August 2020

IST 687 APPLIED DATA SCIENCE PROJECT

ANALYSIS OF AIRLINES DATA

RASHIKA PRAMOD SINGH

SYRACUSE UNIVERSITY

Table of Contents

Problem Statement

Southeast Airlines needs to lower their customer churn i.e. the number of customers that stopped using its service. The airlines are losing customers as even the loyalty program is not helping it to retain customers. The airlines need to identify some of the causes of losing the customers and identify the key metrics which affect the churn rate. The below insights could be used in increasing the customer retention and help the business.

Data set and Descriptive Statistics

The data set used is the survey data collected by Southeast airlines, this data is used to predominantly calculated NPS. It contains thousands of observations and each row represents a flight segment, by airline and each column represents an attribute of that particular flight segment. The dataset has 5,000 rows and 31 columns. Descriptive statistics were used to analyze the overall data set which provided information about the mean, median, maximum and minimum values. For example, the average age of a customer is 46.2 years while the maximum age is 86 years while minimum is 15 years.

```
 Destination.City    Orgin.City        Airline.Status          Age          Gender
 Length:5000        Length:5000        Length:5000        Min.   :15.0   Length:5000
 Class :character   Class :character   Class :character   1st Qu.:33.0   Class :character
 Mode  :character   Mode  :character   Mode  :character   Median :45.0   Mode  :character
                                                          Mean   :46.2
                                                          3rd Qu.:58.0
                                                          Max.   :85.0

 Price.Sensitivity Year.of.First.Flight Flights.Per.Year    Loyalty          Type.of.Travel
 Min.   :0.000     Min.   :2003         Min.   : 0.00    Min.   :-0.97368   Length:5000
 1st Qu.:1.000     1st Qu.:2005         1st Qu.: 8.00    1st Qu.:-0.70000   Class :character
 Median :1.000     Median :2007         Median :17.00    Median :-0.42857   Mode  :character
 Mean   :1.275     Mean   :2007         Mean   :19.86    Mean   :-0.27013
 3rd Qu.:2.000     3rd Qu.:2010         3rd Qu.:29.00    3rd Qu.: 0.05882
 Max.   :4.000     Max.   :2012         Max.   :92.00    Max.   : 1.00000

 Total.Freq.Flyer.Accts Shopping.Amount.at.Airport Eating.and.Drinking.at.Airport    Class
 Min.   : 0.0000        Min.   :  0.00             Min.   :  0.00                 Length:5000
 1st Qu.: 0.0000        1st Qu.:  0.00             1st Qu.: 30.00                 Class :character
 Median : 0.0000        Median :  0.00             Median : 60.00                 Mode  :character
 Mean   : 0.9152        Mean   : 27.09             Mean   : 69.34
 3rd Qu.: 2.0000        3rd Qu.: 30.00             3rd Qu.: 90.00
 Max.   :10.0000        Max.   :540.00             Max.   :535.00

  Day.of.Month   Flight.date        Partner.Code       Partner.Name       Origin.State
 Min.   : 1.00   Length:5000        Length:5000        Length:5000        Length:5000
 1st Qu.: 8.00   Class :character   Class :character   Class :character   Class :character
 Median :16.00   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :15.83
 3rd Qu.:23.00
 Max.   :31.00

 Destination.State  Scheduled.Departure.Hour Departure.Delay.in.Minutes Arrival.Delay.in.Minutes
 Length:5000        Min.   : 1.00           Min.   :  0.00             Min.   :  0.00
 Class :character   1st Qu.: 9.00           1st Qu.:  0.00             1st Qu.:  0.00
 Mode  :character   Median :13.00           Median :  0.00             Median :  0.00
                    Mean   :12.96           Mean   : 14.93             Mean   : 15.59
                    3rd Qu.:17.00           3rd Qu.: 13.00             3rd Qu.: 15.00
                    Max.   :23.00           Max.   :514.00             Max.   :624.00
                                            NA's   :90                 NA's   :96
 Flight.cancelled   Flight.time.in.minutes Flight.Distance  Likelihood.to.recommend     olong
 Length:5000        Min.   : 14.0          Min.   :  67.0   Min.   : 2.000          Min.   :-165.39
 Class :character   1st Qu.: 61.0          1st Qu.: 370.0   1st Qu.: 7.000          1st Qu.:-111.93
 Mode  :character   Median : 91.0          Median : 621.0   Median : 9.000          Median : -89.51
                    Mean   :113.5          Mean   : 804.1   Mean   : 7.428          Mean   : -95.12
                    3rd Qu.:144.0          3rd Qu.:1024.0   3rd Qu.: 9.000          3rd Qu.: -81.61
                    Max.   :389.0          Max.   :3302.0   Max.   :10.000          Max.   : -66.12
                    NA's   :96
```

```
      olat           dlong              dlat      
 Min.   :18.02   Min.   :-161.78   Min.   :18.02  
 1st Qu.:33.82   1st Qu.:-111.93   1st Qu.:33.82  
 Median :37.67   Median : -90.34   Median :37.67  
 Mean   :37.10   Mean   : -95.45   Mean   :37.09  
 3rd Qu.:40.72   3rd Qu.: -81.64   3rd Qu.:41.07  
 Max.   :71.29   Max.   : -66.12   Max.   :71.29  
```

Business Questions

Some business questions that can be used to analyze likelihood to recommend:

1. What age groups travel the most so that effective marketing strategies can be used?
2. Which gender is the promoter for the airlines, and which is detractor?
3. What is the NPS across different airlines?
4. What is the reason for low NPS?
5. Does type of travel affect likelihood to recommend?
6. Which airlines can Southeast Airlines partner with?

Data Cleaning

There was a need to check NA values in the data set. The NA values were replaced with the mean values of that attribute. There were NA values in Arrival.Delay.in.Miutes, Flight.time.in.minutes, Departure, Delay.in.Minutes.
I also had to change the data type of certain variables for analysis like changing numeric values to categorical or creating subgroups of attributes such as age.
There were also attributes added to the data which included arrival delay greater than 5 minutes which was if the arrival delay was greater than 5 minutes or not. Similarly, it was calculated if the departure delay is more than 5 minutes or not. Another attribute was converting the likelihood into categorical values i.e. if the likelihood is less than 7, the customer is a detractor, greater than or equal to 7 and less than 8 is a passive and greater than 8 is a promoter. Net promoter score (NPS) was calculated as subtracting detractors from promoters.

