IST 687 Applied Data Science

Analysis of Airline Survey dataset

M004 GROUP 1

Sangam Ludhani

Trisha Chakraborty

Aditi Shrivastava

Rahul Rathod

Zhida Liu

Nahnsan Guseh

Data Cleaning

The dataset for this project consisted of 129880 observations and 26 attributes.

Our approach included detecting and correcting inaccurate records to ease the process of analysis. Data cleaning is necessary before we can explore and create models. We focused on narrowing the data to the airlines with lower median value of customer satisfaction.

We started the data munging process by altering the names of the dataset columns so that the dots can be removed from column names, deleted values with decimals and then re-numbered the rows. Then we converted the columns to characters and numeric to ease the application of functions on each of them. The date format of flight date column was inconsistent so we changed the format of this column to a standard date format.

When flight cancelled (= yes), there were NA values in 2 columns (ArrivalDelayinMinutes, Flighttimeinminutes), we dropped or omitted these rows (2400) from the dataset.
When flight cancelled (= no), there were NA values in 2 columns (ArrivalDelayinMinutes, Flighttimeinminutes), we changed the NA values to the mean of the column in order to keep the distribution of the column as same.

Code:

```r
# 1) Data cleaning
# changing column names of dataset to remove dots
colnames(df.sample) <- c("Satisfaction","AirlineStatus","Age","Gender","PriceSensitivity","YearofFirstFlight",
                         + "NoofFlightspa","percentofFlightwithotherAirlines","TypeofTravel","No.ofotherLoyaltyCards",
                         + "ShoppingAmountatAirport","EatingandDrinkingatAirport", "Class", "DayofMonth","Flightdate",
                         + "AirlineCode","AirlineName","OrginCity","OriginState","DestinationCity","DestinationState",
                         + "ScheduledDepartureHour","DepartureDelayinMinutes","ArrivalDelayinMinutes","Flightcancelled",
                         + "Flighttimeinminutes","FlightDistance","ArrivalDelaygreater5Mins")

# Column satisfaction is having 3 unnecessary values. Finding indexes and deleting these rows from dataset
unique(df.sample$Satisfaction)               # Finding these values (4.00.5 , 4.00.2.00,  4.00.2.00)
x <- grep("*.00.*", df.sample$Satisfaction)   # Finding their row number in dataset by grep (pattern matching)
df.sample <- df.sample[-x,]                      # deleting row numbers (38898 38899 38900)
row.names(df.sample) <- NULL                            # renumbering the rows
y <- grep("*.5", df.sample$Satisfaction)
df.sample <- df.sample[-y,]                      # deleting row numbers (1 3 7 52712 52714 52718)
nrow(df.sample)

# function to correct and convert columns to string and remove whitespaces
correcttostring <- function(col)
{
  col <- as.character(col)
  col <- trimws (col, which = c("both"))
  col <- as.factor(col)
  return (col)
}

# changing date format to a standard format
df.sample$Flightdate <- gsub("/","-",df.sample$Flightdate)
df.sample$Flightdate <- as.Date(df.sample$Flightdate,tryFormats = c("%m-%d-%y", "%m/%d/%y"))

#converting columns to character
df.sample$AirlineStatus <- correcttostring(df.sample$AirlineStatus)
df.sample$Gender <- correcttostring(df.sample$Gender)
df.sample$TypeofTravel <- correcttostring(df.sample$TypeofTravel)
df.sample$Class <- correcttostring(df.sample$Class)
df.sample$AirlineCode <- correcttostring(df.sample$AirlineCode)
df.sample$AirlineName <- correcttostring(df.sample$AirlineName)
df.sample$OrginCity <- correcttostring(df.sample$OrginCity)
```

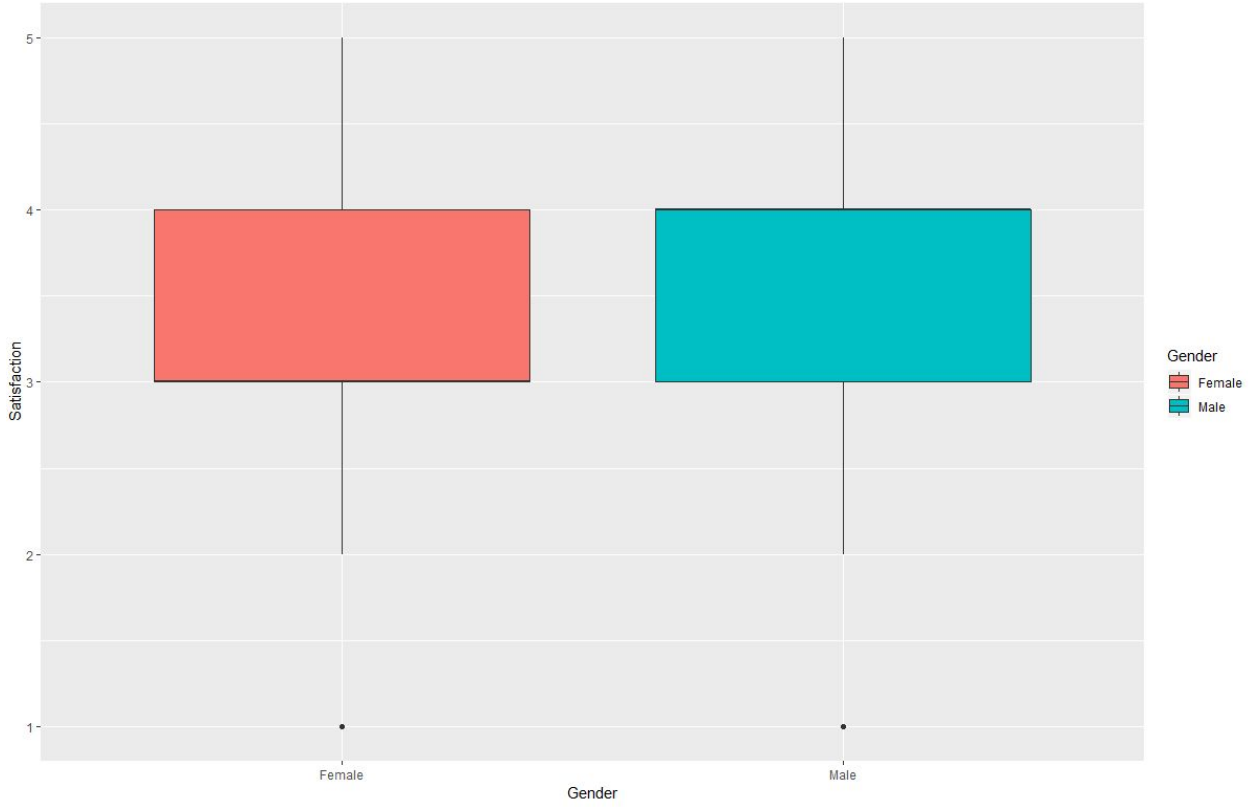
```r
df.sample$OriginState <- correcttostring(df.sample$OriginState)
df.sample$DestinationCity <- correcttostring(df.sample$DestinationCity)
df.sample$DestinationState <- correcttostring(df.sample$DestinationState)
df.sample$ArrivalDelaygreater5Mins <- correcttostring(df.sample$ArrivalDelaygreater5Mins)
df.sample$Flightcancelled <- correcttostring(df.sample$Flightcancelled)

str(df.sample)
#df$Genderlabel <- factor(df$Gender,labels = c(0,1))

# converting columns to numeric
# when you use as.numeric on a factor then you get the underlying integers
df.sample$Satisfaction <- as.numeric(as.character(df.sample$Satisfaction))
df.sample$Age <- as.numeric(df.sample$Age)
df.sample$PriceSensitivity <- as.numeric(df.sample$PriceSensitivity)
df.sample$YearofFirstFlight <- as.numeric(df.sample$YearofFirstFlight)
df.sample$NoofFlightspa<-as.numeric(df.sample$NoofFlightspa)
df.sample$percentofFlightwithotherAirlines <- as.numeric(df.sample$percentofFlightwithotherAirlines)
df.sample$No.ofotherLoyaltyCards <- as.numeric(df.sample$No.ofotherLoyaltyCards)
df.sample$ShoppingAmountatAirport <- as.numeric(df.sample$ShoppingAmountatAirport)
df.sample$EatingandDrinkingatAirport <- as.numeric(df.sample$EatingandDrinkingatAirport)
df.sample$ScheduledDepartureHour <- as.numeric(df.sample$ScheduledDepartureHour)
df.sample$FlightDistance <- as.numeric(df.sample$FlightDistance)
df.sample$DepartureDelayinMinutes <- as.numeric(df.sample$DepartureDelayinMinutes)
df.sample$ArrivalDelayinMinutes <- as.numeric(df.sample$ArrivalDelayinMinutes)
df.sample$Flighttimeinminutes <- as.numeric(df.sample$Flighttimeinminutes)
df.sample$DayofMonth <- as.numeric(df.sample$DayofMonth)

df.m <- df.sample[,c(-16,-18,-20,-27)]
# Deleted Day of Month,Airline Code,Orgin City,Destination City,Flight Distance

# replacing blank values in column (ArrivalDelayinMinutes,Flighttimeinminutes) when (flightcancelled=No) with mean of column

df.fcyes <- filter(df.m,Flightcancelled=="Yes")
df.fcno <- filter(df.m,Flightcancelled=="No")
df.fcno$ArrivalDelayinMinutes[is.na(df.fcno$ArrivalDelayinMinutes)] <- floor(mean(df.fcno$ArrivalDelayinMinutes , na.rm = TRUE))
df.fcno$Flighttimeinminutes[is.na(df.fcno$Flighttimeinminutes)] <- floor(mean(df.fcno$Flighttimeinminutes , na.rm = TRUE))

df.cleaned <- rbind(df.fcno,df.fcyes)
df <- df.cleaned
df.model <- na.omit(df)
```

```r
str(df.model)
summary(df.model)
```

Business Questions

1. **Which gender either male or female has the higher customer satisfaction rating value?**

On plotting a box plot between gender and satisfaction, we observed that the median satisfaction value of males is higher than that of female.

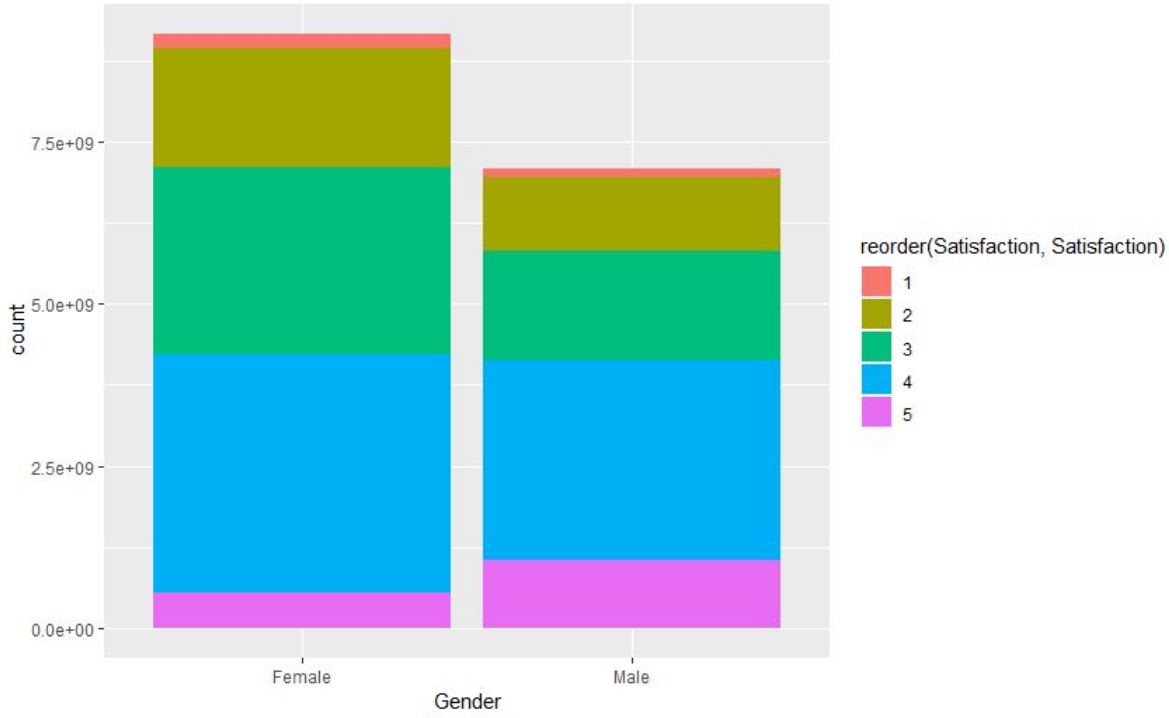


	Gender	AveSat	MedianSat	count
1	Female	3.270754	3	71866
2	Male	3.531485	4	55614

Code:

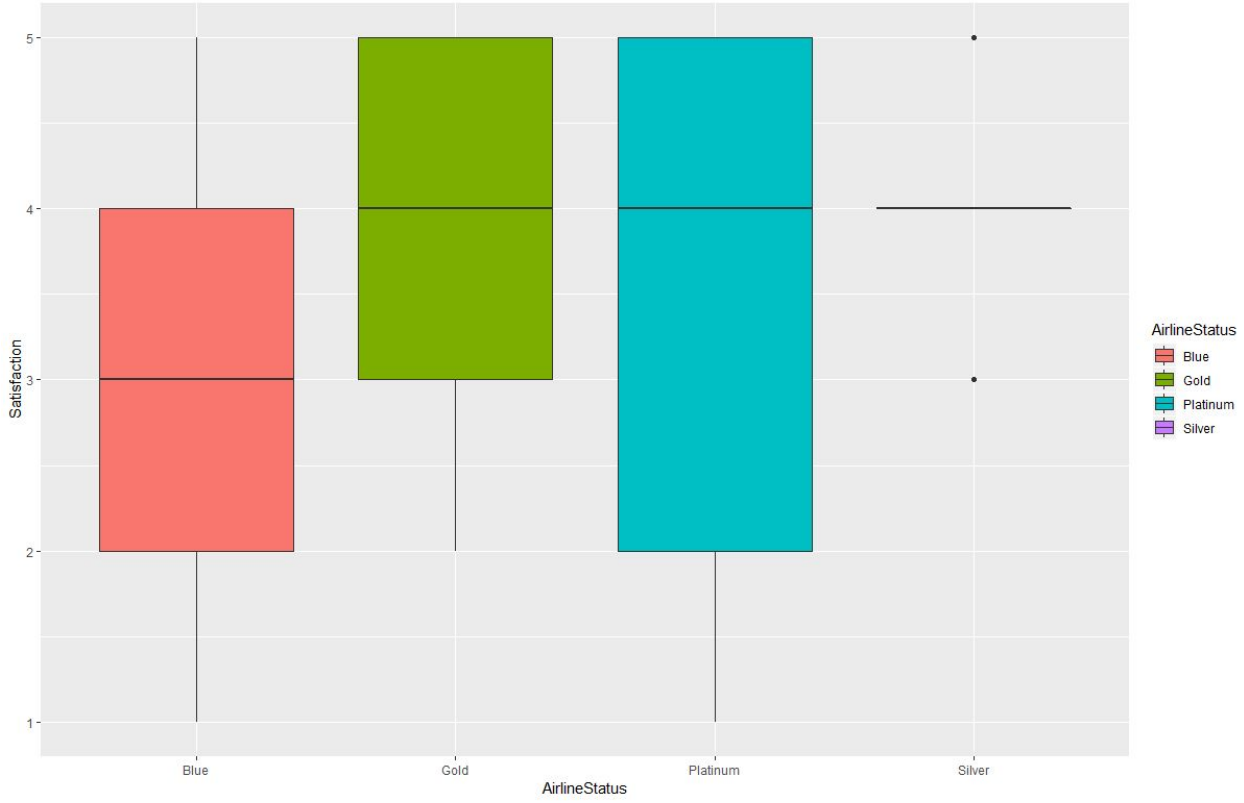
```
#analysis the relationship between Satisfaction and Gender
df.Gender <- group_by(df.model,Gender)
z.Gender <- summarise(df.Gender, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction),count=n())
View(z.Gender)
myboxPop <- ggplot(df.model,aes(x=Gender,y=Satisfaction,fill=Gender))+ geom_boxplot()
myboxPop
```

Other than this, we also plotted bar chart for Gender with total count and filled it with satisfaction. It shows females are giving more low ratings of (1, 2, 3) compared to male.



2. **Which type of airline status has the highest customer satisfaction rating?**

We plotted a boxplot between Airlinestatus and customer satisfaction and on analysis of this plot we found that the gold, platinum and silver airline statuses have highest median value of satisfaction while blue airline has the lowest.

