated to departure.delay
#Flight.distance is highly correlated to flight.time
```



	X	Age	Flights.Per.Year	Loyalty	Total.Freq.Flyer.Accts	Shopping.Amo
X	1.0000000000	0.0106506856	0.030977536	-0.0129674391	0.004148359	
Age	0.0106506856	1.0000000000	0.255878657	-0.2786852730	-0.431811441	
Flights.Per.Year	0.0309775363	0.2558786573	1.000000000	-0.7047154128	-0.212276226	
Loyalty	-0.0129674391	-0.2786852730	-0.704715413	1.0000000000	0.375301812	
Total.Freq.Flyer.Accts	0.0041483588	-0.4318114406	-0.212276226	0.3753018118	1.000000000	
Shopping.Amount.at.Airport	0.0062109503	-0.0067630454	-0.062815692	0.0075844927	0.007443536	
Eating.and.Drinking.at.Airport	-0.0166307864	0.1276520967	-0.087863444	0.0082376585	-0.052939361	
Departure.Delay.in.Minutes	0.0080139947	-0.0058774299	-0.007746739	0.0066117706	0.006625870	
Arrival.Delay.in.Minutes	0.0086970994	-0.0053145964	-0.002879135	0.0052716148	0.002696698	
Flight.time.in.minutes	0.0349749626	0.0132564281	0.009643489	-0.0142461259	-0.024173347	
Flight.Distance	0.0285877564	0.0147740317	0.006132373	-0.0136396046	-0.023479668	
Likelihood.to.recommend	-0.0132749787	-0.2054947945	-0.234736874	0.1589769399	0.077087207	
olong	0.0110674016	-0.0060546415	-0.006956443	-0.0004442256	-0.014607106	
olat	0.0155847152	0.0002495013	0.016522855	-0.0003483570	-0.005390212	
dlong	0.0008812358	0.0025067464	-0.014561974	0.0041660361	-0.016659914	
dlat	-0.0003428261	-0.0164361277	0.001014249	-0.0055759346	0.027935855	

Fig. Correlation Table

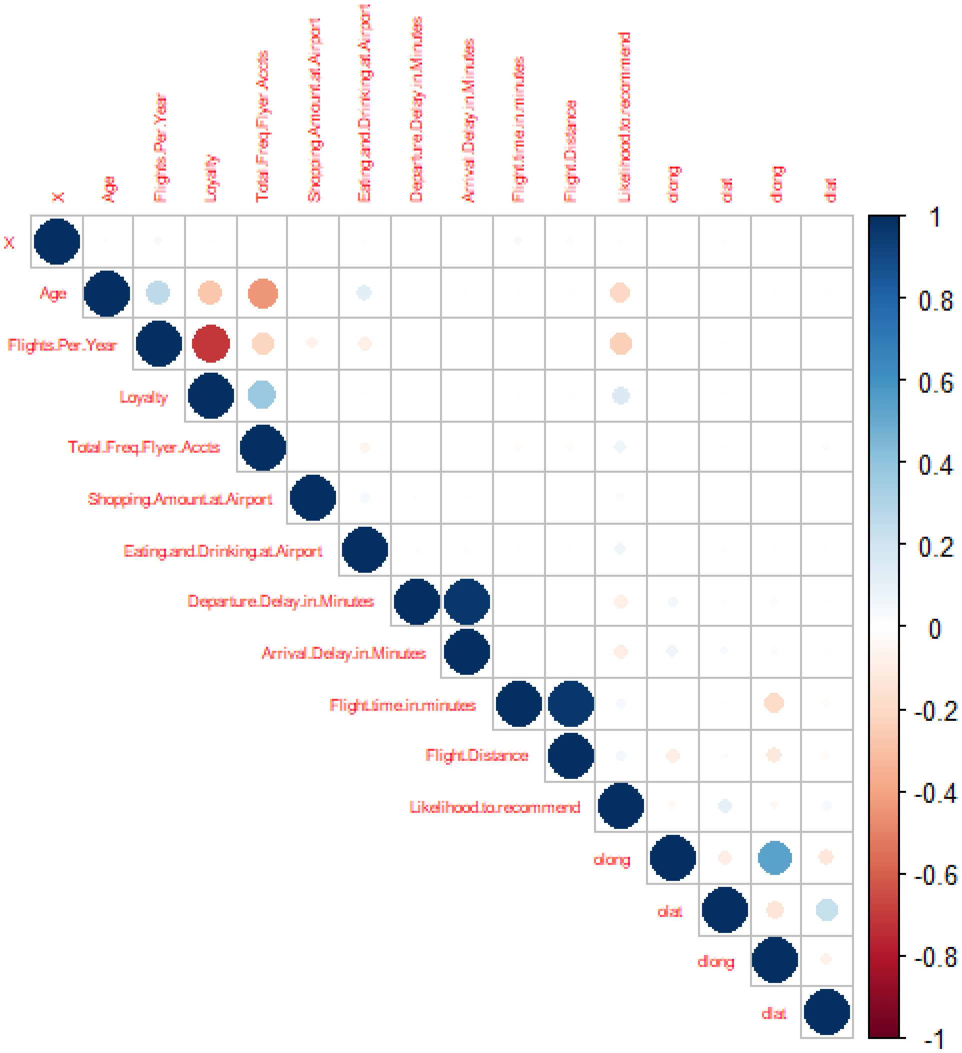


Fig. Correlation Plot

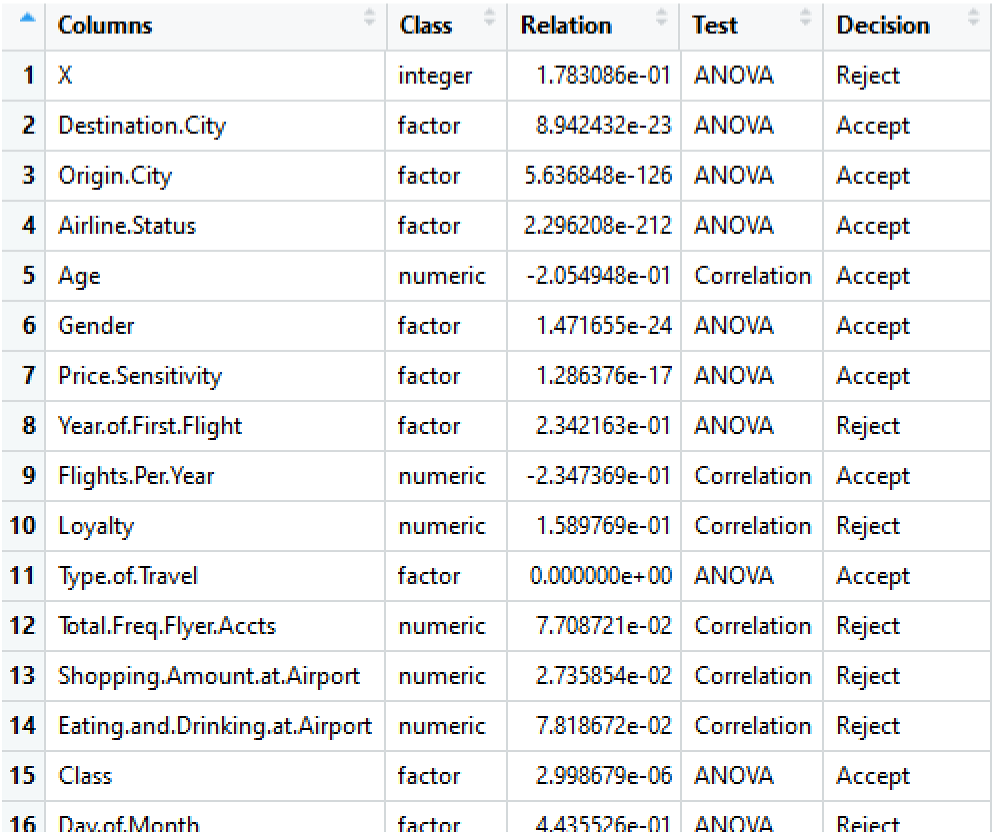
After the correlation test, we zoom in on the relation of our dependent variable with all the independent variables. For numerical variables, we use the correlation again and for the factor variables, we use the **ANOVA** test.

ANOVA test is used to find the relation between factor variables and numerical variables. Since, our dependent variable being numerical, ANOVA helps us to know if the **mean recommendation score** of every level in a categorical variable is different(**variance**) from the other level in the same variable. It works on the principle of **confidence interval**. In this test, we use the confidence level of **95%.**

```
################### Correlation & ANOVA ######################

#Creating a table for association of independent variables with dependent
  AttrSelect <- data.frame(Columns = colnames(AD_Cleaned))
  for (i in 1:length(colnames(AD_Cleaned))){
    AttrSelect$Class[i] <- class(AD_Cleaned[ , i])
  }
  AttrSelect$Relation = 0
  View(AttrSelect)
#Checking the correlation for numeric variables
#Checking the ANOVA test value for factor variables
  for(i in 1:length(AD_Cleaned)){
    if(AttrSelect[i,2] == "numeric"){
      AttrSelect$Relation[i]=cor(AD_Cleaned[,i], AD_Cleaned$Likelihood.to.recommend)
      AttrSelect$Test[i]<-"Correlation"
    }
    else{
      AOVTest <- aov(AD_Cleaned$Likelihood.to.recommend ~ AD_Cleaned[,i])
      AttrSelect$Relation[i] = unlist(summary(AOVTest))["Pr(>F)1"]
      AttrSelect$Test[i]<-"ANOVA"
        }
  }
```

So, we created a dataframe that will contain the results of ANOVA test and correlation test. Next step is to decide which variable to accept and which ones to reject. So, we keep the confidence level to 95% for ANOVA test and since the correlation values are not that high, we keep the correlation parameters flexible for selection.

```
#Deciding the cutoffs for feature selection
  AttrSelect$Decision = "Reject"
  AttrSelect$Decision[AttrSelect$Test == "ANOVA" & AttrSelect$Relation <= 0.05] = "Accept"
  AttrSelect$Decision[AttrSelect$Test == "Correlation" &
                    (AttrSelect$Relation>=0.10  & AttrSelect$Realtion <=0.80 )]  = "Accept"
  AttrSelect$Decision[AttrSelect$Test == "Correlation" &
                    (AttrSelect$Relation<=-0.10  & AttrSelect$Relation >=-0.80 )]  = "Accept"

  View(AttrSelect)
```



	Columns	Class	Relation	Test	Decision
1	X	integer	1.783086e-01	ANOVA	Reject
2	Destination.City	factor	8.942432e-23	ANOVA	Accept
3	Origin.City	factor	5.636848e-126	ANOVA	Accept
4	Airline.Status	factor	2.296208e-212	ANOVA	Accept
5	Age	numeric	-2.054948e-01	Correlation	Accept
6	Gender	factor	1.471655e-24	ANOVA	Accept
7	Price.Sensitivity	factor	1.286376e-17	ANOVA	Accept
8	Year.of.First.Flight	factor	2.342163e-01	ANOVA	Reject
9	Flights.Per.Year	numeric	-2.347369e-01	Correlation	Accept
10	Loyalty	numeric	1.589769e-01	Correlation	Reject
11	Type.of.Travel	factor	0.000000e+00	ANOVA	Accept
12	Total.Freq.Flyer.Accts	numeric	7.708721e-02	Correlation	Reject
13	Shopping.Amount.at.Airport	numeric	2.735854e-02	Correlation	Reject
14	Eating.and.Drinking.at.Airport	numeric	7.818672e-02	Correlation	Reject
15	Class	factor	2.998679e-06	ANOVA	Accept
16	Day.of.Month	factor	4.435526e-01	ANOVA	Reject

Fig. View of "AttrSelect" dataframe

Chi-Square test:

For this project we did not use the chi-square test for feature selection, but we still performed it to check the independence of factor independent variables from one another.