Code snippets:

```
#Data loading and cleaning
airData<-fromJSON("project.json")
df<-data.frame(airData)
View(df)
str(df)
summary(df)
#contains 5000 rows and 31 columns

sapply(airData,function(x)sum(is.na(x))) #check for null values
#columns with null values
#Departure.Delay.in.Minutes
#airData$Arrival.Delay.in.Minutes
#Flight.time.in.minutes

#Removing null values
airData$Arrival.Delay.in.Minutes[is.na(airData$Arrival.Delay.in.Minutes)]=round(mean(airData$Arrival.Delay.in.Minutes, na.rm=TRUE))
airData$Flight.time.in.minutes[is.na(airData$Flight.time.in.minutes)]=round(mean(airData$Flight.time.in.minutes, na.rm=TRUE))
airData$Departure.Delay.in.Minutes[is.na(airData$Departure.Delay.in.Minutes)]=round(mean(airData$Departure.Delay.in.Minutes, na.rm=TRUE))
View(df)

#Adding attributes to the data
df1<-df
for (i in 1:length(df1$Arrival.Delay.in.Minutes)) #calculating arrival delay if greater than 5 minutes
{
  if (df1$Arrival.Delay.in.Minutes[i] > 5)
  {
    df1$ArrivalDelay[i] <- "Yes"
  }
  else
  {
    df1$ArrivalDelay[i] <- "No"
  }
}

#calculating if departure delay is greater than 5 minutes
for (i in 1:length(df1$Departure.Delay.in.Minutes))
{
  if (df1$Departure.Delay.in.Minutes[i] > 5)
  {
    df1$DepartDelay[i] <- "Yes"
  }
  else
  {
    df1$DepartDelay[i] <- "No"
  }
}
```

```r
#Converting likelihood into categorical values
for (i in 1:length(df1$Likelihood.to.recommend))
{
  if (df1$Likelihood.to.recommend[i] < 7)
  {
    df1$likelihood[i] <- "Detractor"
  }
  else if (df$Likelihood.to.recommend[i] >= 7 & df$Likelihood.to.recommend[i] <= 8)
  {
    df1$likelihood[i] <- "Passive"
  }
  else
  {
    df1$likelihood[i] <- "Promoter"
  }
}

#calculating NPS
count <- table(df1$Partner.Name,df1$likelihood)
dim(count)
npsdata <- data.frame(Airlines = unique(rownames(count)),Detractor = count[,1],Passive = count[,2],Promoters = count[,3])
npsdata$NPS <- ((npsdata$Promoters - npsdata$Detractor)/(npsdata$Promoters + npsdata$Detractor + npsdata$Passive))*100
```

Exploratory Data Analysis

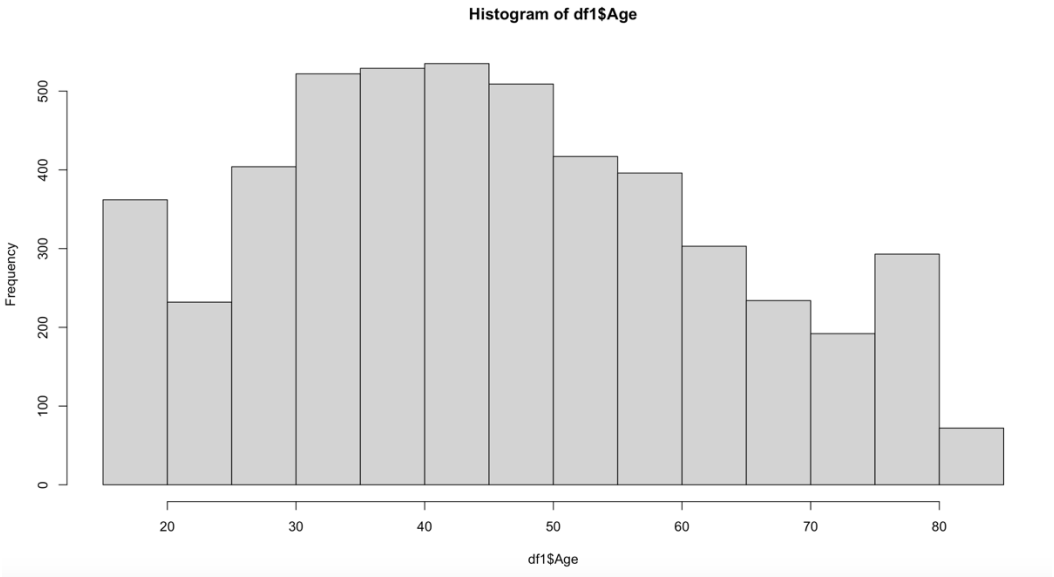
Exploratory data analysis was used to identify some of the underlying features and get a better understanding of the data. Some of the business questions were answered using this analysis.

Code

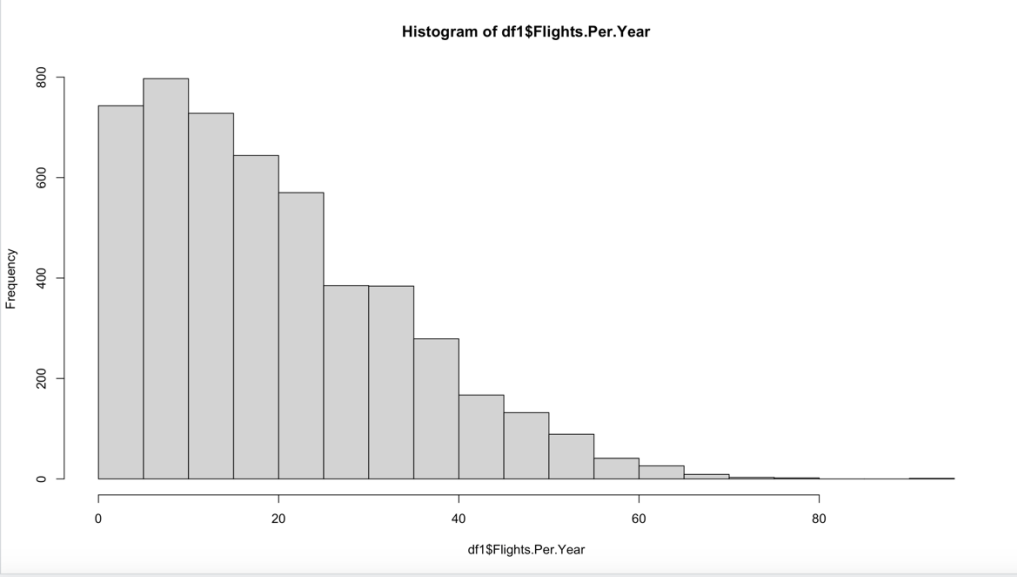
```
#examining relation between likelihood to recommend and price sensitivity
price<-ggplot(df1,aes(y=Likelihood.to.recommend,x=Price.Sensitivity,fill=Flight.cancelled))+geom_boxplot() #create a boxplot
price<-price+ ggtitle("Likelihood to recommend based on price sensitivity")
price

#examining relation between type of travel and likelihood to recomend
travel <- ggplot(df1,aes(x=Type.of.Travel, y=Likelihood.to.recommend))
travel <- travel + geom_col()
travel <- travel + theme(axis.text.x = element_text(angle = 90, hjust = 1))
travel

#create histogram to examine relation between gender and likelihood
ggplot(df1,aes(x=Gender)) + geom_histogram(stat="count",color="black",aes(fill=likelihood))

#create maps to examine likelihood by state
us <- map_data("state")
df1$Origin.State <- tolower(df1$Origin.State)
map1<- ggplot(df1, aes(map_id = Origin.State, label = Origin.State)) #ggplot for adding the data and map_id for specifying the data for map
map1<- map1 + geom_map(map = us, aes(fill=likelihood, x=dlong, y=dlat)) #adding the map as us
map1
```

Histograms for numeric variables

I used histograms to plot the counts of numeric variables such as age and number of flights per year to understand the distribution of the values across these variables.



The maximum number of travelers are between the age group 30 to 50 years old.