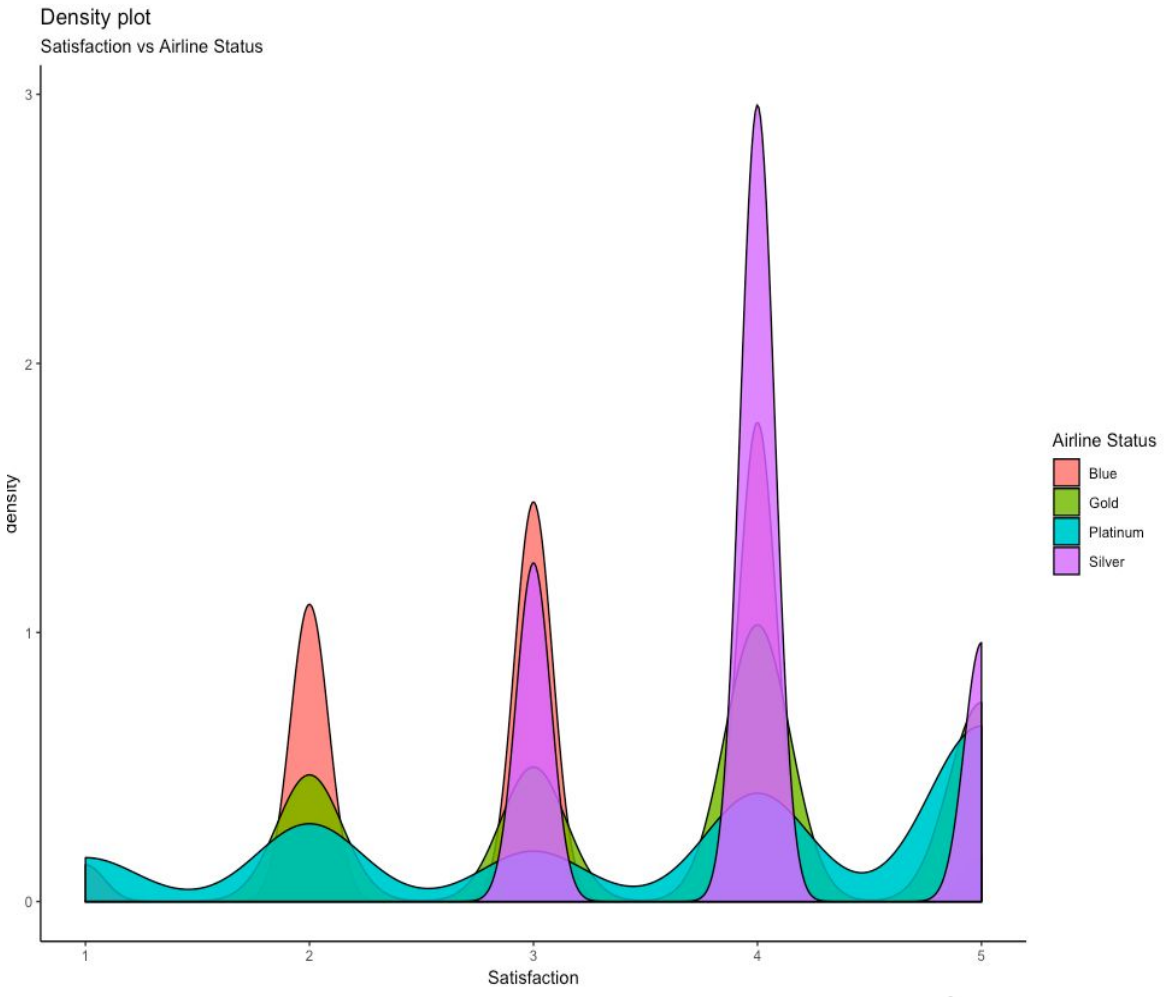


AirlineStatus	AveSat	MedianSat	count
Blue	3.161516	3	87081
Gold	3.752266	4	10701
Platinum	3.659600	4	4104
Silver	3.945300	4	25594

Code:

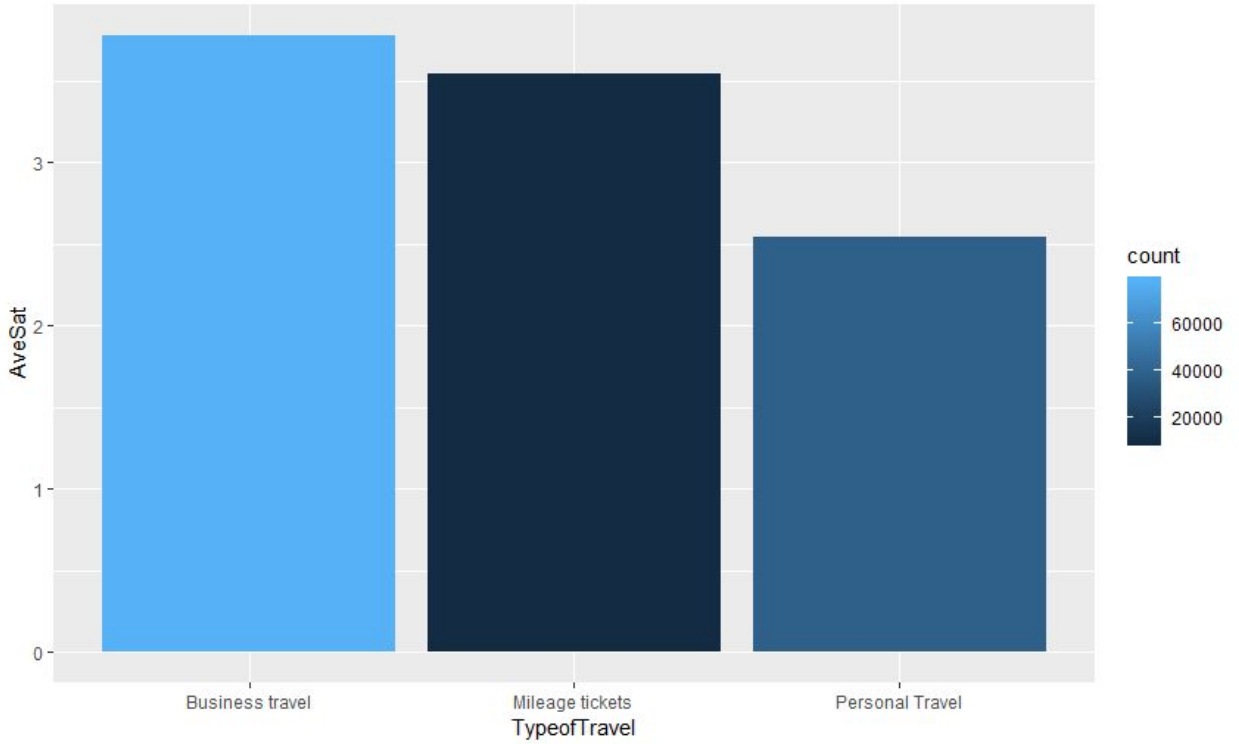
```
#analysis the relationship between Satisfaction and Airlinestatus
df.AirlineStatus <- group_by(df.model,AirlineStatus)
z.AirlineStatus <- summarise(df.AirlineStatus, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction),count=n())
View(z.AirlineStatus)
myboxPop <- ggplot(df.model,aes(x=AirlineStatus,y=Satisfaction,fill=AirlineStatus))+ geom_boxplot()
myboxPop
```

Other than this, we also plotted a density plot for different airline status and observed that for airline status as blue, customer satisfaction is spread in ratings of (1, 2, and 3)



3. **Which travel types have the highest and lowest customer satisfaction rating**?

On plotting a bar chart between type of travel and satisfaction, we can conclude that business and mileage travel type has highest average customer satisfaction rating. While, personal travel type has the lowest average customer satisfaction rating.

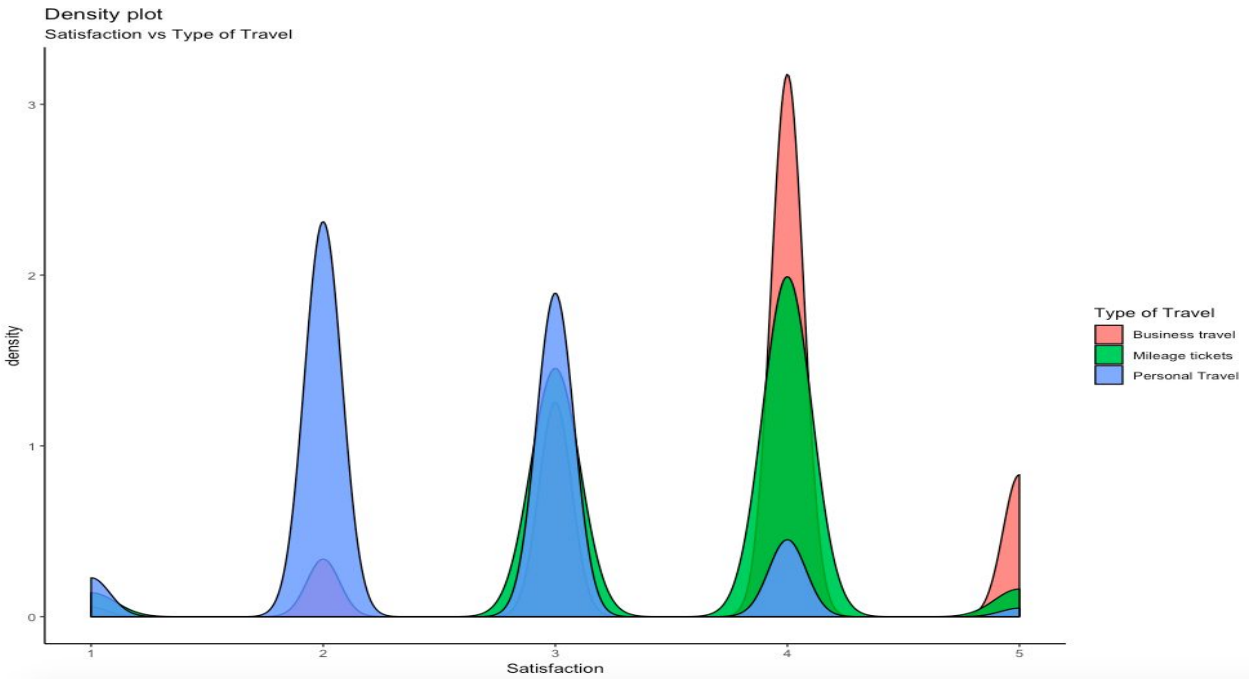


	TypeofTravel	AveSat	MedianSat	count
1	Business travel	3.781724	4	78584
2	Mileage tickets	3.540807	4	9839
3	Personal Travel	2.545894	2	39057

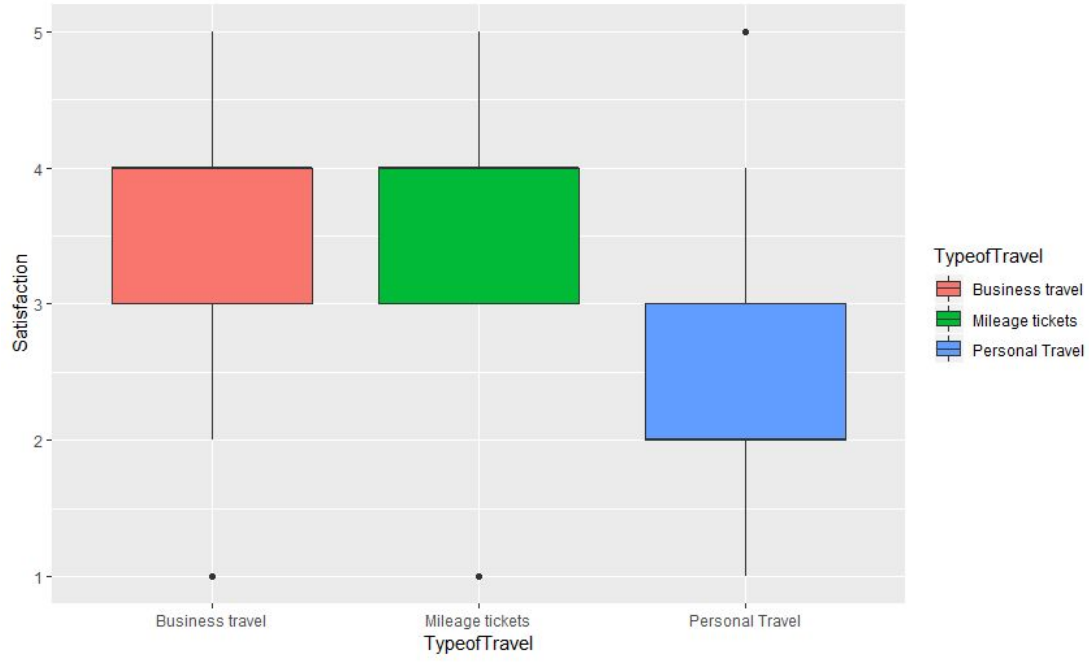
Code:

```
### analysis the relationship between Satisfaction and TypeofTravel
df.TypeofTravel <- group_by(df.model,TypeofTravel)
z.TypeofTravel <- summarise(df.TypeofTravel, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction),count=n())
View(z.TypeofTravel)
g_TypleOfTravel <- ggplot(df.model, aes(x=TypeofTravel, y=reorder(Satisfaction,Satisfaction) ,fill=Satisfaction)) +geom_col() +ylab("Satisfaction")
g_TypleOfTravel
```

Other than this, we also plotted a density plot for different type of travel and observed that for type of travel as personal, customer satisfaction is mostly spread in the ratings of (1, 2, and 3)

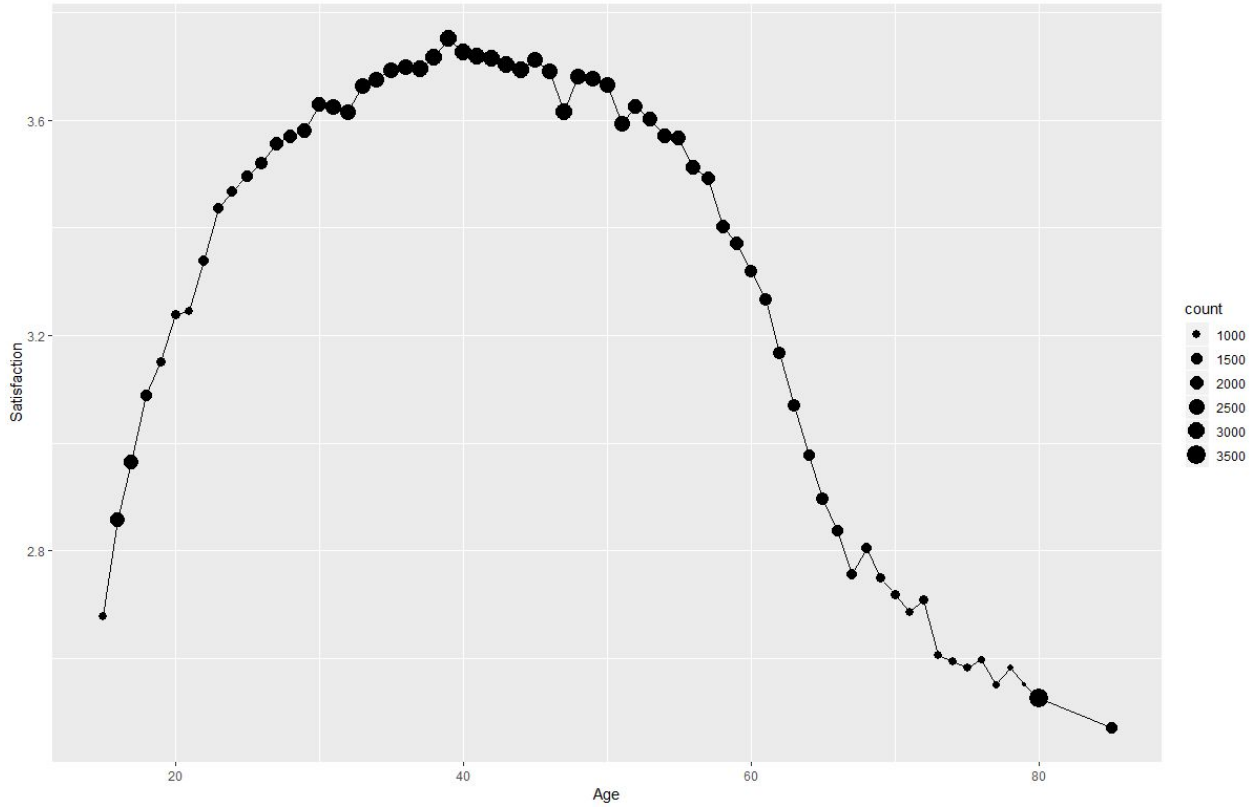


On plotting boxplot, we observed that for personal type of travel median satisfaction rating is low compared to business and mileage tickets.



