PROJECT REPORT
IST 687
GROUP 4

Prepared by:
Gaurav Salvi
Nikhil Patil
Purva Kedari
Shuying Zhao
Yue Wang

TABLE OF CONTENTS

1. Problem Statement……………………………………………………………………………………………………2
2. Scope of the Project……………………………………………………………………………………………………2
3. Context……2
4. Business Questions……………………………………………………………………………………………………2
5. Data Munging………2
6. Selecting a Client (Cheapseats Airlines)………………………………………………………………………3
7. Cleaning the dataset…………………………………………………………………………………………………6
8. Descriptive Statistics…………………………………………………………………………………………………7
9. Linear Regression Modeling………………………………………………………………………………………14
10. Association Rules……………………………………………………………………………………………………20
11. Support Vector Machines…………………………………………………………………………………………28
12. Actionable Insights…………………………………………………………………………………………………29
13. Recommendations……………………………………………………………………………………………………31
14. Appendix………31

Problem Statement

The project is aimed at analyzing the data from the dataset of customers flying within United States and to generate actionable insights by predicting customers with low satisfaction. The aim is to predict low satisfied customers for one of the clients from the dataset and identify factors affecting the satisfaction. Also suggest feedback or suggestions to improve the satisfaction for those with low satisfaction.

Scope of Project

Following tasks have been considered during this project to gain actionable insights from the data. After selecting one client airways from the dataset given, following tasks will be executed.

- Analyzing critical business questions.
- Cleaning the data set available.
- Analyze the dataset and discuss about the prospective columns which will help us gain insights on customer satisfaction.
- Predict customers with low satisfaction.
- Creating visualizations to support business questions and provide visual insights.
- Providing recommendations/suggestions to the client to increase the customer satisfaction.

Business Questions

- Predict customers with low satisfaction.
- What factors affects the satisfaction of customers?
- Out of all the factors, which attribute has the most influence on customer satisfaction and why?
- What recommendations and suggestions should be provided to the client to improve their customer satisfaction?

Data Munging

Importing the Dataset

First step is to import the dataset available on our system. The file available is .csv type file which contains all the data. We install the necessary libraries required for performing the analysis on the data. Once we import the data we put the data in a variable and convert the file into a data frame for further analysis. We also run the structure and summary functions to know the dataset and get familiar with the values in it.

Code Snippet -

```r
library(readr)
setwd("~/Desktop")
Satisfaction_Survey <- read.csv("Satisfaction Survey.csv", stringAsFactors = FALSE)
str(Satisfaction_Survey)
```

```
## 'data.frame':    129889 obs. of  28 variables:
##  $ Satisfaction                  : Factor w/ 10 levels "1","2","2.5",..: 9 6 3 6 
10 10 5 6 6 6 ...
##  $ Airline.Status                : Factor w/ 4 levels "Blue","Gold",..: 1 1 1 1 
4 2 2 4 1 1 ...
##  $ Age                           : int  31 56 21 43 49 49 35 33 44 51 ...
##  $ Gender                        : Factor w/ 2 levels "Female","Male": 2 2 1 2 2 
1 2 2 1 1 ...
##  $ Price.Sensitivity             : int  1 2 2 1 1 1 1 1 1 1 ...
##  $ Year.of.First.Flight          : int  2007 2006 2006 2007 2006 2010 2011 2010 
2003 2005 ...
##  $ No.of.Flights.p.a.            : int  28 41 8 9 14 0 15 4 8 12 ...
##  $ X..of.Flight.with.other.Airlines: int  7 3 7 9 10 4 5 17 6 7 ...
##  $ Type.of.Travel                : Factor w/ 3 levels "Business travel",..: 1 1 
3 1 1 1 1 1 1 1 ...
##  $ No..of.other.Loyalty.Cards    : int  2 0 0 2 0 1 0 2 0 0 ...
##  $ Shopping.Amount.at.Airport    : int  0 15 0 10 8 0 0 0 0 25 ...
##  $ Eating.and.Drinking.at.Airport  : int  75 60 135 45 26 65 60 90 90 80 ...
##  $ Class                         : Factor w/ 3 levels "Business","Eco",..: 1 1 1 
2 2 2 2 2 2 2 ...
##  $ Day.of.Month                  : int  18 11 25 20 25 16 6 5 21 19 ...
##  $ Flight.date                   : Factor w/ 90 levels "1/1/14","1/10/14",..: 69 
3 18 44 49 8 87 55 14 11 ...
##  $ Airline.Code                  : Factor w/ 14 levels "AA","AS","B6",..: 9 9 9 
9 9 9 9 9 9 9 ...
##  $ Airline.Name                  : Factor w/ 14 levels "Cheapseats Airlines Inc.
",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Orgin.City                    : Factor w/ 295 levels "Aberdeen, SD",..: 169 1
69 179 169 179 169 169 169 179 169 ...
##  $ Origin.State                  : Factor w/ 52 levels "Alabama","Alaska",..: 51 
51 51 51 51 51 51 51 51 51 ...
##  $ Destination.City              : Factor w/ 296 levels "Aberdeen, SD",..: 73 73 
73 73 73 73 73 73 73 73 ...
##  $ Destination.State             : Factor w/ 52 levels "Alabama","Alaska",..: 44 
44 44 44 44 44 44 44 44 44 ...
##  $ Scheduled.Departure.Hour      : int  15 11 12 11 12 18 6 18 12 18 ...
##  $ Departure.Delay.in.Minutes    : int  0 2 34 26 0 0 0 0 0 0 ...
##  $ Arrival.Delay.in.Minutes      : int  3 5 14 39 0 0 0 1 0 0 ...
##  $ Flight.cancelled              : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 
1 1 1 ...
##  $ Flight.time.in.minutes        : int  134 120 122 141 144 123 119 138 114 118 
...
##  $ Flight.Distance               : int  821 821 853 821 853 821 821 821 853 821 
...
##  $ Arrival.Delay.greater.5.Mins  : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 
1 1 1 ...
```

```r
summary(Satisfaction_Survey$Satisfaction)
```

```
##         1         2       2.5         3       3.5         4 4.00.2.00 
##      2999     23587         2     36984         2     53758         2 
##   4.00.5       4.5         5 
##         1         2     12552
```

Selecting the client

Now that we have our dataset, we must select one client that we are going to provide insights to about their customers. Now to select that one client, we perform quantitative analysis to see which airlines has the lowest number of unsatisfied customers.

Now to select the airline, we first see which airline has the how many number of customers. To know that we group the dataset by the airline names and then see the number of rows for each airline, count them and display that for each airline. By doing this we see how many customers each individual airline has.

Code Snippet:

```
data_full<-group_by(dataset, Airline.Name)
data_summ_full<-summarise(data_full, Total_Customer=n())
View(data_summ_full)
```

	Airline.Name	Total_Customer
1	Cheapseats Airlines Inc.	26058
2	Cool&Young Airlines Inc.	1288
3	EnjoyFlying Air Services	8927
4	FlyFast Airways Inc.	15407
5	FlyHere Airways	2481
6	FlyToSun Airlines Inc.	3407
7	GoingNorth Airlines Inc.	1568
8	Northwest Business Airlines Inc.	13840
9	OnlyJets Airlines Inc.	5395
10	Oursin Airlines Inc.	10968
11	Paul Smith Airlines Inc.	12248
12	Sigma Airlines Inc.	17037
13	Southeast Airlines Co.	9577
14	West Airways Inc.	1688

Now we see that Cheapseat Airlines has the largest number of total customers as well as the customers with low customer satisfaction (satisfaction <4). We can see the ratio as well and see that almost all the airlines have 50% ratio of unsatisfied customers. Hence on basis of large number of customer and unsatisfied customers present for the Cheapseat Airlines, we decided to select Cheapseat Airlines as our client to provide insights to.

Code Snippet:

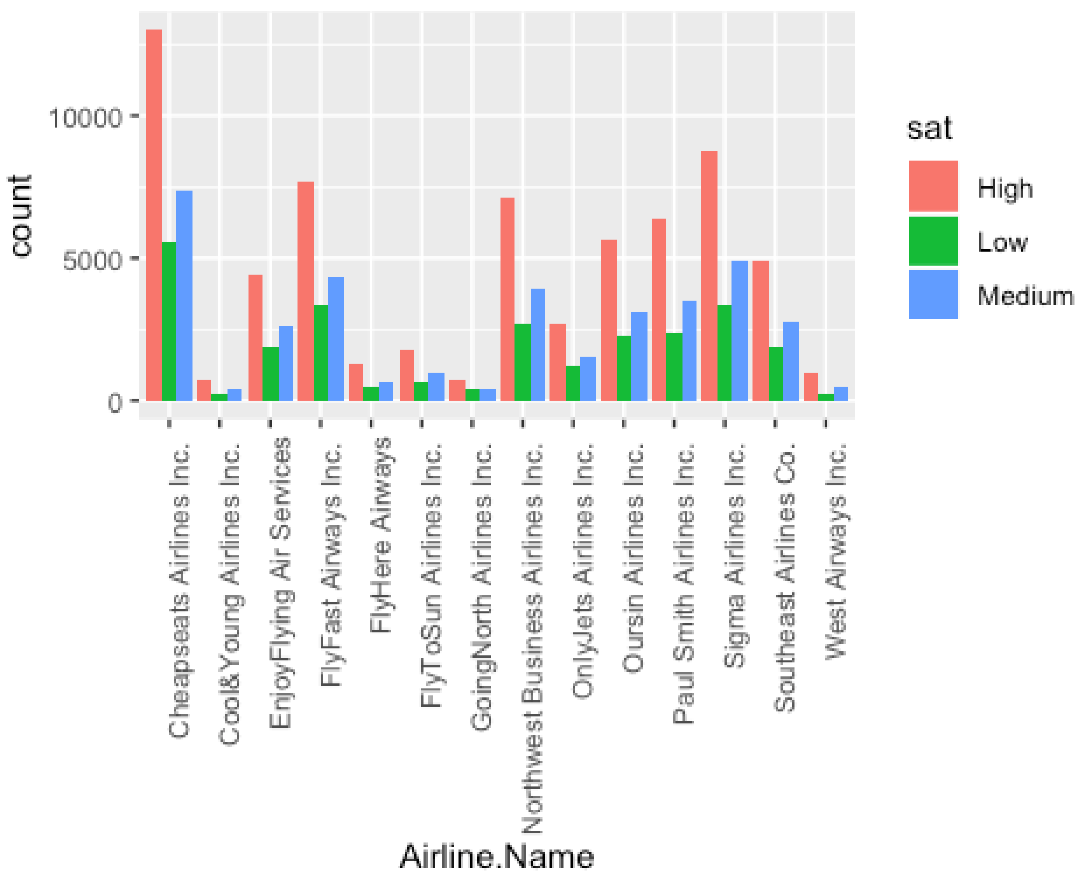
```
data_low <- dataset[dataset$Satisfaction<4,]
data_Name<-group_by(data_low,Airline.Name)
data_summ<-summarise(data_Name ,Low_Customer=n())
View(data_summ)
data_comp<-merge(data_summ,data_summ_full)
```

data_comp$low_ratio<-(data_comp$Low_Customer/data_comp$Total_Customer)*100
View(data_comp)
data_clean<-dataset[(trimws(dataset$Airline.Name,which="right")=="Cheapseats Airlines Inc."),]

	Airline.Name	Low_Customer	Total_Customer	low_ratio
1	Cheapseats Airlines Inc.	13008	26058	49.91941
2	Cool&Young Airlines Inc.	583	1288	45.26398
3	EnjoyFlying Air Services	4484	8927	50.22964
4	FlyFast Airways Inc.	7700	15407	49.97728
5	FlyHere Airways	1186	2481	47.80331
6	FlyToSun Airlines Inc.	1592	3407	46.72733
7	GoingNorth Airlines Inc.	822	1568	52.42347
8	Northwest Business Airlines Inc.	6701	13840	48.41763
9	OnlyJets Airlines Inc.	2730	5395	50.60241
10	Oursin Airlines Inc.	5319	10968	48.49562
11	Paul Smith Airlines Inc.	5874	12248	47.95885
12	Sigma Airlines Inc.	8224	17037	48.27141
13	Southeast Airlines Co.	4617	9577	48.20925
14	West Airways Inc.	734	1688	43.48341

Plot for a better visual representation of customers and airlines.

```
ggplot(Satisfaction_Survey,aes(x=Airline.Name,fill=sat))+geom_bar(position='dodge')+theme(axis.text.x = element_text(angle=90,hjust=1))
```



After selecting Cheapseat Airlines as our client, we make a dataset just for the Cheapseat airlines and now we perform the data cleaning part to start our analysis on the dataset.

