;





Customer Churn Analysis

for Southeast Airlines

Group 5

Chenghao Wu | Zhengyang Xu | Irene Lu | Paridhi Rajyaguru | Yeswanth Reddy Velapalem

IST 687 - Introduction to Data Science

Content

* **Overview**
* **Data Prepossessing**
* **Demographics**
* **Further Cleaning**
* **Models**
* **Interpretation/ Insight**
* **Appendix and Reference**

# **Overview**

For our project, we did a comprehensive study of the customer experience for Southeast Airlines and its regional partners. **Southeast wants to minimise its “Customer Churn”** i.e, prevent its customer from leaving their services.

Traditionally, Southeast has focused on an elaborate Loyalty Program to entice its customers. This entailed that customers who flew frequently with Southeast would be rewarded with “frequent flyer miles” that could be redeemed for discounts or free travel. However this is no evidence suggesting that this approach is reaping valuable results.   
  
Moreover, *customer churn is actually a lagging indicator*, meaning the loss has already occurred. The aim is really to avoid the loss from occurring, by recognizing some metrics that may help identify dissatisfied customers, and in turn prevent them from leaving by taking appropriate actions to improve service.

## Our Goal

In our analysis we focus on the **“Net Promoter Score”** for Southeast and its regional partner airlines. For our dataset, the customers were asked to respond on a scale of 1-10, to a simple question:

“How likely is it that you will recommend our airline to a friend or colleague?”

If respondents score less than 7, they’re detractors. If they scored above an 8, they’re promoters. In the middle range (a score of 7 or 8), they’re “passive”. We then calculate the NPS as the difference between the percentage of promoters and the percentage of detractors in the group.  
  
NPS gives us a fairly good idea about our customer pool. In fact, it has been shown that it is highly sensitive to customer churn, more so than loyalty or customer satisfaction. A positive NPS indicates that we have greater number of promoters and vice-versa. It is good to retain promoters, and important to convert detractors. Detractors may be problematic, as they may actively tell their social connections not to use our service in addition to stop themselves.   
  
**Therefore, our goal for this project is to focus on the “Detractor” type of customers. We want to seek airlines with a poor NPS (the ones with majority of detractors), and find some valuable insights into why their customers are dissatisfied.**

## Available Data

Our dataset is the product of recently completed customer surveys of Southeast Airlines and its partners. The data captures:

* **Flight related data**, such as: origin, destination, flight time, distance, airline status, class of travel, flight delay, cancellation etc. and,
* **Customer related data**: age, gender, price sensitivity, shopping and eating habits at the airport and so on.

This data really encompasses everything within a customer’s experience, from their purchase of ticket to landing. We have a total of 10282 observations with 32 columns, wherein each column represents a specific customer or flight related attribute.

Through our analysis, we attempt to spot the most significant variables which weigh heavily on customer satisfaction and NPS. As we proceed, we use only select such attributes for a more in depth study.

We first calculated NPS for each individual airline and identified those with the poorest scores, shown below:

|  |  |
| --- | --- |
| **Airline** | **NPS** |
| Oursin Airlines | 0.31 |
| Cheapseats Airlines | 2.02 |
| Northwest Business Airlines | -57.90 |
| FlyFast Airways Inc. | -19.26 |
| Going North Airlines Inc. | -11.11 |

Next, we created a brand new dataset with information for only the above mentioned airlines. As mentioned earlier we want to focus on airlines with more detractors and understand why that is the case. Through our analysis we attempt to gain valuable insights and provide suggestions on how these airlines could improve their performance and retain customers.

# 

# Getting Data and Preprocessing

Dataset was provided by the professor in classroom.The original data had 10282 observations and 32 columns. We calculated the NPS for each airline and decided to consider five airlines with the lowest NPS scores since it is our main objective to improve the NPS. The data of the selected five airlines alone constituted 60% of the data.

Before using the data for visualization and modelling, we had to clean the data.There were so many missing values in columns like Departure.Delay.in.Minutes, Arrival.Delay.in.Minutes,

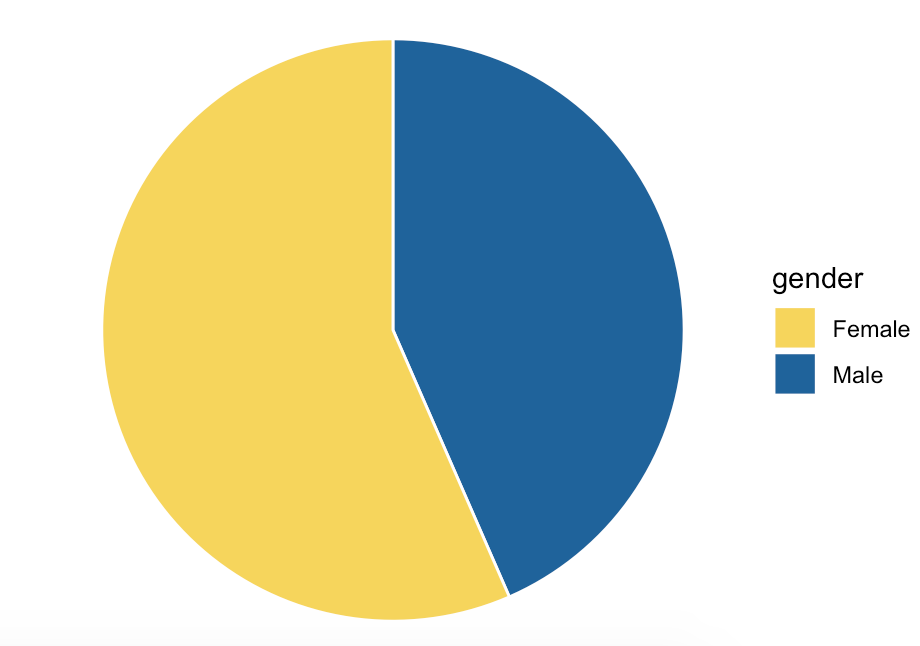
Flight.Time.in.Minutes etc.,. In Data Cleaning we replaced the missing values with the mean of the entire column. We found some undesired columns in the data but we have included all the columns because each model might use different columns. We decided to use only relevant columns in the models and remove the undesired models when needed. Also we removed all the cases where flight is cancelled, which contributed to most of the missing values and also to make sure there is no bias in the feedback of customers.

Then we did some data munging to transform and convert the raw data into usable format.