```
############## Chi-Square Test #################
#Checking the association between independent factor variables
 FactorD <- AD_Cleaned[sapply(AD_Cleaned, is.factor)]
 dim(FactorD)
 ChiResult <- matrix(nrow = 16, ncol = 16)
 for (i in 1:length(FactorD)) {
   for (j in 1:length(FactorD)) {
     CTest <- chisq.test(table(FactorD[ , i], FactorD[ , j]))
     ChiResult[i,j] <- unlist(CTest)["p.value"]
   }
 }
 ChiResult <- as.data.frame(ChiResult)
 colnames(ChiResult) <- colnames(FactorD)
 rownames(ChiResult) <- colnames(FactorD)
 View(ChiResult)
```

Outlier treatment:

We have created two functions in R, to treat the numerical variables if the contain the outliers. First function treat the outliers by upper fence and lower fence value and the second function treat the outliers by median value.

Lower fence Upper fence Method:

```
#Outlier Treatment by Winsorizing
Out_Treat_W = function(x){
  Q1 = quantile(x, 0.25)
  Q3 = quantile(x, 0.75)
  IQR = Q3 - Q1
  LC = Q1 - 1.5*IQR
  UC = Q3 + 1.5*IQR
  Out_Count = sum(x > UC | x < LC)
  UOut <- which(x > UC)
  LOut <- which(x < LC)
  for (i in 1:length(UOut)){
    x[UOut[i]] <- UC
  }
  for (i in 1:length(LOut)){
    x[LOut[i]] <- LC
  }
```

```
  OutInfo = list(TotalOutliers = Out_Count, LCutoff = LC, UCutoff = UC)
  print(OutInfo)
  return(x)
}
```

Median imputation method:

```
#Outlier Treatment by Median
Out_Treat_M = function(x){
  Q1 = quantile(x, 0.25)
  Q3 = quantile(x, 0.75)
  IQR = Q3 - Q1
  LC = Q1 - 1.5*IQR
  UC = Q3 + 1.5*IQR
  Out_Count = sum(x > UC | x < LC)
  TOut <- which(x > UC | x < LC)
  for (i in 1:length(TOut)){
    x[TOut[i]] <- median(x)
  }
  OutInfo = list(TotalOutliers = Out_Count, LCutoff = LC, UCutoff = UC)
  print(OutInfo)
  return(x)
}
```

Graphical data analysis:

```
#Destination City
```

```
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Destination.City, y = AD_Cleaned$Likelihood.to.recommend)) +
  geom_point(stat = "identity") + theme(axis.text.x=element_text(angle=90))
#Can't tell anything, Flat distribution, Constant median

#Origin City
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Origin.City, y = AD_Cleaned$Likelihood.to.recommend)) +
  geom_point(stat = "identity") + theme(axis.text.x=element_text(angle=90))
#Can't tell anything, Flat distribution, Constant median
```

```
#Destination state
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Destination.State, y = AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point") +
  theme(axis.text.x=element_text(angle=90))
#Origin state
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Origin.State, y = AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point") +
  theme(axis.text.x=element_text(angle=90))
#Can't tell anything, Flat distribution, Constant median
```

```
#Airline Status
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Airline.Status, y = AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#We can clearly see that Silver, Gold and Platinum status people rate the airlines very high as compared to the Blue status
```

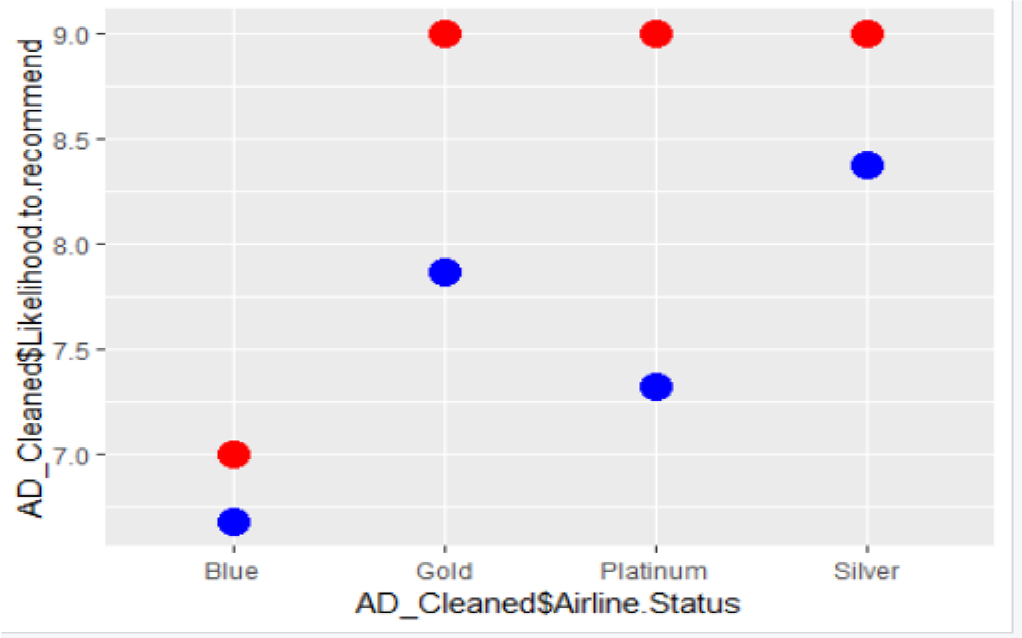


Fig. Airline Status vs Recommendation Score

```
#Age
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Age, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", geom = "line") +
  stat_summary(fun.y = "mean", colour = "blue", geom = "line")
#All values are from 20-85, so no chance of outlier
#It can be seen that the middle age people give pretty high ratings to the airlines,
#so, if we classify the age variable in 3 categories, it might help
```

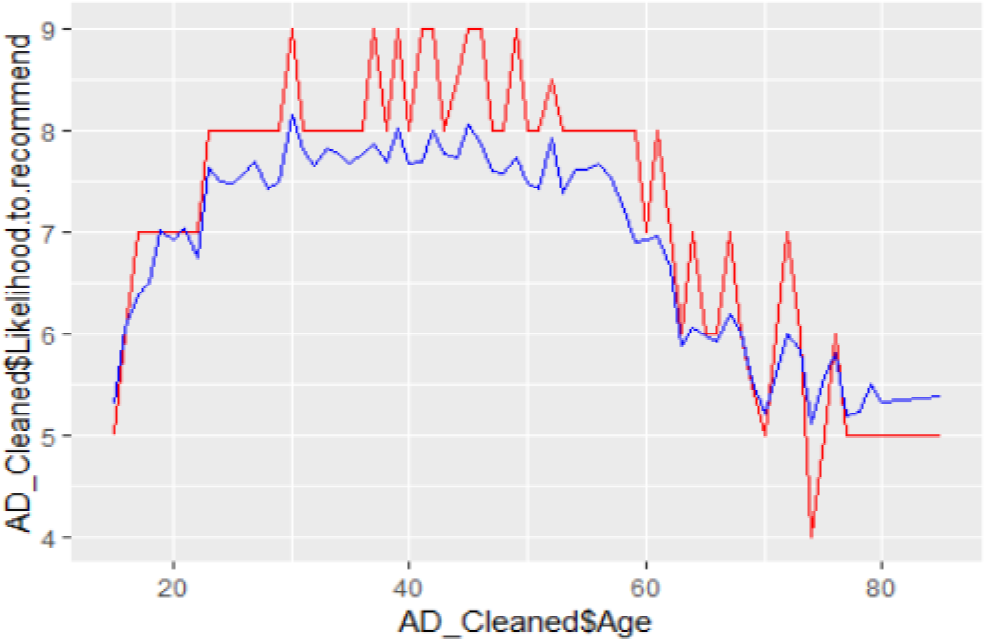


Fig. Age vs Recommendation Score

```
#Gender
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Gender, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#So, male customers give higher ratings than the female customers
#male data skewed
#female data close to normal distribution
```

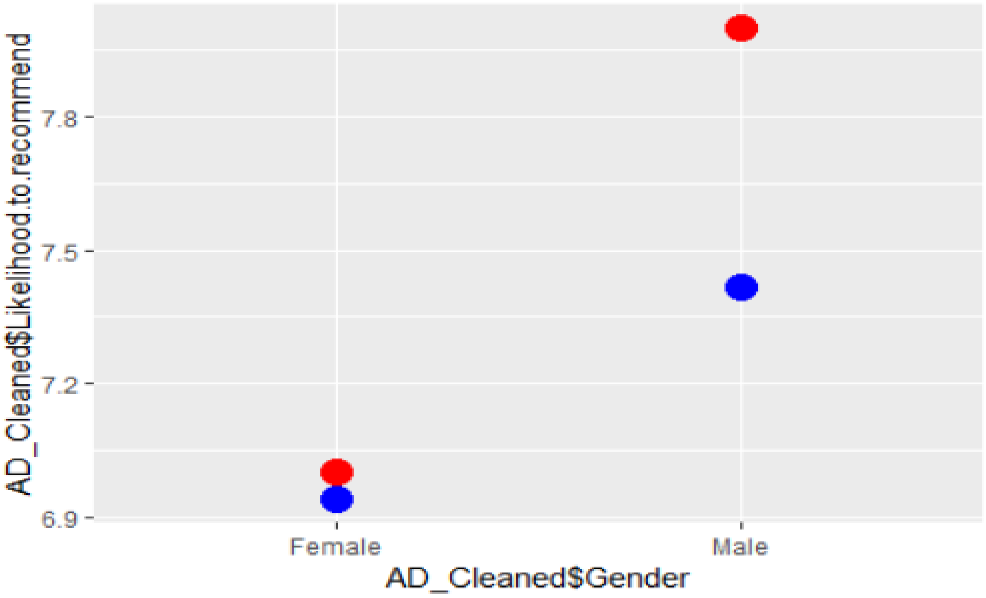


Fig. Gender vs Recommendation Score