4. **Which age ranges of customers have the highest and lowest customer satisfaction rating?**

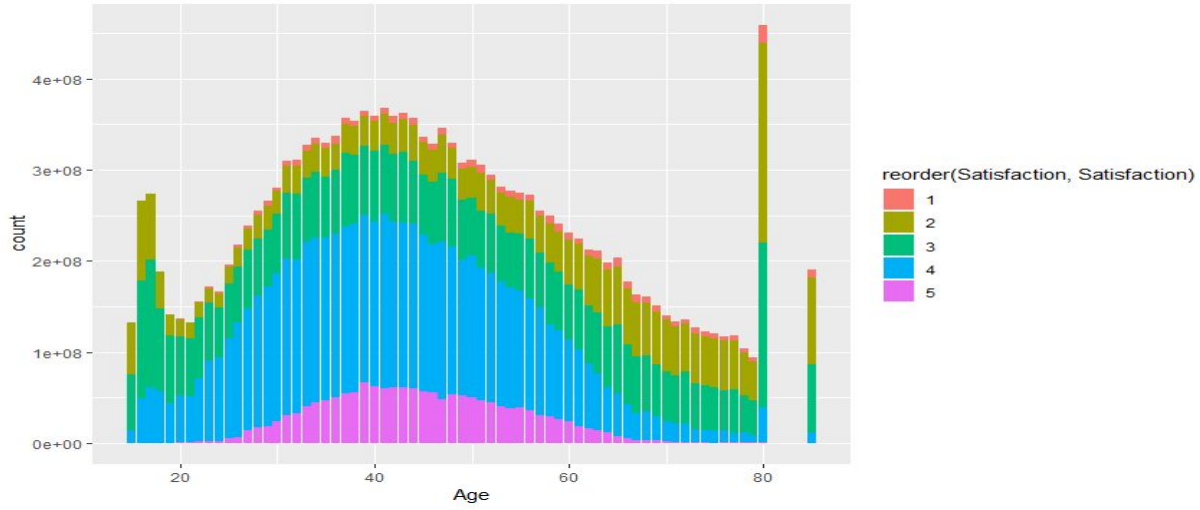
On plotting a scatterplot between ages and average satisfaction, we observed that the average satisfaction increases till the age of 40 and then it starts decreasing with age. Especially, for age group 60-85 average satisfaction is less.



Code:

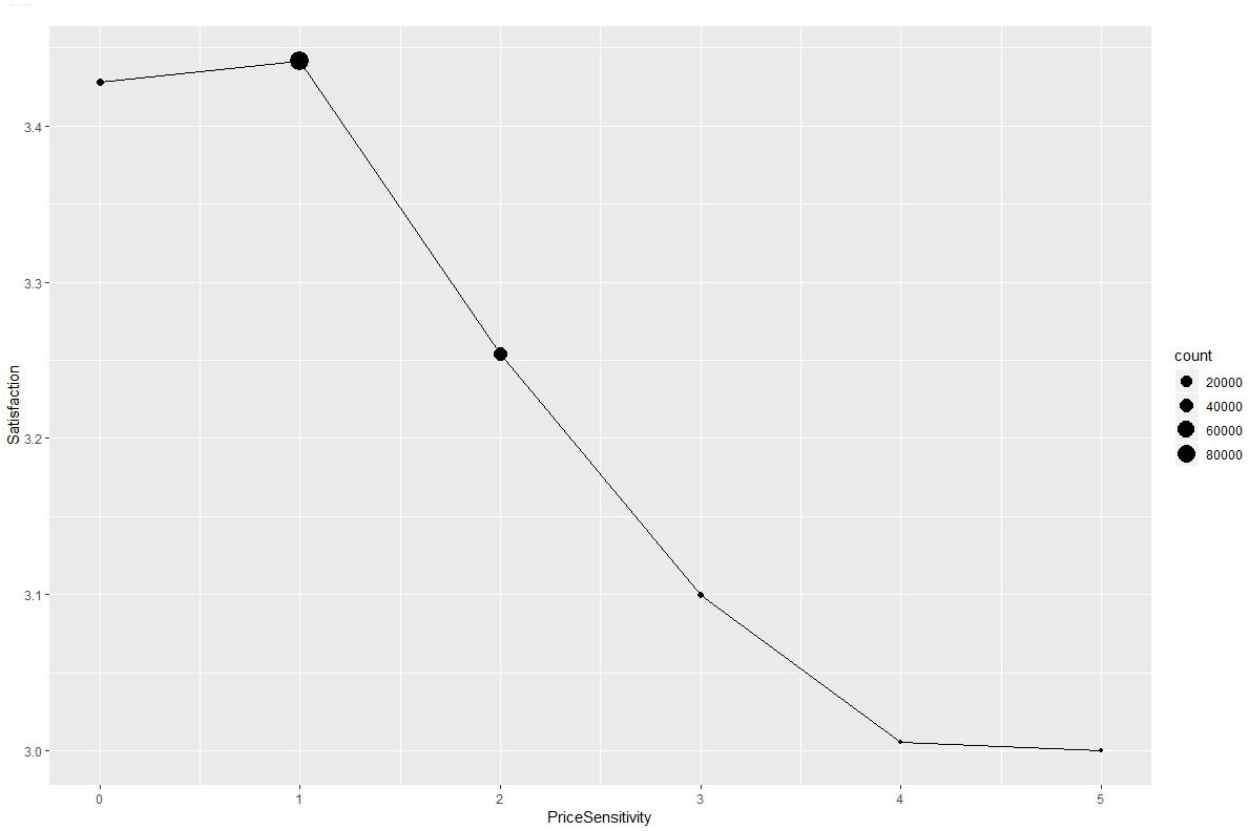
```
#analysis the relationship between Satisfaction and Age
df.age <- group_by(df.model,Age)
z.Age <- summarise(df.age, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction),count=n())
View(z.Age)
gplot_Age <- ggplot(z.Age,aes(x=Age, y=AveSat))
gplot_Age <- gplot_Age + geom_line() +geom_point(aes(size=count))
gplot_Age
```

Other than this, we have also plotted a bar chart of Age with total count and filled it with satisfaction. It can be seen that people in age group (60-85) are mostly giving ratings of 1,2 and 3.



5. **How is price sensitivity related to customer satisfaction?**

On plotting a scatterplot between price sensitivity and customer satisfaction, we observed that the customer satisfaction rating corresponding to price sensitivity of 1 is highest and that of 5 is lowest. We can conclude that if price sensitivity and average customer satisfaction is inversely related.



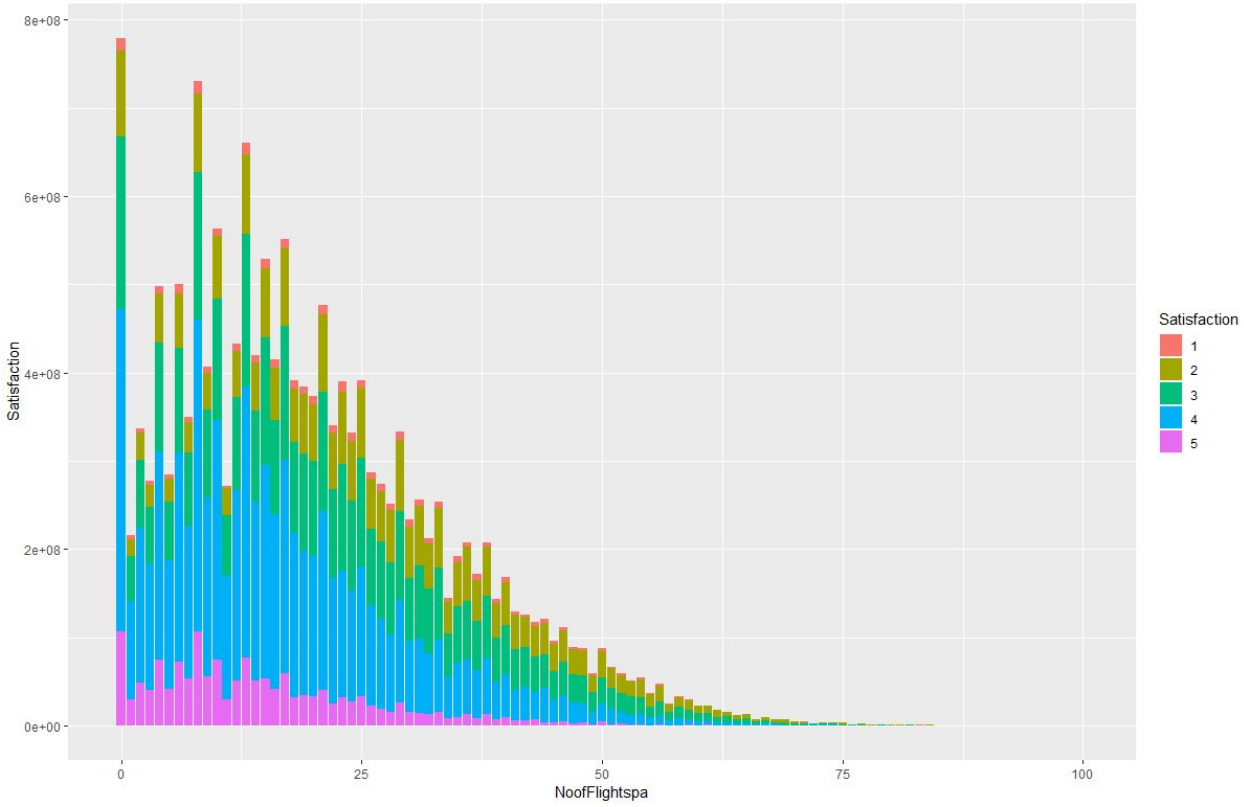
	PriceSensitivity	AveSat	MedianSat	count
1	0	3.427966	4	4012
2	1	3.441723	4	86500
3	2	3.254342	3	35063
4	3	3.099825	3	1713
5	4	3.005236	3	191
6	5	3.000000	3	1

Code:

```
#analysis the relationship between Satisfaction and Price Sensitivity
df.PriceSensitivity <- group_by(df.model,PriceSensitivity)
z.PriceSensitivity <- summarise(df.PriceSensitivity, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction), count = n())
View(z.PriceSensitivity)

# visualization
ggplot_PriceSensitivity <- ggplot(z.PriceSensitivity,aes(x=PriceSensitivity, y=AveSat))
ggplot_PriceSensitivity <- ggplot_PriceSensitivity + geom_line() +geom_point(aes(size=count)) +ylab("Satisfaction")
ggplot_PriceSensitivity
```

6. How does number of flights per annum alter the customer satisfaction rating?

On plotting a barchart between number of flights per annum and customer satisfaction, we observed that when the number of flights per annum decreases customer satisfaction increases.



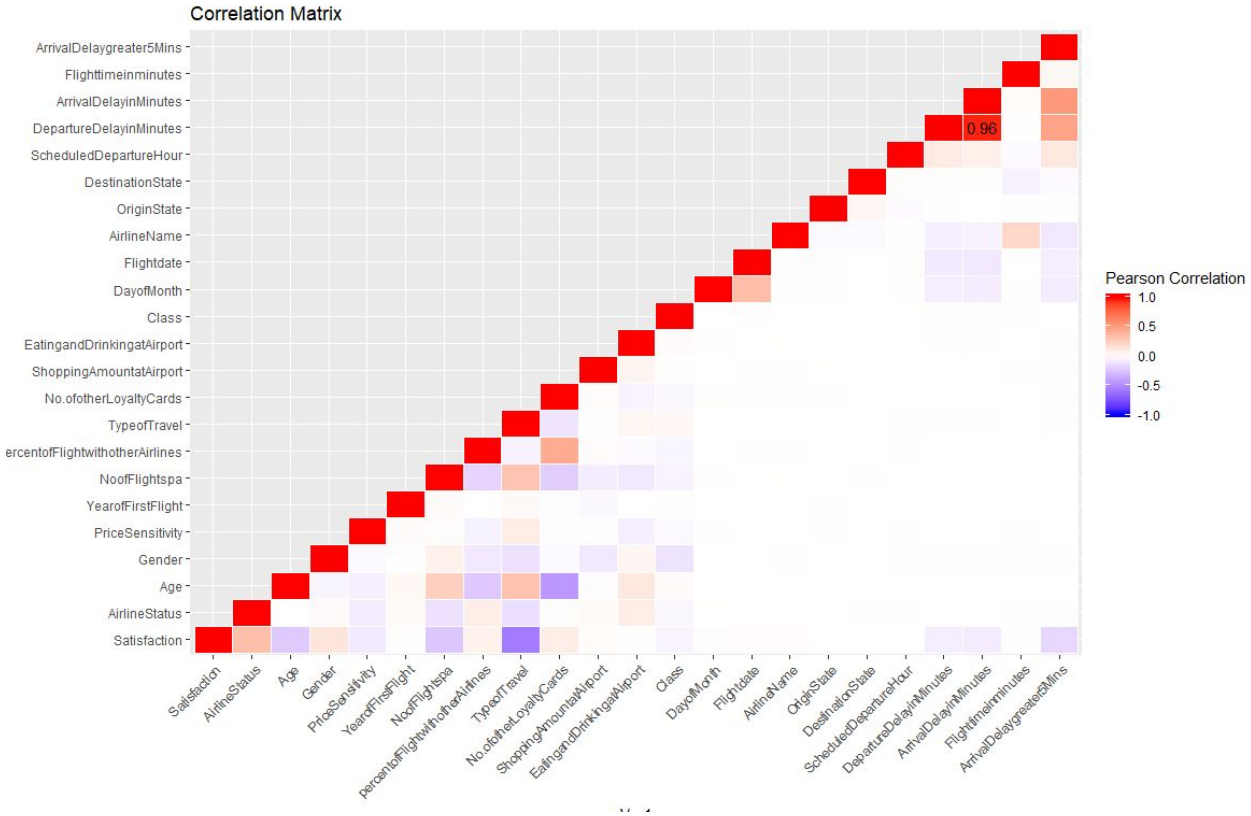
Code:

```
#analysis the relationship between Satisfaction and NoofFlightsp.a.
df.NoofFlightspa <- group_by(df.model,NoofFlightspa)
z.NoofFlightspa <- summarise(df.NoofFlightspa, AveSat = mean(Satisfaction), MedianSat = median(Satisfaction), count = n())
View(z.NoofFlightspa)

ggplot_NoOfFlightspa <- ggplot(df.model,aes(x=NoofFlightspa, y=nrow(df.model)))
ggplot_NoOfFlightspa <- ggplot_NoOfFlightspa + geom_col(aes(fill=reorder(Satisfaction,Satisfaction)))
ggplot_NoOfFlightspa <- ggplot_NoOfFlightspa  +ylab("Satisfaction") + guides(fill=guide_legend(title="Satisfaction"))
ggplot_NoOfFlightspa
```

Narrowing down the data set

Correlation Matrix

Correlation matrix is used to investigate dependencies between multiple variables at the same time. It results in a table of correlation coefficients between each variables. Correlation coefficient measures the percentage of fluctuation in one variable that can be explained by another variable. A correlation of 1 means the variables move in perfect unison, a correlation of -1 means the variables move in the complete opposite direction, and a correlation of 0 means there is no relationship at all between the two variables.