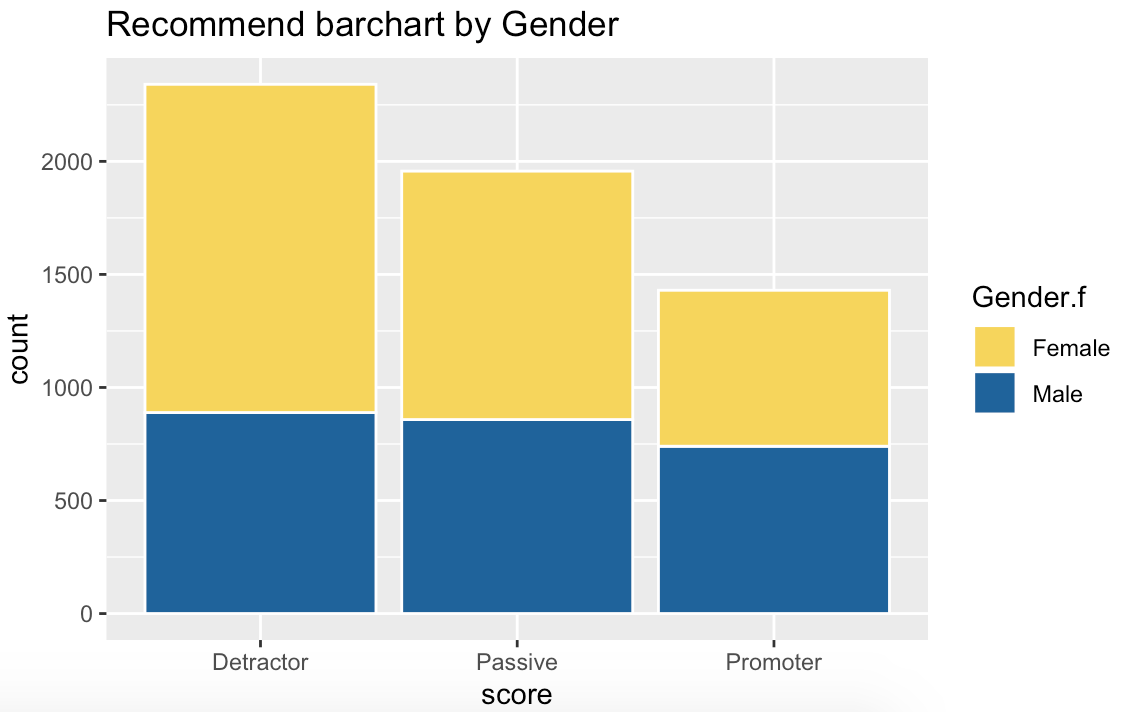
# Demographics

Gender

Pie chart of gender



This pie chart shows the gender ratio in our dataset. We have 56.54% female and 43.46% male customers according to our dataset. More specifically, there are 3175 females and 2440 males.

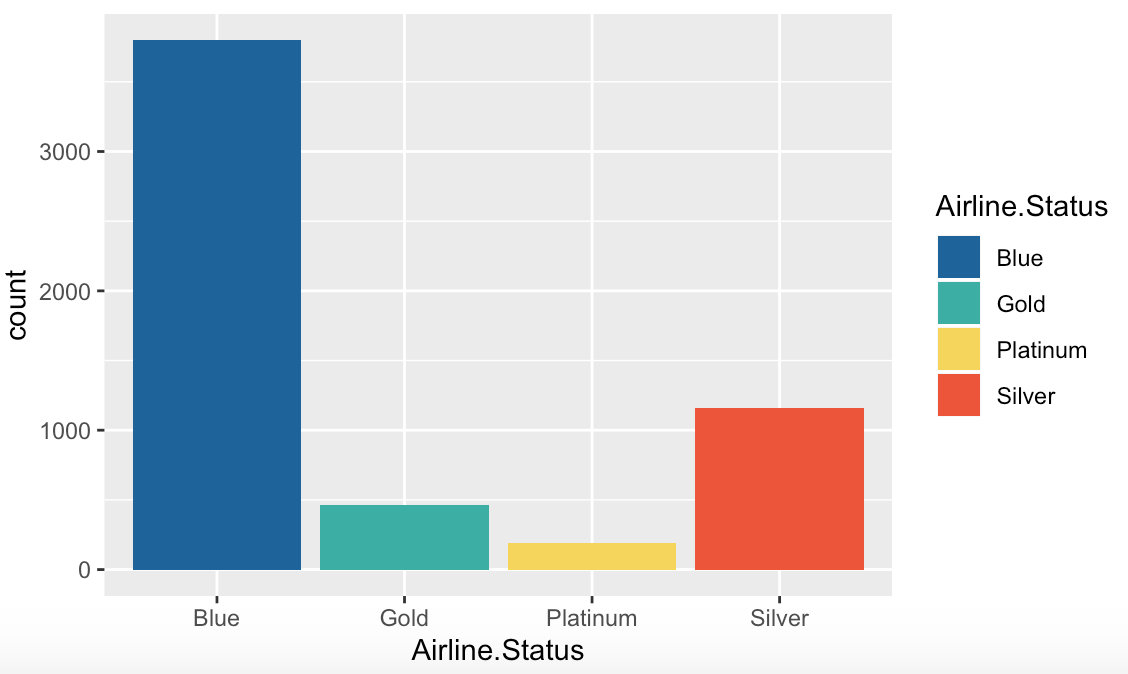
Barchart of Gender vs Type of Customer

This barchart shows type of customer vs gender. Where promoters seems evenly distributed among females and males, detractors and passive customers are taken mainly by females.