```
#Price Sensitivity
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Price.Sensitivity, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#We can see a decreasing trend in ratings with the increase in price sensitivity
#No, outliers as values take from 0 to 5 only.
```

```
#Year of first flight
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Year.of.First.Flight, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Flat distribution, Constant median
```

```
#Flights per year
boxplot(AD_Cleaned$Flights.Per.Year, horizontal = T)
```

```
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Flights.Per.Year, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", geom = "line") +
  stat_summary(fun.y = "mean", colour = "blue", geom = "line")
#Decreasing trend with increase in Flights per year
```

```
#Loyalty
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Loyalty, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", geom = "line")
boxplot(AD_Cleaned$Loyalty, horizontal = T)
#We can try using the group by function, create 2 groups; -1 - 0, 0 - 1
#Negative vs Positive
#No outliers, but data is slightly skewed
#Very random Recommendation Score but slight increasing trend
```

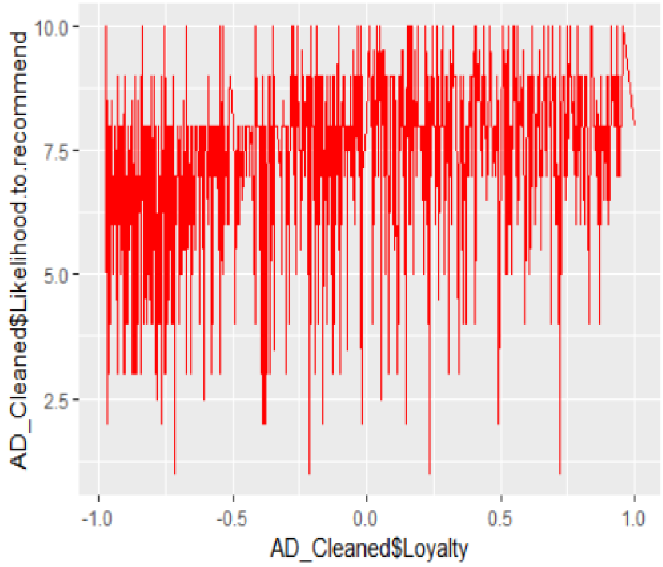


Fig. Loyalty vs R. Score

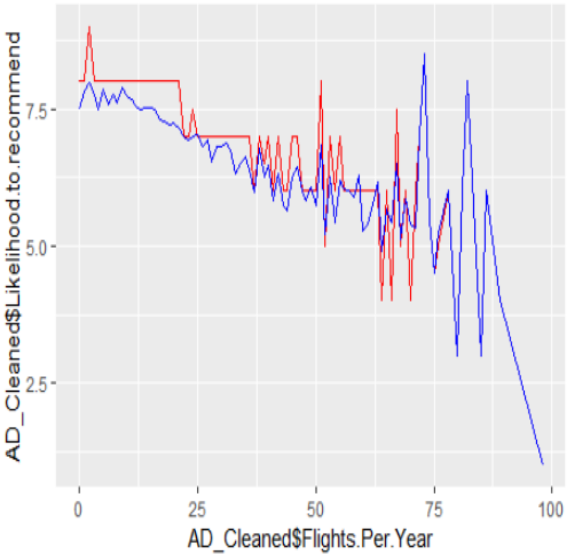


Fig. Flights per year vs R. Score

```
#Type of travel
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Type.of.Travel, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Huge diffrence between categories of 3 people, personal travel people are definitely detractors
```

```
#Total Freq Flyer Accounts
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Total.Freq.Flyer.Accts, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
boxplot(AD_Cleaned$Total.Freq.Flyer.Accts, horizontal = T)
summary(AD_Cleaned$Total.Freq.Flyer.Accts)
#Values are from 0 to 10, but after 5, data is scattered
```

```
tempD$Total.Freq.Flyer.Accts <- Out_Treat_W(tempD$Total.Freq.Flyer.Accts)
```

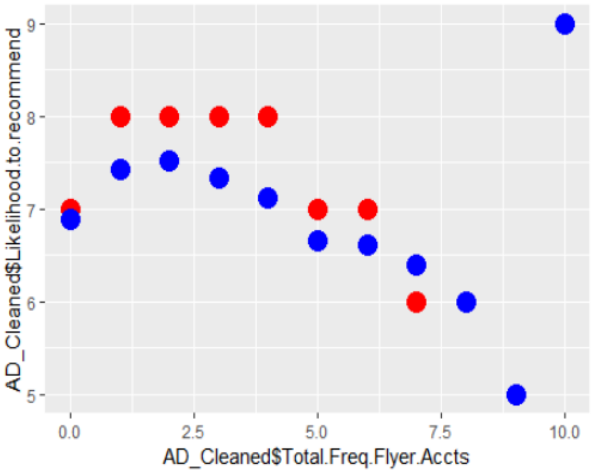


Fig. Total Accounts vs R. Score

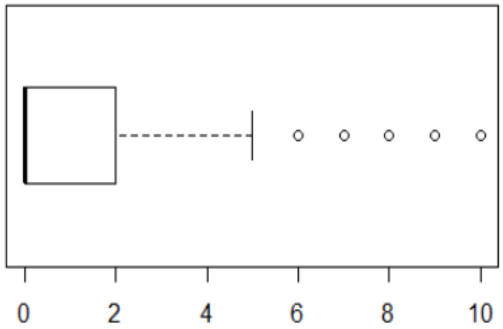


Fig. Boxplot of Totol.Freq.Flyer.Accounts

```
#Shopping
hist(AD_Cleaned$Shopping.Amount.at.Airport)
boxplot(AD_Cleaned$Shopping.Amount.at.Airport, horizontal = T)
#In this attribte the outliers should definitely be treated
#People who don't do shopping at the airport are too much
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Shopping.Amount.at.Airport, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", geom = "line")
#Flat distribution, No use in linear regression
#No point in treating outliers
```

```
#Eating and Drinking
hist(AD_Cleaned$Eating.and.Drinking.at.Airport)
boxplot(AD_Cleaned$Eating.and.Drinking.at.Airport, horizontal = T)
#In this attribte the outliers should definitely be treated
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Eating.and.Drinking.at.Airport, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", geom = "line")
```

```
#Winsorized outliers treatment used
```

```
AD_Cleaned$Eating.and.Drinking.at.Airport <-
Out_Treat_W(AD_Cleaned$Eating.and.Drinking.at.Airport)
```

```
#Class
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Class, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Eco-Plus gives less Recommendation Score
```

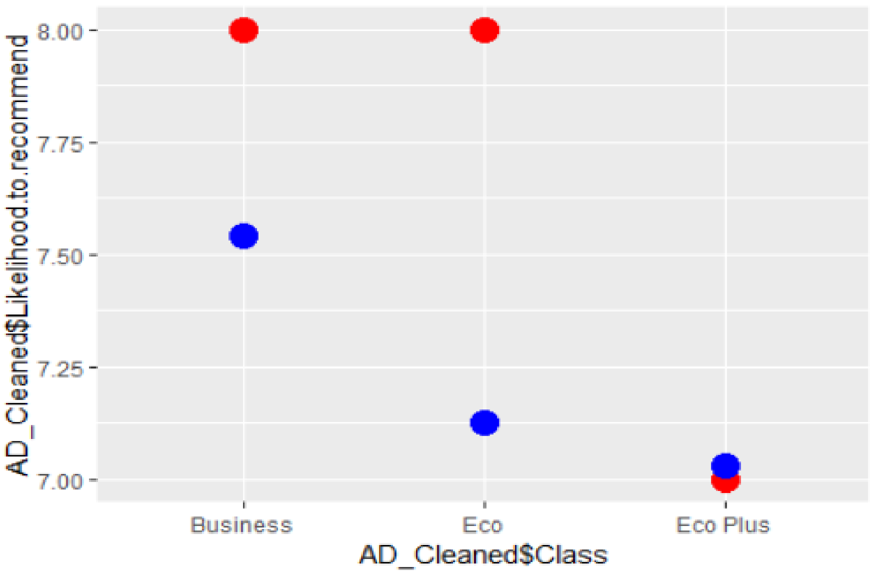


Fig. Class vs Recommendation Score

```
#Day of Month
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Day.of.Month, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Flat distribution

#Flight Date
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Flight.date, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Flat distrbution

#Scheduled depart hour
```

```
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Scheduled.Departure.Hour, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Flat distribution
```

```
#Depart delay min
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Departure.Delay.in.Minutes, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
#Messed up distribution

#Arrival delay min
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Arrival.Delay.in.Minutes, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
#Messed up distribution
```

```
#Flight Cancelled
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Flight.cancelled, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point") +
  stat_summary(fun.y = "mean", colour = "blue", size = 5, geom = "point")
#Flight canceled people rate less
#Interesting fact: The diffrence is very less
```

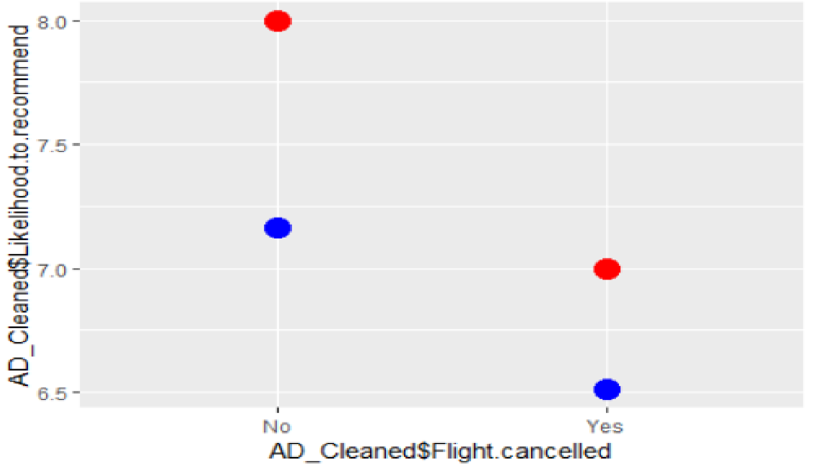


Fig. Flight Cancelled vs R. Score

```
#Flight Time
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Flight.time.in.minutes, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
```

```
#can divide into 2 parts, 0-200(flat distribution), 200-400(messy distribution)

#Flight dist
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$Flight.Distance, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
#might be good because upward trend
```

```
#olong
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$olong, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
```

```
############### olat
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$olat, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
############### dlong
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$dlong, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
############### dlat
ggplot(data = AD_Cleaned, mapping = aes(x = AD_Cleaned$dlat, y =
AD_Cleaned$Likelihood.to.recommend)) +
  stat_summary(fun.y = "median", colour = "red", size = 5, geom = "point")
#dlat can be an excellent predictor
#But it does make sense to include only dlat in model
#It will make the model overfitted
```

#EDA - WITHIN INDEPENDENT VARIABLES