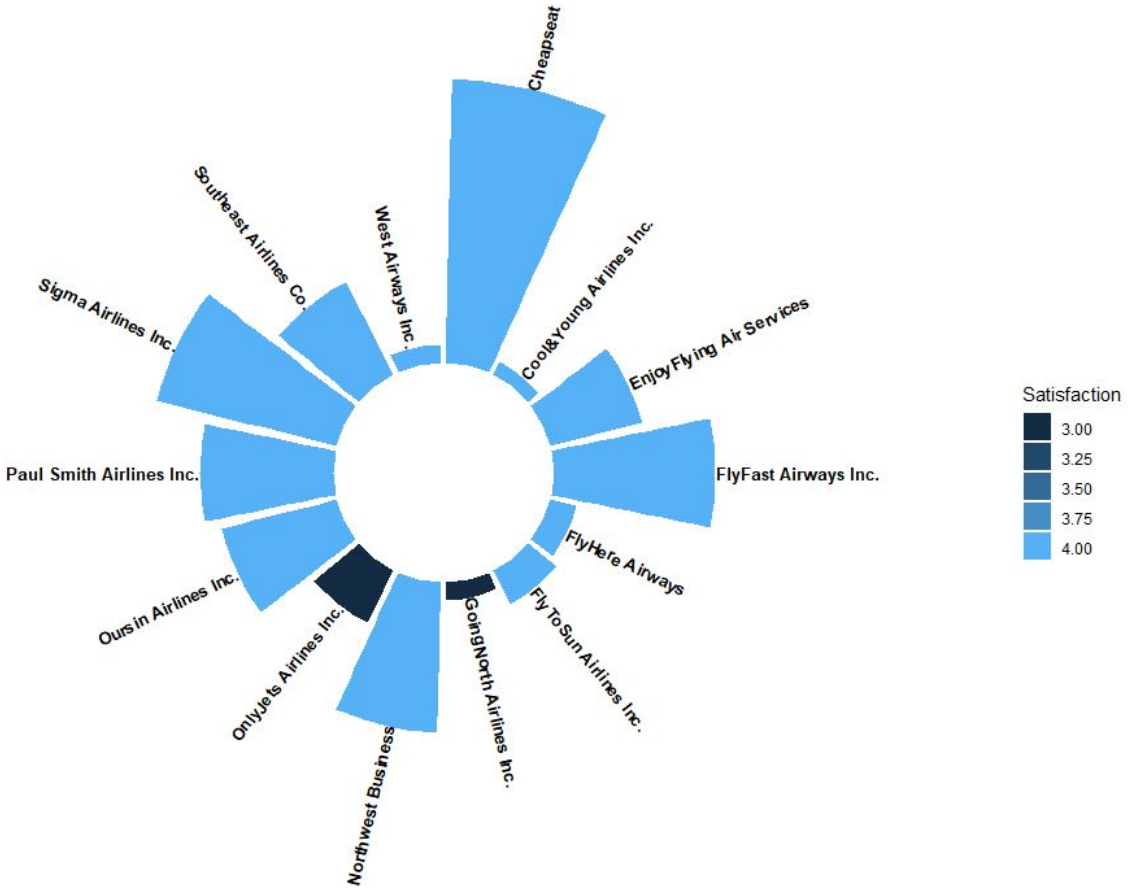
From the below plot, we can observe the below dependencies and relationship between variables with customer satisfaction,

Direct Dependency	Indirect Dependency	No Dependency
AirlineStatus	PriceSensitivity	YearOfFirstFlight
Gender	NoOfFlightsPerAnnum	EatingandDrinkingAtAirport
PercentageOfAirlineWithOtherAirlines	TypeOfTravel	DayOfMonth
NoOfOtherLoyaltyCards	DepartureDelay	FlightDate
	ArrivalDelay	AirlineName
	ArrivalDelayGreaterThan5mins	OriginState
	ShoppingAmountAtAirport	DestinationState
		ScheduledDepartureHour
		FlightTimeinMins
		Class

Code:

```
# Start of Correlation Matrix ------------------------------------------------------------

corr <- df.model %>% sapply(., as.numeric) %>% as.data.table()
corr <- cor(corr, use = 'pairwise.complete.obs')
corr[upper.tri(corr)] <- NA
corr <- melt(corr, na.rm = T) %>% as.data.table() %>% setorder(-value)
corr$text <- ifelse(abs(corr$value) >= .8 & corr$value != 1, round(corr$value, 2), '')

ggplot(data = corr, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = 'white') +
  geom_text(aes(label = text)) +
  scale_fill_gradient2(low = 'blue', high = 'red', mid = 'white',
                       midpoint = 0, limit = c(-1, 1),
                       name = 'Pearson Correlation') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = 'Correlation Matrix')
```

After analyzing and visualizing the dataset, we got a clear idea about the data. Next we moved on focusing on two particular airlines namely OnlyJetInc airlines and GoingNorth airlines because the median customer satisfaction for both these airlines was low as compared to other airlines as can be seen in below circular boxplot. Its scaling is as per the count of each airline and color fill is as per the median satisfaction value. OnlyJet and GoingNorth has lowest median satisfaction value of 3 & hence different in color from the rest.



Code:

```
#-------------------------------------------------------------------------------------------------------------
# Creating dataset for circular boxplot

df.Airline <- group_by(df.model,AirlineName)
z <- summarize(df.Airline, mediansatisfaction = median(Satisfaction), count = n())
#b <- z[order(-z$count),]
b <- z
b$AirlineName<- as.character(b$AirlineName)
b$AirlineName[b$AirlineName == "Cheapseats Airlines Inc."] <- c("Cheapseat")
b$AirlineName[b$AirlineName == "Northwest Business Airlines Inc."] <- c("Northwest Business")
b$AirlineName<- as.factor(b$AirlineName)

#unique(b$AirlineName)

data = data.frame(id=seq(1,14), individual = paste(b$AirlineName, sep=""), value= b$count)

# ----- This section prepare a dataframe for labels ---- #
# Get the name and the y position of each label
label_data=data

# calculate the ANGLE of the labels
number_of_bar=nrow(label_data)
angle= 90 - 360 * (label_data$id-0.5) /number_of_bar     # I substract 0.5 because the letter must have the angle of the center of the bars. Not extreme right(1)

# calculate the alignment of labels: right or left
# If I am on the left part of the plot, my labels have currently an angle < -90
label_data$hjust<-ifelse( angle < -90, 1, 0)

# flip angle BY to make them readable
label_data$angle<-ifelse(angle < -90, angle+180, angle)
# ----- ------------------------------------------- ---- #

# Start the plot
p = ggplot(data, aes(x=as.factor(id), y=value)) +       # Note that id is a factor. If x is numeric, there is some space between the first bar

  # This add the bars with a blue color
  geom_bar(stat="identity", aes(fill= b$mediansatisfaction))+ guides(fill = guide_legend(title = "Satisfaction"))+    #alpha("skyblue", 0.7)) +

  # Limits of the plot = very important. The negative value controls the size of the inner circle, the positive one is useful to add size over each bar
  ylim(-10000,30000) +
```

```
  ylim(-10000,30000) +

  # Custom the theme: no axis title and no cartesian grid
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    plot.margin = unit(rep(-1,4), "cm")      # Adjust the margin to make in sort labels are not truncated!
  ) +

  # This makes the coordinate polar instead of cartesian.
  coord_polar(start = 0) +

  # Add the labels, using the label_data dataframe that we have created before
  geom_text(data=label_data, aes(x=id, y=value+10, label=individual, hjust=hjust), color="black", fontface="bold",alpha=1,
            +size=3.5, angle= label_data$angle, inherit.aes = FALSE )
p
```

Linear Modelling

Linear modelling is used to predict the value of an outcome variable Y based on one or more input predictor variables X. Here, we can establish a linear relationship between the two variables where a continuous variable is modelled as a mathematical function of one or more variable which are known.

On the entire dataset, we performed forward linear modelling and added one-by-one variable and kept on checking increasing R square value. Finally, after creating 22 models we got 8 out of 28 columns which were significant and impacting (+ve and –ve manner) customer satisfaction column.

- Airline Status
- Gender
- Age
- Price sensitivity
- Number of flights per annum
- Type of travel
- Shopping Amount at airport
- Arrival delay greater than 5 minutes

The above factors are considered to be independent variables and customer satisfaction will be the dependent variable.

Code:

```
lm.Data.model<- lm(formula=Satisfaction ~ AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel
                   + ShoppingAmountatAirport + ArrivalDelaygreater5Mins ,data = df.model)
summary(lm.Data.model)
```

Output:

```
lm(formula = Satisfaction ~ AirlineStatus + Gender + Age + PriceSensitivity +
    NoofFlightspa + TypeofTravel + ShoppingAmountatAirport +
    ArrivalDelaygreater5Mins, data = df.model)

Residuals:
     Min       1Q   Median       3Q      Max
-3.10091 -0.41383  0.08041  0.47060  2.86686

Coefficients:
                                Estimate Std. Error  t value Pr(>|t|)
(Intercept)                    3.831e+00  8.607e-03  445.060  < 2e-16 ***
AirlineStatusGold              4.429e-01  7.454e-03   59.424  < 2e-16 ***
AirlineStatusPlatinum          2.658e-01  1.161e-02   22.901  < 2e-16 ***
AirlineStatusSilver            6.208e-01  5.190e-03  119.596  < 2e-16 ***
GenderMale                     1.315e-01  4.153e-03   31.662  < 2e-16 ***
Age                           -2.248e-03  1.273e-04  -17.662  < 2e-16 ***
PriceSensitivity              -3.947e-02  3.738e-03  -10.559  < 2e-16 ***
NoofFlightspa                 -3.236e-03  1.519e-04  -21.311  < 2e-16 ***
TypeofTravelMileage tickets   -1.467e-01  7.776e-03  -18.868  < 2e-16 ***
TypeofTravelPersonal Travel   -1.078e+00  4.970e-03 -216.996  < 2e-16 ***
ShoppingAmountatAirport        1.541e-04  3.827e-05    4.026 5.67e-05 ***
ArrivalDelaygreater5Minsyes   -3.363e-01  4.228e-03  -79.524  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7196 on 127468 degrees of freedom
Multiple R-squared:  0.4456,    Adjusted R-squared:  0.4456
F-statistic:  9315 on 11 and 127468 DF,  p-value: < 2.2e-16
```

We observe that the R Squared value is 0.4456 whiich means that a combination of all the seven factors account for 44% of the customer satisfaction rating.

Linear Modeling on OnlyJets Airlines:

Then, after due deliberation we conducted linear modelling on two airlines , the first one being Only Jet customers with customer satisfaction ratings based on below factors :

- Airline Status
- Gender
- Age
- Price sensitivity
- Number of flights per annum
- Type of travel
- Shopping Amount at airport
- Arrival delay greater than 5 minutes

Here, we observed that the R Squared value is highest for the combination of below factors:

- Airline Status
- Gender
- Age
- Price sensitivity
- Number of flights per annum
- Type of travel
- Arrival delay greater than 5 minutes

Code:

```
# Preparation of dataset of OnlyJets airline
#unique(df.model$AirlineName)
df.onlyjet <- filter(df.model, AirlineName == "OnlyJets Airlines Inc.")

# Apply linear model to onlyjet
lm.onlyjet <- lm(formula=Satisfaction ~ AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel
                 + ShoppingAmountatAirport + ArrivalDelaygreater5Mins ,data = df.onlyjet)
summary(lm.onlyjet)                #Multiple R-squared:  0.462
```

Output:

```
lm(formula = Satisfaction ~ AirlineStatus + Gender + Age + PriceSensitivity +
    NoofFlightspa + TypeofTravel + ShoppingAmountatAirport +
    ArrivalDelaygreater5Mins, data = df.onlyjet)

Residuals:
     Min       1Q   Median       3Q      Max
-3.03831 -0.40950  0.04461  0.45966  2.91114

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    3.8883710  0.0422657  91.998  < 2e-16 ***
AirlineStatusGold              0.4317964  0.0361858  11.933  < 2e-16 ***
AirlineStatusPlatinum          0.3142080  0.0556936   5.642 1.77e-08 ***
AirlineStatusSilver            0.6458252  0.0259400  24.897  < 2e-16 ***
GenderMale                     0.1336080  0.0204437   6.535 6.94e-11 ***
Age                           -0.0034478  0.0006252  -5.515 3.65e-08 ***
PriceSensitivity              -0.0530736  0.0181606  -2.922 0.003488 **
NoofFlightspa                 -0.0027499  0.0007516  -3.659 0.000256 ***
TypeofTravelMileage tickets   -0.1497830  0.0378875  -3.953 7.81e-05 ***
TypeofTravelPersonal Travel   -1.0950156  0.0243538 -44.963  < 2e-16 ***
ShoppingAmountatAirport        0.0003420  0.0001942   1.761 0.078264 .
ArrivalDelaygreater5Minsyes   -0.3675005  0.0203828 -18.030  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7188 on 5260 degrees of freedom
Multiple R-squared:  0.462,     Adjusted R-squared:  0.4609
F-statistic: 410.6 on 11 and 5260 DF,  p-value: < 2.2e-16
```

We observe that the R Squared value is 0.462 which means that a combination of all the seven factors account for 46% of the customer satisfaction rating.

Linear modelling on GoingNorth airlines

Here, we observed that the R Squared value is highest for the combination of below factors :

- Airline Status
- Gender
- Type of travel
- Arrival delay greater than 5 minutes

Code:

```
# Preparation of dataset of GoingNorth airline
df.GoingNorth <- filter(df.model, AirlineName == "GoingNorth Airlines Inc.")

# Apply linear model to GoingNorth
lm.GoingNorth <- lm(formula=Satisfaction ~ AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel
                    + ShoppingAmountatAirport + ArrivalDelaygreater5Mins ,data = df.GoingNorth)
summary(lm.GoingNorth)                #Multiple R-squared:  0.4586
```

Output:

```
lm(formula = Satisfaction ~ AirlineStatus + Gender + Age + PriceSensitivity +
    NoofFlightspa + TypeofTravel + ShoppingAmountatAirport +
    ArrivalDelaygreater5Mins, data = df.GoingNorth)

Residuals:
     Min       1Q   Median       3Q      Max
-2.90023 -0.40264 -0.01678  0.52827  2.50973

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  3.7350527  0.0830068  44.997  < 2e-16 ***
AirlineStatusGold            0.3695293  0.0688701   5.366 9.29e-08 ***
AirlineStatusPlatinum        0.2841611  0.1121101   2.535  0.01135 *
AirlineStatusSilver          0.6967049  0.0496503  14.032  < 2e-16 ***
GenderMale                   0.1238522  0.0393699   3.146  0.00169 **
Age                         -0.0018469  0.0012130  -1.523  0.12805
PriceSensitivity            -0.0264705  0.0362088  -0.731  0.46486
NoofFlightspa               -0.0010062  0.0014213  -0.708  0.47912
TypeofTravelMileage tickets -0.2014477  0.0741220  -2.718  0.00665 **
TypeofTravelPersonal Travel -1.1575502  0.0466389 -24.819  < 2e-16 ***
ShoppingAmountatAirport      0.0004101  0.0003437   1.193  0.23295
ArrivalDelaygreater5Minsyes -0.3661973  0.0382071  -9.585  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7503 on 1550 degrees of freedom
Multiple R-squared:  0.4586,    Adjusted R-squared:  0.4548
F-statistic: 119.4 on 11 and 1550 DF,  p-value: < 2.2e-16
```

We observe that the R Squared value is 0.4586 which means that a combination of all the seven factors account for 45% of the customer satisfaction rating.

Association Rules

Association Rule Mining is a common technique used to find associations between many variables. It is used when you want to find an association between different objects in a set, find frequent patterns in a transaction database, relational databases or any other information repository.

Here, we have used association rule to determine the factors which would have maximum impact on the customer satisfactions. We set RHS in the model as customer satisfaction in order to analyze the factors which made for high or low customer rating. Based on this, we considered some rules and picked up top two rules which had the highest “Lift” value.

This helped us to understand the combination of best factors which would be useful to convert a low customer satisfaction rating to high customer satisfaction rating.

Association Rules on entire data set:

Code:

```
createBuckets <-function(vec)
{
  q <- quantile(vec, c(0.4, 0.6))
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec <= q[1]] <- "Low"
  vBuckets[vec > q[2]] <- "High"
  return (vBuckets)
}

summary(df.model$Satisfaction)
df.model$satis_ <- replicate(length(df.model$Satisfaction), "Average")
df.model$satis_[df.model$Satisfaction >= 4] <- "High" #values higher than 7 are termed as high
df.model$satis_[df.model$Satisfaction <= 2] <- "Low"
unique(df.model$satis_)

hist(df.model$Age)
unique(df.model$Age)
median(df.model$Age)
df.model$age_ <- replicate(length(df.model$Age), "Medium")         # 30 to 60 as medium
df.model$age_[df.model$Age >= 15 & df.model$Age <= 30] <- "Young"  #15  to 30 as Young
df.model$age_[df.model$Age > 60 & df.model$Age <= 85] <- "Aged"   # 60 to 85 as Aged

summary(df.model$PriceSensitivity)
hist(df.model$PriceSensitivity)
unique(df.model$PriceSensitivity) ## 0 to5
df.model$PriceSensitivity_<-createBuckets(df.model$PriceSensitivity)

hist(df.model$NoofFlightspa)
unique(df.model$NoofFlightspa) ##
median(df.model$NoofFlightspa)
df.model$NoofFlightspa_ <-createBuckets(df.model$NoofFlightspa)

unique(df.model$ShoppingAmountatAirport) ##
summary(df.model$ShoppingAmountatAirport)
hist(df.model$ShoppingAmountatAirport)
df.model$ShoppingAmountatAirport_<-createBuckets(df.model$ShoppingAmountatAirport)


ruleDF <- data.frame(df.model$satis_,df.model$AirlineStatus, df.model$Gender,df.model$age_, df.model$PriceSensitivity_,
                     + df.model$NoofFlightspa_,df.model$TypeofTravel,df.model$ArrivalDelaygreater5Mins, df.model$ShoppingAmountatAirport_)
summary(ruleDF)
View(ruleDF)
nrow(ruleDF)

aRuleDF <- as(ruleDF,"transactions")
summary(itemFrequency(aRuleDF))
itemFrequencyPlot(aRuleDF)
inspect(aRuleDF)
summary(aRuleDF)
10/length(aRuleDF)
```

High Customer Satisfaction Rules:

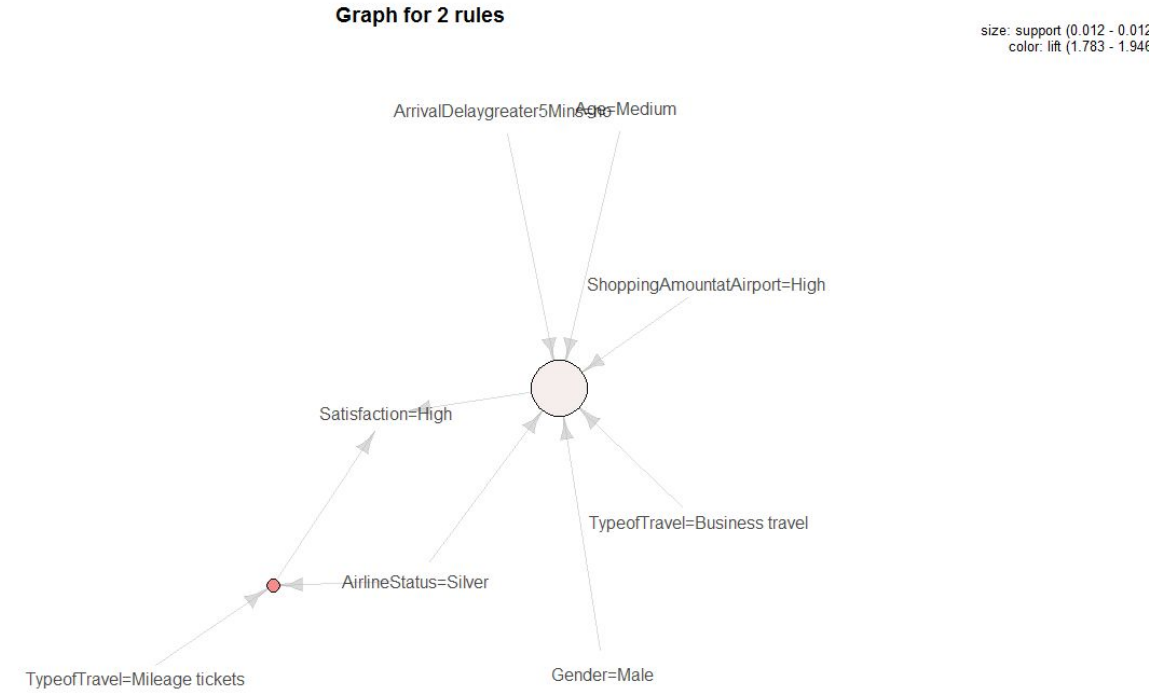
```
ruleset <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("df.model.satis_=High")))
summary(ruleset)
inspect(ruleset)

a.highRules <- sort(ruleset, decreasing =TRUE, by = "lift")
b.highRules <- head(a.highRules, n= 5)
inspect(head(b.highRules[1:2]))

plot(b.highRules[1:2], method="graph", control=list(type="items"))
plot(b.highRules[1:2], method="paracoord", control=list())
```

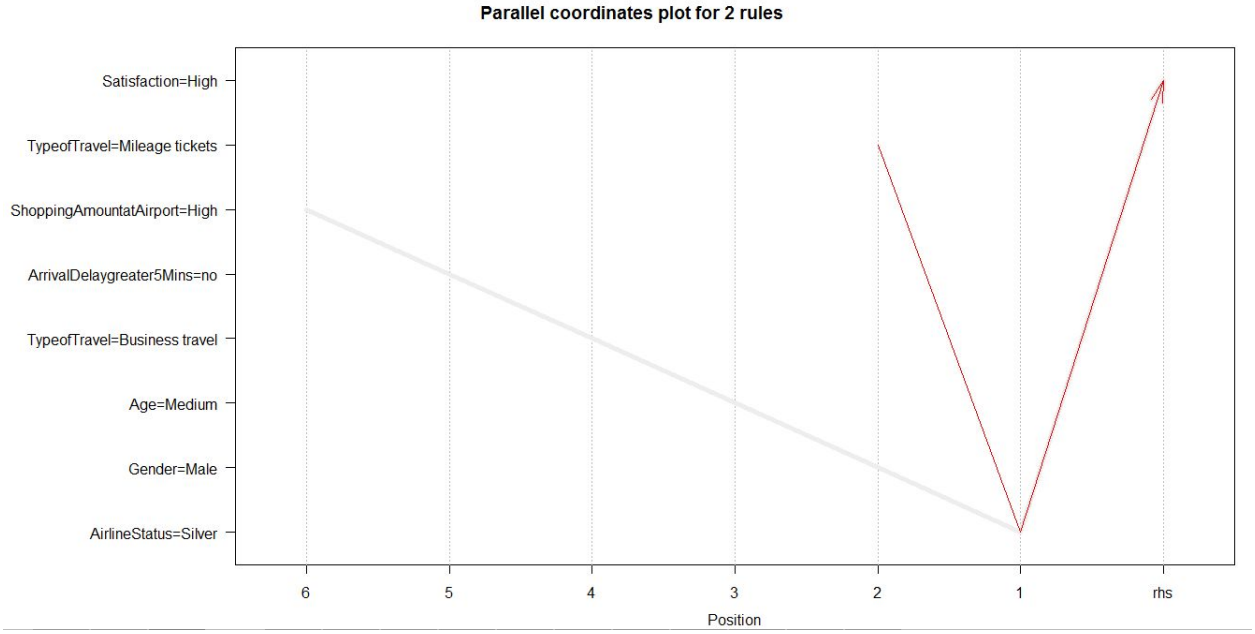
As per our model and combination of our support and confidence values for high customer satisfaction around 2290 rules were generated. Based on the highest values of lift we have taken 2 rules.

Rules:

```
> inspect(head(b[1:2]))
    lhs                                  rhs                    support confidence     lift count
[1] {AirlineStatus=Blue,
     Age=Aged,
     PriceSensitivity=Low,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.01286476  0.9969605 4.879166  1640
[2] {AirlineStatus=Blue,
     Age=Aged,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.01935990  0.9963666 4.876260  2468
```

Plots:





On running the association rule mining for highly satisfied customers, we have taken top 2 rules and plotted the below graph to determine the factors which influence customer satisfaction in airlines.

From the above plot we can determine the most significant attributes which influence a customer to become a highly satisfied customer are -

- Age - Medium (30-60)
- Gender - Male
- Type of travel - Business, mileage
- Airline Status - Silver
- Price sensitivity - Low
- Arrival delay greater than 5 minutes - No
- Shopping amount at Airport - High

This shows that the above factors help the customer to be highly satisfied.

Low satisfied Customer:

Code:

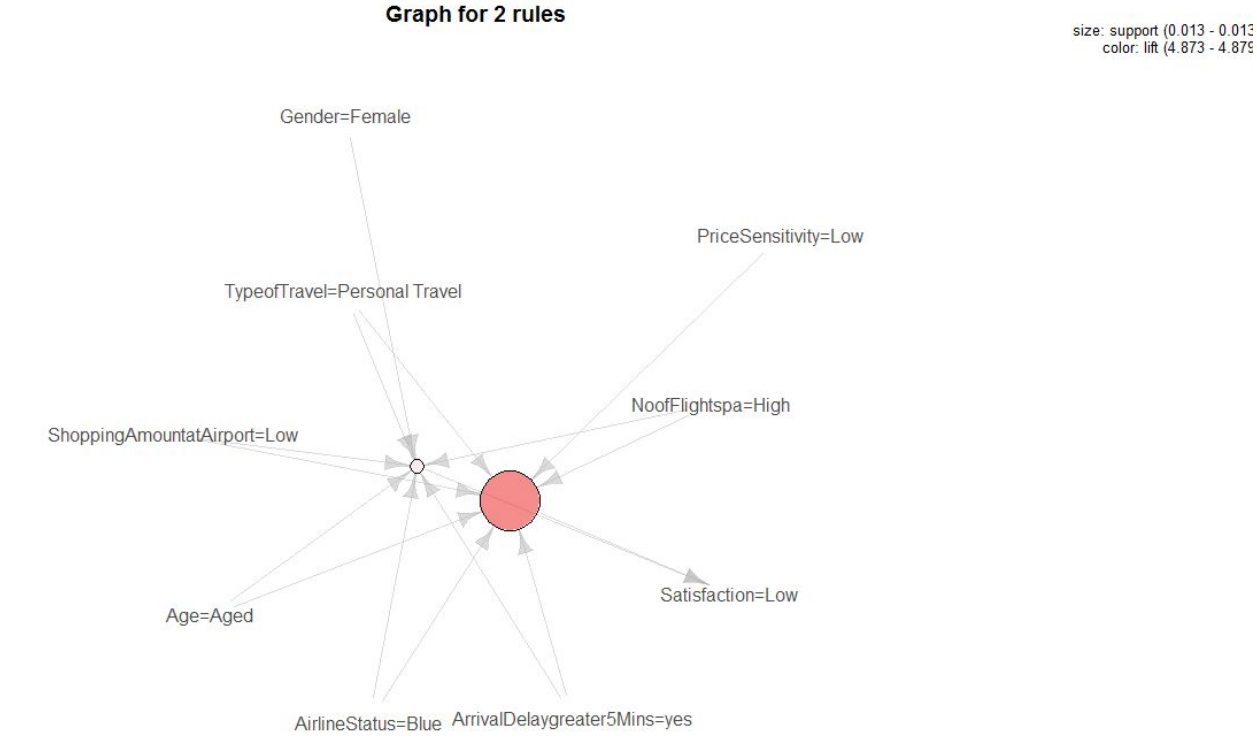
```
ruleset <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("df.model.satis_=Low")))
a.lowRules <- sort(ruleset, decreasing =TRUE, by = "lift")
b.lowRules<- head(a.lowRules, n= 5)
inspect(head(b[1:2]))
plot(b.lowRules[1:2], method="graph", control=list(type="items"))
plot(b.lowRules[1:2], method="paracoord", control=list())

#
```

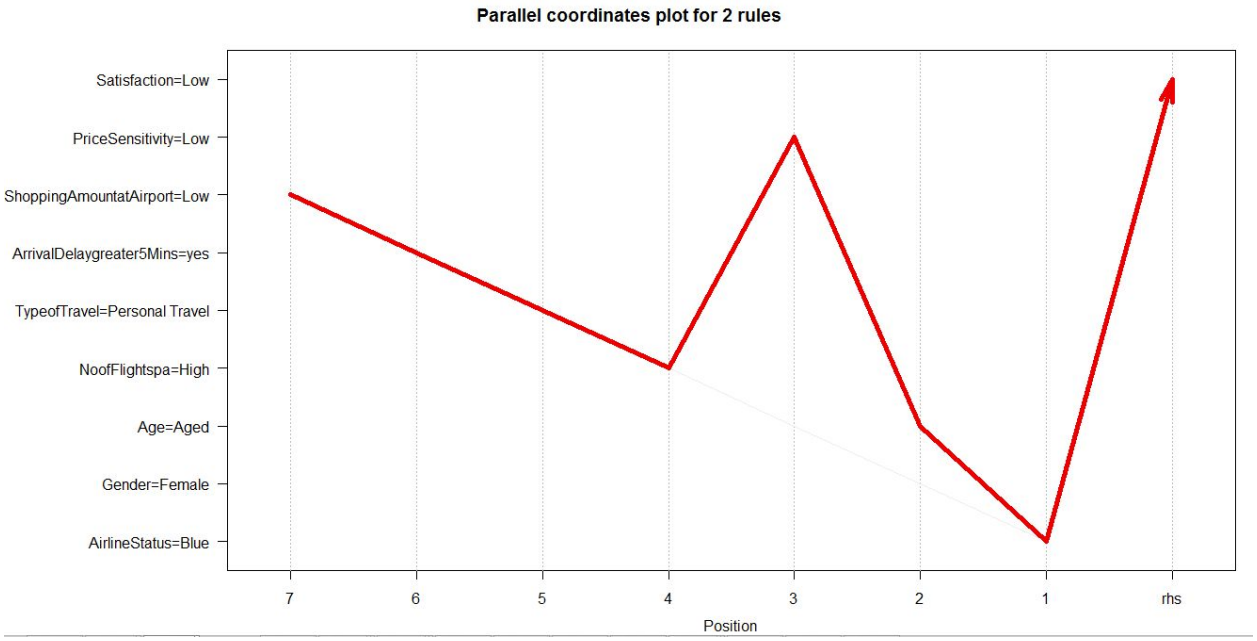
Rules:

As per our model and combination of our support and confidence values for high customer satisfaction around 465 rules were generated. Based on the highest values of lift we have taken 1st and 4th rule.

```
    lhs                                  rhs                   support confidence     lift count
[1] {AirlineStatus=Blue,
     Age=Aged,
     PriceSensitivity=Low,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.01286476  0.9969605 4.879166  1640
[2] {AirlineStatus=Blue,
     Gender=Female,
     Age=Aged,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.01267650  0.9956870 4.872934  1616
```

Plot:





From the above plot we can determine the most significant attributes which influence a customer to become low satisfied customer are -

- Age- Aged(60-85)
- Airline Status - Blue

- Price sensitivity - low
- Number of flights per annum - high
- Arrival delay greater than 5 minutes - Yes
- Shopping amount at Airport - Low
- Type of travel - personal travel
- Gender - Female

Association Rules on OnlyJet Airways:

Code:

```
# Applying Association rules to onlyjet

createBuckets <-function(vec)
{
  q <- quantile(vec, c(0.4, 0.6))
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec <= q[1]] <- "Low"
  vBuckets[vec > q[2]] <- "High"
  return (vBuckets)
}

summary(df.onlyjet$Satisfaction)
df.onlyjet$satis_ <- replicate(length(df.onlyjet$Satisfaction), "Average")
df.onlyjet$satis_[df.onlyjet$Satisfaction >= 4] <- "High" #values higher than 7 are termed as high
df.onlyjet$satis_[df.onlyjet$Satisfaction <= 2] <- "Low"
unique(df.onlyjet$satis_)

hist(df.onlyjet$Age)
unique(df.onlyjet$Age)
median(df.onlyjet$Age)
df.onlyjet$age_ <- replicate(length(df.onlyjet$Age), "Medium")          # 30 to 60 as medium
df.onlyjet$age_[df.onlyjet$Age >= 15 & df.onlyjet$Age <= 30] <- "Young"  #15  to 30 as Young
df.onlyjet$age_[df.onlyjet$Age > 60 & df.onlyjet$Age <= 85] <- "Aged"   # 60 to 85 as Aged

summary(df.onlyjet$PriceSensitivity)
hist(df.onlyjet$PriceSensitivity)
unique(df.onlyjet$PriceSensitivity) ## 0 to5
df.onlyjet$PriceSensitivity_<-createBuckets(df.onlyjet$PriceSensitivity)

hist(df.onlyjet$NoofFlightspa)
unique(df.onlyjet$NoofFlightspa) ##
median(df.onlyjet$NoofFlightspa)
df.onlyjet$NoofFlightspa_ <-createBuckets(df.onlyjet$NoofFlightspa)

unique(df.onlyjet$ShoppingAmountatAirport) ##
summary(df.onlyjet$ShoppingAmountatAirport)
hist(df.onlyjet$ShoppingAmountatAirport)
df.onlyjet$ShoppingAmountatAirport_<-createBuckets(df.onlyjet$ShoppingAmountatAirport)
```

```
ruleDF <- data.frame(df.onlyjet$satis_,df.onlyjet$AirlineStatus, df.onlyjet$Gender,df.onlyjet$age_, df.onlyjet$PriceSensitivity_,
                     df.onlyjet$NoofFlightspa_,df.onlyjet$TypeofTravel,df.onlyjet$ArrivalDelaygreater5Mins, df.onlyjet$ShoppingAmountatAirport_)
summary(ruleDF)
View(ruleDF)
nrow(ruleDF)

aRuleDF <- as(ruleDF,"transactions")
summary(itemFrequency(aRuleDF))
itemFrequencyPlot(aRuleDF)
inspect(aRuleDF)
summary(aRuleDF)
10/length(aRuleDF)
```

High Customer Satisfaction Rules:

```
ruleset.high <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("df.onlyjet.satis_=High")))

r1.high <- sort(ruleset.high, decreasing =TRUE, by = "lift")
r1.high.head <- head(r1.high, n= 5)
inspect(head(r1.high.head[c(4,6)]))

inspect(head(r1.high.head[c(4,6)]))

#plot(r1.high.head)
plot(r1.high.head[c(4,6)], method="graph", control=list(type="items"))
plot(r1.high.head[c(4,6)], method="paracoord", control=list(reorder = TRUE))
```