Airline Flyer Status

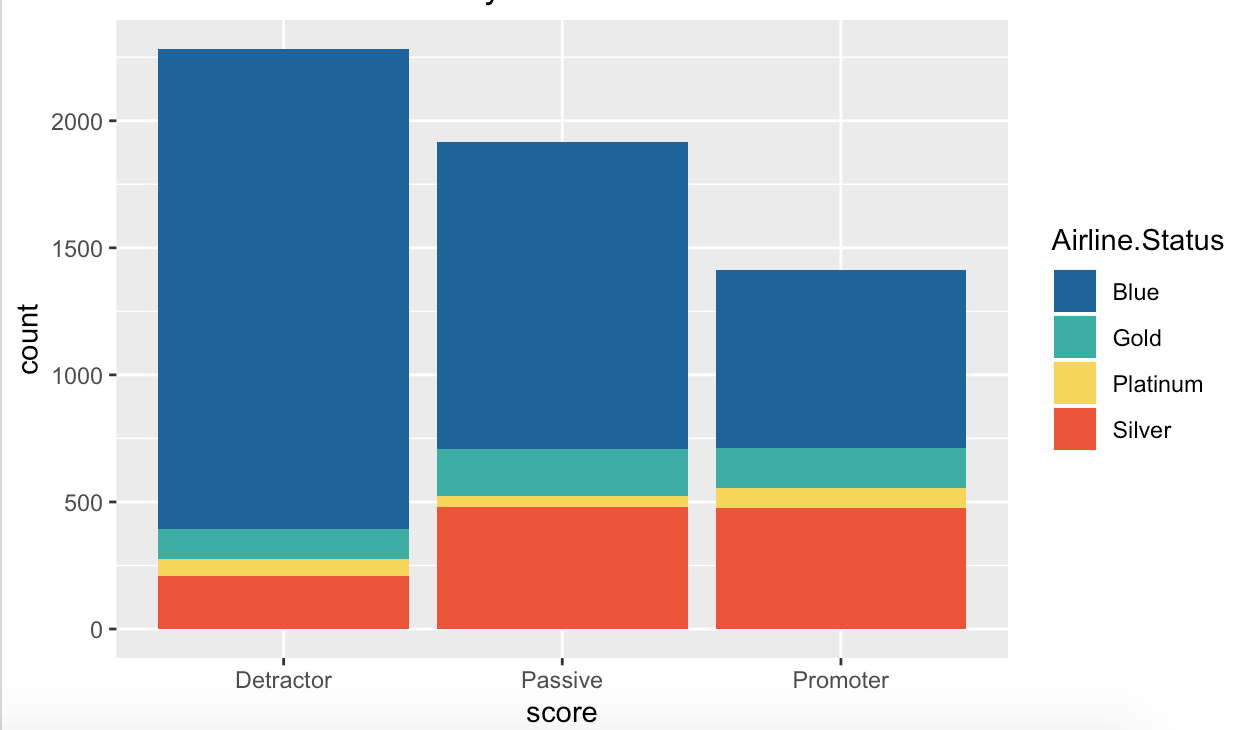
We have 4 levels of airline flyer status from low to high: Blue, Silver, Gold and Platinum.

Barchart of Airline Status



This bar chart shows the number of customers in each airline status. We have 3798 Blue, 1162 Silver, 464 Gold, and 191 Platinum. As flyer status becomes higher, there are fewer people.

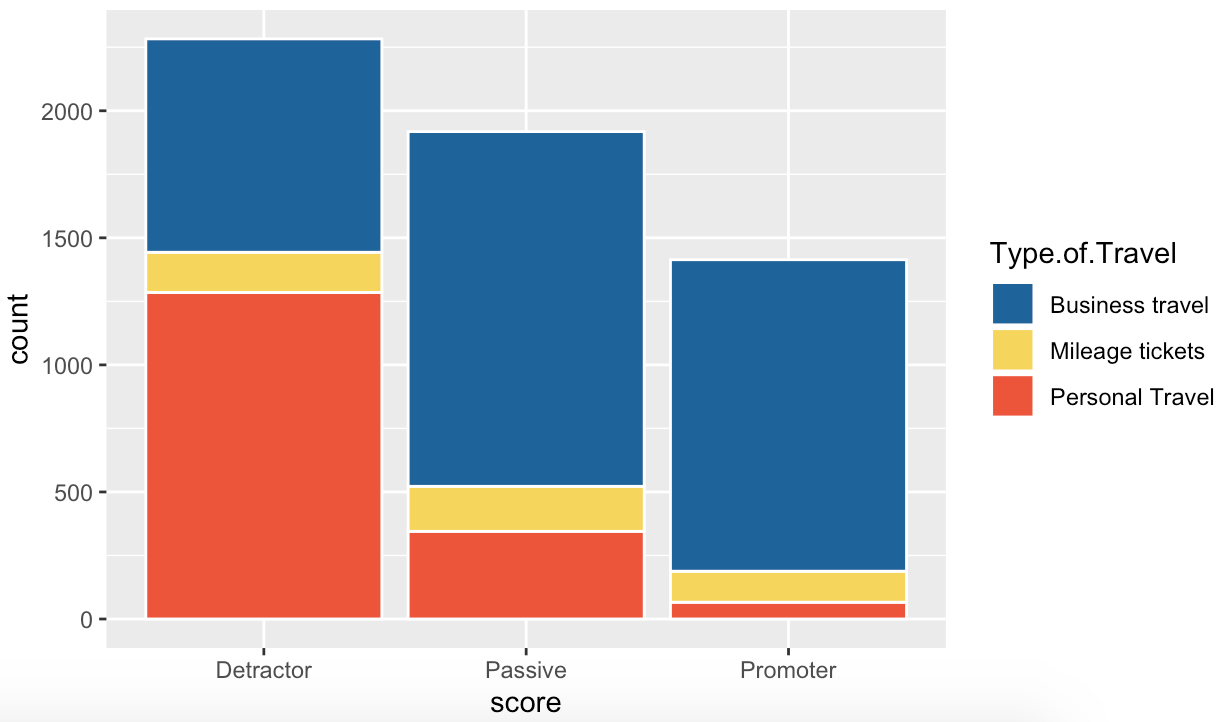
Airline status vs type of customers



This bar graph shows type of customers (detractors, passive, promoter) vs airline flyer status. Although we have more blue status customers in our dataset, we can see that blue status tend to be detractors than passive or promoter. For silver and gold customers, they are more towards the positive side.