```
#Flight per year VS Loyalty
ggplot(AD_Cleaned, aes(AD_Cleaned$Flights.Per.Year, AD_Cleaned$Loyalty)) +
  geom_point() #Strong Negative correlation
```

```
#olong vs dlong
ggplot(AD_Cleaned, aes(AD_Cleaned$olong, AD_Cleaned$dlong)) +
  geom_point() #Strong positive correlation
```

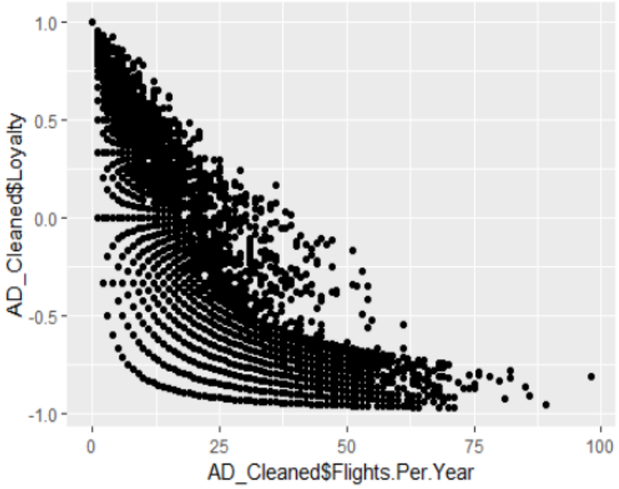


Fig. Flights.Per.Year vs Loyalty

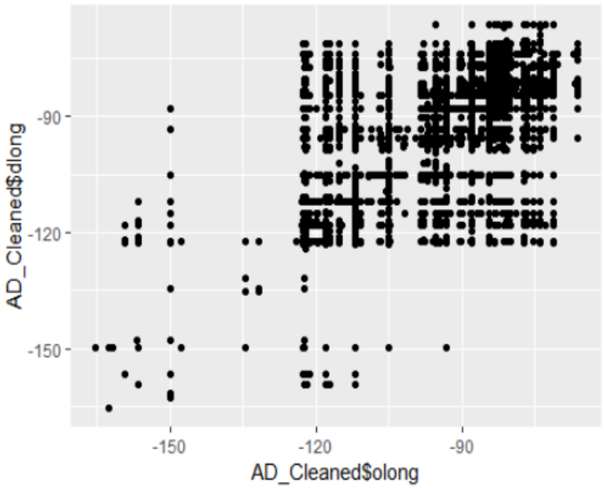


Fig. olong vs dlong

```
#Arrival delay vs departure delay
ggplot(AD_Cleaned, aes(AD_Cleaned$Arrival.Delay.in.Minutes,
AD_Cleaned$Departure.Delay.in.Minutes)) +
  geom_point()
#0.97 - Strong positive correlation
```

```
#Distance vs flight time
ggplot(AD_Cleaned, aes(AD_Cleaned$Flight.Distance, AD_Cleaned$Flight.time.in.minutes)) +
  geom_point()
#0.97 - Strong positive correlation
```

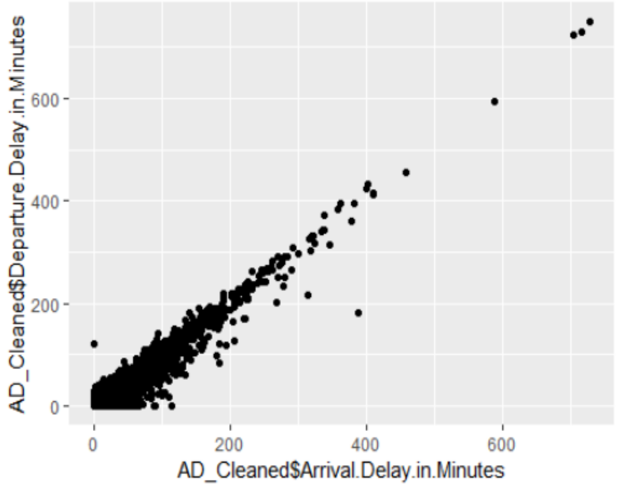


Fig. Arrival delay vs Departure delay

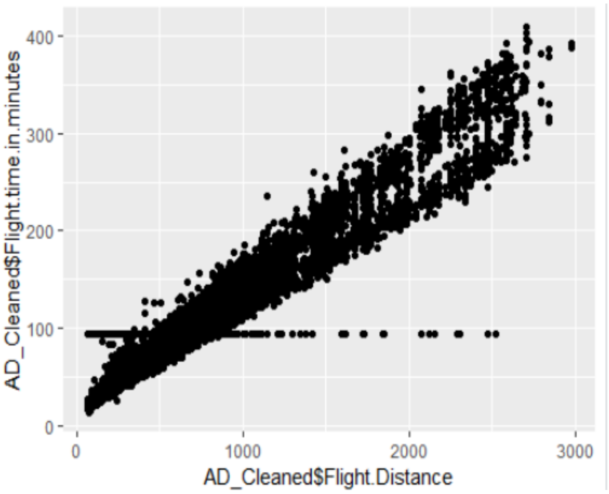


Fig. Flight distance vs Flight time

MODELLING TECHNIQUES

Model 1 – Linear Model

Our aim here is to build a linear regression model that will successfully predict the whether a person will be a promoter or detractor with respect to the airline. We have cleaned the dataset and performed the exploratory data analysis. Now using all those insights, we built a linear regression model.

```
tempD <- AD_Cleaned
```

Copying the dataset into a new dataframe, ‘tempD’, because we will need this dataset again later into the project. Now based on our analysis, we will select only the significant variables for our linear model and delete the variables that do not help us in predicting.

For doing that, we use the insights that we obtained from exploratory data analysis which includes, correlation values, ANOVA values and p-values that we obtained from performing multiple iterations of linear regression model. We checked adjusted R-squared and the p-values of variables every time, after trying out diffrent permutations and combinations of attributes.

```
#Selecting the variables based on:
#1] CORRELATION TEST
#2] ANOVA TEST
#3] MULTICOLLINEARITY
#4] EXPLORATORY DATA ANALYSIS
#5] P-VALUES

#Deleting "time" related variables
#Reason: pvalue & EDA & ANOVA
tempD <- tempD[, - which(colnames(tempD) == "Year.of.First.Flight")]
tempD <- tempD[, - which(colnames(tempD) == "Flight.date")]
tempD <- tempD[, - which(colnames(tempD) == "Scheduled.Departure.Hour")]
tempD <- tempD[, - which(colnames(tempD) == "Day.of.Month")]
```

So first, we delete all the variables that have “time” factor associated with them. EDA showed us that these variables don’t really help in prediction of the recommendation score of the customer.

Next, we delete the variables which contain the information about origin city and destination city of the flight. ANOVA test showed us that these variables have satisfactory variation of recommendation score for different cities, but in EDA we observed that it does not make any sense to take a categorical variable that has so may levels. Also, the p-values were not significant.

```
#Deleting city information variables
#Reason: pvalue & EDA
tempD <- tempD[, - which(colnames(tempD) == "Destination.City")]
tempD <- tempD[, - which(colnames(tempD) == "Origin.City")]

#Reason: Multi-collinearity
tempD <- tempD[, - which(colnames(tempD) == "Departure.Delay.in.Minutes")]
tempD <- tempD[, - which(colnames(tempD) == "Flight.Distance")]
#Reason: Chi-Square Test
tempD <- tempD[, - which(colnames(tempD) == "Partner.Name")]

#Location variables
#Reason: pvalues & EDA
tempD <- tempD[, - which(colnames(tempD) == "olong")]
tempD <- tempD[, - which(colnames(tempD) == "olat")]
tempD <- tempD[, - which(colnames(tempD) == "dlong")]
tempD <- tempD[, - which(colnames(tempD) == "dlat")]

#Reason: pvalue & ANOVA TEST
tempD <- tempD[, - which(colnames(tempD) == "Flight.cancelled")]
```

Multicollinearity: This phenomenon tells us that if there is a high correlation between two numerical independent variables, then we should delete one variable (the one which is less significant) between two of them. This is because their interdependence might affect our prediction of dependent variable.

So, we observed that there is **0.97** correlation between "*Arrival.delay.in.minutes*" and "*Departure.Delay.in.Minutes*" column and **0.97** correlation between "*Flight.Distance*" and "*Flight.time.in.minutes*" column. After trying out the combinatons, we decide to delete "*Departure.Delay.in.Minutes*" and "*Flight.Distance*" variables.

Location variables: We found that "*dlat*" – destination latitude variable is a very good predictor of recommendation score, but it does not make any sense to add that variable into the model. The reason is that **correlation does not imply causation**. Sometimes, two variables might be correlated but it does not mean that one variable is causing that change in other variable.

For "*Flight.cancelled*" variable we observe from EDA that the median value for both the groups (YES/NO) does not differ much, and the p-value is also less for the variable.

Grouping:

We observed the EDA for “Age” variable and conclude that the slope for age across the distribution changes drastically twice during age increasing from 0 to 85. The recommendation score goes on increasing from **0 to 30**, then stays flat from **30 to 50** and then starts decreasing from **50 to 85**.

So, we definitely know that “Age” plays a huge factor in recommendation score. We needed to capture this effect of **changing slope**. Hence, we decided to use the concept of **interaction variable**.

First we grouped the numerical age variable into categories as **young, old and middle** aged people. Then we introduced the **dummy variables** for different age groups. This helps us in differentiating between different categories, but to capture the effect of slope (trend) in the variable, we need a variable that captures the interaction between actual age variable and dummy variables. These variables are called **interaction variables**. They are calculated as the **product** of Actual age variable and dummy variables.

```
########################## GROUPING #############################
#AGE
#Grouping Age variable by creating dummies for Young, Middle and Old people

for (i in 1:length(tempD$Age)) {
  if (tempD$Age[i] <= 30){
    tempD$Age_Young[i] = 1
  }
  else{
    tempD$Age_Young[i] = 0
  }
}
str(tempD$Age_Young)

for (i in 1:length(tempD$Age)) {
  if (tempD$Age[i] >= 50){
    tempD$Age_Old[i] = 1
  }
  else{
    tempD$Age_Old[i] = 0
  }
}

#Looking at the EDA of Age variable
#Drastic change in slope twice
#SO, creating the interaction variable (Age * Dummy)
tempD$Young_Inter <- tempD$Age * tempD$Age_Young
tempD$Old_Inter <- tempD$Age * tempD$Age_Old
```

Next, is creating a quadratic variable for “**Total.Freq.Flyer.Accts**”. This variable follows **quadratic trend** with respect to recommendation score. Hence, we introduce a **second degree** variable of “Total.Freq.Flyer.Accts”.