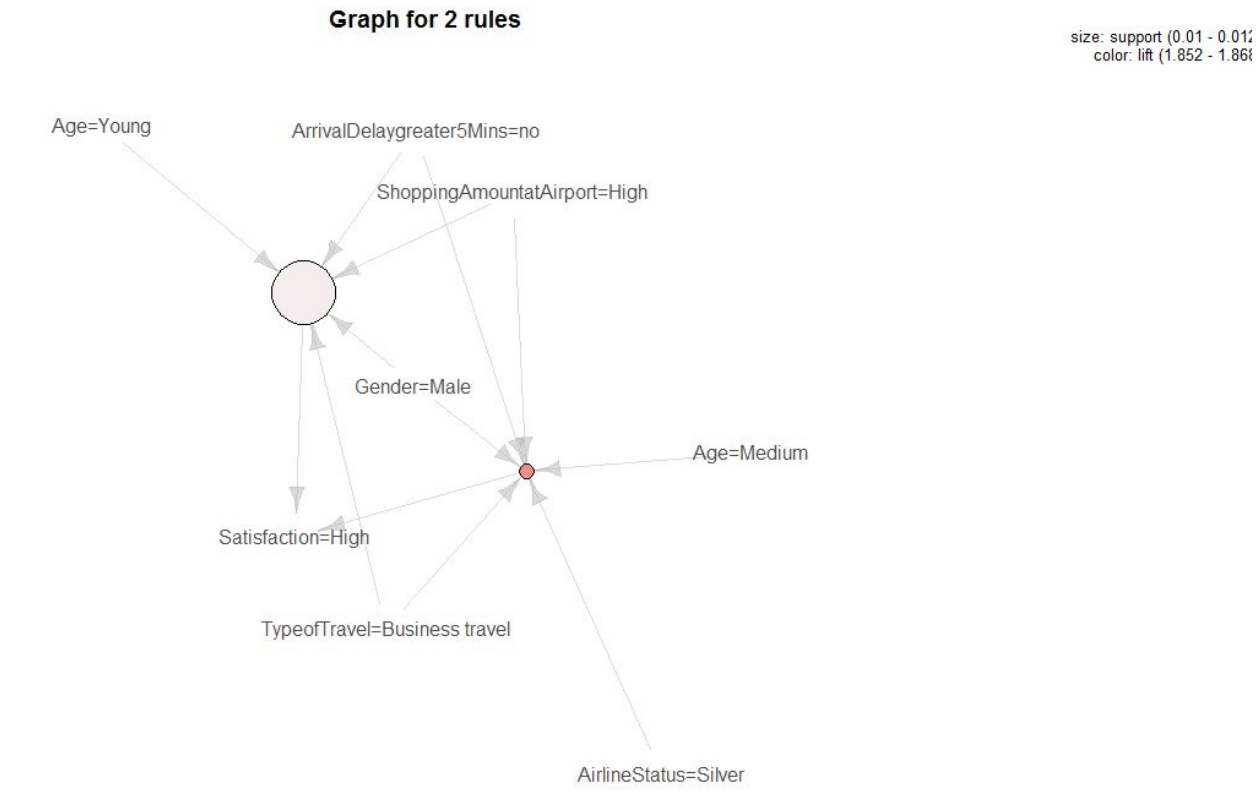
As per our model and combination of our support and confidence values for high customer satisfaction around 2204 rules were generated. Based on the highest values of lift we have taken 2 rules.

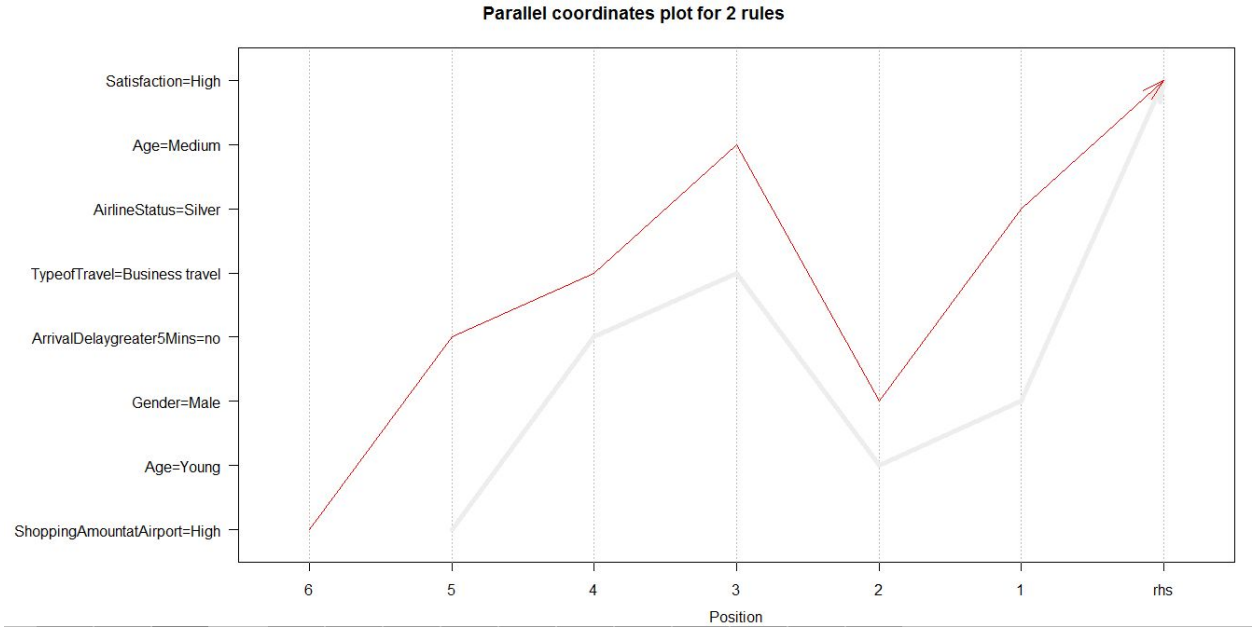
Rules:

```
> inspect(head(r1.high.head[c(4,6)]))
    lhs                                rhs                    support confidence     lift count
[1] {AirlineStatus=Silver,
     Gender=Male,
     Age=Medium,
     TypeofTravel=Business travel,
     ArrivalDelaygreater5Mins=no,
     ShoppingAmountatAirport=High} => {Satisfaction=High} 0.01043247  0.9322034 1.867950    55
[2] {Gender=Male,
     Age=Young,
     TypeofTravel=Business travel,
     ArrivalDelaygreater5Mins=no,
     ShoppingAmountatAirport=High} => {Satisfaction=High} 0.01157056  0.9242424 1.851998    61
```

Plot:

On running the association rule mining for highly satisfied customers, we have taken top 2 rules and plotted the below graph to determine the factors which influence customer satisfaction in airlines.


Graph for 2 rules
size: support (0.01 - 0.012
color: lift (1.852 - 1.868
Age=Young
ArrivalDelaygreater5Mins=no
ShoppingAmountatAirport=High
Gender=Male
Age=Medium
Satisfaction=High
TypeofTravel=Business travel
AirlineStatus=Silver


Parallel coordinates plot for 2 rules
Satisfaction=High
Age=Medium
AirlineStatus=Silver
TypeofTravel=Business travel
ArrivalDelaygreater5Mins=no
Gender=Male
Age=Young
ShoppingAmountatAirport=High
6
5
4
3
2
1
rhs
Position

From the above plot we can determine the most significant attributes which influence a customer to become a highly satisfied customer are -

- Age - Young (15-30)
- Gender - Male
- Airline Status - Silver
- Type of Travel - Business travel
- Arrival delay greater than 5 minutes - No
- Shopping amount at Airport - High

This shows that the above factors help the customer to be highly satisfied.

Low Customer Satisfaction Rules:

Code:

```
ruleset.low <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("df.onlyjet.satis_=Low")))
r2.low <- sort(ruleset.low, decreasing =TRUE, by = "lift")
r2.low.head <- head(r2.low, n = 10)
inspect(r2.low.head)

#plot(r1.high.head)
plot(r2.low.head[c(4,7)], method="graph", control=list(type="items"))
plot(r2.low.head[c(4,7)], method="paracoord", control=list(reorder = TRUE))
```

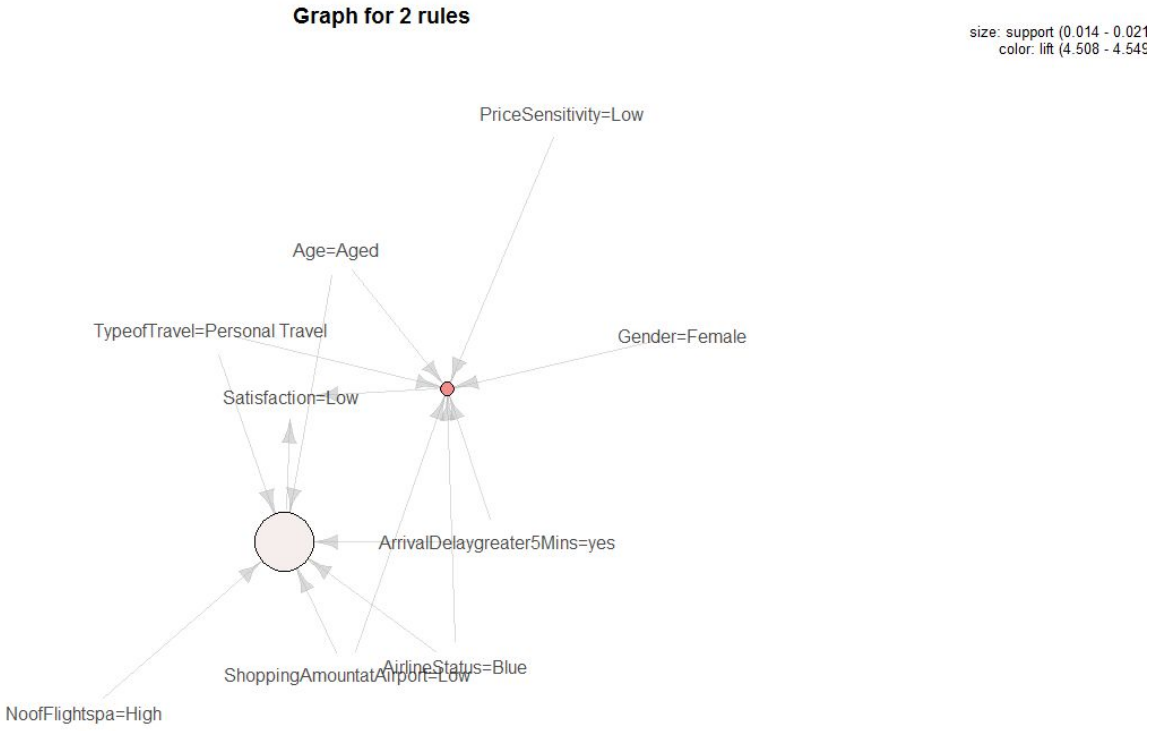
As per our model and combination of our support and confidence values for low customer satisfaction around 563 rules were generated. Based on the highest values of lift we have taken 2 rules.

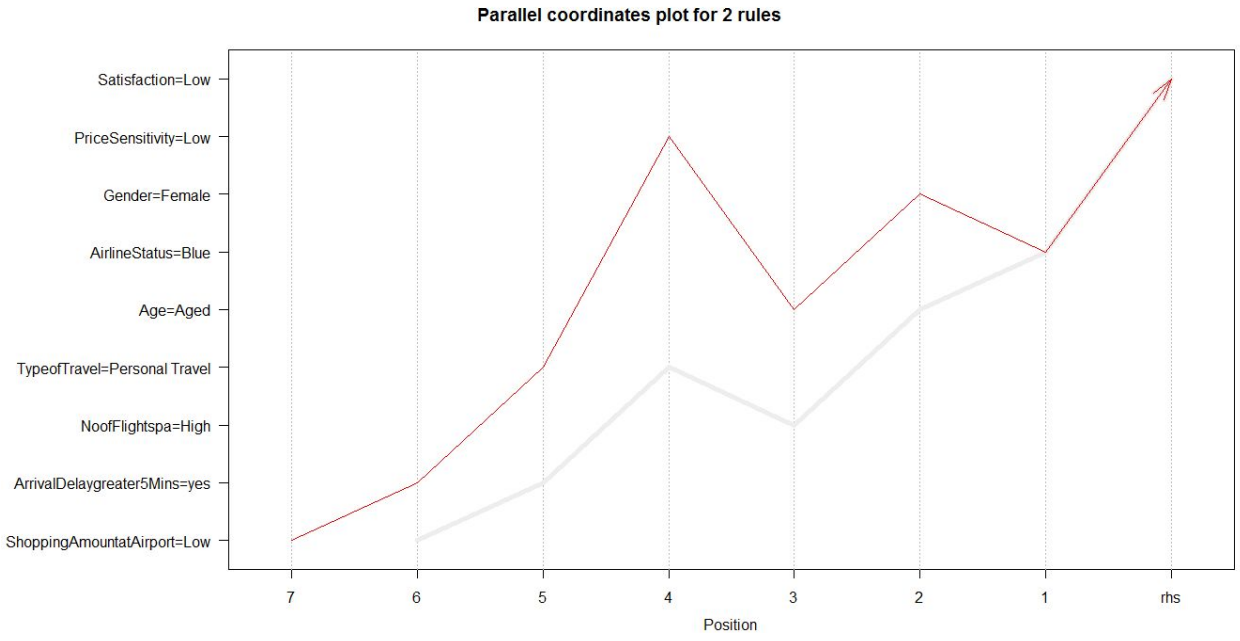
Rules:

```
> inspect(r2.low.head[c(4,7)])
    lhs                                  rhs                    support confidence     lift count
[1] {AirlineStatus=Blue,
     Gender=Female,
     Age=Aged,
     PriceSensitivity=Low,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.01422610  1.0000000 4.548749    75
[2] {AirlineStatus=Blue,
     Age=Aged,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes,
     ShoppingAmountatAirport=Low}  => {Satisfaction=Low} 0.02124431  0.9911504 4.508495   112
>
```

Plot:

On running the association rule mining for the Only Jet lowest satisfied customers , we have taken top 2 rules and plotted the below graph to determine the factors which influence customer satisfaction in airlines.





From the above plot we can determine the most significant attributes which influence a customer to become low satisfied customer are -

- Age- Aged(60-85)
- Gender- Female
- Airline Status - Blue

- Price sensitivity - low
- Arrival delay greater than 5 minutes - Yes
- Shopping amount at Airport - Low
- Type of travel - personal travel

Association Rules on GoingNorth Airways:

Code:

```
# Applying Association rules to GoingNorth

createBuckets <-function(vec)
{
  q <- quantile(vec, c(0.4, 0.6))
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec <= q[1]] <- "Low"
  vBuckets[vec > q[2]] <- "High"
  return (vBuckets)
}

summary(df.GoingNorth$Satisfaction)
df.GoingNorth$satis_ <- replicate(length(df.GoingNorth$Satisfaction), "Average")
df.GoingNorth$satis_[df.GoingNorth$Satisfaction >= 4] <- "High" #values higher than 7 are termed as high
df.GoingNorth$satis_[df.GoingNorth$Satisfaction <= 2] <- "Low"
unique(df.GoingNorth$satis_)

hist(df.GoingNorth$Age)
unique(df.GoingNorth$Age)
median(df.GoingNorth$Age)
df.GoingNorth$age_ <- replicate(length(df.GoingNorth$Age), "Medium")         # 30 to 60 as medium
df.GoingNorth$age_[df.GoingNorth$Age >= 15 & df.GoingNorth$Age <= 30] <- "Young"  #15  to 30 as Young
df.GoingNorth$age_[df.GoingNorth$Age > 60 & df.GoingNorth$Age <= 85] <- "Aged"   # 60 to 85 as Aged

summary(df.GoingNorth$PriceSensitivity)
hist(df.GoingNorth$PriceSensitivity)
unique(df.GoingNorth$PriceSensitivity) ## 0 to5
df.GoingNorth$PriceSensitivity_<-createBuckets(df.GoingNorth$PriceSensitivity)

hist(df.GoingNorth$NoofFlightspa)
unique(df.GoingNorth$NoofFlightspa) ##
median(df.GoingNorth$NoofFlightspa)
df.GoingNorth$NoofFlightspa_ <-createBuckets(df.GoingNorth$NoofFlightspa)

unique(df.GoingNorth$ShoppingAmountatAirport) ##
summary(df.GoingNorth$ShoppingAmountatAirport)
hist(df.GoingNorth$ShoppingAmountatAirport)
df.GoingNorth$ShoppingAmountatAirport_<-createBuckets(df.GoingNorth$ShoppingAmountatAirport)


ruleDF <- data.frame(df.GoingNorth$satis_,df.GoingNorth$AirlineStatus, df.GoingNorth$Gender,df.GoingNorth$age_, df.GoingNorth$PriceSensitivity_,
                     +df.GoingNorth$NoofFlightspa_,df.GoingNorth$TypeofTravel,df.GoingNorth$ArrivalDelaygreater5Mins,
                     +df.GoingNorth$ShoppingAmountatAirport_)
summary(ruleDF)
View(ruleDF)
nrow(ruleDF)

aRuleDF <- as(ruleDF,"transactions")
summary(itemFrequency(aRuleDF))
itemFrequencyPlot(aRuleDF)
inspect(aRuleDF)
summary(aRuleDF)
10/length(aRuleDF)
```