Type of Travel

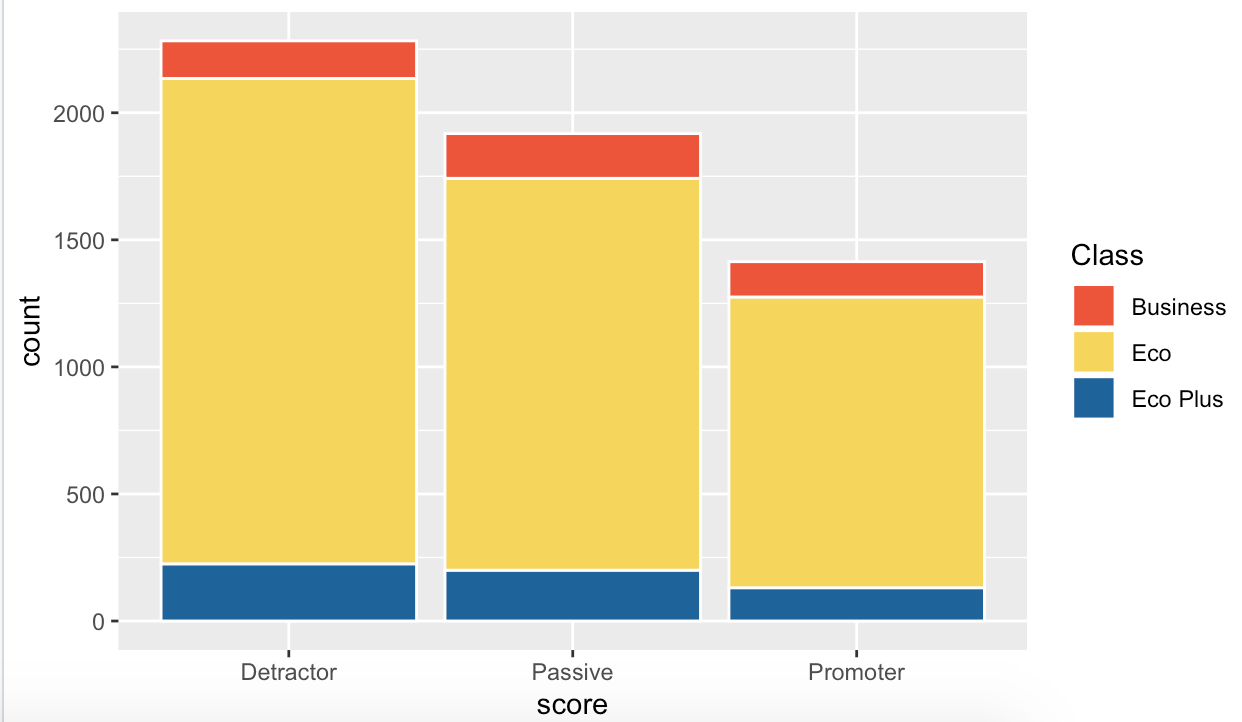
Type of Customer vs Type of Travel



Based on our dataset, our customers has three types of travel: Business travel, mileage tickets, and personal travel. The majority is business travel. Second largest group is personal travel. From this bar chart we can see Business travellers tend to give higher ratings, while personal travellers are mostly detractors.

Class

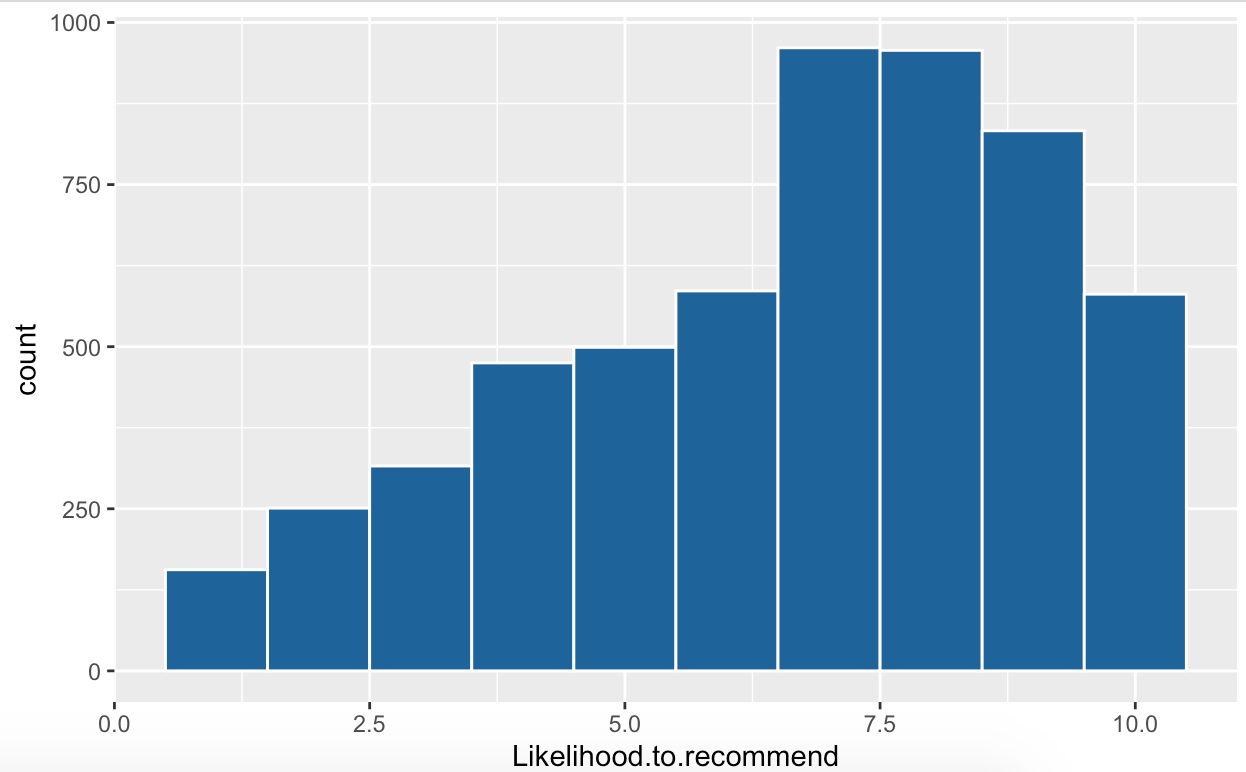
Bar chart of Type of Customer vs Class



According to our dataset, our customers are mainly from Eco class. And Eco class customers are more likely to be detractors/passive than promoter.