```
#Total.Freq.Flyer.Accts
#Total accounts dosent follow linear trend, instead it follows quadratic
#Creating a quadratic variable for total accounts

tempD$TFFA2 <- tempD$Total.Freq.Flyer.Accts * tempD$Total.Freq.Flyer.Accts
```

Dividing the data into train data and test data:

When someone builts a model, they get adjusted R-squared explaining how much of variation in Y variable is explained by X variables. But, to know if the model will work on an unknown dataset i.e. the dataset that out model has never seen before, we need to test it on a complete new dataset. So, we divide the dataset into two categories, namely **Train (70%)** and **test (30%)** of complete observations. We use “**sample_frac**” function of “**dplyr**” library which helps us in dividing the dataset equally (each level of categorical variables) into two datasets.
Then we create a new test dataset which will not have the dependent variable values. The reason is that we will be predicting those values with the help of our model.

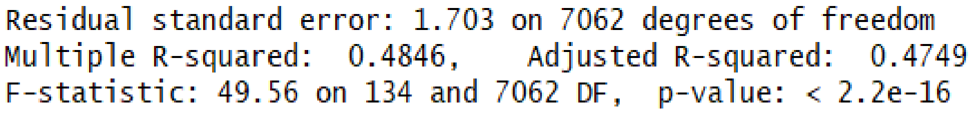
```
############################ Train -Test ############################
library(dplyr)
set.seed(121)

stratified_sample <- sample_frac(tempD, 0.7)
View(stratified_sample)

TrainD <- tempD[stratified_sample$X,]
Test <- tempD[-stratified_sample$X,]

View(TrainD)
View(Test)

TestD <- Test[,- which(colnames(Test) == "Likelihood.to.recommend")]
View(TestD)
str(TestD)
```

Model Application:

We have our final dataset, we apply the model and check the summary of the model.

```
######################## LINEAR MODEL #############################
```

```
LetsRegret <- lm(Likelihood.to.recommend ~. ,data = TrainD)
summary(LetsRegret)
```



```
Residual standard error: 1.703 on 7062 degrees of freedom
Multiple R-squared:  0.4846,    Adjusted R-squared:  0.4749
F-statistic: 49.56 on 134 and 7062 DF,  p-value: < 2.2e-16
```

Fig. Summary of the model

We checked the adjusted R-squared value of our model. So we can say that 47.49% of the variation in Y-variable is explained by our model.

Prediction:

We predict the recommendation score using the linear model we built. For that, we use the predict function and our test dataset.

```
mypred <- round(predict(LetsRegret, TestD))

Results <- data.frame(Actual = Test$Likelihood.to.recommend, Prediction = mypred)
View(Results)
```

Now, in the "Results" dataframe we have the actual recommendation scores and the predicted recommendation scores. We need to make two more columns which will contain the actual customer type and predicted customer type (Promoter or a detractor).
Note that here we did not consider the passive type of customers because lot of customers had their recommendation scores between 6 to 8, so most of the predictions would have been passive. Since, our main aim of the project is to determine the number of promoters and detractors, we classify the dependent variable into 2 categories only.
To create the column actual customer type we use for-loop;

```
for (i in 1:length(Results$Actual)) {
  if (Results$Actual[i] >= 8){
    Results$Actual_Type[i] = "Promoter"
  }
  else {
    ResultsActual_Type[i] = "Detractor"
  }
}
```

Similarly, for predicted customer type we use for-loop again;

```
for (i in 1:length(Results$Prediction)) {
  if (Results$Prediction[i] >= 8){
    Results$Prediction_Type[i] = "Promoter"
  }
  else {
    Results$Prediction_Type[i] = "Detractor"
  }
}
View(Results)
```

Calculating the Accuracy:

For predicting the accuracy, we need to calculate that out of all the predictions that we made, how many of those predictions were true.

```
Correct_V <- which(results$actual_Type==results$pred_Type)

Accuracy <- (length(Correct_V) * 100)/length(results$actual)
Accuracy
```

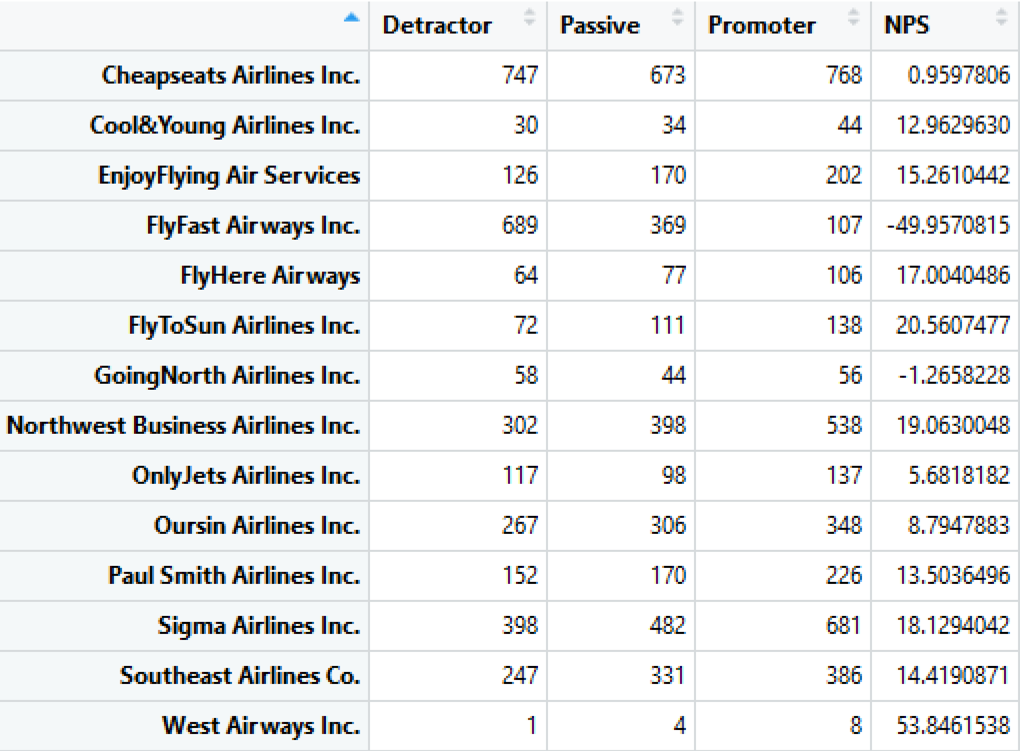
Net Promoter Score

For calculating the net promoter score, we first need to classify the data as Promoters, Passive customers and detractors. For that, we use the recommendation score and create a new variable called customer type.

```
#Adding a variable classified as Promoter, Passive and Detractor
#Based on recommendation score

for (i in 1:length(AD_Cleaned$Likelihood.to.recommend)) {
  if (AD_Cleaned$Likelihood.to.recommend[i] > 8) {
    AD_Cleaned$Customer.Type[i] = "Promoter"
  }
  else if (AD_Cleaned$Likelihood.to.recommend[i] < 7) {
    AD_Cleaned$Customer.Type[i] = "Detractor"
  }
  else {
    AD_Cleaned$Customer.Type[i] = "Passive"
  }
}
```

Now, to calculate the net promoter score, we will subtract the number of detractors from number of promoters and then divide it by the total number of customers for that particular airline.

```
#Calculating the Net-Promoter score for each partner airline

NPS <- as.data.frame.matrix(table(AD_Cleaned$Partner.Name, AD_Cleaned$Customer.Type))

NPS$NPS <- ((NPS$Promoter - NPS$Detractor)*100)/(NPS$Detractor + NPS$Passive +
NPS$Promoter)
View(NPS)
```



	Detractor	Passive	Promoter	NPS
Cheapseats Airlines Inc.	747	673	768	0.9597806
Cool&Young Airlines Inc.	30	34	44	12.9629630
EnjoyFlying Air Services	126	170	202	15.2610442
FlyFast Airways Inc.	689	369	107	-49.9570815
FlyHere Airways	64	77	106	17.0040486
FlyToSun Airlines Inc.	72	111	138	20.5607477
GoingNorth Airlines Inc.	58	44	56	-1.2658228
Northwest Business Airlines Inc.	302	398	538	19.0630048
OnlyJets Airlines Inc.	117	98	137	5.6818182
Oursin Airlines Inc.	267	306	348	8.7947883
Paul Smith Airlines Inc.	152	170	226	13.5036496
Sigma Airlines Inc.	398	482	681	18.1294042
Southeast Airlines Co.	247	331	386	14.4190871
West Airways Inc.	1	4	8	53.8461538

Fig. Net Promoter Score for all airlines

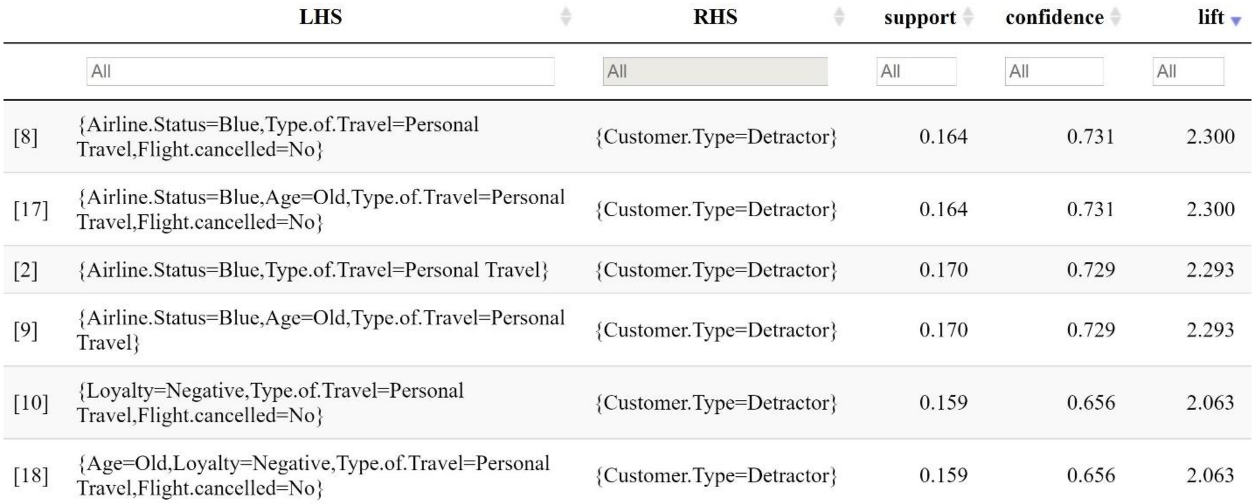
Model – 2 Association Rules Mining

Apriori algorithm provides different associations between attributes in our dataset. We tried to generate associations between the promotors and detractors with the remaining attributes of the survey. We found interesting associations between gender, class, purpose(type of travel), airline status with whether they are promoters or detractors.

Customers who fly on tickets where airline status is Blue tend to usually be detractors. Female customers on personal travel tend to be detractors. Customers whose purpose of travel is business generally tend to be promoters as long as there isn’t a huge arrival delay. All these conclusions are drawn for associations with a high lift value.