High Customer satisfaction Rules:

Code:

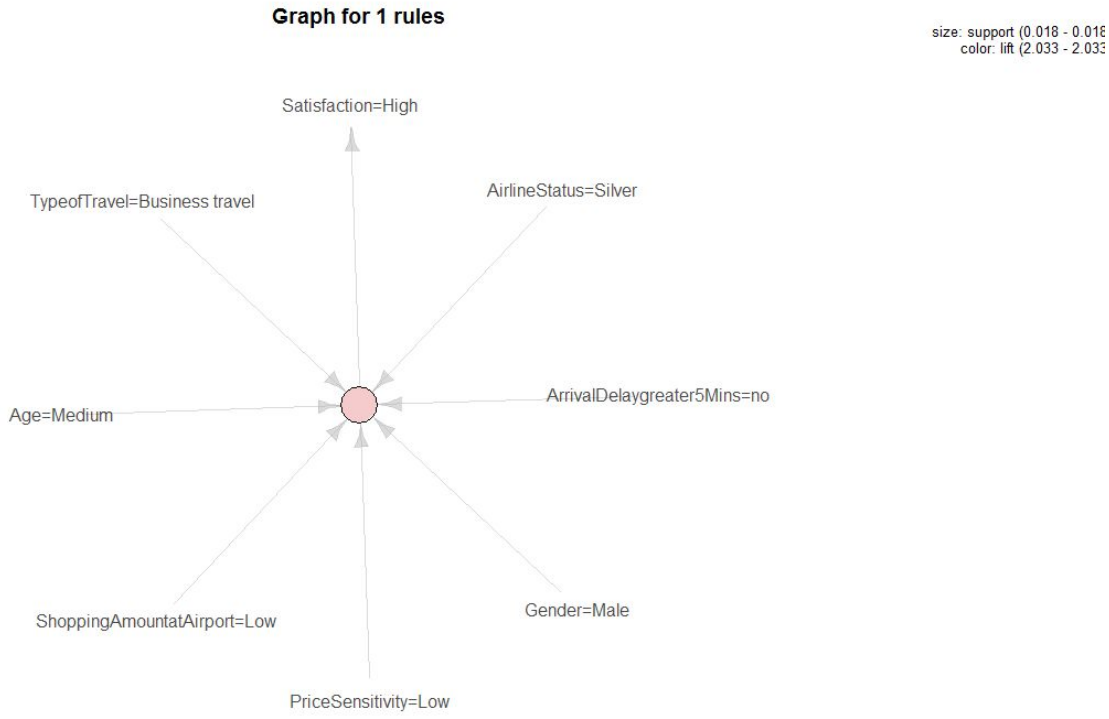
```
ruleset.high <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("df.GoingNorth.satis_=High")))

r1.high <- sort(ruleset.high, decreasing =TRUE, by = "lift")
r1.high.head <- head(r1.high, n= 10)
inspect(head(r1.high.head[c(4,6)]))

#plot(r1.high.head)
plot(r1.high.head[c(4,6)], method="graph", control=list(type="items"))
plot(r1.high.head[c(4,6)], method="paracoord", control=list(reorder = TRUE))
```

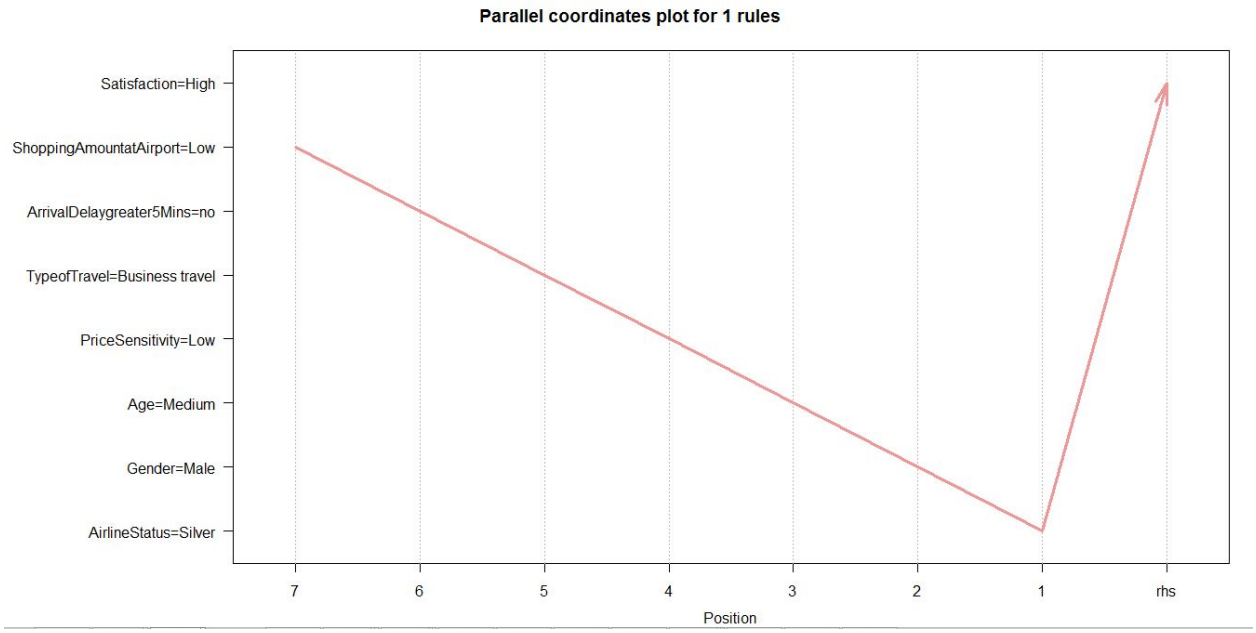
As per our model and combination of our support and confidence values for low customer satisfaction around 2054 rules were generated. Based on the highest values of lift we have taken 1 rule.

```
    lhs                                  rhs                     support confidence     lift count
[1] {AirlineStatus=Silver,
     Gender=Male,
     Age=Medium,
     PriceSensitivity=Low,
     TypeofTravel=Business travel,
     ArrivalDelaygreater5Mins=no,
     ShoppingAmountatAirport=Low}  => {Satisfaction=High} 0.01792574  0.9655172 2.032531    28
> |
```

Plot:

On running the association rule mining for the GoingNorth highly satisfied customers , we have taken top 1 rules and plotted the below graph to determine the factors which influence customer satisfaction in airlines.





From the above plot we can determine the most significant attributes which influence a customer to become highly satisfied customer are -

- Age- Medium(30-60)
- Gender- Male
- Airline Status - Silver
- Price sensitivity - low
- Arrival delay greater than 5 minutes - No
- Shopping amount at Airport - Low
- Type of travel - Business travel

Low Customer Satisfaction Rules:

Code

```
ruleset.low <- apriori(aRuleDF, parameter = list(support=0.01, confidence = 0.5), appearance = list(default="lhs",rhs=("Satisfaction=Low")))
r2.low <- sort(ruleset.low, decreasing =TRUE, by = "lift")
r2.low.head <- head(r2.low, n = 10)
inspect(r2.low.head[c(3,6)])
```

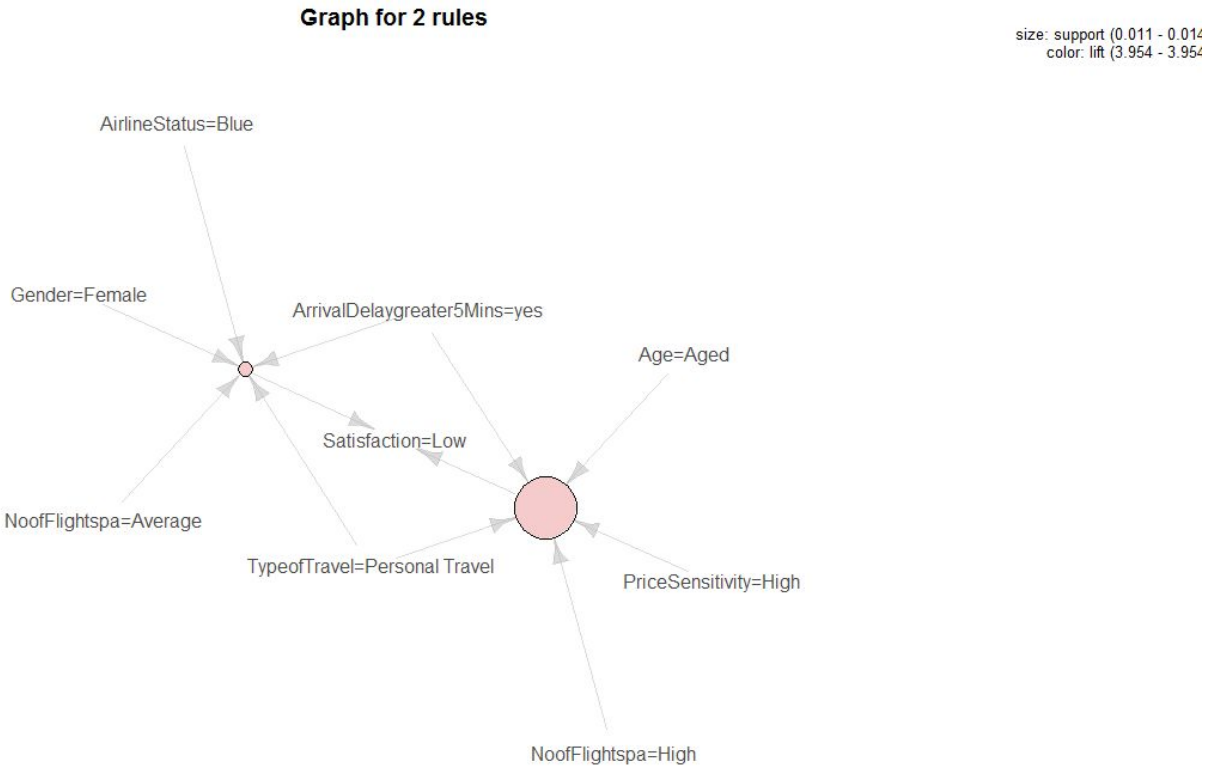
As per our model and combination of our support and confidence values for low customer satisfaction around 721 rules were generated. Based on the highest values of lift we have taken 2 rules.

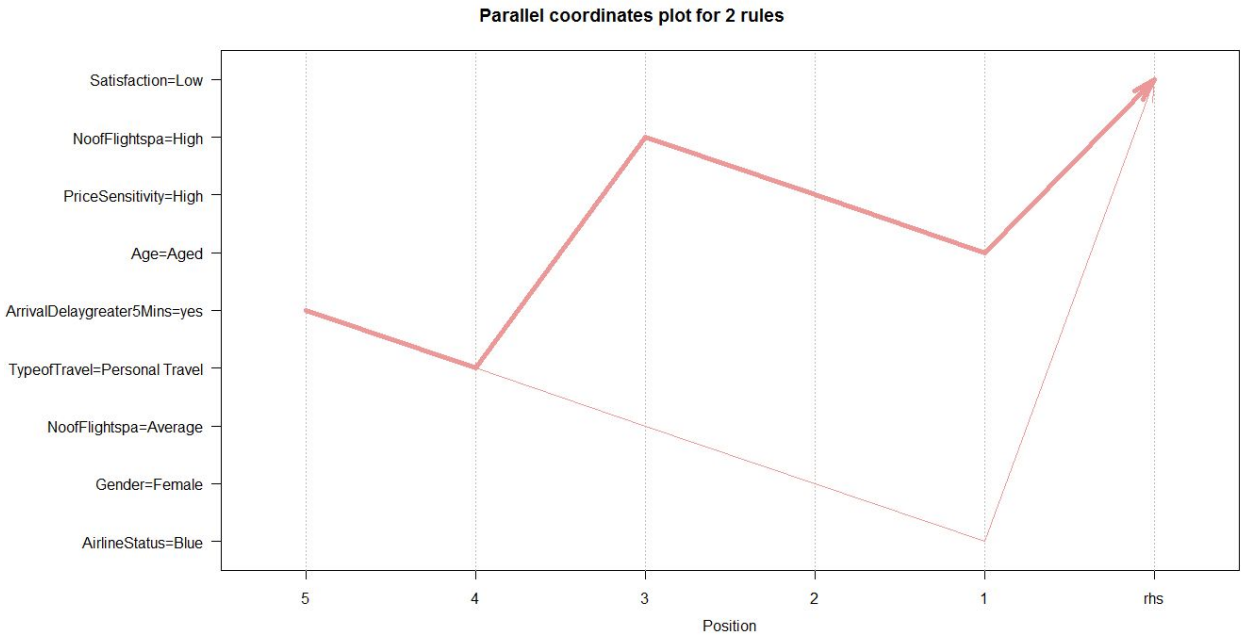
Rules:

```
> inspect(r2.low.head[c(3,6)])
    lhs                                rhs                    support confidence    lift count
[1] {AirlineStatus=Blue,
     Gender=Female,
     NoofFlightspa=Average,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes} => {Satisfaction=Low} 0.01088348          1 3.95443    17
[2] {Age=Aged,
     PriceSensitivity=High,
     NoofFlightspa=High,
     TypeofTravel=Personal Travel,
     ArrivalDelaygreater5Mins=yes} => {Satisfaction=Low} 0.01408451          1 3.95443    22
```

Plot:

On running the association rule mining for the Only Jet lowest satisfied customers , we have taken top 2 rules and plotted the below graph to determine the factors which influence customer satisfaction in airlines.





From the above plot we can determine the most significant attributes which influence a customer to become low satisfied customer are -

- Age- Aged(60-85)
- Gender- Female
- Airline Status - Blue
- Price sensitivity - High
- Arrival delay greater than 5 minutes - Yes
- Type of travel - personal travel

Support Vector Machine

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Here, we have considered variables from the above three methods i.e. linear modelling, association rules mining and correlation matrix and applied them to support vector machine to cross validate the significance of the variables.

The outcomes will highlight if the accuracy of SVM model is high then the selected variables would be highly significant.

Support Vector Machine on sampled dataset (of 50,000 rows)

Code:

```
#Build a Model using ksvm( ) for sampled random dataset of 50,000 rows
Svmmodel1 <- ksvm(CustSat~AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTrav
                  + ArrivalDelaygreater5Mins, data = trainData, kernel="rbfdot", kpar="automatic", C

Svmmodel1   #Training error :  0.199622
#Cross validation error : 0.205142
#predict the outcome for sampled DF
SvmPred <- predict(Svmmodel1, testData, type = "response")
pred <- data.frame(SvmPred)
table(testData$CustSat)
table(pred)
compTable <- data.frame(testData$CustSat,pred$SvmPred)
table(compTable)
#Calculate error rate for this sampled dataset
Error <- (table(compTable)[1,2]+table(compTable)[2,1])/nrow(testData)
Error  #0.2045084
Accuracy <- (table(compTable)[1,1]+table(compTable)[2,2])/nrow(testData)
Accuracy #0.7954916
```

Here, we have sampled 50000 attributes from the whole dataset and divided into test data and training data. Then we used this training data to build SVM model and applied test data to predict our results i.e. the accuracy reflects out to be 79%.