Likelihood to recommend

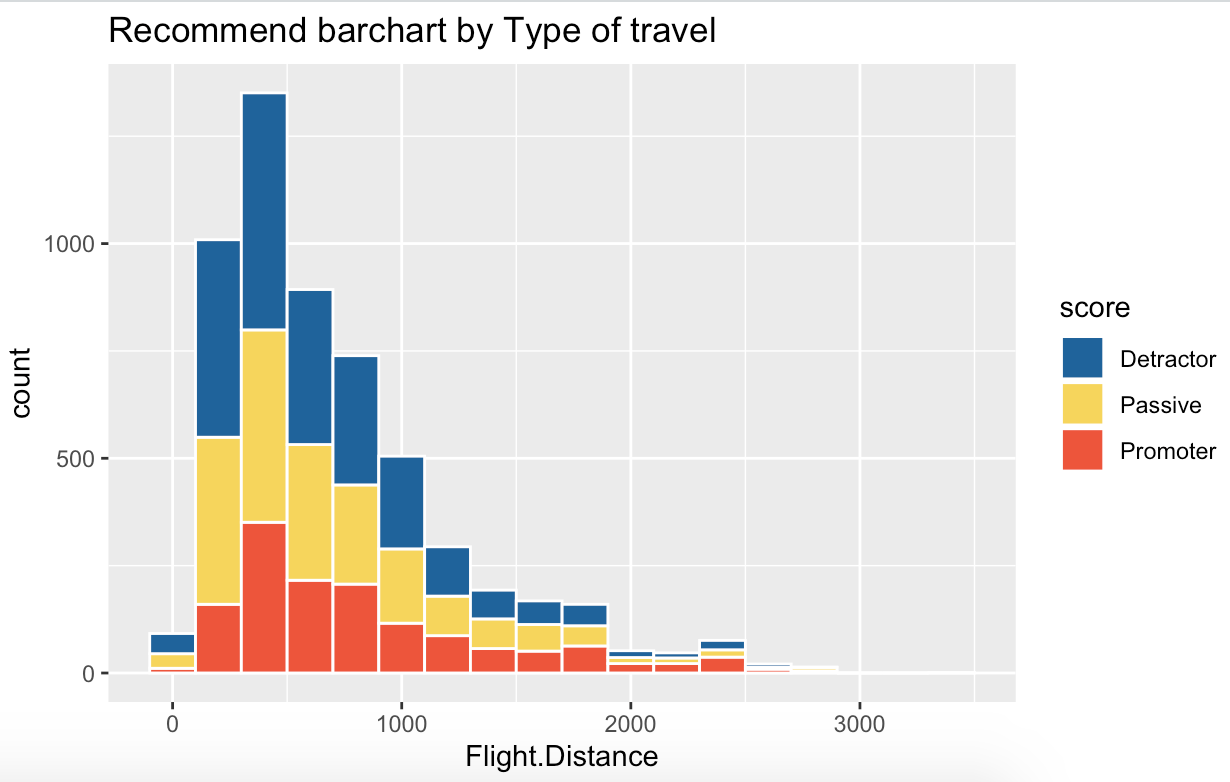
Histogram of Likelihood to recommend



This is a histogram of likelihood to recommend. Customers will give a score from 1 to 10 *,* which shows how likely the customer is to recommend the airline to their friends (10 is very likely, and 1 is not very likely). According to our histogram, we can see that the distribution is left-skewed. Since we classify score 0-6 as detractors, there are more detractors.

Flight Distance

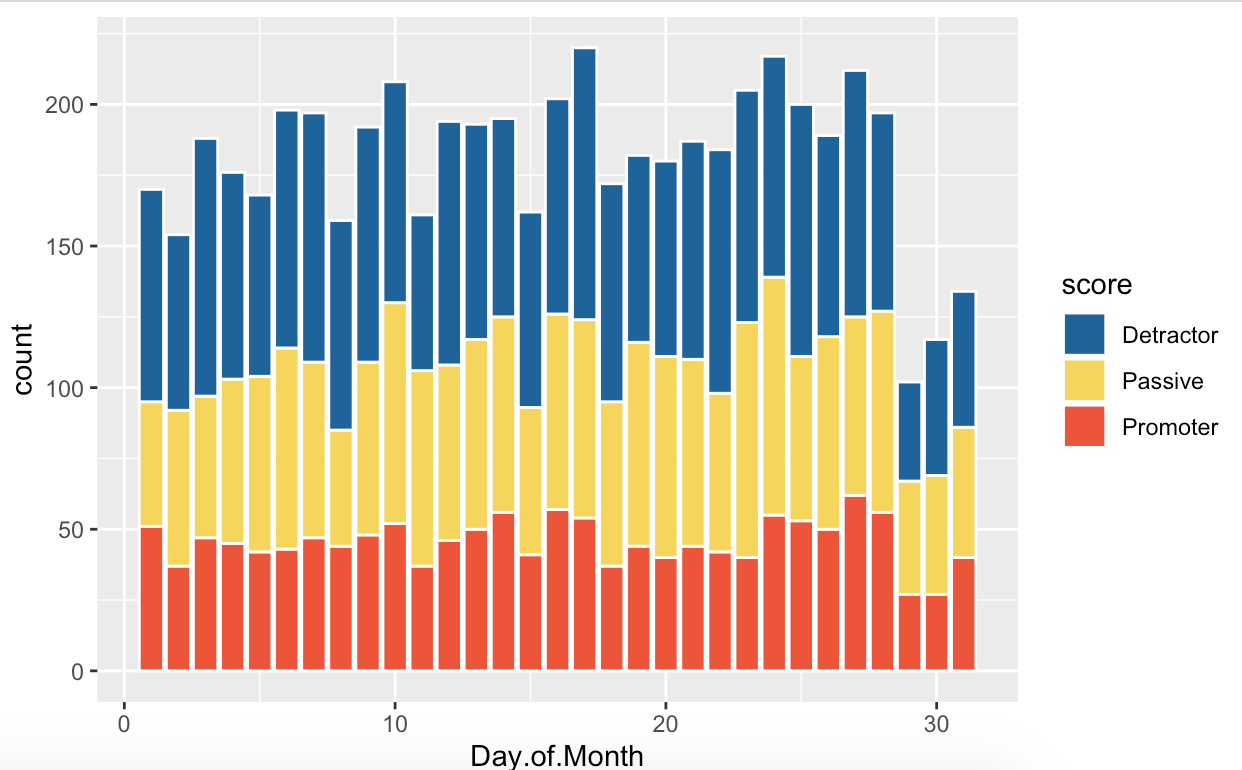
Histogram of Flight Distance with Type of Customer



This histogram shows that promoter falls between flight distance of 0 to 1000. However, there are way more detractors between this range as well.

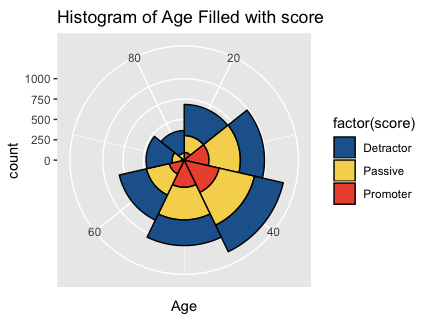
Day of Month

Barchart of day of month vs type of customer



There is no strong relationship between day of month and type of customers.