```
library(RJSONIO)
library(jsonlite)
library(tidyverse)
library(corrplot)
library(ggplot2)
library(kernlab)
library(arules)
library(arulesViz)


# Read the data
df <- read.csv('ARM_Data.csv')
View(df)


# remove states attributes
df <- df[,c(-10,-11,-12)]
str(df)


# Generate transactions and apriori algorithm
```

```
df$Price.Sensitivity <- as.factor(df$Price.Sensitivity)
trans <- as(df, "transactions")
inspect(trans)
itemFrequency(trans)
itemFrequencyPlot(trans)
ruleset <- apriori(trans,
                   parameter=list(support=0.15,confidence=0.5),
                   appearance = list(default="lhs", rhs=("Customer.Type=Detractor")))
inspect(ruleset)
inspectDT(ruleset)
plot(ruleset, method = "paracoord") # generate a plot for association rules

ruleset <- apriori(trans,
                   parameter=list(support=0.3,confidence=0.5),
                   appearance = list(default="lhs", rhs=("Customer.Type=Promoter")))
inspect(ruleset)
inspectDT(ruleset)
plot(ruleset, method = "paracoord") # generate a plot for association rules
```



	LHS	RHS	support	confidence	lift
	All	All	All	All	All
[8]	{Airline.Status=Blue,Type.of.Travel=Personal Travel,Flight.cancelled=No}	{Customer.Type=Detractor}	0.164	0.731	2.300
[17]	{Airline.Status=Blue,Age=Old,Type.of.Travel=Personal Travel,Flight.cancelled=No}	{Customer.Type=Detractor}	0.164	0.731	2.300
[2]	{Airline.Status=Blue,Type.of.Travel=Personal Travel}	{Customer.Type=Detractor}	0.170	0.729	2.293
[9]	{Airline.Status=Blue,Age=Old,Type.of.Travel=Personal Travel}	{Customer.Type=Detractor}	0.170	0.729	2.293
[10]	{Loyalty=Negative,Type.of.Travel=Personal Travel,Flight.cancelled=No}	{Customer.Type=Detractor}	0.159	0.656	2.063
[18]	{Age=Old,Loyalty=Negative,Type.of.Travel=Personal Travel,Flight.cancelled=No}	{Customer.Type=Detractor}	0.159	0.656	2.063

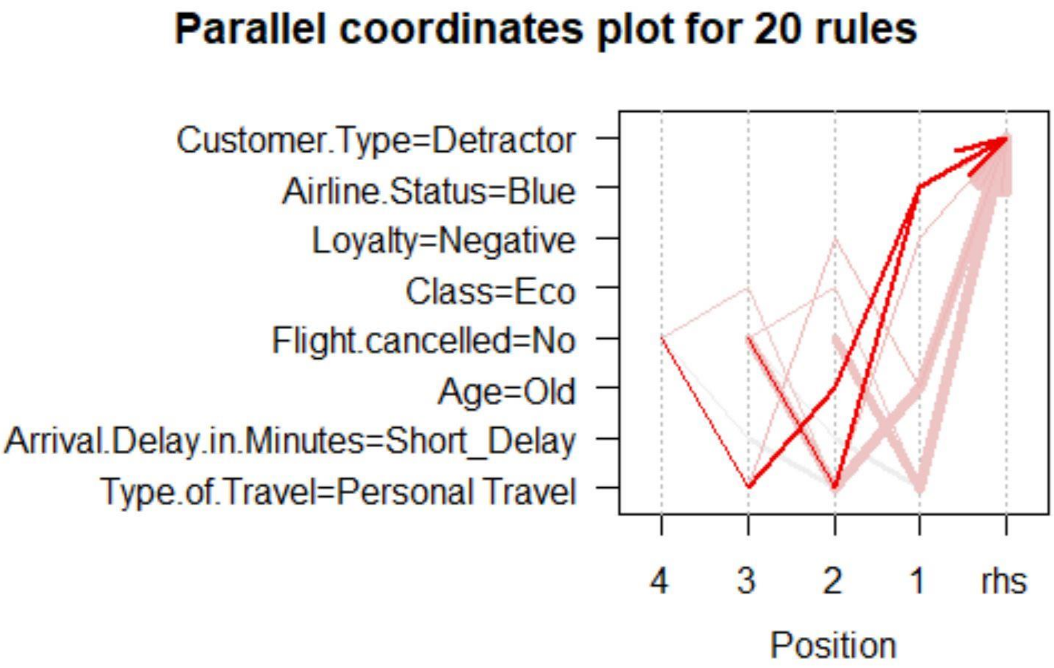
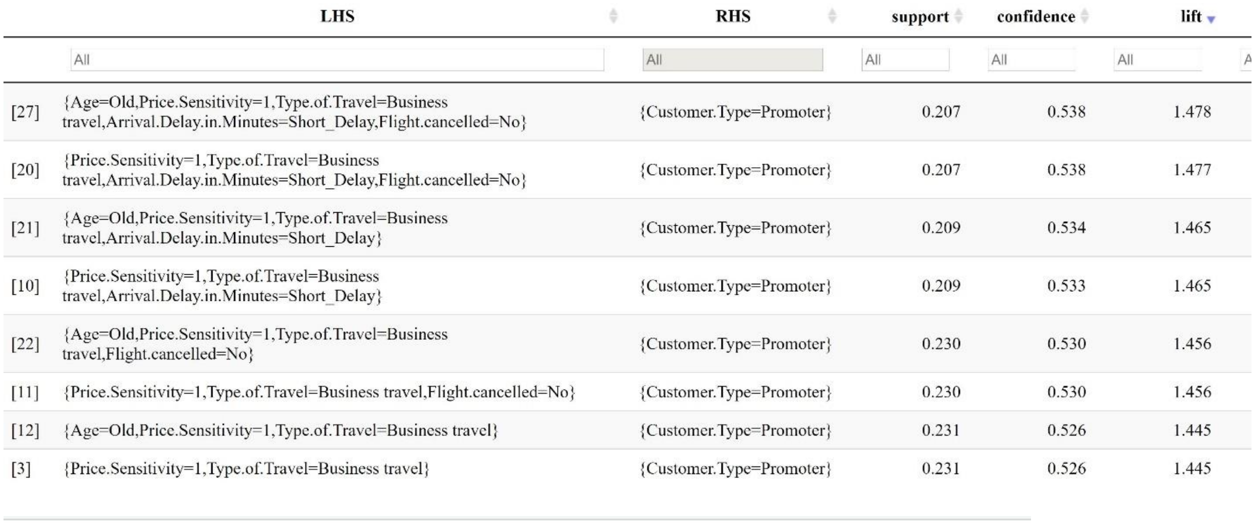
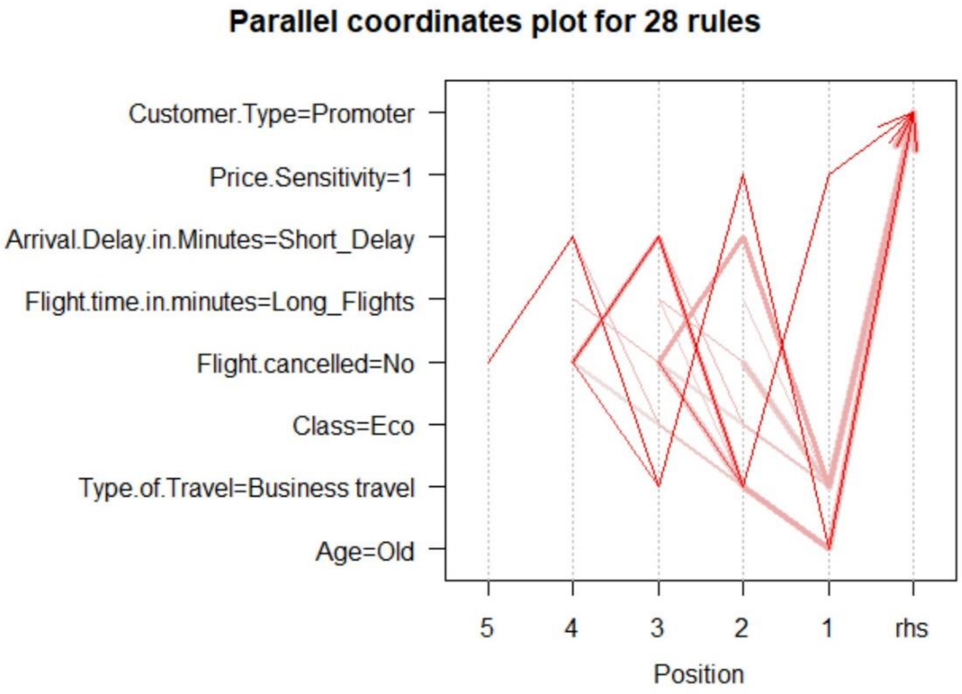


Fig: Association Rules for Detractors



	LHS	RHS	support	confidence	lift
	All	All	All	All	All
[27]	{Age=Old,Price.Sensitivity=1,Type.of.Travel=Business travel,Arrival.Delay.in.Minutes=Short_Delay,Flight.cancelled=No}	{Customer.Type=Promoter}	0.207	0.538	1.478
[20]	{Price.Sensitivity=1,Type.of.Travel=Business travel,Arrival.Delay.in.Minutes=Short_Delay,Flight.cancelled=No}	{Customer.Type=Promoter}	0.207	0.538	1.477
[21]	{Age=Old,Price.Sensitivity=1,Type.of.Travel=Business travel,Arrival.Delay.in.Minutes=Short_Delay}	{Customer.Type=Promoter}	0.209	0.534	1.465
[10]	{Price.Sensitivity=1,Type.of.Travel=Business travel,Arrival.Delay.in.Minutes=Short_Delay}	{Customer.Type=Promoter}	0.209	0.533	1.465
[22]	{Age=Old,Price.Sensitivity=1,Type.of.Travel=Business travel,Flight.cancelled=No}	{Customer.Type=Promoter}	0.230	0.530	1.456
[11]	{Price.Sensitivity=1,Type.of.Travel=Business travel,Flight.cancelled=No}	{Customer.Type=Promoter}	0.230	0.530	1.456
[12]	{Age=Old,Price.Sensitivity=1,Type.of.Travel=Business travel}	{Customer.Type=Promoter}	0.231	0.526	1.445
[3]	{Price.Sensitivity=1,Type.of.Travel=Business travel}	{Customer.Type=Promoter}	0.231	0.526	1.445



Model – 3 Support Vector Machine (SVM)

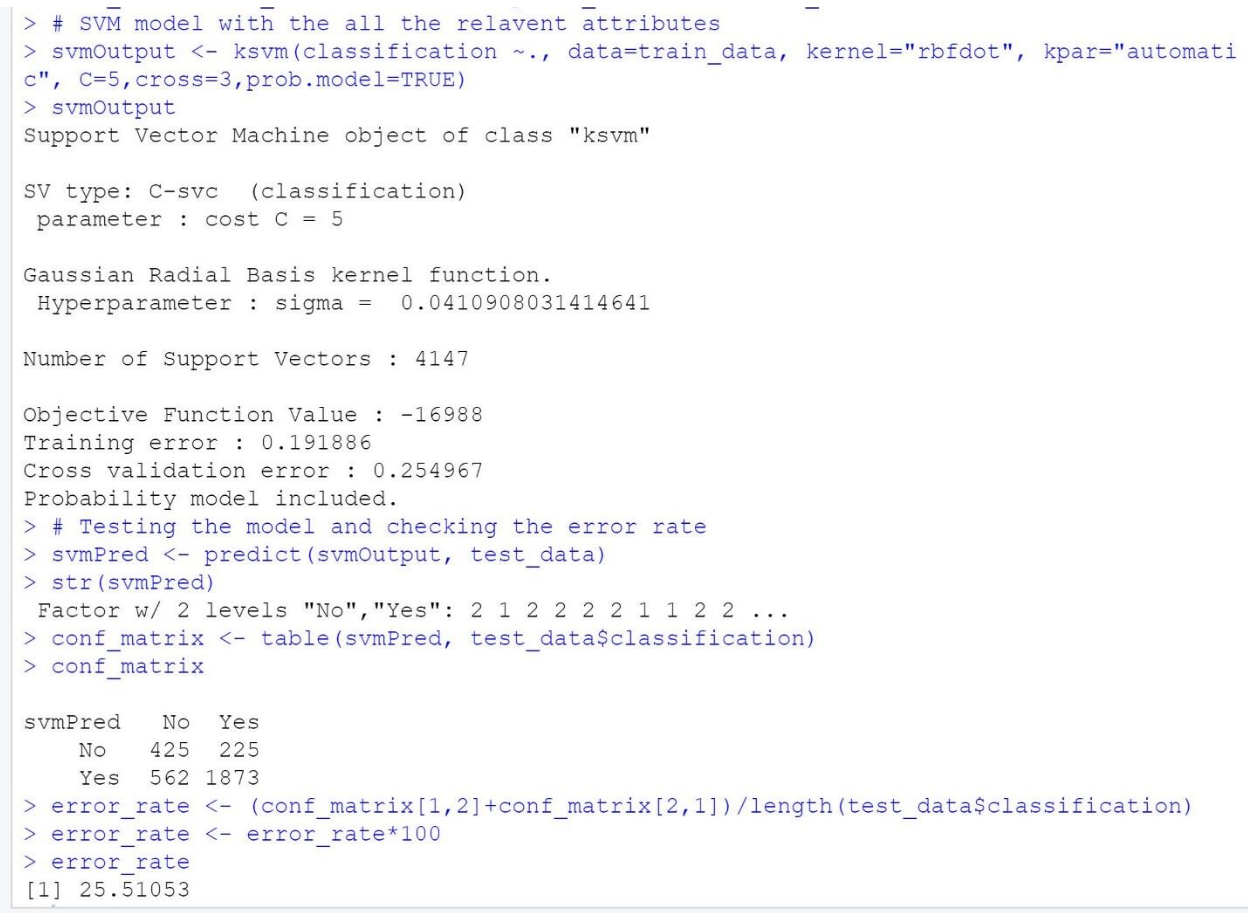
The cleaned dataset is read and prepared for building the SVM model. All the attributes determined as relevant from correlation, ANOVA tables are copied into a new dataframe. The data is split into train and test data in a 70:30 proportion. The customers are segregated into two categories Promoter or Detractor. We tried multiple different C values in order to make sure that the accuracy of the model is maximized. We are using only 2 level classification in order to improve the accuracy of the model. The models accuracy rate is 74.5%.