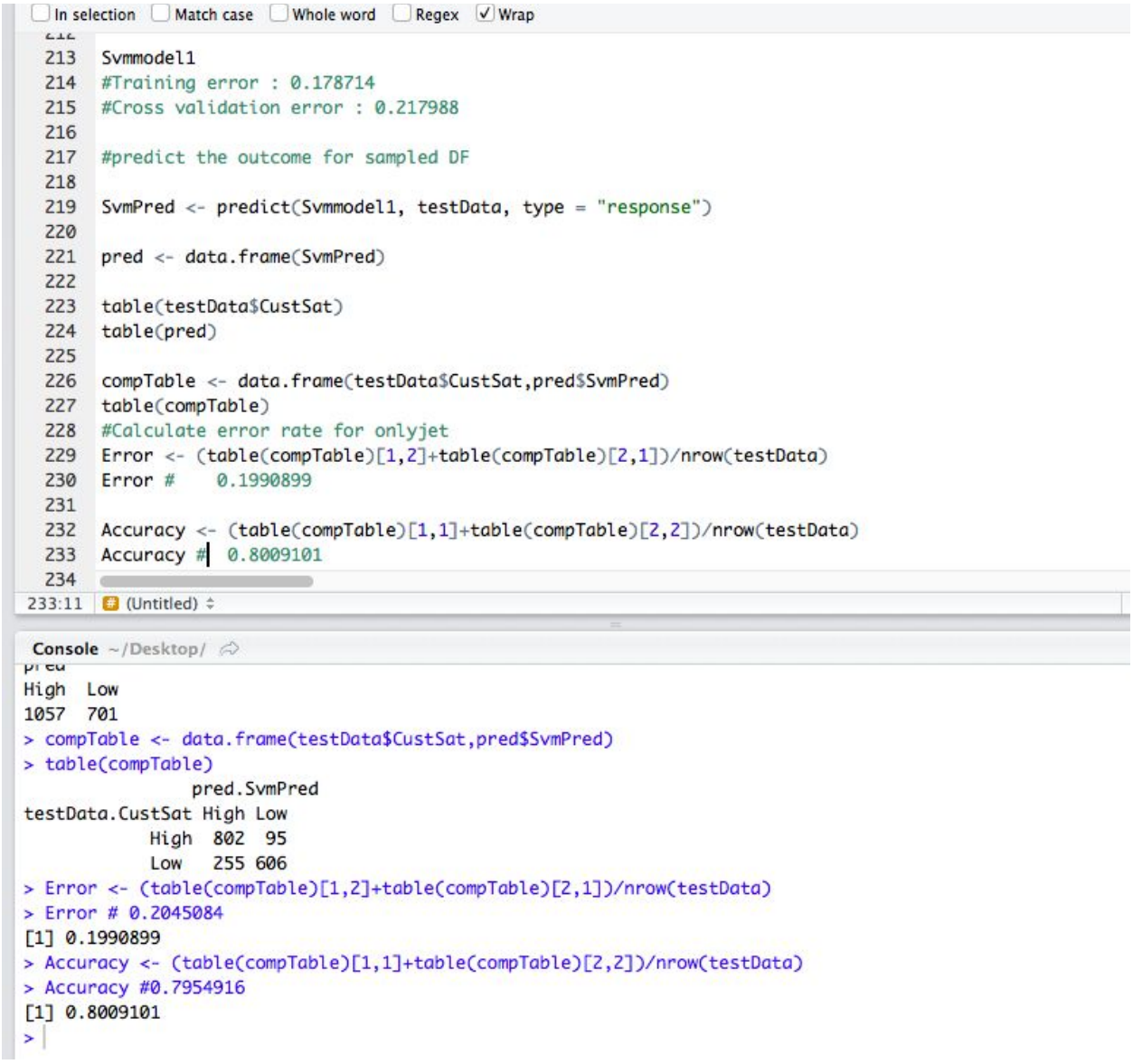
Support Vector Machine on Going North Airline

```
#Build a Model using ksvm( )
Svmmodel1 <- ksvm(CustSat~AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel+ ShoppingAmountatAirport
                  + ArrivalDelaygreater5Mins, data = trainData, kernel="rbfdot", kpar="automatic", C=5, cross=3, prob.model=TRUE)



Svmmodel1
#Training error : 0.144092
#Cross validation error : 0.222863

#predict the outcome for sampled DF

SvmPred <- predict(Svmmodel1, testData, type = "response")

pred <- data.frame(SvmPred)

table(testData$CustSat)
table(pred)

compTable <- data.frame(testData$CustSat,pred$SvmPred)
table(compTable)
#Calculate error rate for onlyjet
Error <- (table(compTable)[1,2]+table(compTable)[2,1])/nrow(testData)
Error #    0.2284069

Accuracy <- (table(compTable)[1,1]+table(compTable)[2,2])/nrow(testData)
Accuracy #  0.7715931
```

```
> Svmmodel1
Support Vector Machine object of class "ksvm"

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

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.167529198810679

Number of Support Vectors : 529

Objective Function Value : -1924.741
Training error : 0.144092
Cross validation error : 0.222863
Probability model included.
> SvmPred <- predict(Svmmodel1, testData, type = "response")
> pred <- data.frame(SvmPred)
> table(testData$CustSat)

High  Low
 249  272
> table(pred)
pred
High  Low
 286  235
> compTable <- data.frame(testData$CustSat,pred$SvmPred)
> table(compTable)
               pred.SvmPred
testData.CustSat High Low
            High  208  41
            Low    78 194
> Error <- (table(compTable)[1,2]+table(compTable)[2,1])/nrow(testData)
> Error #    0.1990899
[1] 0.2284069
> Accuracy <- (table(compTable)[1,1]+table(compTable)[2,2])/nrow(testData)
> Accuracy #  0.8009101
[1] 0.7715931
>
```

Here, we filtered the dataset for Going North airline and divided into test data and training data. Then we used this training data to build SVM model and applied test data to predict our results i.e. the accuracy reflects out to be 77%.

Support Vector Machine on OnlyJets



```
In selection  Match case  Whole word  Regex  Wrap
213  Svmmodel1
214  #Training error : 0.178714
215  #Cross validation error : 0.217988
216
217  #predict the outcome for sampled DF
218
219  SvmPred <- predict(Svmmodel1, testData, type = "response")
220
221  pred <- data.frame(SvmPred)
222
223  table(testData$CustSat)
224  table(pred)
225
226  compTable <- data.frame(testData$CustSat,pred$SvmPred)
227  table(compTable)
228  #Calculate error rate for onlyjet
229  Error <- (table(compTable)[1,2]+table(compTable)[2,1])/nrow(testData)
230  Error #    0.1990899
231
232  Accuracy <- (table(compTable)[1,1]+table(compTable)[2,2])/nrow(testData)
233  Accuracy #  0.8009101
234
233:11  (Untitled)

Console ~/Desktop/
pred
High  Low
1057  701
> compTable <- data.frame(testData$CustSat,pred$SvmPred)
> table(compTable)
               pred.SvmPred
testData.CustSat High Low
            High  802  95
            Low   255 606
> Error <- (table(compTable)[1,2]+table(compTable)[2,1])/nrow(testData)
> Error # 0.2045084
[1] 0.1990899
> Accuracy <- (table(compTable)[1,1]+table(compTable)[2,2])/nrow(testData)
> Accuracy #0.7954916
[1] 0.8009101
>
```

Here, we filtered the dataset for OnlyJets airline and divided into test data and training data. Then we used this training data to build SVM model and applied test data to predict our results i.e. the accuracy reflects out to be 80%.

Decision Tree

Decision Tree on entire dataset

Decision tree algorithm helps us explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome. A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

To further validate our results, we used decision tree model to check the accuracy of the significant variables.

```
##Decision Tree-----------------------------------------------
#-----------------------------------------------------------
df.model$Target <- df.model$Satisfaction
df.model$Target <- ifelse(df.model$Satisfaction >3,"1","0")
View(df.model)
tree1<- rpart(Target~AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel+ ShoppingAmountatAirport
              + ArrivalDelaygreater5Mins, data = trainData )
tpred <- predict(tree1,testData)
View(tpred)
# trainData and testData
randIndex <- sample(1:dim(df.model[1]))   # indexes of sample dataset
length(randIndex)                          # length of randIndex is same as total rows of hotel survey dataset
# 3. create a training data set and a test data set.
cutPoint2_3 <- floor(2*dim(df.model[1])/3)
trainData <- df.model[randIndex[1:cutPoint2_3[1]],]   #Build our training set from the first 6666 rows of hotelSurvey Dataset
testData <- df.model[randIndex[(cutPoint2_3[1]+1):dim(df.model)[1]],]

dim(trainData)
dim(testData)

tpred <-data.frame(tpred)
tpred <- round(tpred)
View(tpred)
compTable <- table(testData$Target,tpred$X1)
#Accura
Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
Accuracy #0.7847696
```

```
> compTable

       0     1
  0 12961  7621
  1  1525 20387
> (compTable)[1,1]
[1] 12961
> #Accura
> Accuracy <- (compTable)[1,1]+(compTable)[2,2]/nrow(testData)
> Accuracy
[1] 12961.48
> nrow(testData)
[1] 42494
> #Accura
> Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
> Accuracy
[1] 0.7847696
```

So here, with decision tree model, a tree of all variables is constructed and then variable condition is checked that starts from the root of the tree and progresses towards the stem of the tree.

Decision Tree on Only Jets

```
#=======================
#####for OnlyJets
df.onlyjet$Target <- df.onlyjet$Satisfaction
df.onlyjet$Target <- ifelse(df.onlyjet$Satisfaction >3,"1","0")
View(df.onlyjet)
# trainData and testData
randIndex <- sample(1:dim(df.onlyjet[1]))   # indexes of sample dataset
length(randIndex)                           # length of randIndex is same as total rows of hotel survey dataset
# 3. create a training data set and a test data set.
cutPoint2_3 <- floor(2*dim(df.onlyjet[1])/3)
trainData <- df.onlyjet[randIndex[1:cutPoint2_3[1]],]   #Build our training set from the first 6666 rows of hotelSurvey Dataset
testData <- df.onlyjet[randIndex[(cutPoint2_3[1]+1):dim(df.onlyjet)[1]],]

dim(trainData)
dim(testData)
##
tree1<- rpart(Target~AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel+ ShoppingAmountatAirport
              + ArrivalDelaygreater5Mins, data = trainData )
tpred <- predict(tree1,testData)
tpred <-data.frame(tpred)
tpred <- round(tpred)
View(tpred)
compTable <- table(testData$Target,tpred$X1)
compTable
#Accura
Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
Accuracy # 0.7758817
```

```
> compTable <- table(testData$Target,tpred$X1)
> compTable

     0   1
  0 565 305
  1  89 799
> #Accura
> Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
> Accuracy #0.7847696
[1] 0.7758817
```

Here, with decision tree model, a tree of Only Jet dataset is constructed and then variable condition is checked that starts from the root of the tree and progresses towards the stem of the tree.

Decision Tree on Going North

```
#=======================
#####for GoingNorth
df.GoingNorth$Target <- df.GoingNorth$Satisfaction
df.GoingNorth$Target <- ifelse(df.GoingNorth$Satisfaction >3,"1","0")
View(df.GoingNorth)
# trainData and testData
randIndex <- sample(1:dim(df.GoingNorth[1]))   # indexes of sample dataset
length(randIndex)                             # length of randIndex is same as total rows of hotel survey dataset
# 3. create a training data set and a test data set.
cutPoint2_3 <- floor(2*dim(df.GoingNorth[1])/3)
trainData <- df.GoingNorth[randIndex[1:cutPoint2_3[1]],]   #Build our training set from the first 6666 rows of hotelSurvey Dataset
testData <- df.GoingNorth[randIndex[(cutPoint2_3[1]+1):dim(df.GoingNorth)[1]],]

dim(trainData)#1041   25
dim(testData)#521  25

tree1<- rpart(Target~AirlineStatus + Gender+ Age + PriceSensitivity + NoofFlightspa+ TypeofTravel+ ShoppingAmountatAirport
              + ArrivalDelaygreater5Mins, data = trainData )
tpred <- predict(tree1,testData)
tpred <-data.frame(tpred)
tpred <- round(tpred)
View(tpred)
compTable <- table(testData$Target,tpred$X1)
compTable
#Accura
Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
Accuracy # 0.7619962
```

```
> compTable <- table(testData$Target,tpred$X1)
> compTable

    0   1
0 184  92
1  32 213
> #Accura
> Accuracy <- ((compTable)[1,1]+(compTable)[2,2])/nrow(testData)
> Accuracy # 0.7758817
[1] 0.7619962
```

Here, with decision tree model, a tree of Going North dataset is constructed and then variable condition is checked that starts from the root of the tree and progresses towards the stem of the tree.

Validation

- Significant variables observed from Linear modelling, visualizations and correlation matrix (mentioned below) are validated using 3 modelling techniques of Association rules, Support vector machine and Decision tree.
- Significant Variables on which these models are applied:
    - Airline Status
    - Gender
    - Age
    - Price sensitivity
    - Number of flights per annum
    - Type of travel
    - Shopping Amount at airport
    - Arrival delay greater than 5 minutes

- Models:

1) **In Association rules**, we got combination of rules with these significant variables.

Customer Satisfaction = Low when,
Age = Age range (60-85)
Gender = Female
Airline Status = Blue
Price sensitivity = High
Arrival delay greater than 5 minutes = Yes
Type of travel = personal travel
This proves the insights we are getting from visualizations of these columns.

2) **In SVM & Decision tree**, we built a model using these significant variables as input (training data) and checked accuracy of the models with prediction on testing data

Prediction Efficiency from SVM: 80.09%
Prediction Efficiency from Decision Tree: 77.58

With higher accuracy, these variables are proved to be important and impacting customer satisfaction the most.

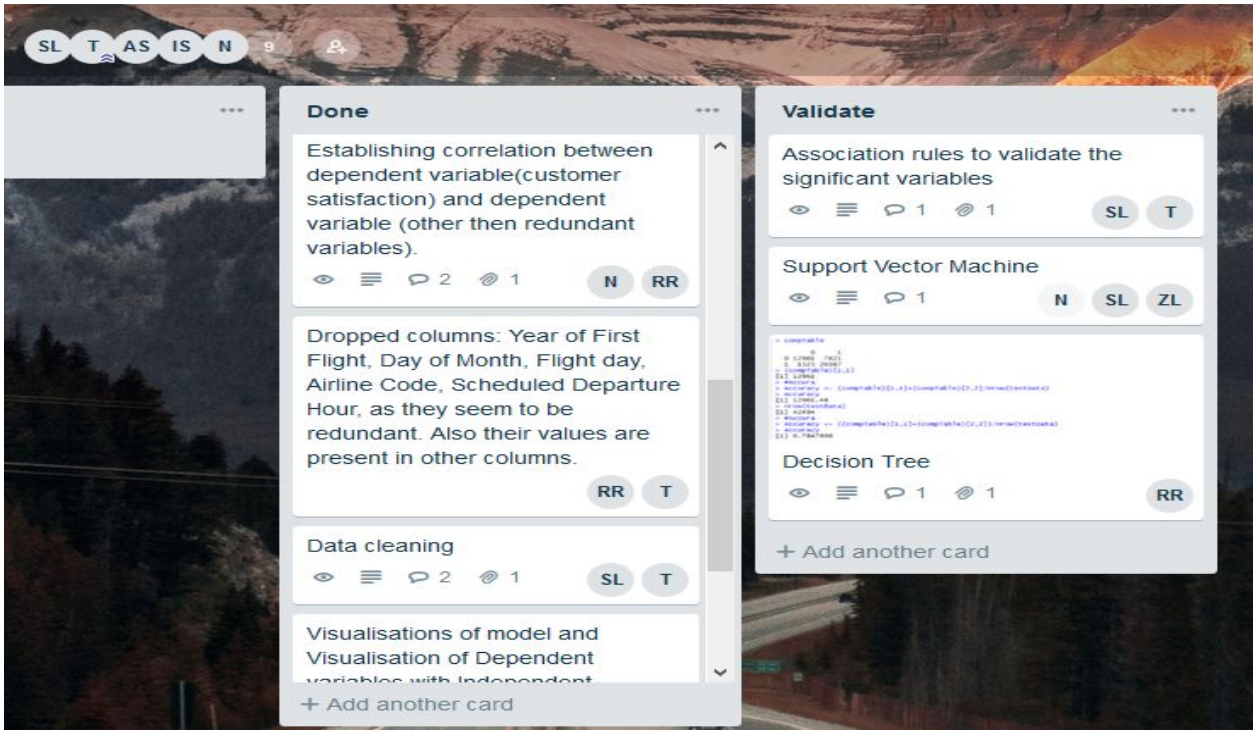
Actionable Insights

Through Visualizations & modelling these are inferences,

- Female customers are giving lower satisfaction ratings and could be offered better deals for improvement
- Aged (60-85) customers are using “Personal type of travel” & their average satisfaction is low. Hence, personal travel amenities for senior customers should be increased and they should be well-maintained.
- Premium status classes likes Gold, Platinum are having good customer satisfaction whereas Blue has lower ratings. Therefore, extra services should be provided for convenience of customers travelling through Blue status.
- Customers taking more flights per year are giving low satisfaction ratings. Thus, loyal customers should be provided with discounts and points which might satisfy them in long run.

Trello Update

Task/ Process	Details	Group Members
Data Cleaning	Column renaming, treating NA values, correcting column datatypes	Sangam, Trisha
Linear Modelling	Created more than 10 models to check if R square value with forward modelling. This helped us in filtering down number of significant variables.	Trisha, Aditi
Visualisations	Plotted maps, density plots, ggplots to understand trends and dependency	Aditi, Zhida, Rahul
Correlation Matrix	Created the correlation matrix and established relationship between dependent variable (Satisfaction) and independent variables. This helped us in filtering down number of significant variables.	Rahul, Nahnsan
Association Rules	Created rules and was able to validate the significant variables from the dataset and their impact on Customer Satisfaction. Performed this step on entire dataset and specific client airline data	Sangam, Trisha
Support Vector Machine	We have sampled 70000 attributes from the whole dataset and divided into test data and training data . Then we used this training data to build SVM model and applied test data to predict our results i.e. the accuracy reflects out to be 79%.	Zhida, Nahnsan
Decision Tree	Build a model using decision tree (rpart) and check for accuracy of model for input (significant variables)	Rahul



Attached R code:

Final project File.R