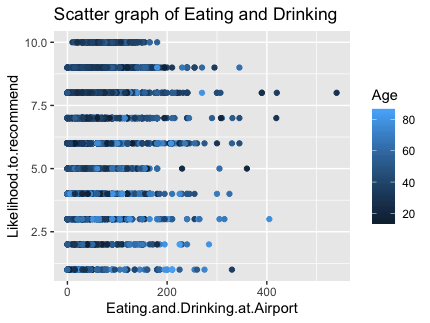
## Age



From the Graph we can clearly see that the most customers are from around 25-55 years old, the percentage of detractor is pretty high at the range 50-80 and 15-25 years old.

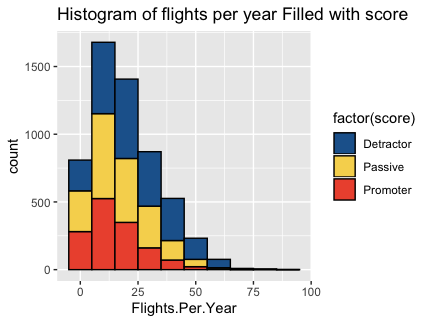
## 

## Eating and Drinking



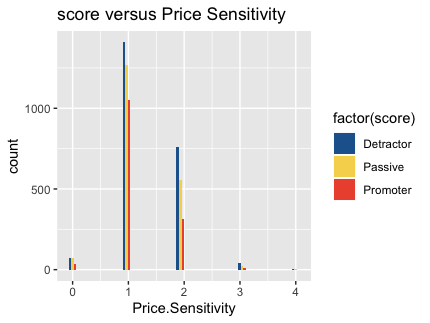
We also use scatter graph to demonstrate the relationship between the money they spend on eating and drinking at the airport and likelihood to recommend and age. The customers who tend to give a 10 score all spend less than 200 dollars. But we cannot draw a conclusion that the less they spend, the more saficified they are. And customers who are younger than around 30 years old spend not much.

## Flight Per year



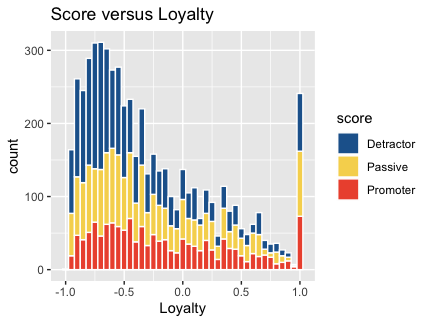
According to the percentage of detractor is increasing(or the promoter is decreasing ) as their fights per years increase, we can say that the more fights they have per year, the less satisfied they are.

## Price Sensitivity



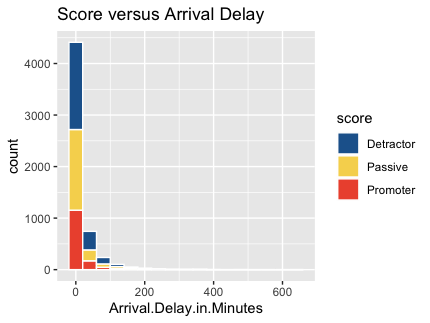
The bar chart shows that when price sensitivity is higher, the satisfaction will be lower. Besides, we can see that most customers are not really price sensitive since most of them are at level 1 and 2.

## Loyalty



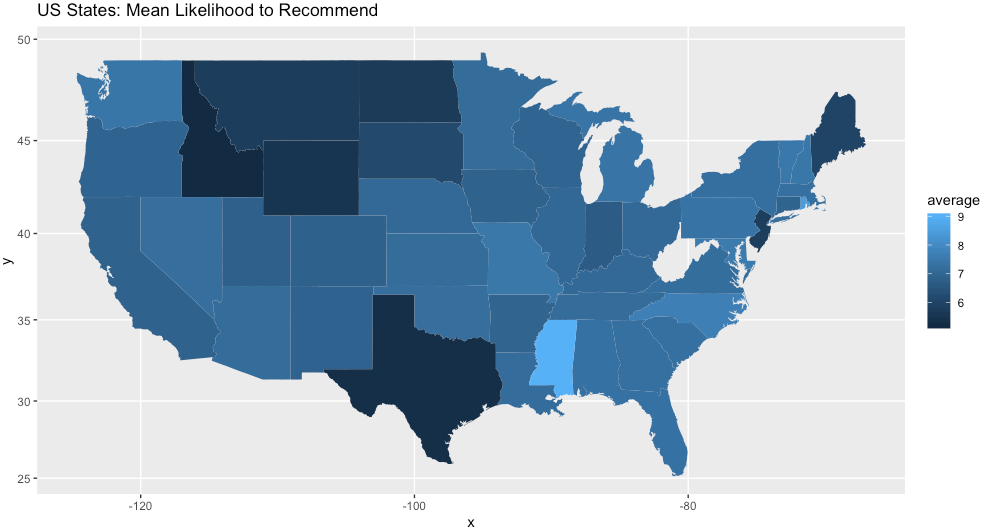
From the histogram we can see that if the customer has more loyalty, they are more likely to be the promoter. Contrastly, if he/she is not loyal, he/she tends to be a detractor.

## Arrival Delay



From the graph we can see that most of the delay are less than 50 minutes, and the longer the fight delay, the more detractors there are.