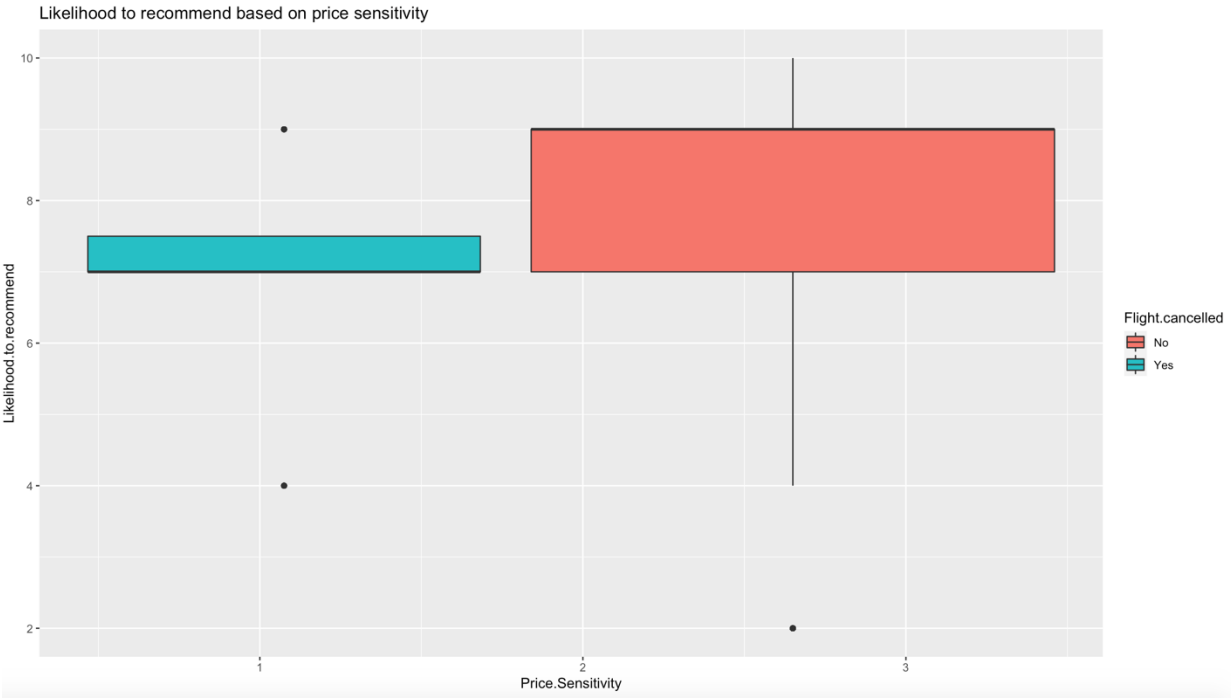
The maximum count of flights per year is mainly between 5-10.

Tables for categorical variables

Tables are created to get the count of categorical variables like there are more females than males and the maximum number of customers are in the Blue airline status with maximum passengers traveling business class.

```
Female   Male
  2820   2180
> table(df1$Airline.Status) #There are 4 categories of Airline status

    Blue     Gold Platinum   Silver
    3442      421      150      987
> table(df1$Type.of.Travel)

Business travel Mileage tickets Personal Travel
           3041             389            1570
```

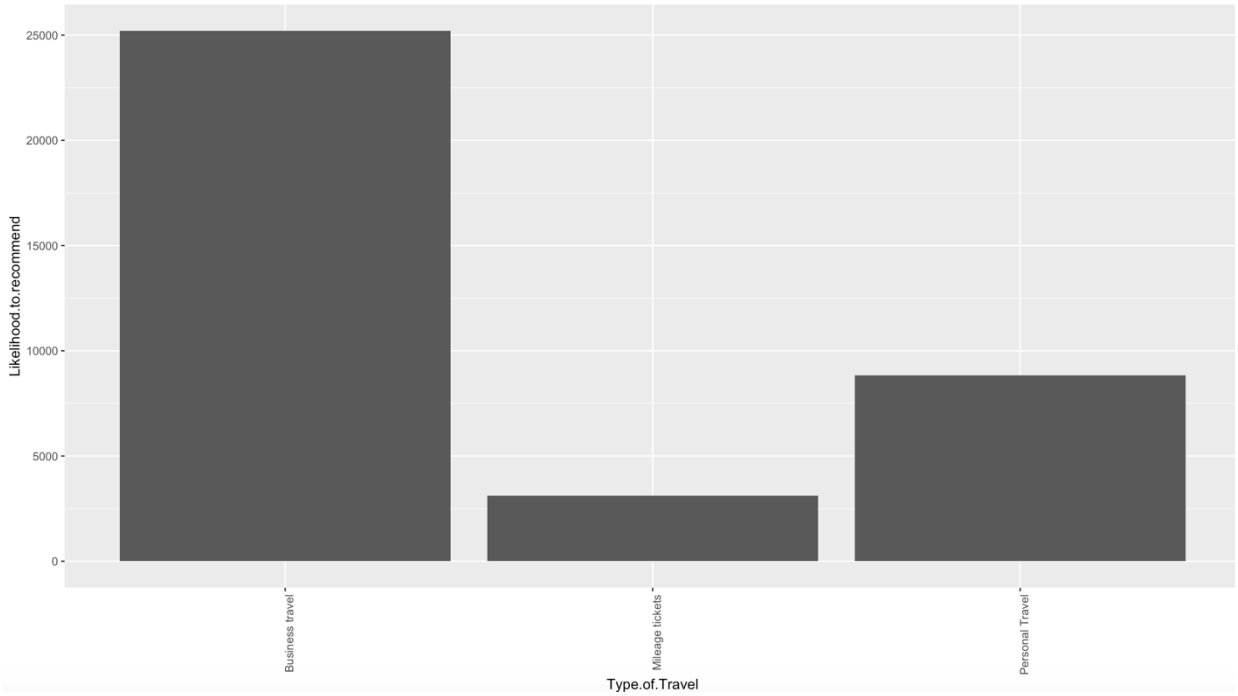
Boxplots for analyzing features

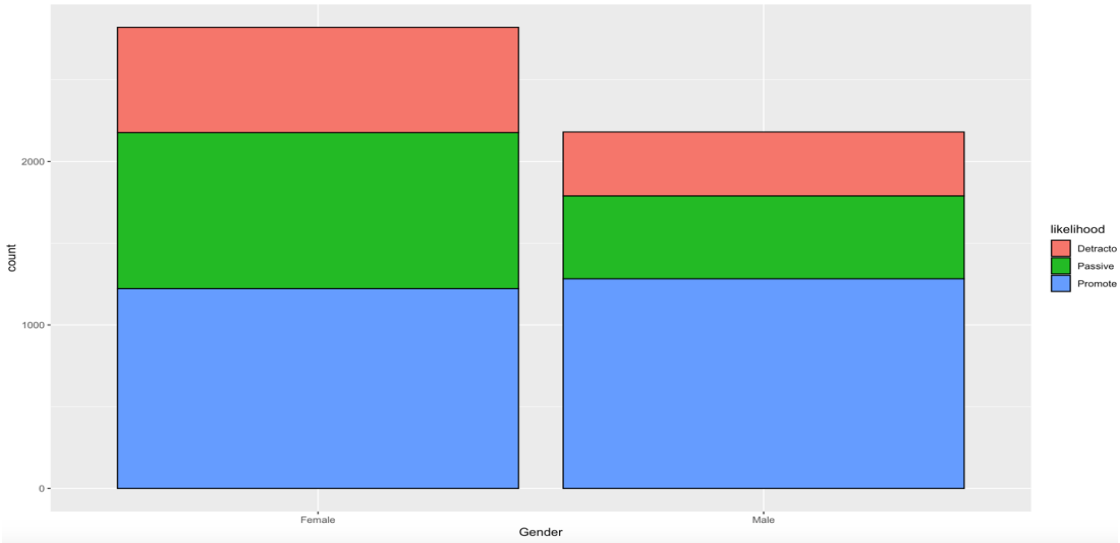


Customers whose flights are not cancelled are more likely to recommend the airlines