```
library(RJSONIO)
library(jsonlite)
library(tidyverse)
library(corrplot)
library(ggplot2)
library(kernlab)

# Reading the csv with the data and viewing it
df <- read.csv('Cleaned_Data.csv')
View(df)
str(df)

# Creating a duplicate of the original data
# Removing the attributes which were determined to be unimportant for analysis
df_duplicate <- df
df_duplicate <- df_duplicate[,c(-1,-2,-7,-11,-15,-16,-18,-21,-22,-24,-26,-28,-29,-30,-31)]
View(df_duplicate)
str(df_duplicate)
```

```
# Convert attributes to numeric type

df_duplicate$Airline.Status <- as.numeric(df_duplicate$Airline.Status)

df_duplicate$Age <- as.numeric(df_duplicate$Age)

df_duplicate$Gender <- as.numeric(df_duplicate$Gender)

df_duplicate$Price.Sensitivity <- as.numeric(df_duplicate$Price.Sensitivity)

df_duplicate$Flights.Per.Year <- as.numeric(df_duplicate$Flights.Per.Year)

df_duplicate$Loyalty <- as.numeric(df_duplicate$Loyalty)

df_duplicate$Type.of.Travel <- as.numeric(df_duplicate$Type.of.Travel)

column_names <- colnames(df_duplicate)

for (i in 1:length(column_names)){

  df_duplicate[,column_names[i]] <- as.numeric(df_duplicate[,column_names[i]])

}

str(df_duplicate)


# Converting the Likelihood to recommend score to promoter or detractor

df_duplicate$classification <- "Yes"

index <- which(df$Likelihood.to.recommend <7)

df_duplicate$classification[index] <- "No"

df_dd <- df_duplicate[,-16]

View(df_dd)


# Generating a random index for Train and test data segementation

randIndex <- sample(1:dim(df_dd)[1])

length(randIndex)
```

```
# Creating Train and Test data with 60:40 split
cutpoint_train <- floor(length(randIndex)*0.6)
train_data <- df_dd[randIndex[1:cutpoint_train],]
test_data <- df_dd[randIndex[(cutpoint_train+1):dim(df_dd)[1]],]

# SVM model with the all the relavent attributes
svmOutput <- ksvm(classification ~., data=train_data, kernel="rbfdot", kpar="automatic", C=5,cross=3,prob.model=TRUE)
svmOutput

# Testing the model and checking the error rate
svmPred <- predict(svmOutput, test_data)
str(svmPred)
conf_matrix <- table(svmPred, test_data$classification)
conf_matrix
error_rate <- (conf_matrix[1,2]+conf_matrix[2,1])/length(test_data$classification)
error_rate <- error_rate*100
error_rate

# SVM 2. Creating another svm model with fewer attributes as predictors
svmOutput_2 <- ksvm(classification ~ Airline.Status+Age+Gender+Type.of.Travel+Eating.and.Drinking.at.Airport+Class+Partner.Code, data=train_data, kernel="rbfdot", kpar="automatic", C=5,cross=3,prob.model=TRUE)
svmOutput_2

# Testing the model and checking the error rate
```

```
svmPred <- predict(svmOutput_2, test_data)

str(svmPred)

conf_matrix <- table(svmPred, test_data$classification)

conf_matrix

error_rate <- (conf_matrix[1,2]+conf_matrix[2,1])/length(test_data$classification)

error_rate <- error_rate*100

error_rate
```



```
> # SVM model with the all the relavent attributes
> svmOutput <- ksvm(classification ~., data=train_data, kernel="rbfdot", kpar="automati
c", C=5,cross=3,prob.model=TRUE)
> svmOutput
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0410908031414641

Number of Support Vectors : 4147

Objective Function Value : -16988
Training error : 0.191886
Cross validation error : 0.254967
Probability model included.
> # Testing the model and checking the error rate
> svmPred <- predict(svmOutput, test_data)
> str(svmPred)
 Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 1 1 2 2 ...
> conf_matrix <- table(svmPred, test_data$classification)
> conf_matrix

svmPred   No  Yes
    No   425  225
    Yes  562 1873
> error_rate <- (conf_matrix[1,2]+conf_matrix[2,1])/length(test_data$classification)
> error_rate <- error_rate*100
> error_rate
[1] 25.51053
```

SVM (3 Level Classification)

Another SVM model with 3 levels of classification (Promoter, Passive, Detractor) was built and tested. The accuracy reduced when compared to the binary classification model to 57.4%.

```
# SVM 3 Level Classification
df <- read.csv('ARM_Data.csv')
passive_index <- which(df$Customer.Type=="Passive")
#df$Customer.Type[passive_index] <- "Promoter"
#df$Customer.Type <- as.numeric(df$Customer.Type)
df$Customer.Type <- as.factor(df$Customer.Type)
str(df)
column_names <- colnames(df)
for(i in 1:(length(column_names)-1)){
  df[,i] <- as.numeric(df[,i])
}
str(df)


randIndex <- sample(1:dim(df)[1])
length(randIndex)


cutpoint_train <- floor(length(randIndex)*0.6)
train_data <- df[randIndex[1:cutpoint_train],]
test_data <- df[randIndex[(cutpoint_train+1):dim(df)[1]],]


# SVM 3
svmOutput <- ksvm(Customer.Type ~., data=train_data, kernel="rbfdot", kpar="automatic", C=10,cross=3,prob.model=TRUE)
```

```
svmOutput


prediction <- predict(svmOutput, test_data)

xtab <- table(test_data$Customer.Type, prediction)

xtab

success_rate <- (xtab[1,1]+xtab[2,2]+xtab[3,3])/length(test_data$Customer.Type)

success_rate <- success_rate*100

success_rate

error_rate <- 100-success_rate

error_rate
```

Sentiment Analysis

During our study towards the customer reviews for every flight, we decided to use sentimental analysis to find out whether the reviews were positive or negative. First part in this analysis is to load the data and clean data. Also, we loaded and cleaned files for positive and negative keywords.