## Likelihood to Recommend

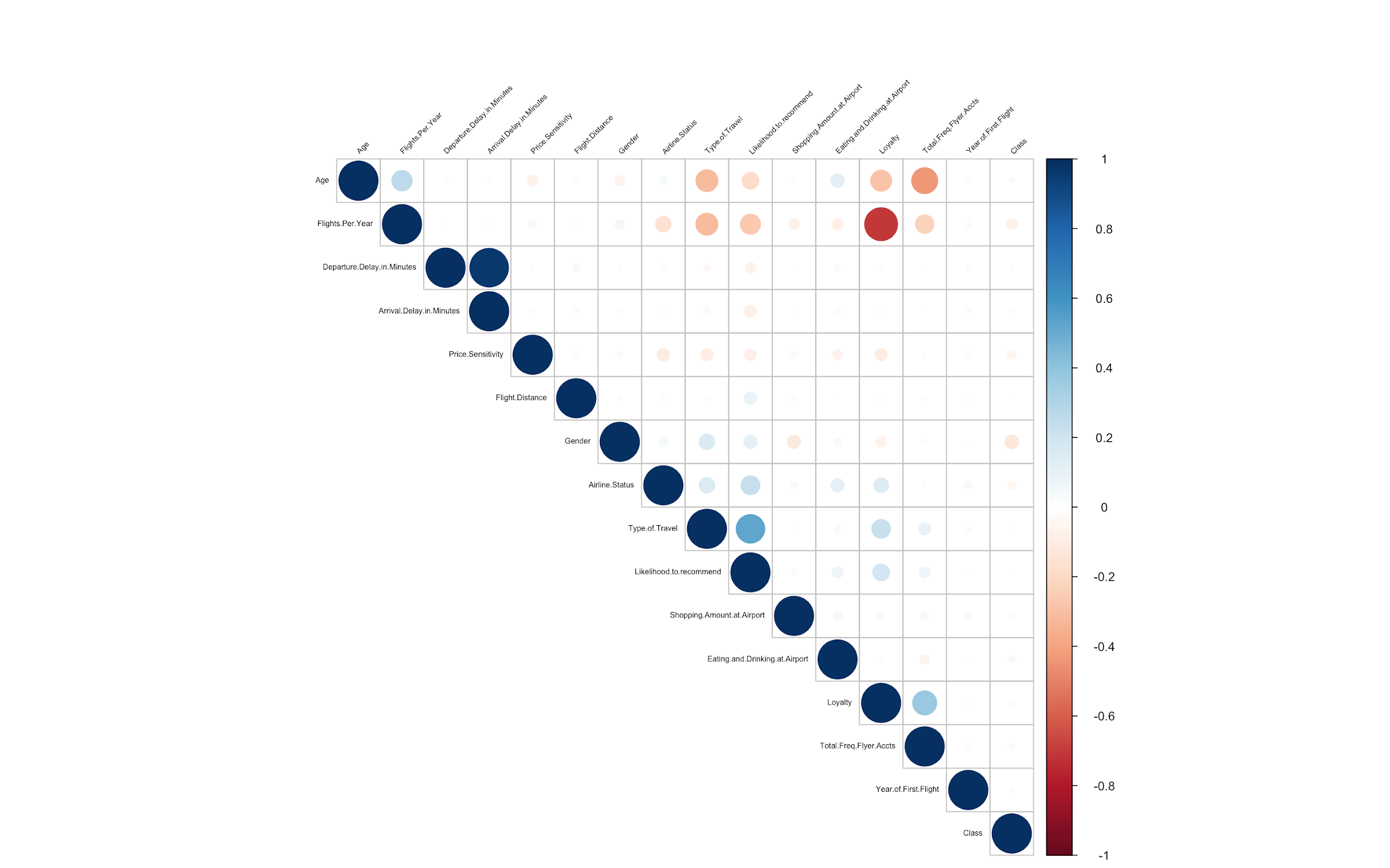


We can see the average “Likelihood to Recommend” scores by US states in the above map. This gives us an idea of customer satisfaction by regions serviced by Southeast Airlines.

The darker regions represent areas with a poor score, and consequently regions wherein the airline service is not upto standard. We observe that the states of Texas, North and South Dakota, Montana and Wyoming have the lowest scores. From our data, we know that Cheapseat Airlines and OurSin Airlines account for a majority of flights in these states.

Correlation:

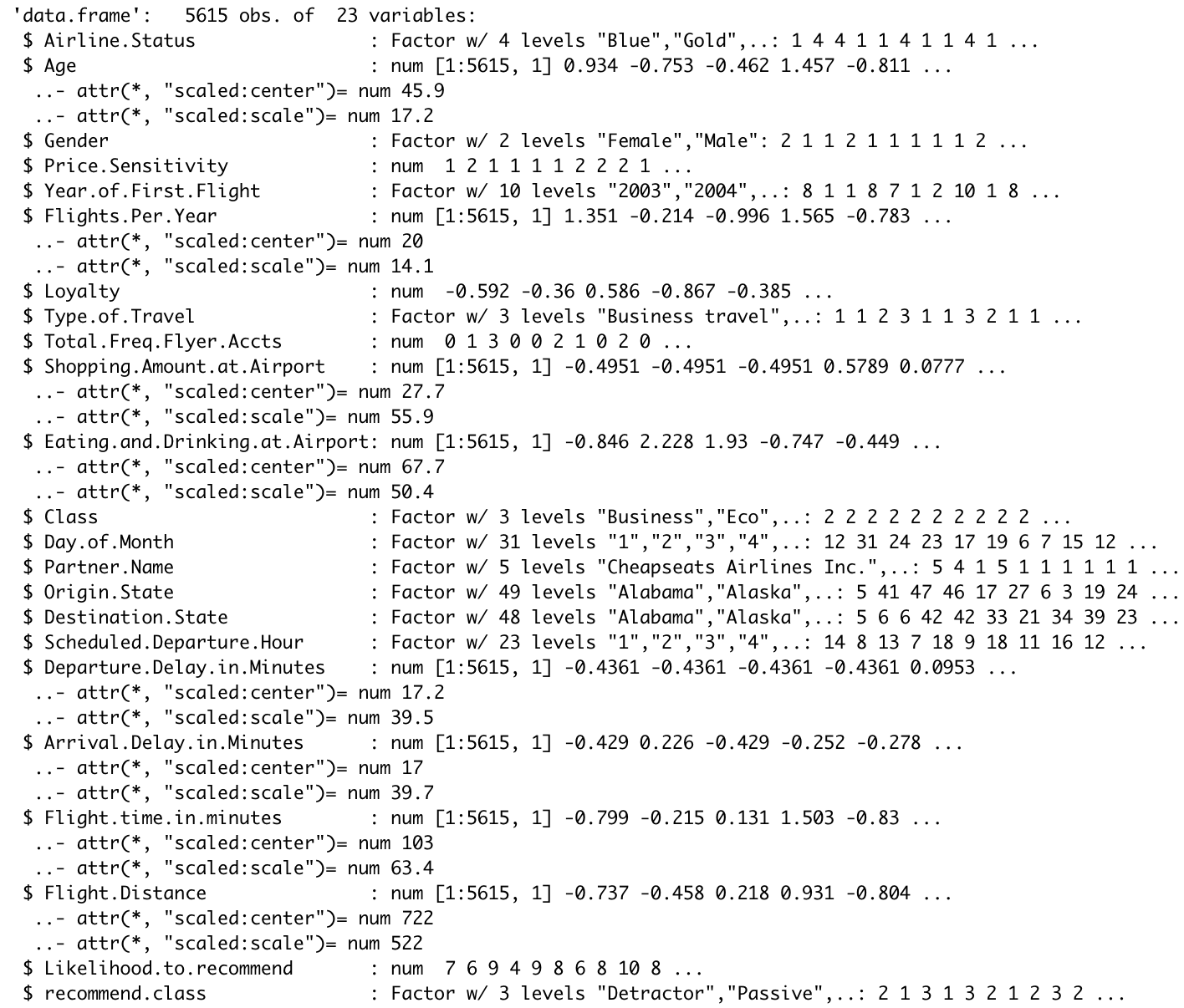
Correlation plot was generated to get an initial reference of variables that might be affecting the likelihood to recommend value