Cleaning dataset

First, we see all the NA values present in our data. Once we see which attributes have NA's, we now remove these NA's. We remove the NA's present in the attribute 'Arrival Delay In Minutes'. Once we do that we see that all the NA's in the dataset get removed too.

Code Snippet:

```
colSums(is.na(data_clean))
data_clean <- filter(data_clean,  !is.na(Arrival.Delay.in.Minutes))
colSums(is.na(data_clean))
```

```
> colSums(is.na(data_clean))
                 Satisfaction                   Airline.Status                             Age
                            0                                0                               0
                       Gender                Price.Sensitivity             Year.of.First.Flight
                            0                                0                               0
             No.of.Flights.p.a. X..of.Flight.with.other.Airlines                  Type.of.Travel
                            0                                0                               0
      No..of.other.Loyalty.Cards       Shopping.Amount.at.Airport Eating.and.Drinking.at.Airport
                            0                                0                               0
                        Class                     Day.of.Month                     Flight.date
                            0                                0                               0
                 Airline.Code                     Airline.Name                     Orgin.City
                            0                                0                               0
                 Origin.State                 Destination.City               Destination.State
                            0                                0                               0
      Scheduled.Departure.Hour         Departure.Delay.in.Minutes         Arrival.Delay.in.Minutes
                            0                              316                             389
             Flight.cancelled            Flight.time.in.minutes                Flight.Distance
                            0                              389                               0
     Arrival.Delay.greater.5.Mins
                            0
```

After removing NA's from one attribute, all the NA's get removed.

```
> colSums(is.na(data_clean))
                 Satisfaction                   Airline.Status                             Age
                            0                                0                               0
                       Gender                Price.Sensitivity             Year.of.First.Flight
                            0                                0                               0
             No.of.Flights.p.a. X..of.Flight.with.other.Airlines                  Type.of.Travel
                            0                                0                               0
      No..of.other.Loyalty.Cards       Shopping.Amount.at.Airport Eating.and.Drinking.at.Airport
                            0                                0                               0
                        Class                     Day.of.Month                     Flight.date
                            0                                0                               0
                 Airline.Code                     Airline.Name                     Orgin.City
                            0                                0                               0
                 Origin.State                 Destination.City               Destination.State
                            0                                0                               0
      Scheduled.Departure.Hour         Departure.Delay.in.Minutes         Arrival.Delay.in.Minutes
                            0                                0                               0
             Flight.cancelled            Flight.time.in.minutes                Flight.Distance
                            0                                0                               0
     Arrival.Delay.greater.5.Mins
                            0
>
```

Now we clean the names of the columns. Since using '.' As a conjunction for attribute names is not a good way to write column names as the code also includes '.', we remove all the '.' And blank spaces ' ' in the column names and replace them with an underscore '_'.

Code Snippet:

unclean_names <- colnames(data_clean)
clean_names <- gsub("\\.", "_", unclean_names)
colnames(data_clean) <- clean_names

After we remove the NA's, we move on to removing any abnormal or garbage data values present in the dataset. In the customer satisfaction column, we have abnormal values that we have removed. We see that we have some abnormal values when we use the 'unique' function in Rstudio. We remove these values be getting the now numbers for them and them deleting these rows.

```
index_1 <- which(data_clean$Satisfaction=='4.00.2.00')
index_2 <- which(data_clean$Satisfaction=='4.00.5')
index_1

## [1] 38899 38900

index_2

## [1] 38898

data_clean <- data_clean [-38900:-38899, ]
data_clean <- data_clean [-38898, ]
nrow(data_clean)

## [1] 129886

data_clean $Satisfaction<-as.numeric(as.character(data_clean $Satisfaction)
)

unique(data_clean $Satisfaction)

## [1] 4.5 4.0 2.5 5.0 3.5 2.0 3.0 1.0
```

Descriptive Statistics

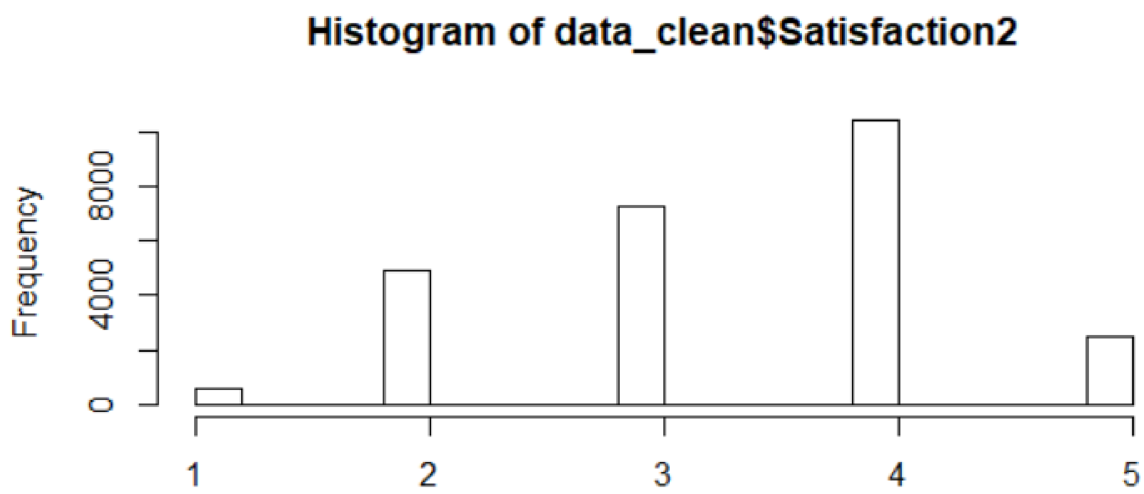
Now that we have our data prepped to perform analysis, we study the data and the attributes and get a general idea about the dataset.

a) Bucketing the satisfaction variable.

We create buckets for different levels of satisfaction.

```
createBucketsSurvey<-function(vec){
  vBuckets <- replicate(length(vec), "Medium")
  vBuckets[vec<3 ] <- "Low"
  vBuckets[vec>3 ] <- "High"
```

```
  return(vBuckets)
}

sat<-createBucketsSurvey(Satisfaction_Survey$Satisfaction)
```



We can see that for the client selected the number highest satisfaction rate is for 4 followed by 3 , 2, 5 ,1.

By further analyzing the data we can find the number of customers for each level of satisfaction.

Satisfaction Level 1 = 607, Satisfaction Level 2 = 4929, Satisfaction Level 3 = 7231, Satisfaction Level 4 = 10424, Satisfaction Level 5 = 2478.

Code Snippet:

```
data_clean$Satisfaction2 <- as.numeric(data_clean$Satisfaction)
hist(data_clean$Satisfaction2)
sat1 <- data_clean[data_clean$Satisfaction=='1',]
nrow(sat1)
sat2 <- data_clean[data_clean$Satisfaction=='2',]
nrow(sat2)
sat3 <- data_clean[data_clean$Satisfaction=='3',]
nrow(sat3)
sat4 <- data_clean[data_clean$Satisfaction=='4',]
nrow(sat4)
sat5 <- data_clean[data_clean$Satisfaction=='5',]
nrow(sat5)
```

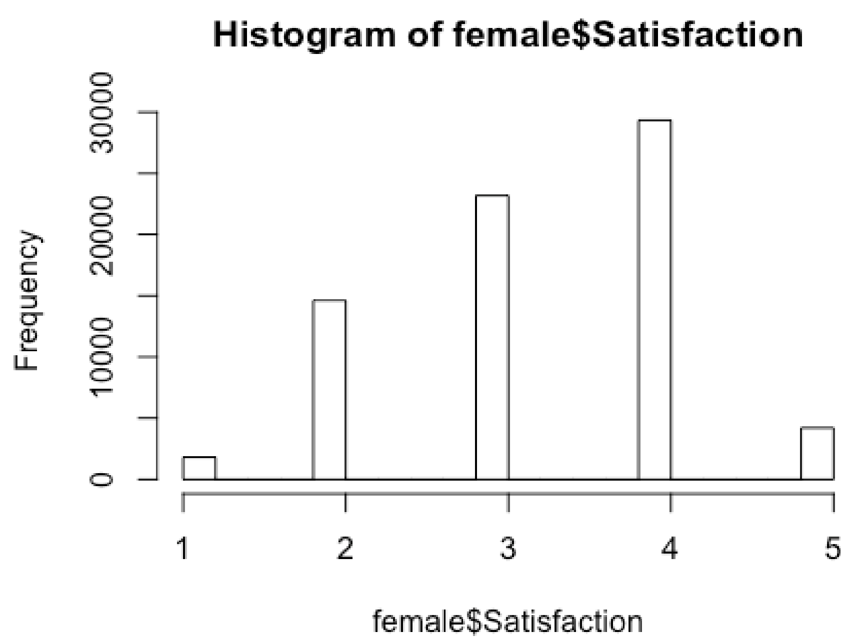
b) Satisfaction with gender

Now we compare the satisfaction levels with the Gender attribute.

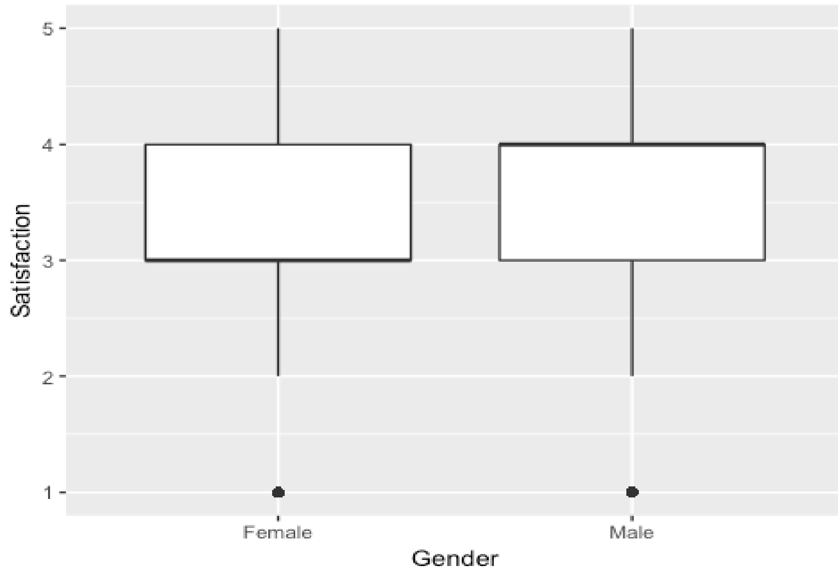
```
male<- Satisfaction_Survey[Satisfaction_Survey$Gender=='Male',]
View(male)
nrow(male)
```

```
## [1] 56359

hist(male$Satisfaction)
```

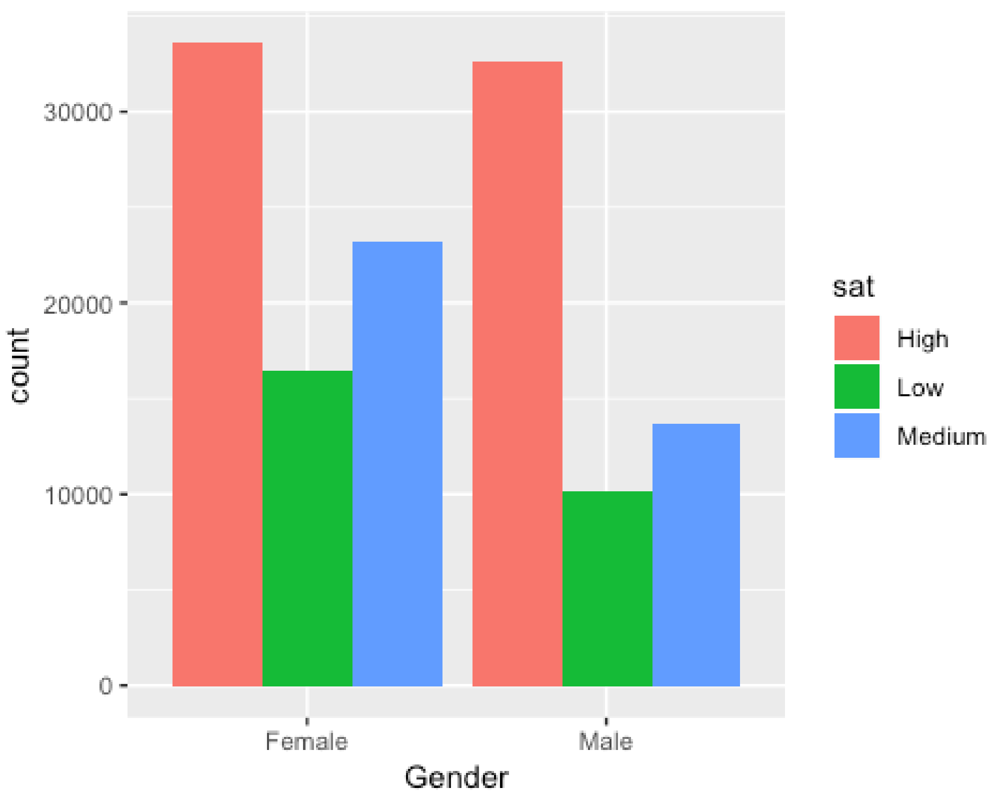

```
female <- Satisfaction_Survey[Satisfaction_Survey$Gender=='Female',]
View(female)
nrow(female)

## [1] 73190

hist(female$Satisfaction)
```



```
ggplot(Satisfaction_Survey,aes(x=Gender,y=Satisfaction))+geom_boxplot()
```




```
ggplot(Satisfaction_Survey,aes(x=Gender,fill=sat))+geom_bar(position='dodge')
```