Barplots and histograms

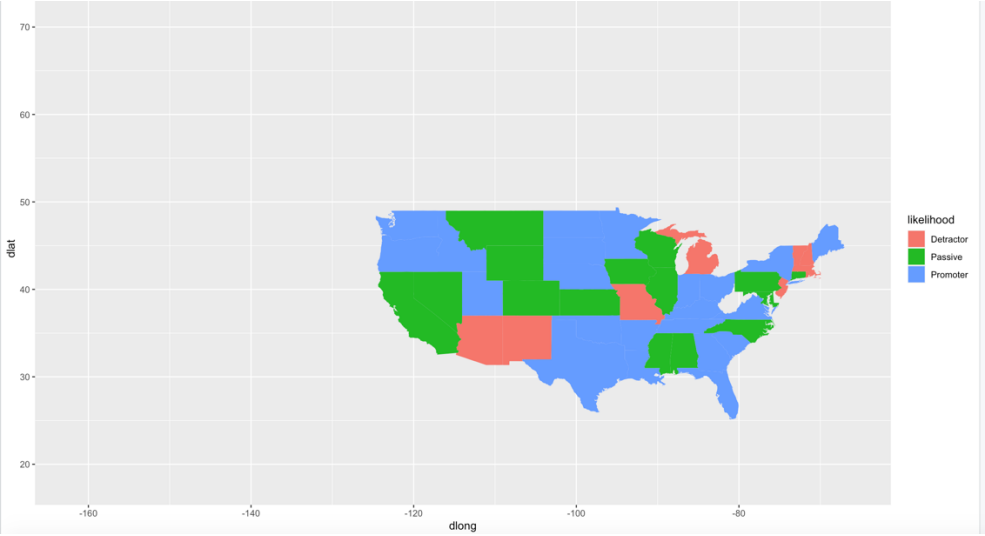
The histogram shows that maximum of the business travelers is likely to recommend the airlines.





The histogram shows that the number of promoters in male passengers and female passengers are almost the same.

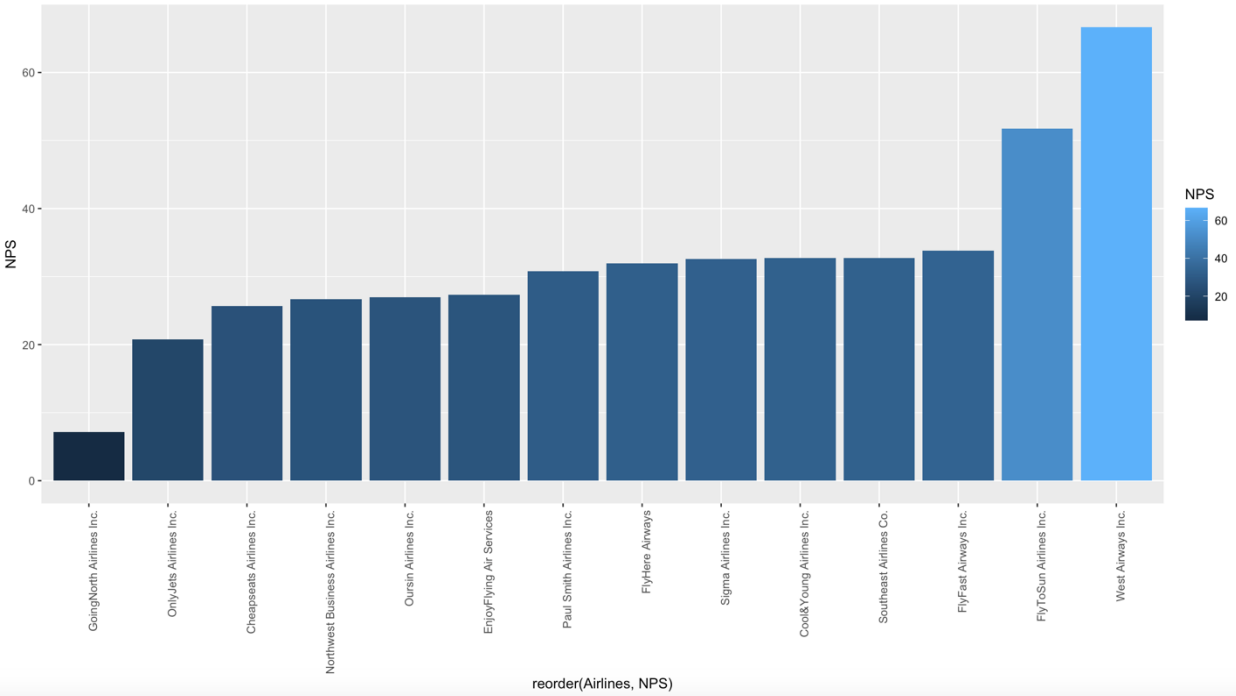
Maps for location data



Customers at the southern part of USA are more likely to recommend the airlines.

Using NPS to plot visualizations

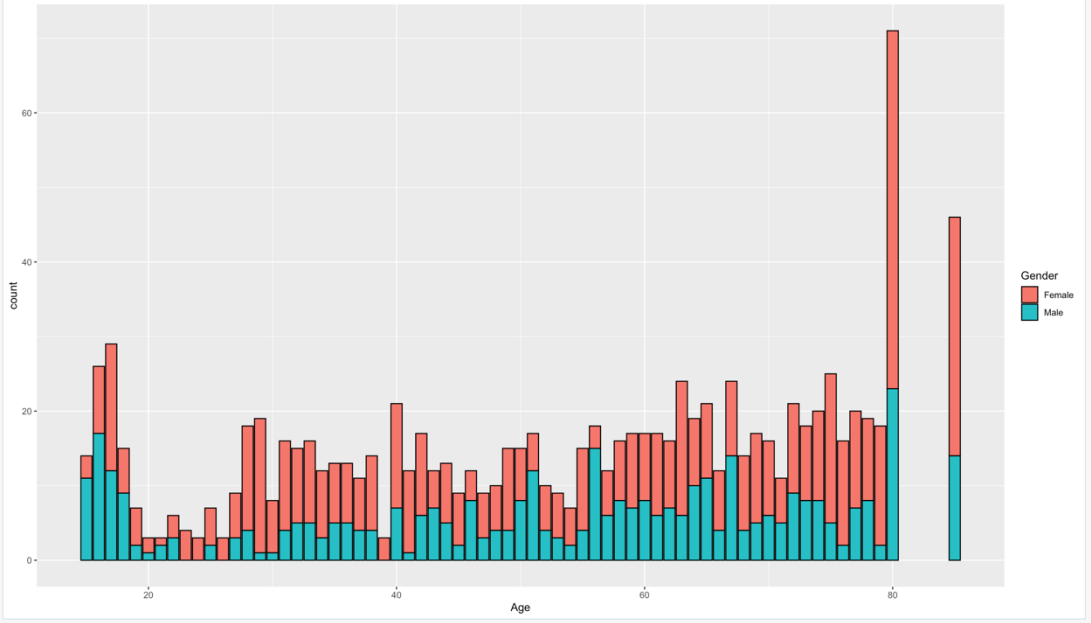
West Airways has the greatest number of NPS.



Creating separate promoter and detractor data and visualizing the data.

Examining the promoter data across different genders.