```
#install and library the reuired packages

install.packages("tidyverse")

install.packages("tm")

install.packages("wordcloud")

install.packages('sentimentr')

install.packages("ggplot2")

library("tidyverse")

library("tm")

library("wordcloud")

library('sentimentr')

library(ggplot2)


#load the airline data file and file which contains positive and negative words

AD <- read.csv("AirplaneData.csv")

posWords <- scan("positive-words.txt", character(0), sep = "\n")

negWords <- scan("negative-words.txt", character(0), sep = "\n")
```

```
#taking freeText column from the data

AD <- AD %>% filter(freeText != "NA")
```

```
charVector <- AD$freeText
```

We used to below code to convert all the words from data to term document matrix and then we took sum of word occurrences in the comments for each unique word and plotted a word cloud for all the words. As we can see from the word cloud **“flight”** is the most used keyword in the comments

```
#creating word corpus from the data
words.vec <-VectorSource(charVector)
words.corpus <- Corpus(words.vec)
words.corpus <- tm_map(words.corpus,content_transformer(tolower))
words.corpus <- tm_map(words.corpus,removePunctuation)
words.corpus <- tm_map(words.corpus,removeNumbers)
words.corpus <- tm_map(words.corpus,removeWords,stopwords(kind="en"))

#creating termdocumentmatrix of the word corpus
tdm <- TermDocumentMatrix(words.corpus)
str(tdm)
tdm <- tdm[-1159,]
tdm <- tdm[-1340,]
tdm <- tdm[-1791,]
tdm
inspect(tdm)
```

```
#coverting tdm into a matrix and finding count of words

m <- as.matrix(tdm)

wc <- rowSums(m)

wc <- sort(wc,decreasing = TRUE)

head(wc)


#creating word cloud for of all the unique words in the dataset

cf <- data.frame(word=names(wc),freq=wc)

wordcloud(cf$word,cf$freq)
```

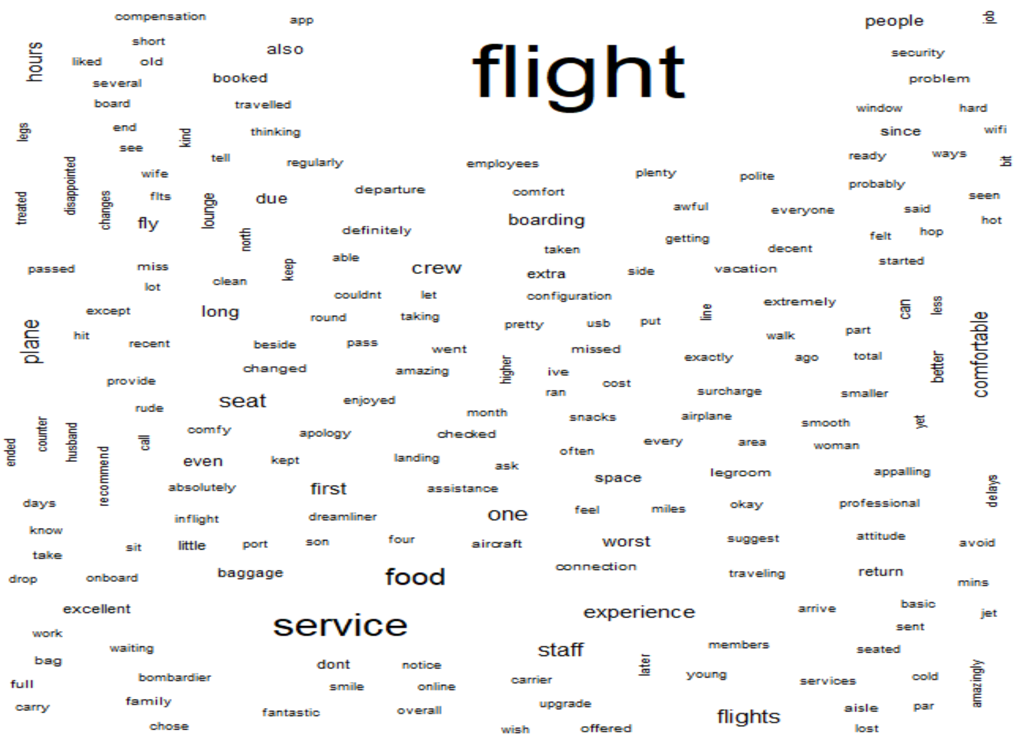


Fig: Word cloud for all the unique words

In our sentimental analysis we use the R package names **“sentimentr”,** which gives a sentiment score for a text. Using this we generated a sentiment score for all the text lines in the comments and took a sum to get one score for each review comment. Using a threshold of **0.7** we classified score greater than 0.7 as score for positive comment and as negative comment for a score less than 0.7

```
#finding positive and negative review of partner airlines using package sentimentr

senti <- AD[,c("Partner.Name","freeText")] # creating dataframe containing partner flight names and commnets

senti$freeText <- as.character(senti$freeText)

ft <- as.character(AD[1,"freeText"])



# using sentimentr packgae calculate sentiment score of each comment

ll <- c()

txt <- senti$freeText

for(val in txt){



  ll <- c(ll,sum(sentiment(val)[,sentiment]))

}

senti$sentimentScore <- ll



#setting threshold as 0.7 os score to mark comments as positive or negative

senti$recommend[which(senti$sentimentScore>=0.7)] <- "positive"

senti$recommend[which(senti$sentimentScore<0.7)] <- "negative"
```

We grouped data based on partner airline and positive and negative comments and took count of number of positive and negative comments each airline has. We then calculated proportion of the comments that is **count of comments/total number of comments in data.** We then use ggplot to plot this data on a bar graph. From this chart we can see airlines with proportion of positive and negative comments.

```
#getting total number of positive and negative comments for each airline

dfscore <- senti[,c("Partner.Name","recommend")] %>%
group_by(Partner.Name,recommend) %>% count(recommend)

colnames(dfscore) <- c("airline","comments","cnt")

# getting propotion of comments on basis of total comments

prop <- (dfscore[,"cnt"]/282)*100

dfscore$prop <- (dfscore[,"cnt"]/282)*100



#creating a plot to view number postive and negative for each airlines

dfs <- data.frame(dfscore$airline,dfscore$comments,dfscore$cnt,prop)

colnames(dfs) <- c("airline","comments","cnt","prop")

p <- ggplot(dfs, aes(fill=comments, y=prop, x=airline)) +

  geom_bar(position="dodge", stat="identity") + theme(axis.text.x = element_text(angle
= 90, hjust = 1))

p <- p + ggtitle("Airline Feedback Comments") + xlab("Airlines") + ylab("Proportion of
Comments")

p
```

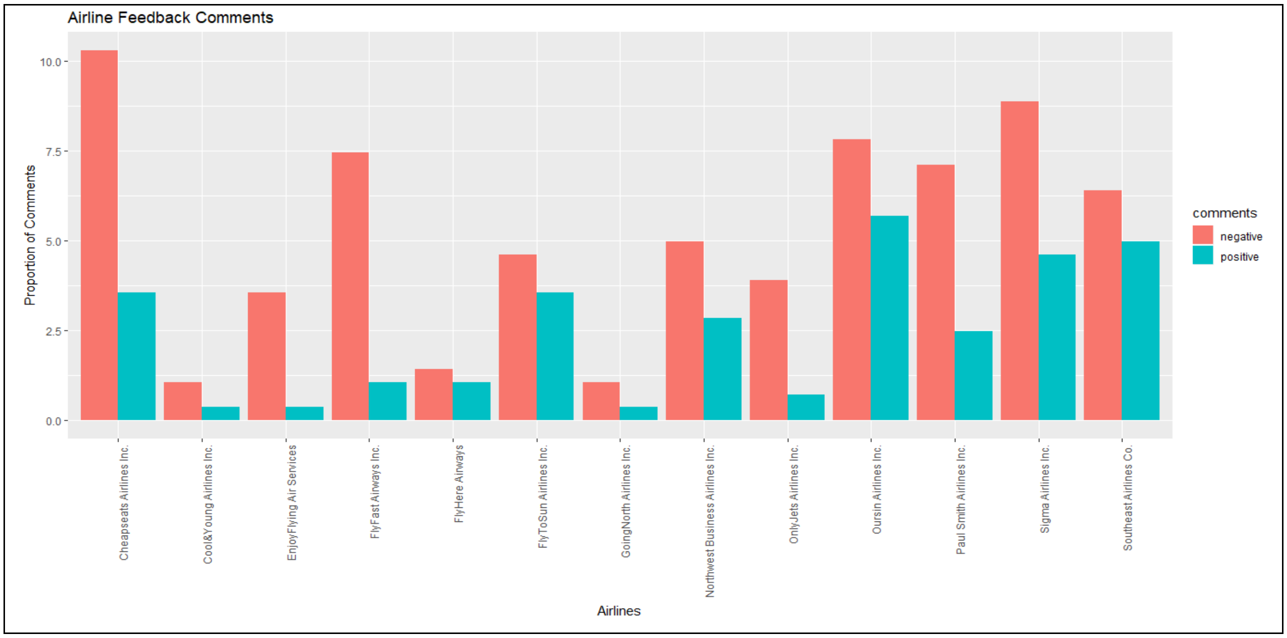


Fig: Plot for positive and negative comments of each partner airline

ACTIONABLE INSIGHTS:

1. Business travelers travelling by Blue status airline tend to be promoters whereas Personal travelers with Blue status airline tend to be detractors. The airlines should focus on retaining the customer satisfaction of Business travelers and at the same time conduct a survey for people travelling for personal reasons to check which attributes need to be addressed.
2. People who are classified as Frequent flyers tend to give higher ratings regardless of whether they have a frequent flyer account with Southeast airlines. It is also observed that frequent flyers who have more than two frequent flyer accounts tend to be detractors. So rather than targeting frequent flyers with loyalty programs we should target them with reduced airfare or fast check-in to ensure they are promoters. This will ensure the airlines don't incur additional frequent flyer program debt thereby increasing it's overall profits.
3. Middle aged people (ages in the range of 30-50) are more likely to give higher ratings as compared to younger and old aged people. The airlines should try to retain middle aged customers' engagement and should focus on old and young aged people. Adding facilities to help old people travel easily from one place to another and providing them with personnel who can carry their luggage would help in increasing customer satisfaction. The airlines can provide more entertainment services which can help in engaging the younger age group.
4. People with airline status Blue are less likely to give higher rating as compared to Gold, Silver and Platinum status airlines. However, the frequency of customers traveling by Blue status airline is high. Thus, Southeast airline should focus more on the service improvement of Blue status airlines and people travelling by Blue status airlines should be given some extra benefits to improve customer satisfaction.
5. Categorization of arrival delay in minutes into 2 categories: arrival delay greater than 5 minutes (Long delay) and arrival delay less than 5 minutes (Short delay) showed us that customer satisfaction is higher when delays are short. The airlines should try to increase their operational efficiency in order to increase customer satisfaction.
6. Female customers tend to give less recommendation scores as compared to male customers which reflects the airlines may lack certain facilities for females. Female customers happen to be promoters for airlines where price sensitivity is 1. This indicates that females are affected by the price sensitivity attribute. Females who travel by Blue status airlines for personal reasons tend to be detractors. In order to target female gender group, the airlines should conduct a survey for female travelers to see which factors affect customer satisfaction.
7. Eco and Eco plus class customers generally give lower ratings as compared to customers traveling by Business class.
8. People who travel by Blue status airlines and spend relatively less amount on eating and drinking at airport tend to be detractors. Thus, the airlines should focus on improving the services of Blue status airlines.

9. Customers tend to be promoters when price sensitivity is 1 and as price sensitivity increases, customers tend to be detractors. In order to engage and satisfy more customers, small discounts or bonuses can be beneficial.
10. The number of frequent flyer accounts of customers should be limited to 2 as it is observed that as the number of frequent flyer accounts increase over 2, the customers tend to become detractors.