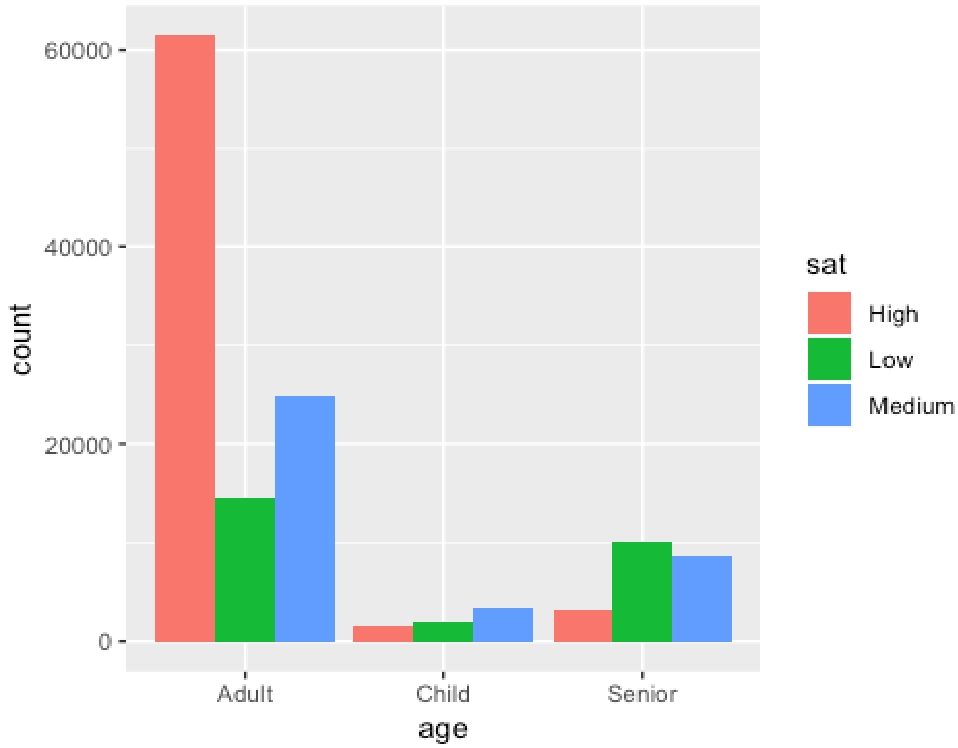


We see that there is no visually difference between the two genders and both genders are almost same in the numbers of satisfaction levels they give.

c) Satisfaction with Age

```
agefunction<-function(vec){
  vBuckets <- replicate(length(vec), "Adult")
  vBuckets[vec <= 18] <- "Child"
```

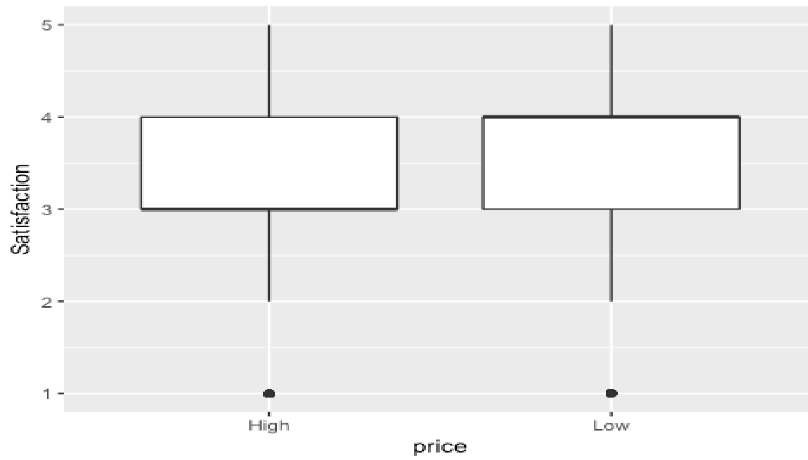
```
  vBuckets[vec >= 65] <- "Senior"
  return(vBuckets)
}

age<-agefunction(Satisfaction_Survey$Age)
ggplot(Satisfaction_Survey,aes(x=age,fill=sat))+geom_bar(position='dodge')
```



We see that people who are adults i.e. between the age group of 18-65 are more likely to give higher rating compared to children and senior people. After further analysis of Age attribute, we see that people within the age group of 35-50 tend to give higher rating.

d) Satisfaction with Price Sensitivity

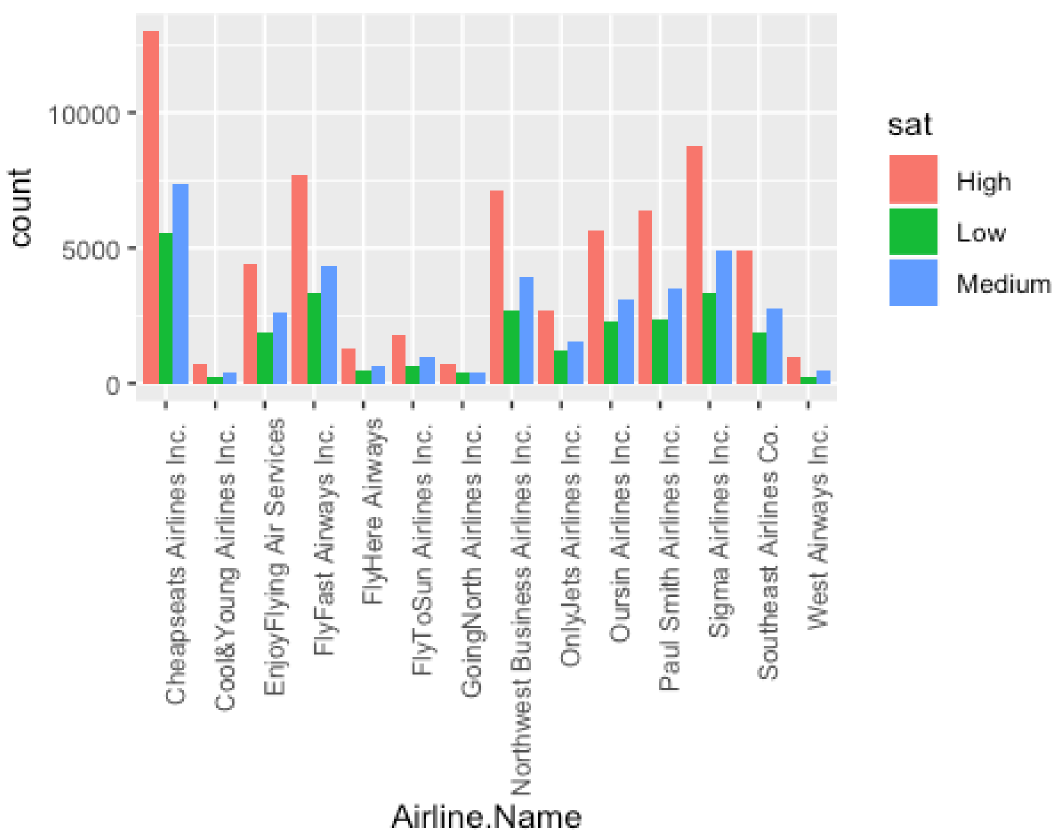
```
createBuckets<-function(vec){
  q <- quantile(vec, c(0.4, 0.6))
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec <= q[1]] <- "Low"
  vBuckets[vec > q[2]] <- "High"
  return(vBuckets)
}

price<-createBuckets(Satisfaction_Survey$Price.Sensitivity)
ggplot(Satisfaction_Survey,aes(x=price,y=Satisfaction))+geom_boxplot()
```



We see that lower the price sensitivity, higher is the customer satisfaction.

e) Satisfaction with different Airlines

```
ggplot(Satisfaction_Survey,aes(x=Airline.Name,fill=sat))+geom_bar(position='d
odge')+theme(axis.text.x = element_text(angle=90,hjust=1))
```

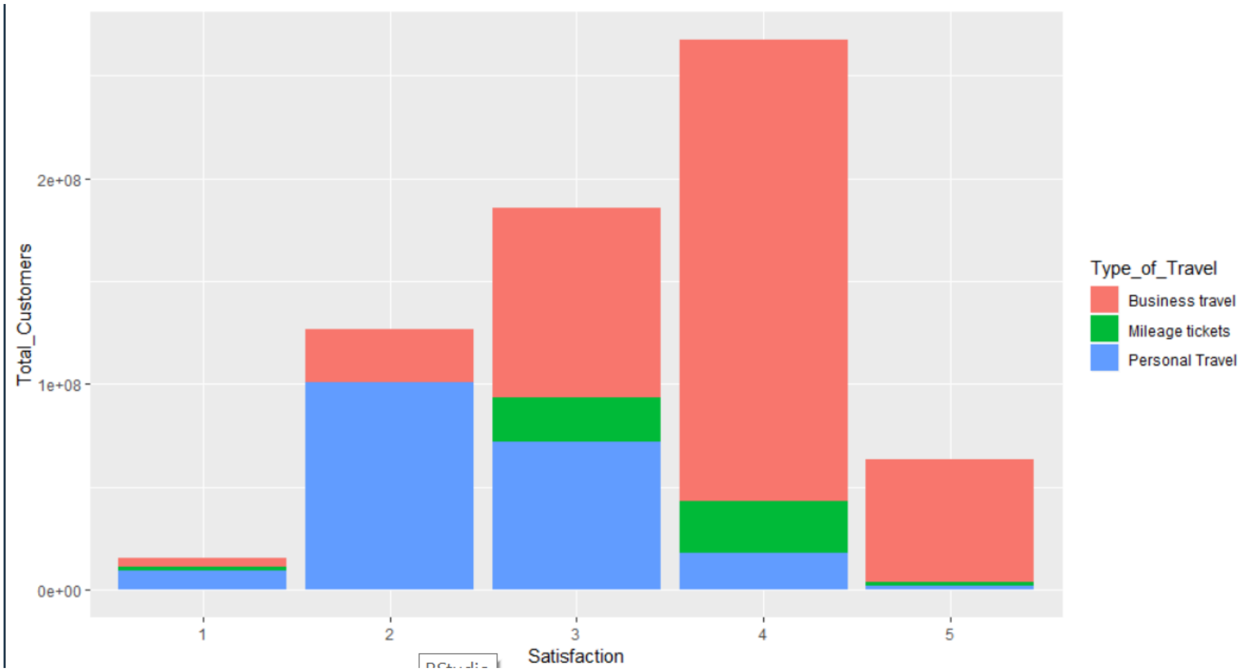


In the graphs, we can see that Cheapseats Airline Inc. has the large amount of the low satisfaction as before.

f) Satisfaction with Type of Travel

Total_Customers <- nrow(data_clean)
travel <- ggplot(data_clean, aes(x=Satisfaction, y=Total_Customers ,fill=Type_of_Travel)) + geom_col()

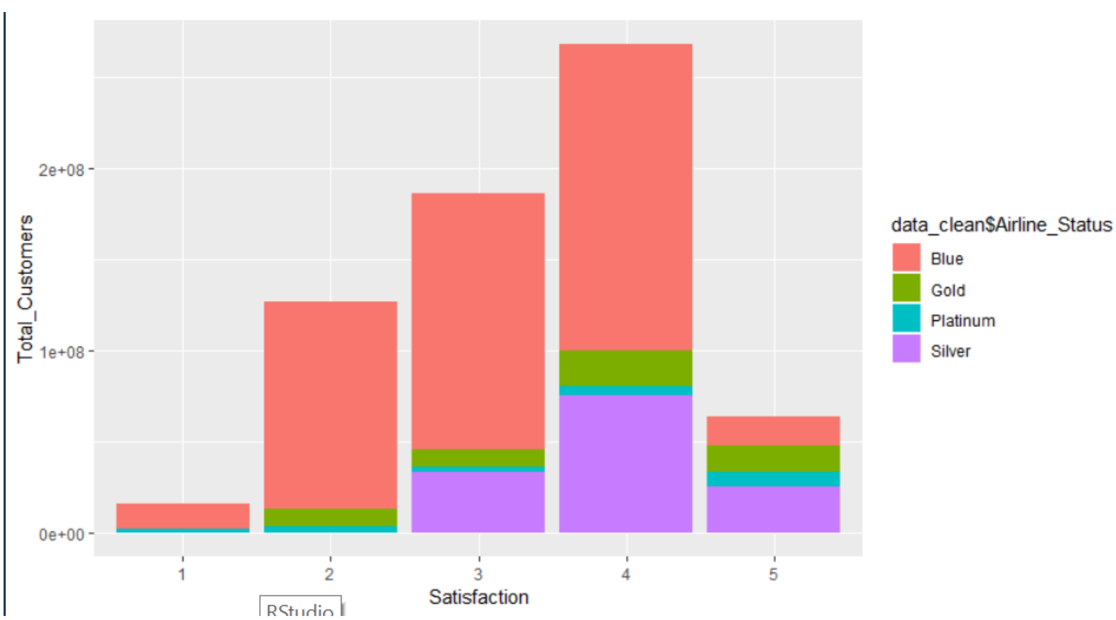
travel



We see that people travelling for business reasons give a higher rating as opposed to people travelling for personal reasons who mostly give a lower rating.

g) Satisfaction vs Airline Status

stts <- ggplot(data_clean, aes(x=Satisfaction, y=Total_Customers ,fill=data_clean$Airline_Status)) + geom_col()
stts



We see that customers in the blue status are the largest and give low satisfaction rating, whle the people in silver status usually give better rating.