Examining the detractor data across different genders.



Models:

Linear Modeling

Linear modeling is used to predict the value of the dependent variable Y based on the independent variables X which can be one or more than one. We try to establish a linear relationship between the variables.
For linear modeling, I needed to convert categorical values into numerical values as type of travel was categorical and gender is also converted into 0,1.
I tried various combinations of variables for the modeling like age, flight per year, price sensitivity, price per year, departure delay, flight distance.

Code

```
#---Linear modeling-------------------------------------------------------------------
df4<-df1

#converting categorical into numeric values
df4$Type.of.Travel <- gsub('Business travel', 0, df4$Type.of.Travel)
df4$Type.of.Travel <- (gsub('Mileage tickets', 1, df4$Type.of.Travel))
df4$Type.of.Travel <- (gsub('Personal Travel', 2, df4$Type.of.Travel))


#converting categorical into numeric values
df4$Gender <- (gsub('Male', 0, df4$Gender))
df4$Gender <- (gsub('Female', 1, df4n$Gender))


#regression model for predicting likelihood for recomendation by age
model1<-lm(formula =Likelihood.to.recommend~Age, data=df1)
summary(model1)

#regression model for predicting likelihood for recomendation by age, flights per year
model2<-lm(formula =Likelihood.to.recommend~Age+Flights.Per.Year, data=df1)
summary(model2)

#regression model for predicting likelihood for recomendation by age, flights per year, price sensitivity
model3<-lm(formula =Likelihood.to.recommend~Age+Flights.Per.Year+Gender+Price.Sensitivity, data=df1)
summary(model3)

#regression model for predicting likelihood for recomendation by age, flights per year, price sensitivity, type of travel, departure delay
model4<-lm(formula =Likelihood.to.recommend~Age+Flights.Per.Year+Gender+Price.Sensitivity+Type.of.Travel+Flight.Distance+Departure.Delay.in.Minutes, data=df1)
summary(model4)
```

R-square value of this model 1 is low and hence we add more predictors.

```
Call:
lm(formula = Likelihood.to.recommend ~ Age, data = df1)

Residuals:
    Min      1Q  Median      3Q     Max
-6.0393 -1.1301  0.6884  1.5660  3.4739

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.826101   0.082400  107.11   <2e-16 ***
Age         -0.030263   0.001668  -18.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.061 on 4998 degrees of freedom
Multiple R-squared:  0.06178,   Adjusted R-squared:  0.0616
F-statistic: 329.1 on 1 and 4998 DF,  p-value: < 2.2e-16
```

Results of model 2 show a better R squared, we add more predictors to examine the results. Adding flights per year improves the r-squared value.

```
Call:
lm(formula = Likelihood.to.recommend ~ Age + Flights.Per.Year,
    data = df1)

Residuals:
   Min     1Q Median     3Q    Max
-6.269 -1.084  0.732  1.485  3.642

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       9.050066   0.084475  107.13   <2e-16 ***
Age              -0.025959   0.001704  -15.23   <2e-16 ***
Flights.Per.Year -0.021286   0.002089  -10.19   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.04 on 4997 degrees of freedom
Multiple R-squared:  0.08088,   Adjusted R-squared:  0.08051
F-statistic: 219.9 on 2 and 4997 DF,  p-value: < 2.2e-16
```

Model 3 shows a better R-Square than model 2 by using age, flights per year, price sensitivity

```
lm(formula = Likelihood.to.recommend ~ Age + Flights.Per.Year +
    Gender + Price.Sensitivity, data = df1)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8529 -1.0794  0.6643  1.4894  3.6567

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        9.271451   0.112534  82.388  < 2e-16 ***
Age               -0.025694   0.001691 -15.195  < 2e-16 ***
Flights.Per.Year  -0.021847   0.002067 -10.571  < 2e-16 ***
GenderMale         0.502618   0.057664   8.716  < 2e-16 ***
Price.Sensitivity -0.346255   0.050959  -6.795 1.21e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.015 on 4995 degrees of freedom
Multiple R-squared:  0.1036,    Adjusted R-squared:  0.1029
F-statistic: 144.4 on 4 and 4995 DF,  p-value: < 2.2e-16
```

Model 4 regression model for predicting likelihood uses age, flights per year, price sensitivity, type of travel, departure delay and shows a better R-squared value.

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    8.838e+00  1.047e-01  84.421  < 2e-16 ***
Age                           -6.388e-03  1.543e-03  -4.140 3.53e-05 ***
Flights.Per.Year              -4.515e-03  1.848e-03  -2.443 0.014583 *
GenderMale                     1.769e-01  5.050e-02   3.502 0.000465 ***
Price.Sensitivity             -1.505e-01  4.433e-02  -3.395 0.000692 ***
Type.of.TravelMileage tickets -2.504e-01  9.481e-02  -2.641 0.008300 **
Type.of.TravelPersonal Travel -2.514e+00  6.050e-02 -41.549  < 2e-16 ***
Flight.Distance               -3.986e-05  4.163e-05  -0.957 0.338469
Departure.Delay.in.Minutes    -4.556e-03  6.529e-04  -6.978 3.40e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.726 on 4901 degrees of freedom
  (90 observations deleted due to missingness)
Multiple R-squared:  0.3448,    Adjusted R-squared:  0.3438
F-statistic: 322.5 on 8 and 4901 DF,  p-value: < 2.2e-16
```

Association rule mining

Association rules are used to find association between the variables by finding frequent patterns in the data set such as a person likely to buy x is also likely to buy y. I have used association rules to determine the factors which have maximum impact for promoters and detractors.

Support for a rule is frequency of co-occurrence: LHS and RHS together. Confidence of a rule is proportion of time that LHS and RHS occur together vs. the total number of appearances of LHS. Lift is the confidence/probability of the RHS occurring.

First, I removed all the unnecessary columns and grouped the variables to create subgroups like flight distance is average, low or high and then used apriori for getting a ruleset. I also converted the data frame as an input to apriori() by coercing it to transaction class.

Code

```
#---Association rule mining----------------------------------------------------------
dfnew<-df1

#Removing  columns not necessary for analysis
dfnew <- subset(dfnew, select = -Origin.State)
dfnew <- subset(dfnew, select = -Destination.State)
dfnew <- subset(dfnew, select = -Destination.City)
dfnew <- subset(dfnew, select = -Orgin.City)
dfnew <- subset(dfnew, select = -Day.of.Month)
dfnew <- subset(dfnew, select = -olong)
dfnew <- subset(dfnew, select = -olat)
dfnew <- subset(dfnew, select = -dlong)
dfnew <- subset(dfnew, select = -dlat)
dfnew <- subset(dfnew, select = -Partner.Name)
dfnew <- subset(dfnew, select = -Airline.Status)
View(dfnew)

#create groups with the columns as average, low and high
creategroups<-function(columns)
{
  parts=quantile(columns,c(0.45,0.55))
  groups<-rep("Average",length(columns))
  groups[columns<=parts[1]]<-"Low"
  groups[columns>=parts[2]]<-"High"
  return(as.factor(groups))
}

#using the function to create subgroups in the columns
dfnew$Loyalty<-creategroups(dfnew$Loyalty)
dfnew$Eating.and.Drinking.at.Airport<-creategroups(dfnew$Eating.and.Drinking.at.Airport)
dfnew$Flight.Distance<-creategroups(dfnew$Flight.Distance)
dfnew$Total.Freq.Flyer.Accts<-creategroups(dfnew$Total.Freq.Flyer.Accts)
dfnew$Price.Sensitivity<-creategroups(dfnew$Price.Sensitivity)
dfnew$Flights.Per.Year<-creategroups(dfnew$Flights.Per.Year)

#converting dataframe into factors
dfnew<-mutate_all(dfnew,as.factor)
#converting to transaction matrix
dfnew_x<-as(dfnew,"transactions")
summary(dfnew_x)
#inspect the matrix
inspect(dfnew_x)
```

```
#inspect the matrix
inspect(dfnew_x)

itemFrequency(dfnew_x)

#apriori for association rule where the lhs is default and rhs is likehood to recommend for promoters
ruleset<-apriori(dfnew_x,parameter=list(support=0.04,maxlen=20, confidence=0.5),appearance = list(default="lhs",rhs=("likelihood=Promoter")))

#apriori for association rule where the lhs is default and rhs is likehood to recommend for detractors
rule<-apriori(dfnew_x,parameter=list(support=0.04,maxlen=20, confidence=0.5),appearance = list(default="lhs",rhs=("likelihood=Detractor")))

#inspect promoters in rhs
inspectDT(ruleset)

#inspect detractors in rhs
inspectDT(rule)
```

Rule set for promoters. Some common trends include promoters have average number of flights per year and have taken their first flight a few years ago and partner code of OU.

[1]	{}	{likelihood=Promoter}	0.501	0.501	1.000	1.000	2,504.000
[2]	{Type.of.Travel=Mileage tickets}	{likelihood=Promoter}	0.045	0.573	0.078	1.145	223.000
[3]	{Flights.Per.Year=Average}	{likelihood=Promoter}	0.045	0.567	0.079	1.132	225.000
[4]	{Class=Business}	{likelihood=Promoter}	0.050	0.597	0.084	1.191	250.000
[5]	{Partner.Code=US}	{likelihood=Promoter}	0.044	0.519	0.084	1.036	219.000
[6]	{Year.of.First.Flight=2008}	{likelihood=Promoter}	0.048	0.529	0.090	1.056	238.000
[7]	{Year.of.First.Flight=2007}	{likelihood=Promoter}	0.049	0.529	0.093	1.057	245.000
[8]	{Year.of.First.Flight=2009}	{likelihood=Promoter}	0.049	0.528	0.093	1.054	246.000
[9]	{Partner.Code=OU}	{likelihood=Promoter}	0.047	0.507	0.093	1.013	237.000
[10]	{Likelihood.to.recommend=10}	{likelihood=Promoter}	0.094	1.000	0.094	1.997	468.000

Rule set for detractors include personal travelers with partner code WN and male passengers.

	LHS	RHS	support	confidence	coverage	lift	count
	All	All	All	All	All	All	All
[1]	{Likelihood.to.recommend=4}	{likelihood=Detractor}	0.183	1.000	0.183	4.831	917.000
[2]	{Type.of.Travel=Personal Travel}	{likelihood=Detractor}	0.159	0.506	0.314	2.443	794.000
[3]	{Partner.Code=WN,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.046	1.000	0.046	4.831	229.000
[4]	{Type.of.Travel=Personal Travel,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.146	1.000	0.146	4.831	731.000
[5]	{Gender=Male,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.068	1.000	0.068	4.831	339.000
[6]	{Flight.Distance=High,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.088	1.000	0.088	4.831	440.000
[7]	{Eating.and.Drinking.at.Airport=Low,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.077	1.000	0.077	4.831	385.000
[8]	{Loyalty=Low,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.106	1.000	0.106	4.831	528.000
[9]	{Flight.Distance=Low,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.077	1.000	0.077	4.831	385.000
[10]	{Loyalty=High,Likelihood.to.recommend=4}	{likelihood=Detractor}	0.059	1.000	0.059	4.831	294.000

Support vector machine

Support vector machine is mathematical description of the position and orientation of the planar separation. Novel data point is to be mapped via SVM into higher-dimensional space and indicate whether the point is above or below the planar separation. For this, I created groups for age and removed the unwanted columns. Then I divided the data set into training data and testing data where train data is 2/3rd cut point of data and test data is 1/3rd of the data. I have used it to predict the likelihood of recommendation using attributes such as age, gender, type of travel, flight distance, departure delay in minutes.

Steps include:

Divide into a “training” and “test” data set:
Suggested delineation
-2/3-> Training
-1/3-> Test
Create 2/3 cut point
Create trainData (using 2/3 cut point) and testData (using rest of the data) and the previously created random indexes

Code

```
#---Support vector machine---------------------------------------------------------------------
df3 <- df1

#creating groups for age
for (i in 1:length(df3$Age))
{
  if (df3$Age[i] >=15 & df3$Age[i] <= 29)
  {
    df3$Age_group[i] <- "Age between 15 and 29"
  }
  else if (df3$Age[i] >=30 & df3$Age[i] <= 54)
  {
    df3$Age_group[i] <- "Age between 30 and 54"
  }
  else
  {
    df3$Age_group[i] <- "Age above 54"
  }
}

#removing unwanted columns
df3<- subset(df3, select = -Origin.State)
df3 <- subset(df3, select = -Destination.State)
df3 <- subset(df3, select = -Destination.City)
df3 <- subset(df3, select = -Orgin.City)
df3 <- subset(df3, select = -Day.of.Month)
df3 <- subset(df3, select = -olong)
df3 <- subset(df3, select = -olat)
df3 <- subset(df3, select = -dlong)
df3 <- subset(df3, select = -dlat)
df3 <- subset(df3, select = -Partner.Name)
df3 <- subset(df3, select = -Airline.Status)

df3$Gender <- (gsub('Male', 0, df3$Gender))
df3$Gender <- (gsub('Female', 1, df3$Gender))
df3 <- subset(df3, select = -Airline.Status)

df3$Gender <- (gsub('Male', 0, df3$Gender))
df3$Gender <- (gsub('Female', 1, df3$Gender))

#converting categorical into numeric values
df3$Type.of.Travel <- gsub('Business travel', 0, df3$Type.of.Travel)
df3$Type.of.Travel <- (gsub('Mileage tickets', 1, df3$Type.of.Travel))
df3$Type.of.Travel <- (gsub('Personal Travel', 2, df3$Type.of.Travel))

df3$Likelihood.to.recommend[is.na(df3$Likelihood.to.recommend)]=round(mean(df3$Likelihood.to.recommend, na.rm=TRUE))
df3$Age[is.na(df3$Age)]=round(mean(df3$Age, na.rm=TRUE))
df3$Flight.Distance[is.na(df3$Flight.Distance)]=round(mean(df3$Flight.Distance, na.rm=TRUE))
df3$Departure.Delay.in.Minutes[is.na(df3$Departure.Delay.in.Minutes)]=round(mean(df3$Departure.Delay.in.Minutes, na.rm=TRUE))


colnames(df3)
df3 <- df3[,c(-4,-6,-10,-11,-12,-13,-17,-21,-22)]
View(df3)

# creating random sample for training and test dataset
randIndex1 <- sample(1:dim(df3)[1])
cutPoint2_3 <- floor(2 * dim(df3)[1]/3)
trainData <- df3[randIndex1[1:cutPoint2_3],]
testData <- df3[randIndex1[(cutPoint2_3+1):dim(df3)[1]],]

dim(trainData)
#It contains 3333 rows and 23 columns
dim(testData)
#It contains 1667 rows and 23 columns
View(df3)
# running support vector machine
svmOutput <- ksvm(Likelihood.to.recommend~Gender+ Age + Type.of.Travel+Flight.Distance+Departure.Delay.in.Minutes, data = trainData, kernel="rbfdot", kpar="automatic", C=5, cross=2, prob.m
svmOutput