Linear Regression

```
cheaplinear<-lm(cheapseatnew.Satisfaction~.,data=cheaplineardata)
summary(cheaplinear)

##
## Call:
## lm(formula = cheapseatnew.Satisfaction ~ ., data = cheaplineardata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.14453 -0.40108  0.02703  0.49221  2.76267
##
## Coefficients:
##                                          Estimate Std. Error
## (Intercept)                            -3.678e+00  3.051e+00
## numairlineGold                          4.494e-01  1.689e-02
## numairlinePlatinum                      3.018e-01  2.618e-02
## numairlineSilver                        6.454e-01  1.159e-02
## cheapseatnew.Age                       -2.456e-03  3.112e-04
## numgenderMale                           1.252e-01  9.371e-03
## cheapseatnew.Price_Sensitivity         -5.804e-02  8.429e-03
## cheapseatnew.Year_of_First_Flight       3.764e-03  1.520e-03
## cheapseatnew.No_of_Flights_p_a_        -3.047e-03  3.445e-04
## cheapseatnew.X__of_Flight_with_other_Airlines -5.436e-04  5.816e-04
## numttravelMileage tickets              -1.524e-01  1.742e-02
## numttravelPersonal Travel              -1.086e+00  1.111e-02
## cheapseatnew.No__of_other_Loyalty_Cards -2.617e-03  4.799e-03
## numclassEco                            -6.898e-02  1.653e-02
## numclassEco Plus                       -7.067e-02  2.127e-02
## cheapseatnew.Scheduled_Departure_Hour   3.745e-03  1.071e-03
## cheapseatnew.Departure_Delay_in_Minutes 2.218e-04  1.530e-04
## cheapseatnew.Flight_time_in_minutes     5.059e-04  3.437e-04
## cheapseatnew.Flight_Distance           -5.303e-05  4.276e-05
## num5minyes                             -3.555e-01  1.099e-02
##                                        t value Pr(>|t|)
## (Intercept)                             -1.206 0.227954
## numairlineGold                          26.604  < 2e-16 ***
## numairlinePlatinum                      11.525  < 2e-16 ***
## numairlineSilver                        55.662  < 2e-16 ***
```

```
## cheapseatnew.Age                          -7.893 3.08e-15 ***
## numgenderMale                             13.360  < 2e-16 ***
## cheapseatnew.Price_Sensitivity            -6.886 5.87e-12 ***
## cheapseatnew.Year_of_First_Flight          2.475 0.013312 *
## cheapseatnew.No_of_Flights_p_a_           -8.845  < 2e-16 ***
## cheapseatnew.X__of_Flight_with_other_Airlines  -0.935 0.349968
## numttravelMileage tickets                 -8.747  < 2e-16 ***
## numttravelPersonal Travel                -97.811  < 2e-16 ***
## cheapseatnew.No__of_other_Loyalty_Cards   -0.545 0.585569
## numclassEco                               -4.174 3.00e-05 ***
## numclassEco Plus                          -3.322 0.000894 ***
## cheapseatnew.Scheduled_Departure_Hour      3.497 0.000470 ***
## cheapseatnew.Departure_Delay_in_Minutes    1.450 0.147193
## cheapseatnew.Flight_time_in_minutes        1.472 0.141102
## cheapseatnew.Flight_Distance              -1.240 0.214944
## num5minyes                               -32.344  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7204 on 25649 degrees of freedom
## Multiple R-squared:  0.4538, Adjusted R-squared:  0.4534
## F-statistic:  1122 on 19 and 25649 DF,  p-value: < 2.2e-16
```

In the results, we can see that airline status, age, gender, price sensitivity, number of flights, travel mileage ticket, type of travel, class, scheduled department hour and flight delayed greater than 5 minutes are strongly significant to influence the satisfaction because their p-value are quite small. The R squared value are 0.4538 which means that 45% of the satisfaction can be predicted by this model.

Because of the lower R-squared value, we still need to improve the model. Then I use the stepwise model to make the model more perfect.

Stepwise models

```
library('MASS')
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
null<-lm(cheapseatnew.Satisfaction~1,cheaplineardata)
stepAIC(cheaplinear, direction='backward')
```

The results of the backward step is:

```
## Call:
## lm(formula = cheapseatnew.Satisfaction ~ numairline + cheapseatnew.Age +
##     numgender + cheapseatnew.Price_Sensitivity + cheapseatnew.Year_of_First_Flight +
##     cheapseatnew.No_of_Flights_p_a_ + numttravel + numclass +
##     cheapseatnew.Scheduled_Departure_Hour + num5min, data = cheaplineardata)
##
## Coefficients:
##                      (Intercept)
##                        -3.568490
##                   numairlineGold
##                         0.448385
##               numairlinePlatinum
##                         0.300494
##                 numairlineSilver
##                         0.644310
##                 cheapseatnew.Age
##                        -0.002318
##                    numgenderMale
##                         0.126542
##   cheapseatnew.Price_Sensitivity
##                        -0.057088
##  cheapseatnew.Year_of_First_Flight
##                         0.003708
##   cheapseatnew.No_of_Flights_p_a_
##                        -0.002973
##         numttravelMileage tickets
##                        -0.152757
##         numttravelPersonal Travel
##                        -1.087828
##                    numclassEco
##                        -0.069115
##                 numclassEco Plus
##                        -0.069430
## cheapseatnew.Scheduled_Departure_Hour
##                         0.003733
##                      num5minyes
##                        -0.345378
```

Then we summarize the final model after using the backward step.

```
lm_backward <- lm(cheapseatnew.Satisfaction ~ numairline + cheapseatnew.Age +
    numgender + cheapseatnew.Price_Sensitivity + cheapseatnew.Year_of_First_Flight +
    cheapseatnew.No_of_Flights_p_a_ + numttravel + numclass +
    cheapseatnew.Scheduled_Departure_Hour + num5min, data = cheaplineardata)

summary(lm_backward)
```

```
##
## Call:
## lm(formula = cheapseatnew.Satisfaction ~ numairline + cheapseatnew.Age +
##     numgender + cheapseatnew.Price_Sensitivity + cheapseatnew.Year_of_First_Flight +
##     cheapseatnew.No_of_Flights_p_a_ + numttravel + numclass +
##     cheapseatnew.Scheduled_Departure_Hour + num5min, data = cheaplineardata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.15324 -0.40063  0.02483  0.49231  2.76586
##
## Coefficients:
##                                        Estimate Std. Error t value
## (Intercept)                          -3.5684898  3.0503972  -1.170
## numairlineGold                        0.4483849  0.0168591  26.596
## numairlinePlatinum                    0.3004943  0.0261606  11.487
## numairlineSilver                      0.6443097  0.0115574  55.749
## cheapseatnew.Age                     -0.0023181  0.0002828  -8.196
## numgenderMale                         0.1265416  0.0093076  13.596
## cheapseatnew.Price_Sensitivity       -0.0570879  0.0084013  -6.795
## cheapseatnew.Year_of_First_Flight     0.0037077  0.0015201   2.439
## cheapseatnew.No_of_Flights_p_a_      -0.0029733  0.0003401  -8.741
## numttravelMileage tickets            -0.1527574  0.0174133  -8.772
## numttravelPersonal Travel            -1.0878276  0.0110662 -98.302
## numclassEco                          -0.0691150  0.0165268  -4.182
## numclassEco Plus                     -0.0694301  0.0212484  -3.268
## cheapseatnew.Scheduled_Departure_Hour  0.0037330  0.0010518   3.549
## num5minyes                           -0.3453777  0.0094842 -36.416
##                                      Pr(>|t|)
## (Intercept)                          0.242074
## numairlineGold                        < 2e-16 ***
## numairlinePlatinum                    < 2e-16 ***
## numairlineSilver                      < 2e-16 ***
## cheapseatnew.Age                     2.60e-16 ***
## numgenderMale                         < 2e-16 ***
## cheapseatnew.Price_Sensitivity       1.11e-11 ***
## cheapseatnew.Year_of_First_Flight    0.014734 *
## cheapseatnew.No_of_Flights_p_a_       < 2e-16 ***
## numttravelMileage tickets             < 2e-16 ***
## numttravelPersonal Travel             < 2e-16 ***
## numclassEco                          2.90e-05 ***
## numclassEco Plus                     0.001086 **
## cheapseatnew.Scheduled_Departure_Hour 0.000387 ***
## num5minyes                            < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.7204 on 25654 degrees of freedom
## Multiple R-squared:  0.4536, Adjusted R-squared:  0.4534
## F-statistic:  1522 on 14 and 25654 DF,  p-value: < 2.2e-16
```

The variables in backward model are same as the variables in linear model, so them have the same value of R-squared.

Then we use the forward step to test again.

```
stepAIC(null,direction='forward',scope=list(upper=cheaplinear,lower=null))
```

The result is

```
## Call:
## lm(formula = cheapseatnew.Satisfaction ~ numttravel + numairline +
##     num5min + numgender + cheapseatnew.No_of_Flights_p_a_ + cheapseatnew.Age +
##     cheapseatnew.Price_Sensitivity + numclass + cheapseatnew.Scheduled_Departure_Hour +
##     cheapseatnew.Year_of_First_Flight, data = cheaplineardata)
##
## Coefficients:
##                        (Intercept)
##                          -3.568490
##            numttravelMileage tickets
##                          -0.152757
##            numttravelPersonal Travel
##                          -1.087828
##                    numairlineGold
##                           0.448385
##                 numairlinePlatinum
##                           0.300494
##                   numairlineSilver
##                           0.644310
##                       num5minyes
##                          -0.345378
##                     numgenderMale
##                           0.126542
##       cheapseatnew.No_of_Flights_p_a_
##                          -0.002973
##                  cheapseatnew.Age
##                          -0.002318
##        cheapseatnew.Price_Sensitivity
##                          -0.057088
##                      numclassEco
##                          -0.069115
##                  numclassEco Plus
##                          -0.069430
## cheapseatnew.Scheduled_Departure_Hour
##                           0.003733
```

```
##     cheapseatnew.Year_of_First_Flight
##                          0.003708
```

```
lm_forward <- lm(formula = cheapseatnew.Satisfaction ~ numttravel + numairline +
                 num5min + numgender + cheapseatnew.No_of_Flights_p_a_ + cheapseatnew.Age +
                 cheapseatnew.Price_Sensitivity + numclass + cheapseatnew.Scheduled_Departure_Hour +
                 cheapseatnew.Year_of_First_Flight, data = cheaplineardata)
```