#performing prediction
svmPred<-predict(svmOutput, testData)
pred <- data.frame(svmPred)

comparison <- data.frame(testData$Likelihood.to.recommend,svmPred) #create confusion matrix for svmPred against testData$cut
table1<-table(comparison)
View(table1)
```

Comparison of Likelihood to recommend and Prediction

	testData.Likelihood.to.recommend	svmPred	Freq
1	2	3.30525097267905	0
2	4	3.30525097267905	1

Recommendations

- Female customers are more than male customers, but they have a lower likelihood to recommend and hence better marketing ads and services could be given to female customers.
- The age group of customers traveling is in the range of 30-50 years hence, better deals can be given to them to meet their demands.
- West Airways has a good NPS score so the customer can promote the airlines with their acquaintances so South East Airlines can do a market study of the airlines and partner more with the airlines.
- Customers with average number of flights per year are promoters so loyal customers with special benefits so that each customer is interested to fly with the airlines to increase its loyalty.
- A lot of promoter's travel business class so facilities in economy class could be improved in order to attract a greater likelihood of recommendation.
- The Southern part of USA have a number of promoters, but efforts can be taken to improve the quality of service in the other parts.
- Airlines should provide services like medical aid, wheelchairs to people over 50 as they can be detractors.

Code

```
#---Data loading and cleaning----------------------------------------------------------------------

#Installing packages
install.packages("jsonlite")
library(jsonlite)
install.packages("ggplot2")
library(ggplot2)
install.packages("dplyr")
install.packages("dplyr")
library(dplyr)
install.packages("carat")
library(carat)
install.packages("kernlab")
library(kernlab)
install.packages("maps")
library(maps)
install.packages("ggmap")
library(ggmap)
install.packages("arules")
library(arules)
install.packages("arulesViz")
library(arulesViz)
install.packages("corrplot")
library(corrplot)


#Data loading and cleaning
airData<-fromJSON("project.json")
df<-data.frame(airData)
View(df)
str(df)
summary(df)
#contains 5000 rows and 31 columns

sapply(airData,function(x)sum(is.na(x))) #check for null values
#columns with null values
#Departure.Delay.in.Minutes
#airData$Arrival.Delay.in.Minutes
#Flight.time.in.minutes

#Removing null values
airData$Arrival.Delay.in.Minutes[is.na(airData$Arrival.Delay.in.Minutes)]=round(mean(ai
rData$Arrival.Delay.in.Minutes, na.rm=TRUE))
```

```
airData$Flight.time.in.minutes[is.na(airData$Flight.time.in.minutes)]=round(mean(airDat
a$Flight.time.in.minutes, na.rm=TRUE))
airData$Departure.Delay.in.Minutes[is.na(airData$Departure.Delay.in.Minutes)]=round(
mean(airData$Departure.Delay.in.Minutes, na.rm=TRUE))
View(df)

#Adding attributes to the data
df1<-df
for (i in 1:length(df1$Arrival.Delay.in.Minutes)) #calculating arrival delay if greater than 5
minutes
{
  if (df1$Arrival.Delay.in.Minutes[i] > 5)
  {
    df1$ArrivalDelay[i] <- "Yes"
  }
  else
  {
    df1$ArrivalDelay[i] <- "No"
  }
}

#calculating if departure delay is greater than 5 minutes
for (i in 1:length(df1$Departure.Delay.in.Minutes))
{
  if (df1$Departure.Delay.in.Minutes[i] > 5)
  {
    df1$DepartDelay[i] <- "Yes"
  }
  else
  {
    df1$DepartDelay[i] <- "No"
  }
}

#Converting likelihood into categorical values
for (i in 1:length(df1$Likelihood.to.recommend))
{
  if (df1$Likelihood.to.recommend[i] < 7)
  {
    df1$likelihood[i] <- "Detractor"
  }
  else if (df$Likelihood.to.recommend[i] >= 7 & df$Likelihood.to.recommend[i] <= 8)
  {
    df1$likelihood[i] <- "Passive"
  }
  else
```

```
  {
    df1$likelihood[i] <- "Promoter"
  }
}

#calculating NPS
count <- table(df1$Partner.Name,df1$likelihood)
dim(count)
npsdata <- data.frame(Airlines = unique(rownames(count)),Detractor =
count[,1],Passive = count[,2],Promoters = count[,3])
npsdata$NPS <- ((npsdata$Promoters - npsdata$Detractor)/(npsdata$Promoters +
npsdata$Detractor + npsdata$Passive))*100


#---Exploratory data analysis----------------------------------------------------------

#Creating histograms of numeric variables
hist(df1$Age)
hist(df1$Flights.Per.Year)

#Creating tables of categorical variables
table(df1$Gender) #There are 2820 female travelers and 2180 male travelers
table(df1$Airline.Status) #There are 4 categories of Airline status
table(df1$Type.of.Travel)

#examining relation between likelihood to recommend and price sensitivity
price<-
ggplot(df1,aes(y=Likelihood.to.recommend,x=Price.Sensitivity,fill=Flight.cancelled))+ge
om_boxplot() #create a boxplot
price<-price+ ggtitle("Likelihood to recommend based on price sensitivity")
price

#examining relation between type of travel and likelihood to recomend
travel <- ggplot(df1,aes(x=Type.of.Travel, y=Likelihood.to.recommend))
travel <- travel + geom_col()
travel <- travel + theme(axis.text.x = element_text(angle = 90, hjust = 1))
travel

#create histogram to examine relation between gender and likelihood
ggplot(df1,aes(x=Gender)) +
geom_histogram(stat="count",color="black",aes(fill=likelihood))

#create maps to examine likelihood by state
us <- map_data("state")
df1$Origin.State <- tolower(df1$Origin.State)
```

```
map1<- ggplot(df1, aes(map_id = Origin.State, label = Origin.State)) #ggplot for adding the data and map_id for specifying the data for map
map1<- map1 + geom_map(map = us, aes(fill=likelihood, x=dlong, y=dlat)) #adding the map as us
map1

#Calculating NPS

count <- table(df1$Partner.Name,df1$likelihood)
dim(count)
npsdata <- data.frame(Airlines = unique(rownames(count)),Detractor = count[,1],Passive = count[,2],Promoters = count[,3])
npsdata$NPS <- ((npsdata$Promoters - npsdata$Detractor)/(npsdata$Promoters + npsdata$Detractor + npsdata$Passive))*100
ggplot(npsdata,aes(x=reorder(Airlines,NPS),y=NPS)) + geom_col(aes(fill=NPS))+ theme(axis.text.x = element_text(angle = 90, hjust = 1))

#creating promoter data
promoterdata <- subset(df1, likelihood=="Promoter")
View(promoterdata)

#create detractor data
detractordata <- subset(df1, likelihood=="Detractor")
View(detractordata)

#plot for promoter data visualizing age with Gender relation
ggplot(promoterdata,aes(x=Age)) +
geom_histogram(stat="count",color="black",aes(fill=Gender))

#plot for promoter data visualizing age with Gender relation
ggplot(detractordata,aes(x=Age)) +
geom_histogram(stat="count",color="black",aes(fill=Gender))

#---Association rule mining----------------------------------------------------------
dfnew<-df1

#Removing  columns not necessary for analysis
dfnew <- subset(dfnew, select = -Origin.State)
dfnew <- subset(dfnew, select = -Destination.State)
dfnew <- subset(dfnew, select = -Destination.City)
dfnew <- subset(dfnew, select = -Orgin.City)
dfnew <- subset(dfnew, select = -Day.of.Month)
dfnew <- subset(dfnew, select = -olong)
dfnew <- subset(dfnew, select = -olat)
dfnew <- subset(dfnew, select = -dlong)
dfnew <- subset(dfnew, select = -dlat)
```

```
dfnew <- subset(dfnew, select = -Partner.Name)
dfnew <- subset(dfnew, select = -Airline.Status)
View(dfnew)

#create groups with the columns as average, low and high
creategroups<-function(columns)
{
  parts=quantile(columns,c(0.45,0.55))
  groups<-rep("Average",length(columns))
  groups[columns<=parts[1]]<-"Low"
  groups[columns>=parts[2]]<-"High"
  return(as.factor(groups))
}

#using the function to create subgroups in the columns
dfnew$Loyalty<-creategroups(dfnew$Loyalty)
dfnew$Eating.and.Drinking.at.Airport<-
creategroups(dfnew$Eating.and.Drinking.at.Airport)
dfnew$Flight.Distance<-creategroups(dfnew$Flight.Distance)
dfnew$Total.Freq.Flyer.Accts<-creategroups(dfnew$Total.Freq.Flyer.Accts)
dfnew$Price.Sensitivity<-creategroups(dfnew$Price.Sensitivity)
dfnew$Flights.Per.Year<-creategroups(dfnew$Flights.Per.Year)

#converting dataframe into factors
dfnew<-mutate_all(dfnew,as.factor)
#converting to transaction matrix
dfnew_x<-as(dfnew,"transactions")
summary(dfnew_x)
#inspect the matrix
inspect(dfnew_x)

itemFrequency(dfnew_x)

#apriori for association rule where the lhs is default and rhs is likehood to recommend
for promoters
ruleset<-apriori(dfnew_x,parameter=list(support=0.04,maxlen=20,
confidence=0.5),appearance = list(default="lhs",rhs=("likelihood=Promoter")))

#apriori for association rule where the lhs is default and rhs is likehood to recommend
for detractors
rule<-apriori(dfnew_x,parameter=list(support=0.04,maxlen=20,
confidence=0.5),appearance = list(default="lhs",rhs=("likelihood=Detractor")))

#inspect promoters in rhs
inspectDT(ruleset)
```

```
#inspect detractors in rhs
inspectDT(rule)

#---Linear modeling-----------------------------------------------------------------
df4<-df1

#converting categorical into numeric values
df4$Type.of.Travel <- gsub('Business travel', 0, df4$Type.of.Travel)
df4$Type.of.Travel <- (gsub('Mileage tickets', 1, df4$Type.of.Travel))
df4$Type.of.Travel <- (gsub('Personal Travel', 2, df4$Type.of.Travel))


#converting categorical into numeric values
df4$Gender <- (gsub('Male', 0, df4$Gender))
df4$Gender <- (gsub('Female', 1, df4n$Gender))


#regression model for predicting likelihood for recomendation by age
model1<-lm(formula =Likelihood.to.recommend~Age, data=df4)
summary(model1)

#regression model for predicting likelihood for recomendation by age, flights per year
model2<-lm(formula =Likelihood.to.recommend~Age+Flights.Per.Year, data=df4)
summary(model2)

#regression model for predicting likelihood for recomendation by age, flights per year,
price sensitivity
model3<-lm(formula
=Likelihood.to.recommend~Age+Flights.Per.Year+Gender+Price.Sensitivity, data=df4)
summary(model3)

#regression model for predicting likelihood for recomendation by age, flights per year,
price sensitivity, type of travel, departure delay
model4<-lm(formula
=Likelihood.to.recommend~Age+Flights.Per.Year+Gender+Price.Sensitivity+Type.of.Tr
avel+Flight.Distance+Departure.Delay.in.Minutes, data=df4)
summary(model4)


#---Support vector machine----------------------------------------------------------
df3 <- df1

#creating groups for age
for (i in 1:length(df3$Age))
{
  if (df3$Age[i] >=15 & df3$Age[i] <= 29)
```

```
  {
    df3$Age_group[i] <- "Age between 15 and 29"
  }
  else if (df3$Age[i] >=30 & df3$Age[i] <= 54)
  {
    df3$Age_group[i] <- "Age between 30 and 54"
  }
  else
  {
    df3$Age_group[i] <- "Age above 54"
  }
}

#removing unwanted columns
df3<- subset(df3, select = -Origin.State)
df3 <- subset(df3, select = -Destination.State)
df3 <- subset(df3, select = -Destination.City)
df3 <- subset(df3, select = -Orgin.City)
df3 <- subset(df3, select = -Day.of.Month)
df3 <- subset(df3, select = -olong)
df3 <- subset(df3, select = -olat)
df3 <- subset(df3, select = -dlong)
df3 <- subset(df3, select = -dlat)
df3 <- subset(df3, select = -Partner.Name)
df3 <- subset(df3, select = -Airline.Status)

df3$Gender <- (gsub('Male', 0, df3$Gender))
df3$Gender <- (gsub('Female', 1, df3$Gender))

#converting categorical into numeric values
df3$Type.of.Travel <- gsub('Business travel', 0, df3$Type.of.Travel)
df3$Type.of.Travel <- (gsub('Mileage tickets', 1, df3$Type.of.Travel))
df3$Type.of.Travel <- (gsub('Personal Travel', 2, df3$Type.of.Travel))

df3$Likelihood.to.recommend[is.na(df3$Likelihood.to.recommend)]=round(mean(df3$Li
kelihood.to.recommend, na.rm=TRUE))
df3$Age[is.na(df3$Age)]=round(mean(df3$Age, na.rm=TRUE))
df3$Flight.Distance[is.na(df3$Flight.Distance)]=round(mean(df3$Flight.Distance,
na.rm=TRUE))
df3$Departure.Delay.in.Minutes[is.na(df3$Departure.Delay.in.Minutes)]=round(mean(df
3$Departure.Delay.in.Minutes, na.rm=TRUE))


colnames(df3)
df3 <- df3[,c(-4,-6,-10,-11,-12,-13,-17,-21,-22)]
View(df3)
```

```
# creating random sample for training and test dataset
randIndex1 <- sample(1:dim(df3)[1])
cutPoint2_3 <- floor(2 * dim(df3)[1]/3)
trainData <- df3[randIndex1[1:cutPoint2_3],]
testData <- df3[randIndex1[(cutPoint2_3+1):dim(df3)[1]],]

dim(trainData)
#It contains 3333 rows and 23 columns
dim(testData)
#It contains 1667 rows and 23 columns
View(df3)
# running support vector machine
svmOutput <- ksvm(Likelihood.to.recommend~Gender+ Age +
Type.of.Travel+Flight.Distance+Departure.Delay.in.Minutes, data = trainData,
kernel="rbfdot", kpar="automatic", C=5, cross=2, prob.model=TRUE)
svmOutput

#performing prediction
svmPred<-predict(svmOutput, testData)
pred <- data.frame(svmPred)

comparison <- data.frame(testData$Likelihood.to.recommend,svmPred) #comparing the
predicted values with actual values
table1<-table(comparison)
table1<-table(comparison)
View(table1)
```