Then we summary the forward model.

```
summary(lm_forward)

##
## Call:
## lm(formula = cheapseatnew.Satisfaction ~ numttravel + numairline +
##     num5min + numgender + cheapseatnew.No_of_Flights_p_a_ + cheapseatnew.Age +
##     cheapseatnew.Price_Sensitivity + numclass + cheapseatnew.Scheduled_Departure_Hour +
##     cheapseatnew.Year_of_First_Flight, data = cheaplineardata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.15324 -0.40063  0.02483  0.49231  2.76586
##
## Coefficients:
##                                        Estimate Std. Error t value
## (Intercept)                          -3.5684898  3.0503972  -1.170
## numttravelMileage tickets            -0.1527574  0.0174133  -8.772
## numttravelPersonal Travel            -1.0878276  0.0110662 -98.302
## numairlineGold                        0.4483849  0.0168591  26.596
## numairlinePlatinum                    0.3004943  0.0261606  11.487
## numairlineSilver                      0.6443097  0.0115574  55.749
## num5minyes                           -0.3453777  0.0094842 -36.416
## numgenderMale                         0.1265416  0.0093076  13.596
## cheapseatnew.No_of_Flights_p_a_      -0.0029733  0.0003401  -8.741
## cheapseatnew.Age                     -0.0023181  0.0002828  -8.196
## cheapseatnew.Price_Sensitivity       -0.0570879  0.0084013  -6.795
## numclassEco                          -0.0691150  0.0165268  -4.182
## numclassEco Plus                     -0.0694301  0.0212484  -3.268
## cheapseatnew.Scheduled_Departure_Hour  0.0037330  0.0010518   3.549
## cheapseatnew.Year_of_First_Flight     0.0037077  0.0015201   2.439
##                                      Pr(>|t|)
## (Intercept)                          0.242074
## numttravelMileage tickets             < 2e-16 ***
## numttravelPersonal Travel             < 2e-16 ***
## numairlineGold                        < 2e-16 ***
## numairlinePlatinum                    < 2e-16 ***
## numairlineSilver                      < 2e-16 ***
```

```
## num5minyes                          < 2e-16 ***
## numgenderMale                       < 2e-16 ***
## cheapseatnew.No_of_Flights_p_a_     < 2e-16 ***
## cheapseatnew.Age                    2.60e-16 ***
## cheapseatnew.Price_Sensitivity      1.11e-11 ***
## numclassEco                         2.90e-05 ***
## numclassEco Plus                    0.001086 **
## cheapseatnew.Scheduled_Departure_Hour 0.000387 ***
## cheapseatnew.Year_of_First_Flight   0.014734 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7204 on 25654 degrees of freedom
## Multiple R-squared:  0.4536, Adjusted R-squared:  0.4534
## F-statistic:  1522 on 14 and 25654 DF,  p-value: < 2.2e-16
```

The variables in forward model are same as the variables in linear model and variables in backward model, so all of them have the same value of R-squared.

Then we can conclude the significant variables in linear model are airline status, age, gender, price sensitivity, number of flights, type of travel, class, scheduled departure hour and arrival delay greater 5 mins.

Association Rules

We began our exploration of association rules mining using the variables being used in linear model. Before starting the association rule mining, we did data preparation because association rules do not accept numeric or integer variables. The variables we selected are listed below:

Independent variable (1): Satisfaction;

Dependent variable (19): Price_Sensitivity; No_of_other_Loyalty_Cards;

No_of_Flights_p_a_; Type_of_Travel; shopping amount _at_airport; Eating_and_Drinking_at_Airport; Class; Departure_hour; Flight_Distance; Flight_time_in_minutes; Arrival_Delay_greater_5_Mins; Flight_cancelled; Scheduled_Departure_Hour; Departure_Delay_in_Minutes; Arrival_Delay_in_Minutes; Airline_Status; Age; Gender;

We created Buckets function to classify each variable we use. For our independent variable Satisfaction, rating 1-2 points were defined as “low” level , rating 3 point was defined as “average” level and rating 4-5 points were defined as “high” level. Since dependent variable Price_Senstivity rating has the same scale with Satisfaction, we use the same Buckets for these two variables.

```r
# creatBuckets
createBuckets <- function(v){
  vBuckets <- replicate(length(v), "Average")
  vBuckets[v > 3] <- "High"
  vBuckets[v < 3] <- "Low"
  return(vBuckets)
}
# satisfaction
satSuyarule$Satisfaction <- as.numeric(as.character(satSuy$Satisfaction))
satcust <- createBuckets(satSuyarule$Satisfaction)
# price sensitivity
priceSen <- createBuckets(satSuyarule$Price.Sensitivity)
```

```
> str(satcust)
 chr [1:25669] "Low" "High" "High" "High" "Low" "Low" "High" "Low" "Low" "Low" "Low" ...
> str(priceSen)
 chr [1:25669] "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" ...
```

Then we created Buckets Card for variable No_of_other_Loyalty_Cards. Customers with on cards from other company were labeled as “no”, customers with less than 2 cards were labeled as “yes”, customers with more cards were labeled as “more”.

```r
# No..of other loyalty cards
createBucketsCard <- function(v){
  vBuckets <- replicate(length(v), "No")
  vBuckets[v > 0] <- "Yes"
  vBuckets[v >= 2] <- "more"
  return(vBuckets)
}

NumCards <- createBucketsCard(satSuyarule$No..of.other.Loyalty.Cards)
NumCards
```

```
> str(NumCards)
 chr [1:25669] "more" "No" "No" "more" "No" "more" "No" "Yes" "Yes" "Yes" "No" "No" ...
```

We classified variable Age into 3 levels: under 20 years old was defined as “teenager”, over 65 was defined as “senior”, the rest was defined as “adult”. We created BucketsAge.

```r
summary(satSuyarule$Age)
createBucketsAge <- function(v){
  vBuckets <- replicate(length(v), "teenager")
  vBuckets[v >= 20] <- "adult"
  vBuckets[v >= 65] <- "senior"
  return(vBuckets)
}
```

```
> str(age)
 chr [1:25669] "adult" "adult" "adult" "adult" "adult" "adult" "senior" "adult" "adult" ...
```

We created BucketsDelay for variables Departure_Delay_in_Minutes and Arrival_Delay_in_Minutes. Delay zero minutes was defined as “noDelay”, delay less than 17 minutes was defined as “short”, delay minute from 17 to 45 was defined as “middle”, delay more than 45 minutes was defined as “long”.

```
createBucketsDelay <- function(v){
  vBuckets <- replicate(length(v), "NoDelay")
  vBuckets[v > 0] <- "short"
  vBuckets[v >= 17] <- "middle"
  vBuckets[v >= 45] <- "long"
  return(vBuckets)
}
```

```
> str(arrDelay)
 chr [1:129549] "short" "short" "short" "middle" "NoDelay" "NoDelay" "NoDelay" "short" ...
> str(depDelay)
 chr [1:129549] "NoDelay" "short" "middle" "middle" "NoDelay" "NoDelay" "NoDelay" ...
>
```

We created BucketsOther for the other numeric variables. Values less than 40% quartiles were defined as “Low”, values more than 60% quartiles were defined as “High”, others were defined as “Average”.

```
createBucketsOther <- 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)
}
```

```
> str(arrDelay)
 chr [1:129549] "short" "short" "short" "middle" "NoDelay" "NoDelay" "NoDelay" "short" ...
> str(depDelay)
 chr [1:129549] "NoDelay" "short" "middle" "middle" "NoDelay" "NoDelay" "NoDelay" ...
> str(NumFlight)
 chr [1:25669] "Low" "Low" "Low" "Average" "Low" "Low" "High" "High" "High" "Low" ...
> str(shopping)
 chr [1:25669] "Low" "High" "High" "High" "Low" "Low" "Low" "High" "High" "High" "Low" ...
> str(eatdrink)
 chr [1:25669] "Average" "High" "Average" "High" "Average" "Average" "High" "High" ...
> str(departureHour)
 chr [1:25669] "Average" "Low" "Low" "Low" "Low" "Average" "Low" "Low" "Low" "Low" ...
> str(flightTime)
 chr [1:25669] "High" "High" "High" "High" "High" "High" "High" "High" "High" "High" ...
> str(flightDis)
 chr [1:25669] "High" "High" "High" "High" "High" "High" "High" "High" "High" "High" ...
> str(otherAirline)
 chr [1:25669] "High" "Low" "Low" "High" "Low" "Low" "High" "High" "High" "Average" ...
>
```

Finally we created a new dataset data_arules using the 20 categorized variables and turned them into factors. We run str() command and saw there are 25669 observations in our dataset.

```
data_arules <- data.frame(satcust, priceSen, otherAirline,NumCards,age, NumFlight, shopping,
                          eatdrink,departureHour,flightTime,flightDis,depDelay,
                          arrDelay, status, gender, firstflight, typetravel,class, delay5min, cancel )

data_arules[1:20]<-lapply(data_arules[1:20],as.factor)
```

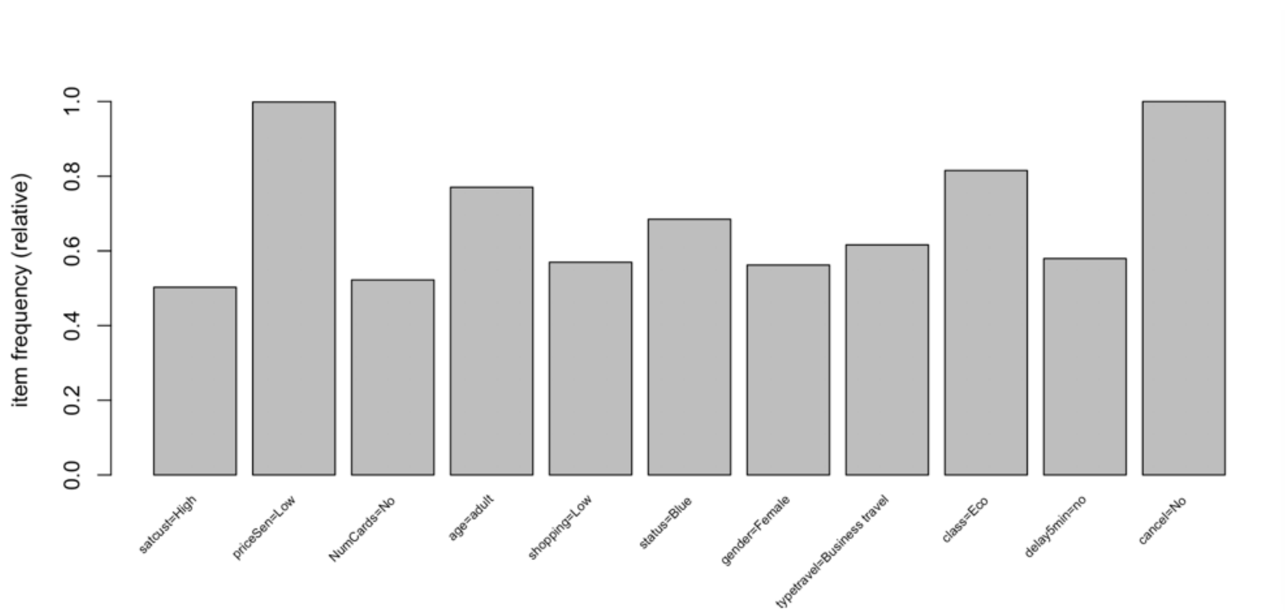
```
> str(data_arules)
'data.frame':   25669 obs. of  20 variables:
 $ satcust      : Factor w/ 2 levels "High","Low": 2 1 1 1 2 2 1 2 2 2 ...
 $ priceSen     : Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 2 2 2 ...
 $ otherAirline : Factor w/ 3 levels "Average","High",..: 2 3 3 2 3 3 2 2 2 1 ...
 $ NumCards     : Factor w/ 3 levels "more","No","Yes": 1 2 2 1 2 1 2 3 3 3 ...
 $ age          : Factor w/ 3 levels "adult","senior",..: 1 1 1 1 1 1 2 1 1 1 ...
 $ NumFlight    : Factor w/ 3 levels "Average","High",..: 3 3 3 1 3 3 2 2 2 3 ...
 $ shopping     : Factor w/ 3 levels "Average","High",..: 3 2 2 2 3 3 3 2 2 2 ...
 $ eatdrink     : Factor w/ 3 levels "Average","High",..: 1 2 1 2 1 1 2 2 3 1 ...
 $ departureHour: Factor w/ 3 levels "Average","High",..: 1 3 3 3 3 1 3 3 3 3 ...
 $ flightTime   : Factor w/ 3 levels "Average","High",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ flightDis    : Factor w/ 3 levels "Average","High",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ depDelay     : Factor w/ 4 levels "long","middle",..: 2 2 3 3 2 2 4 2 4 4 ...
 $ arrDelay     : Factor w/ 4 levels "long","middle",..: 2 4 3 3 4 2 3 2 3 4 ...
 $ status       : Factor w/ 4 levels "Blue","Gold",..: 1 4 1 4 4 1 4 4 1 1 ...
 $ gender       : Factor w/ 2 levels "Female","Male": 1 1 1 1 2 1 1 1 1 1 ...
 $ firstflight  : Factor w/ 10 levels "2003","2004",..: 3 2 2 5 1 3 2 1 10 2 ...
 $ typetravel   : Factor w/ 3 levels "Business travel",..: 3 1 1 1 1 1 2 3 3 3 ...
 $ class        : Factor w/ 3 levels "Business","Eco",..: 1 2 2 2 2 2 2 2 2 2 ...
 $ delay5min    : Factor w/ 2 levels "no","yes": 2 2 1 1 2 2 1 2 1 1 ...
```

When using association rules, we put Satisfaction (“low”, “average”, “high”) on the right hand side trying to find out which variables can affect customer satisfaction.

We specify the minimum level of “support” to 0.5 when using itemFrequencyPlot() function and got a bar graph. The graph shows the relative frequency of occurrence of different items in the dataset.

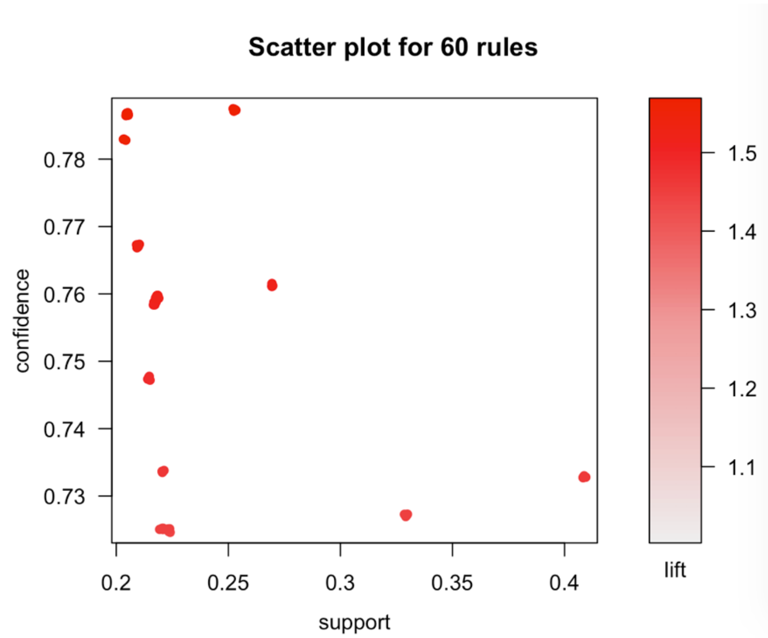
```
data_arules.trans <- as(data_arules, "transactions")

itemFrequencyPlot(data_arules.trans, support=0.5,cex.names=0.6)
```

This command yielded 11 items, support of priceSensitivity, age, airline statue, type of travel, class and flight cancel were all over 50%.



Then we used association rule for customer with high satisfaction level to find out the most important factors which could affect customer satisfaction. We set minimum support to 0.2 and set minimum confidence to 0.2, we will get 256 rules. Then we sort the ruleset by lift. We set lift as more than 1.4. Finally we got 60 goodrules. We plotted the goodrules and inspected the top 10 rules.

```
ruleset_High <- apriori(data_arules.trans, parameter=list(support=0.2, confidence=0.2),
                        appearance=list(default="lhs",rhs=("satcust=High")))
ruleset_high <- sort(ruleset_High, decreasing=TRUE, by="confidence")
goodrules_high <- ruleset_high[quality(ruleset_high)$lift > 1.4]
goodrules_high <- sort(goodrules_high, descreasing=TRUE, by="confidence")
plot(goodrules_high)
inspect(head(goodrules_high,10))
```



Top 10 good rules for satisfaction = high.

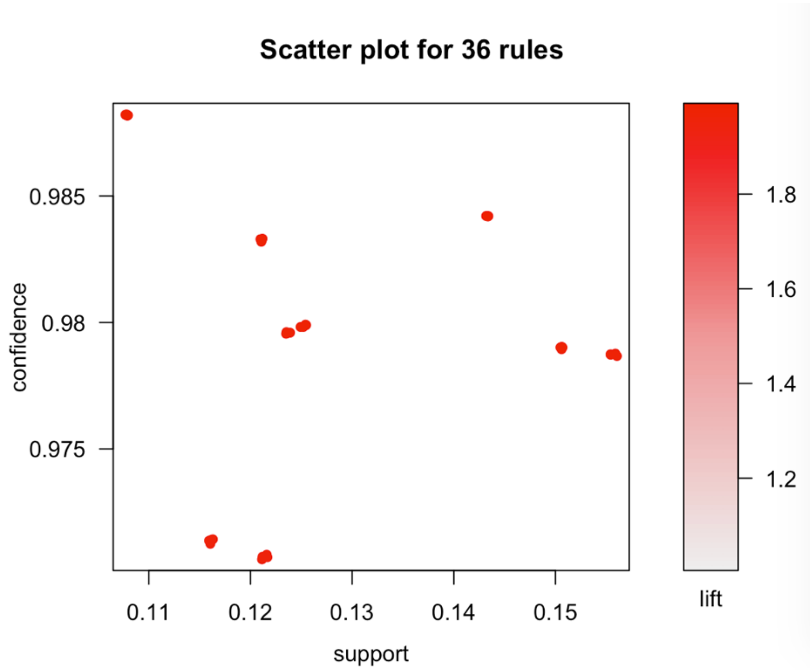
	Lhs	Rhs	Support	Confidence	Lift	Count
[1]	{age=adult, typetravel=Business, delay5min=no}	{satcust=High}	0.2527562	0.7873786	1.565169	5252
[2]	{age=adult, typetravel=Business, delay5min=no, cancel=No}	{satcust=High}	0.2527562	0.7873786	1.565169	6488
[3]	{priceSen=Low, age=adult, typetravel=Business, delay5min=no}	{satcust=High}	0.2524446	0.7873633	1.566488	6480
[4]	{priceSen=Low, age=adult, typetravel=Business, delay5min=no, cancel=No}	{satcust=High}	0.2524446	0.7873633	1.566488	6480

[5]	{priceSen=Low, age=adult, arrDelay=NoDelay, typetravel=Business }	{satcust=High}	0.2046048	0.7866986	1.565166	5252
[6]	{priceSen=Low, age=adult, arrDelay=NoDelay, typetravel=Business, delay5min=no}	{satcust=High}	0.2046048	0.7866986	1.565166	5252
[7]	{priceSen=Low, age=adult, arrDelay=NoDelay, typetravel=Business, cancel=No}	{satcust=High}	0.2046048	0.7866986	1.565166	5252
[8]	{priceSen=Low, age=adult, arrDelay=NoDelay, typetravel=Business, delay5min=no, cancel=No}	{satcust=High}	0.2046048	0.7866986	1.565166	5252
[9]	{age=adult, arrDelay=NoDelay, typetravel=Business }	{satcust=High}	0.2048385	0.7866547	1.565078	5258
[10]	{age=adult, arrDelay=NoDelay, typetravel=Business, delay5min=no}	{satcust=High}	0.2048385	0.7866547	1.565078	5258

In rule [1], the value support shows the proportion of customers who are adult business traveler and have flight delay are less than 5 minutes and meanwhile they have high satisfaction. For customer with high satisfaction, we can calculate the percent of customer who are adult business traveler and have flight delay less than 5 minutes, this percent is value confidence.

Then we used association rule for customer with low satisfaction level to find out the most important factors which could have bad effect on customer satisfaction. We set minimum support to 0.1 and set minimum confidence to 0.5, we will get 1122 rules. Then we sort the ruleset by lift. We set lift as more than 1.95. Finally we got 36 goodrules. We plotted the goodrules and inspected the top 10 rules.

```
ruleset_Low <- apriori(data_arules.trans, parameter=list(support=0.1, confidence=0.5),
                       appearance=list(default="lhs",rhs=("satcust=Low")))
ruleset_low <- sort(ruleset_Low, decreasing=TRUE, by="confidence")
goodrules_low <- ruleset_low[quality(ruleset_low)$lift > 1.95]
goodrules_low <- sort(goodrules_low, descreasing=TRUE, by="confidence")
inspect(head(goodrules_low,10))
plot(goodrules_low)
```



Top 10 goodrules for satisfaction = low.

	Lhs	Rhs	Support	Confidence	Lift	Count
[1]	{NumCards=No, NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1077954	0.9882143	1.986878	2767
[2]	{priceSen=Low, NumCards=No, NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1077954	0.9882143	1.986878	2767
[3]	{NumCards=No, NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1077954	0.9882143	1.986878	2767
[4]	{priceSen=Low, NumCards=No, NumFlight=High, status=Blue, typetravel=Personal, cancel=No}	{satcust=Low}	0.1077954	0.9882143	1.986878	2767
[5]	{NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1432078	0.9842035	1.978814	3676
[6]	{priceSen=Low, NumFlight=High, status=Blue,	{satcust=Low}	0.1432078	0.9842035	1.978814	3676

	typetravel=Personal }					
[7]	{NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1432078	0.9842035	1.978814	3676
[8]	{priceSen=Low, NumFlight=High, status=Blue, typetravel=Personal }	{satcust=Low}	0.1432078	0.9842035	1.978814	3676
[9]	{NumFlight=High, status=Blue, typetravel=Personal, class=Eco}	{satcust=Low}	0.1211578	0.9832438	1.976884	3110
[10]	{priceSen=Low, NumFlight=High, status=Blue, typetravel=Personal, class=Eco}	{satcust=Low}	0.1211578	0.9832438	1.976884	3110

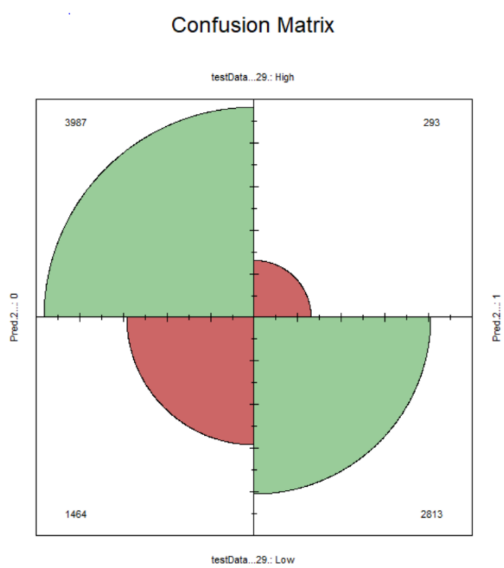
In rule [5], the value support shows the proportion of customers who are frequent personal traveler with blue status and meanwhile they have low satisfaction. For customer with low satisfaction, we can calculate the percent of customer who are frequent personal traveler with blue status, this percent is value confidence.

Using association rules, we can generate customer profile with low satisfaction and high satisfaction. Personal travelers, blue status customers, economy class guests trend to have low satisfaction. Business travelers, adult trend to have high satisfaction. We can also see most influential aspects of the airline service that can affect customer satisfaction. When flight arrival delay is less than 5 minutes or no arrival delay, the customer satisfaction tend to be higher.

Support Vector Machine

```r
table(data_clean$happy_customer)

randIndex <- sample(1:dim(data_clean)[1])
summary(randIndex)
length(randIndex)
head(randIndex)
cutPoint2_3 <- floor(2 * dim(data_clean)[1]/3)
cutPoint2_3
trainData <- data_clean[randIndex[1:cutPoint2_3],]
View(trainData)
testData <-data_clean[randIndex[(cutPoint2_3+1):dim(data_clean)[1]],]
View(testData)
dim(trainData)
dim(testData)

install.packages('kernlab')
library(kernlab)

svmOutput <- ksvm(happy_customer ~ Type_of_Travel + Age + Airline_Status + No_of_Flights_p_a_ + Scheduled_Departure_Hour + Gender
    + Arrival_Delay_greater_5_Mins + Class + Price_Sensitivity + Airline_Status , data=trainData,kernel= "rbfdot",kpar = "automatic",C=5,cross=3,prob.model=TRUE)
svmOutput

Pred <- predict(svmOutput, testData, type = "votes")

compTable2 <- data.frame(testData[,29], Pred[2,])
comp2 <- table(compTable2)
comp2

error_rate_percentage <- (comp2[2,1] + comp2[1,2])/nrow(testData)*100
error_rate_percentage
```

SVM tries to create a “hyperplane” to divide the data. It tries to find Happy customers that have given rating above 3 and separate unhappy customers ie ones with rating below 3 by the support vector machine model. We trained the model with buckets under 3 rating. We received 79.5 % accuracy. The variables used to train the data were Model was based on Age, Airline Status, Type of Travel, Number of Flights, Gender, Delay greater than 5 minutes, Class and Price Sensitivity. Below is the figure represents the number of times the model algorithm predicted customer satisfaction rating accurately.



Prediction Table:

```
> comp2
               Pred.2...
testData...29.    0    1
           High 4061  307
           Low  1451 2738
>
```

The SVM helped us validate the findings from the linear model and Associate Rules. The prediction of a happy customer from the SVM model proved to be accurate almost 80 out of 100 times.

Actionable Insights

Insights from Linear Model

The linear model portrayed the dependency of the customer satisfaction on variables:

- Customers' Age
- Customers' Gender
- Customers; Travel type
- Customers' Ticket Class
- Price Sensitivity
- Number of flights taken by the Customer per year
- Flights delayed more than 5 minutes

The strength of the relation between Customer Satisfaction and the combination of above variables is around 45%

Insights from Association Rules

Attributes pertaining to the highly satisfied customers:

- Adult Customers (Age 18-65)
- Business Travel type
- More number of Loyalty Cards (>2)
- Delay less than 5 mins
- Low Price Sensitivity

Attributes pertaining to the low satisfied customers:

- Children and Senior Customer
- Personal Travel type
- Customer with no loyalty cards
- More flights over the year
- Female Customers
- Economy Class Customers
- Blue status Customer

Insights from Support Vector Machine

The model predicts the likelihood to recommend about customer's Rating : Happy or Unhappy

Accuracy Rate: 79.5 %

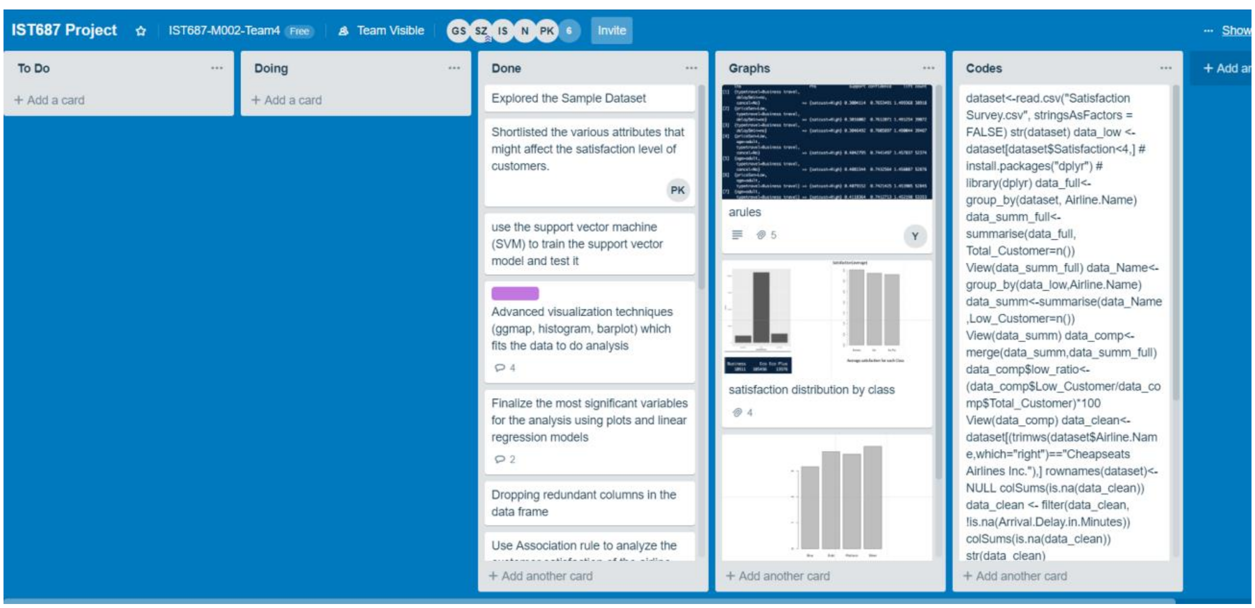
Model was based on Age, Airline Status, Type of Travel, Number of Flights, Gender, Delay greater than 5 minutes, Class and Price Sensitivity

Recommendations for Cheapseats Airline Inc

- Customers travelling for personal travel should be provide better in-flight services
- Better in-flight services like goodies and kids based entertainment services to children. Extended escort services and faster gate access to senior citizens
- Provision of better offers and opportunities to Blue customers for them to upgrade to higher status like Silver status
- Co-passenger preference to female customers travelling alone
- Provision of more loyalty cards to the frequent flyers
- Provision of promotional loyalty cards and occasional free upgrades to customers on economy class
- Provision of free lounge access and stay over facilities to the customers with delayed flights

Appendix

Trello Screenshot



Contributions:

Cleaning data, Data Munging and Selecting the Airline : Nikhil Patil

Descriptive Statistics: Yue Wang, Shu, Gaurav, Purva

Linear Model: Shu, Gaurav

Association Rules: Yue Wang, Nikhil Patil

SVM: Gaurav, Purva

Insights and Recommendation: Entire team

Report Generation: Entire team

Code

```
## Acquiring the datatset ##

dataset<-read.csv("Satisfaction Survey.csv", stringsAsFactors = FALSE)

str(dataset)

## Cleaning the dataset

unique(dataset$Satisfaction)

index_1 <- which(dataset$Satisfaction=='4.00.2.00')
index_2 <- which(dataset$Satisfaction=='4.00.5')




index_1
index_2

dataset <- dataset[-index_1, ]
dataset<- dataset[-index_2, ]
```

```
colSums(is.na(dataset))

data_clean <- filter(dataset,  !is.na(Arrival.Delay.in.Minutes))

colSums(is.na(dataset))


nrow(dataset)


unique(dataset$Satisfaction)


##Selecting an airline


data_low <- dataset[dataset$Satisfaction<4,]


# install.packages("dplyr")

# library(dplyr)


data_full<-group_by(dataset, Airline.Name)

data_summ_full<-summarise(data_full, Total_Customer=n())

View(data_summ_full)


data_Name<-group_by(data_low,Airline.Name)

data_summ<-summarise(data_Name ,Low_Customer=n())

View(data_summ)


data_comp<-merge(data_summ,data_summ_full)

data_comp$low_ratio<-
(data_comp$Low_Customer/data_comp$Total_Customer)*100

View(data_comp)
```

```
data_clean<-dataset[(trimws(dataset$Airline.Name,which="right")=="Cheapseats
Airlines Inc."),]

rownames(dataset)<-NULL


## Data Munging ##


colSums(is.na(data_clean))

data_clean <- filter(data_clean,  !is.na(Arrival.Delay.in.Minutes))

colSums(is.na(data_clean))


str(data_clean)

summary(data_clean)


unclean_names <- colnames(data_clean)

clean_names <- gsub("\\.", "_", unclean_names)

colnames(data_clean) <- clean_names


data_clean$Satisfaction <- as.numeric(data_clean$Satisfaction)


## Descriptive Statistics


#####################################################

data_clean$Satisfaction <- as.numeric(as.character(data_clean$Satisfaction))



# group by the origin state

data_clean.groupByOrigin_City <- group_by(data_clean,Orgin_City)

originCityCount <- summarize(data_clean.groupByOrigin_City,count=n())

View(originCityCount)
```

```
originCityAvgSatisfaction <- summarize(data_clean.groupByOrigin_City,
mean(Satisfaction))

View(originCityAvgSatisfaction)

# group by the destination state

data_clean.groupByDest_City <- group_by(data_clean, Destination_City)

DestCityCount <- summarize(data_clean.groupByDest_City,count=n())

View(DestCityCount)

DestCityAvgSatisfaction <- summarize(data_clean.groupByDest_City,
mean(Satisfaction))

View(DestCityAvgSatisfaction)



colnames(originCityAvgSatisfaction) <- c("Origin_City","Mean_Satisfaction")

colnames(DestCityAvgSatisfaction) <- c("Destination_City","Mean_Satisfaction")

View(originCityAvgSatisfaction)

View(DestCityAvgSatisfaction)



ScatterPlot_Origin_City <- ggplot(originCityAvgSatisfaction,aes(x=Origin_City,
y=Mean_Satisfaction))

ScatterPlot_Origin_City <- ScatterPlot_Origin_City + geom_point() +
ggtitle("Satisfaction across Origin City")

ScatterPlot_Origin_City <- ScatterPlot_Origin_City + theme(axis.text.x =
element_text(angle = 90, hjust = 1))

ScatterPlot_Origin_City #scatterplot of origin city with satisfacion


ScatterPlot_Destination_City <-
ggplot(DestCityAvgSatisfaction,aes(x=Destination_City, y=Mean_Satisfaction))

ScatterPlot_Destination_City <- ScatterPlot_Destination_City + geom_point()+
ggtitle("Satisfaction across Destination City")

ScatterPlot_Destination_City <- ScatterPlot_Destination_City + theme(axis.text.x =
element_text(angle = 90, hjust = 1))
```

```
ScatterPlot_Destination_City #scatterplot of destination city with satisfacion
#####################################################


#Maps


# group by the origin state
data_clean.groupByOrigin <- group_by(data_clean, Origin_State)
originStateCount <- summarize(data_clean.groupByOrigin,count=n())
originStateAvgSatisfaction <- summarize(data_clean.groupByOrigin,
mean(Satisfaction))
View(originStateAvgSatisfaction)


# group by the destination state
data_clean.groupByDest <- group_by(data_clean, Destination_State)
destStateCount <- summarize(data_clean.groupByDest,count=n())
destStateAvgSatisfaction <- summarize(data_clean.groupByDest, mean(Satisfaction))
View(destStateAvgSatisfaction)


Destination_State <- state.name
area <- state.area
center <- state.center
df <- data.frame(Destination_State,area,center)
colnames(destStateAvgSatisfaction) <- c("Destination_State","Mean_of_Satisfaction")
otherDF <- merge(df, destStateAvgSatisfaction, all.x=TRUE)
otherDF$Destination_State <- tolower(otherDF$Destination_State)
us<- map_data("state")#use maps package for plotting with ggplot2


map.simple <- ggplot(otherDF, aes(map_id = Destination_State)) +
guides(fill=guide_legend(title = "Mean of Satisfaction"))
map.simple <- map.simple + geom_map(map =
us,aes(fill=Mean_of_Satisfaction))#plot map on basis of area of states
```

```
map.simple <- map.simple + expand_limits(x = us$long, y = us$lat)+
ggtitle("Distribution across Destination State")#define x and yaxis limits

map.simple


Origin_State <- state.name

area <- state.area

center <- state.center

df <- data.frame(Origin_State,area,center)

colnames(originStateAvgSatisfaction) <- c("Origin_State","Mean_of_Satisfaction")

otherDF <- merge(df, originStateAvgSatisfaction, all.x=TRUE)

View(otherDF)

otherDF$Origin_State <- tolower(otherDF$Origin_State)

us<- map_data("state")#use maps package for plotting with ggplot2

map.simple <- ggplot(otherDF, aes(map_id = Origin_State)) +
guides(fill=guide_legend(title = "Mean of Satisfaction"))

map.simple <- map.simple + geom_map(map =
us,aes(fill=Mean_of_Satisfaction))#plot map on basis of area of states

map.simple <- map.simple + expand_limits(x = us$long, y = us$lat) +
ggtitle("Distribution across Origin State")#define x and yaxis limits

map.simple


# status distribution


ggplot(data=data_clean)+geom_bar(mapping = aes(x=data_clean$Airline_Status)) +
scale_x_discrete("Airline Status")


table(data_clean$Airline_Status, data_clean$Satisfaction)


table(data_clean$Airline_Status)


# satisfaction distribution of Blue
```

```
data_clean_Blue <- data_clean[data_clean$Airline_Status=='Blue',]

hist(data_clean_Blue$Satisfaction)

t_blue <- table(data_clean_Blue$Satisfaction)# 1,2,3,4,5:530 4431 5460 6539 619

n <- names(table(data_clean_Blue$Satisfaction))

d <- c(530, 4431, 5460, 6539, 619)

piepercent <- paste(round(100*d/sum(d),2),"%")



# class

summary(data_clean$Class)

table(data_clean$Class)

# class distribution

ggplot(data=data_clean)+geom_bar(mapping = aes(x=data_clean$Class))+
scale_x_discrete(name="Class") + ggtitle("Class Distribution")

table(data_clean$Class, data_clean$Satisfaction)

# mean by group, group: status
barplot(tapply(data_clean$Satisfaction, data_clean$Airline_Status, mean),col="light
blue")
```

```
table(data_clean$Airline_Status)



ggplot(data_clean_sat,aes(x=Airline_Status,fill=Satisfaction))+geom_bar(position='dodge')+scale_fill_manual(values = colorblue)

str(data_clean_sat)

ggplot(data_clean_sat,aes(x=Airline_Status,fill=Satisfaction))+geom_bar(position='fill')+scale_fill_manual(values = colorblue)


# type of travel visulization

# type of travel distribution

table(data_clean$Type_of_Travel)

barplot(tapply(data_clean$Satisfaction, data_clean$Type_of_Travel, mean), col="light blue")

barplot(table(data_clean$Type_of_Travel), col = "light blue")

table(data_clean$Type_of_Travel, data_clean$Satisfaction)


data_clean_sat <- data_clean

data_clean_sat$Satisfaction <- as.factor(data_clean_sat$Satisfaction)

ggplot(data_clean_sat,aes(x=Type_of_Travel,fill=Satisfaction))+geom_bar(position='dodge')+scale_fill_manual(values = colorblue)

ggplot(data_clean_sat,aes(x=Type_of_Travel,fill=Satisfaction))+geom_bar(position='fill')+scale_fill_manual(values = colorblue)



#ggplot(data_clean, aes(y=Satisfaction, x=Class)) + geom_point() + labs(x="Class", y="customer satisfaction")


data_clean_sat <- data_clean# i will use data_clean_sat, because i has to change satisfaction from numeric to factor
```

```
data_clean_sat$Satisfaction <- as.factor(data_clean_sat$Satisfaction)

ggplot(data_clean_sat,aes(x=Class,fill=Satisfaction))+geom_bar(position='dodge')+scale_fill_manual(values = colorblue)

ggplot(data_clean_sat,aes(x=Class,fill=Satisfaction))+geom_bar(position='fill')+scale_fill_manual(values = colorblue)

createBucketsSurvey<-function(vec)
{ vBuckets <- replicate(length(vec), "Medium")
  vBuckets[vec<3 ] <- "Low"
  vBuckets[vec>3 ] <- "High"
  return(vBuckets) }

sat_viz<-createBucketsSurvey(data_clean$Satisfaction)




male<- data_clean[data_clean$Gender=='Male',]
View(male)
nrow(male)

hist(male$Satisfaction, xlab = "Satisfaction Rating",main = "Male Satisfaction Survey")

female<- data_clean[data_clean$Gender=='Female',]
View(female)
nrow(female)
```

```
hist(female$Satisfaction, xlab = "Satisfaction Rating",main = "Female Satisfaction Survey")


ggplot(data_clean,aes(x=Gender,y=Satisfaction))+geom_boxplot()


agefunction<-function(vec)

{ vBuckets <- replicate(length(vec), "Adult")

  vBuckets[vec <= 18] <- "Child"

  vBuckets[vec >= 65] <- "Senior"

  return(vBuckets) }



age<-agefunction(data_clean$Age)

ggplot(data_clean,aes(x=age,fill=sat_viz))+geom_bar(position='dodge')+
guides(fill=guide_legend(title = "Satisfaction Range"))


createBucketsOther <- 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)

}


price<-createBucketsOther(data_clean$Price_Sensitivity)

ggplot(data_clean,aes(x=price,y=Satisfaction))+geom_boxplot()


sat_vize<-createBucketsSurvey(dataset$Satisfaction)

ggplot(dataset,aes(x=Airline.Name,fill=sat_vize))+geom_bar(position='dodge')+theme(axis.text.x = element_text(angle=90,hjust=1))+guides(fill=guide_legend(title = "Satisfaction Range"))
```

```
##Linear Regression ##

str(data_clean)

cheapseatnew<-data_clean[,c(-11,-12,-16:-21)]

str(cheapseatnew)



which(colnames(cheapseatnew)=="Flight_cancelled")
cheapseatnew <- cheapseatnew[,-(17)]

# cheapseatnew[1:21]<-lapply(cheapseatnew[1:21],as.factor)

str(cheapseatnew)

numairline<-as.factor(cheapseatnew$Airline_Status)
numgender<-as.factor(cheapseatnew$Gender)
numttravel<-as.factor(cheapseatnew$Type_of_Travel)
numclass<-as.factor(cheapseatnew$Class)
num5min<-as.factor(cheapseatnew$Arrival_Delay_greater_5_Mins)

cheaplineardata<-
data.frame(cheapseatnew$Satisfaction,numairline,cheapseatnew$Age,numgender,
cheapseatnew$Price_Sensitivity,cheapseatnew$Year_of_First_Flight,cheapseatnew$N
o_of_Flights_p_a_,
cheapseatnew$X__of_Flight_with_other_Airlines,numttravel,cheapseatnew$No__of_ot
her_Loyalty_Cards,
```

```
numclass,cheapseatnew$Scheduled_Departure_Hour,cheapseatnew$Departure_Dela
y_in_Minutes,

cheapseatnew$Flight_time_in_minutes,cheapseatnew$Flight_Distance,num5min)


str(cheaplineardata)


cheaplinear<-lm(cheapseatnew.Satisfaction~.,data=cheaplineardata)

summary(cheaplinear)


library('MASS')

null<-lm(cheapseatnew.Satisfaction~1,cheaplineardata)

stepAIC(cheaplinear, direction='backward')


lm_backward <- lm(cheapseatnew.Satisfaction ~ numairline + cheapseatnew.Age +

   numgender + cheapseatnew.Price_Sensitivity + cheapseatnew.Year_of_First_Flight
+

   cheapseatnew.No_of_Flights_p_a_ + numttravel + numclass +

   cheapseatnew.Scheduled_Departure_Hour + num5min, data = cheaplineardata)


summary(lm_backward)


stepAIC(null,direction='forward',scope=list(upper=cheaplinear,lower=null))


lm_forward <- lm(formula = cheapseatnew.Satisfaction ~ numttravel + numairline +

              num5min + numgender + cheapseatnew.No_of_Flights_p_a_ +
cheapseatnew.Age +

              cheapseatnew.Price_Sensitivity + numclass +
cheapseatnew.Scheduled_Departure_Hour +

              cheapseatnew.Year_of_First_Flight, data = cheaplineardata)


summary(lm_forward)
```

```
# Association Rules ##

colSums(is.na(data_arules))
data_clean <- filter(data_arules,  !is.na(Arrival_Delay_in_Minutes))
colSums(is.na(data_arules))

createBuckets <- function(v){
  vBuckets <- replicate(length(v), "Average")
  vBuckets[v > 3] <- "High"
  vBuckets[v <= 3] <- "Low"
  return(vBuckets)
}

data_arules = data_clean

str(data_arules)

satcust <- createBuckets(data_arules$Satisfaction)
# satcust

# price sensitivity
priceSen <- createBuckets(data_clean$Price_Sensitivity)
# priceSen



createBucketsCard <- function(v){
  vBuckets <- replicate(length(v), "No")
```

```
  vBuckets[v > 0] <- "Yes"
  vBuckets[v >= 2] <- "more"
  return(vBuckets)
}

# No..of other loyalty cards
createBucketsCard <- function(v){
  vBuckets <- replicate(length(v), "No")
  vBuckets[v > 0] <- "Yes"
  vBuckets[v >= 2] <- "more"
  return(vBuckets)
}

NumCards <- createBucketsCard(data_arules$No__of_other_Loyalty_Cards)
# NumCards

# age
# summary(data_arules$Age)
createBucketsAge <- function(v){
  vBuckets <- replicate(length(v), "teenager")
  vBuckets[v >= 20] <- "adult"
  vBuckets[v >= 65] <- "senior"
  return(vBuckets)
}
age <- createBucketsAge(data_arules$Age)
# age

createBucketsOther <- 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)
}

# No. of flight p.a
NumFlight <- createBucketsOther(data_arules$No_of_Flights_p_a_)
# NumFlight
# shopping amount at airport
shopping <- createBucketsOther(data_arules$Shopping_Amount_at_Airport)
eatdrink <- createBucketsOther(data_arules$Eating_and_Drinking_at_Airport)
# departure hour
departureHour <- createBucketsOther(data_arules$Scheduled_Departure_Hour)
# flight time in minutes

colSums(is.na(data_arules))

flightTime <- createBucketsOther(data_arules$Flight_time_in_minutes)
# flight distance
flightDis <- createBucketsOther(data_arules$Flight_Distance)
# X.. of flight with other airline
otherAirline <- createBucketsOther(data_arules$X__of_Flight_with_other_Airlines)
# otherAirline




# delay in munutes
createBucketsDelay <- function(v){
  vBuckets <- replicate(length(v), "NoDelay")
```

```
 vBuckets[v > 0] <- "small"
 vBuckets[v >= 17] <- "middle"
 vBuckets[v >= 45] <- "long"
 return(vBuckets)
}
# departure delay
depDelay <- createBucketsDelay(data_arules$Departure_Delay_in_Minutes)
# depDelay
# arrive delay
arrDelay <- createBucketsDelay(data_arules$Arrival_Delay_in_Minutes)
# arrDelay

# other categracal variable
status <- data_arules$Airline_Status
gender <- data_arules$Gender
firstflight <- data_arules$Year_of_First_Flight
typetravel <- data_arules$Type_of_Travel
class <- data_arules$Class
airlinename <- data_arules$Airline_Name
dayOfMonth <- data_arules$Day_of_Month
flightdate <- data_arules$Flight_date
depState <- data_arules$Origin_State
arrState <- data_arules$Destination_State
delay5min <- data_arules$Arrival_Delay_greater_5_Mins
cancel <- data_arules$Flight_cancelled



data_arules <- data.frame(satcust, priceSen, otherAirline,NumCards,age, NumFlight,
shopping, eatdrink,departureHour,flightTime,flightDis,depDelay,
              arrDelay, status, gender, firstflight, typetravel,class, delay5min, cancel )
```

```
data_arules[1:20]<-lapply(data_arules[1:20],as.factor)


str(data_arules)


data_arules.trans <- as(data_arules, "transactions")

itemFrequencyPlot(data_arules.trans, support=0.3,cex.names=0.6)


ruleset_High <- apriori(data_arules.trans, parameter=list(support=0.2,
confidence=0.2),
                appearance=list(default="lhs",rhs=("satcust=High")))


ruleset_Low <- apriori(data_arules.trans, parameter=list(support=0.1,
confidence=0.5),
                 appearance=list(default="lhs",rhs=("satcust=Low")))


ruleset_high <- sort(ruleset_High, decreasing=TRUE, by="confidence")

ruleset_low <- sort(ruleset_Low, decreasing=TRUE, by="confidence")


library(grid)

library(arulesViz)


plot(ruleset_high)

plot(ruleset_low)


goodrules_high <- ruleset_high[quality(ruleset_high)$lift > 1.4]

goodrules_high


goodrules_high <- sort(goodrules_high, descreasing=TRUE, by="lift")
```

```
inspect(head(goodrules_high,15))

plot(goodrules_high)

inspect(goodrules_high)

goodrules_low <- ruleset_low[quality(ruleset_low)$lift > 1.95]
goodrules_low

goodrules_low <- sort(goodrules_low, descreasing=TRUE, by="lift")

inspect(head(goodrules_low,20))
plot(goodrules_low)

##Support Vector Machine ##

data_clean$happy_customer <- createBuckets(data_clean$Satisfaction)

dim(data_clean)

table(data_clean$happy_customer)

randIndex <- sample(1:dim(data_clean)[1])
```

```
summary(randIndex)
length(randIndex)
head(randIndex)
cutPoint2_3 <- floor(2 * dim(data_clean)[1]/3)
cutPoint2_3
trainData <- data_clean[randIndex[1:cutPoint2_3],]
View(trainData)
testData <-data_clean[randIndex[(cutPoint2_3+1):dim(data_clean)[1]],]
View(testData)
dim(trainData)
dim(testData)

install.packages('kernlab')
library(kernlab)

str(trainData)

svmOutput <- ksvm(happy_customer ~ Type_of_Travel + Age + Airline_Status +
No_of_Flights_p_a_ + Scheduled_Departure_Hour + Gender +
Arrival_Delay_greater_5_Mins + Class + Price_Sensitivity + Airline_Status ,
data=trainData,kernel= "rbfdot",kpar = "automatic",C=5,cross=3,prob.model=TRUE)
svmOutput

Pred <- predict(svmOutput, testData, type = "votes")

compTable2 <- data.frame(testData[,29], Pred[2,])
comp2 <- table(compTable2)
comp2

fourfoldplot(comp2, color = c("#CC6666", "#99CC99"),
```

```
        conf.level = 0, margin = 1, main = "Confusion Matrix")

error_rate_percentage <- (comp2[2,1] + comp2[1,2])/nrow(testData)*100
error_rate_percentage


## End of Code ##
```