# Further Cleaning

## Feature Scaling

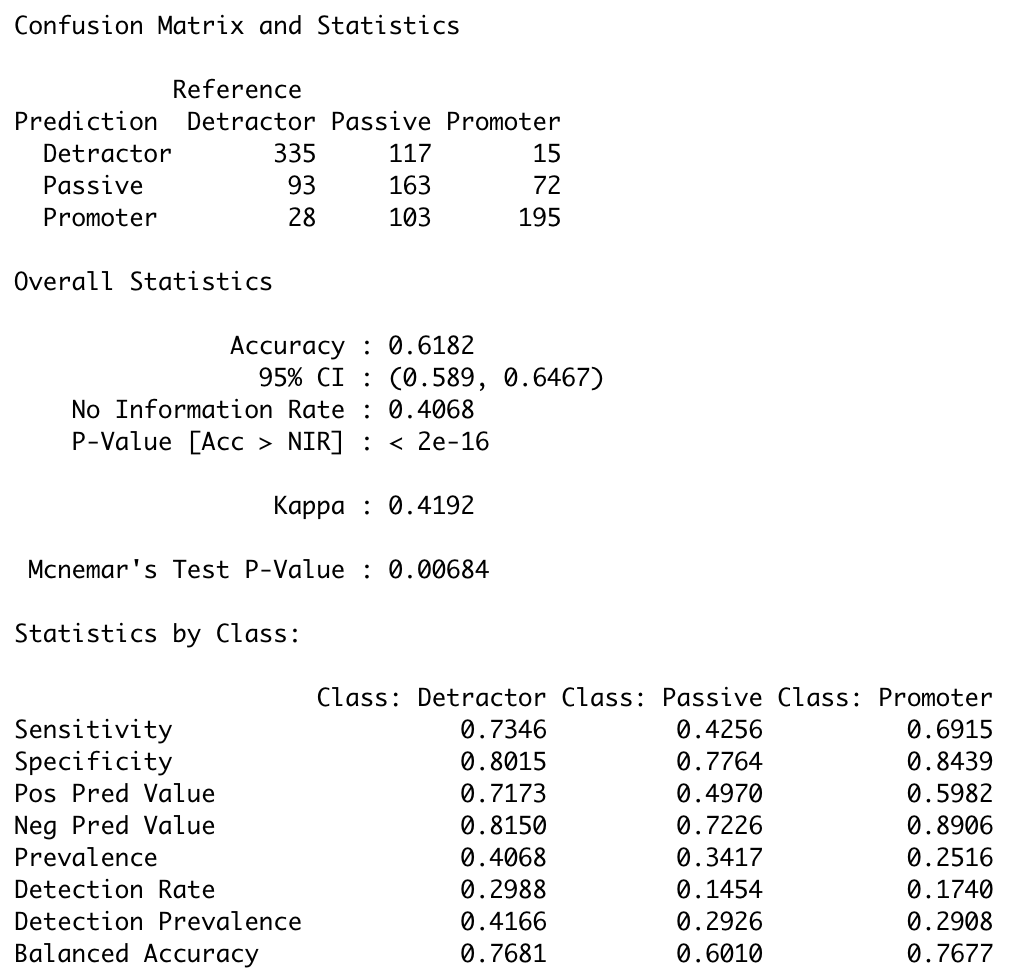
Before we perform models, we also need to do further cleaning data like feature scaling. Feature scaling is used to normalize the range of independent variables of features of data. Since the range of values of raw data varies widely, such as range of shopping amount at the airport is huge different with loyalty variable. In some algorithms model, functions will not work properly without normalization. Below is the description of scaled data.



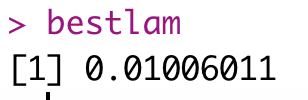
# Models

## Logistic Regression

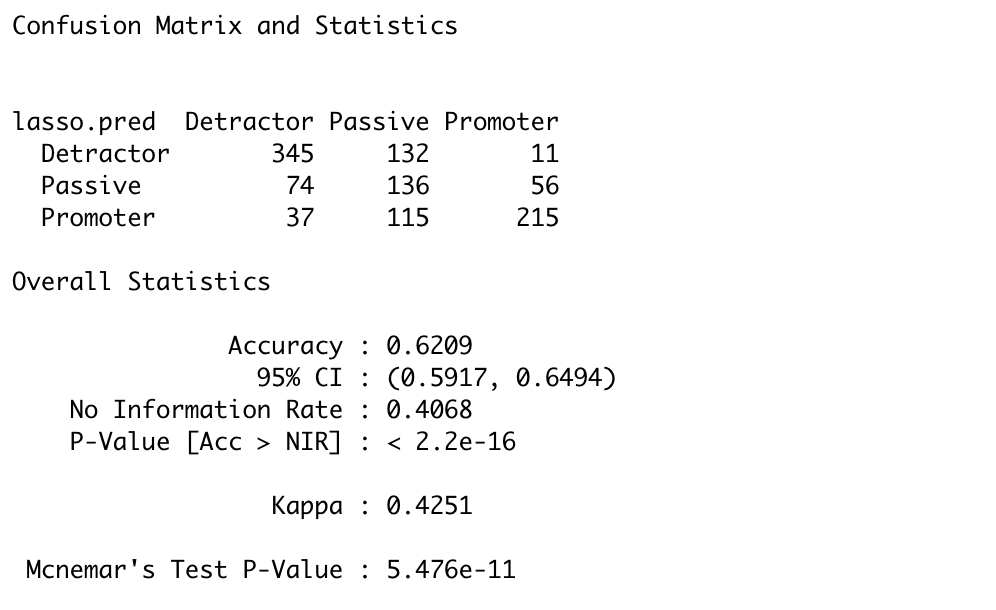
By grouping likelihood to recommend into three groups, one of the best interpretable models for classification is logistic regression model. It allows us to use linear relationship between predictors and classification target. We first split our data set into training set and test set by 0.8 ratio. We use the model training from training set to predict test set recommend class, and compare with our real test set recommend class. After constructing logistic regression, we are not getting pretty good accuracy rate. By confusion matrix below, the accuracy is about 61.8 percent.



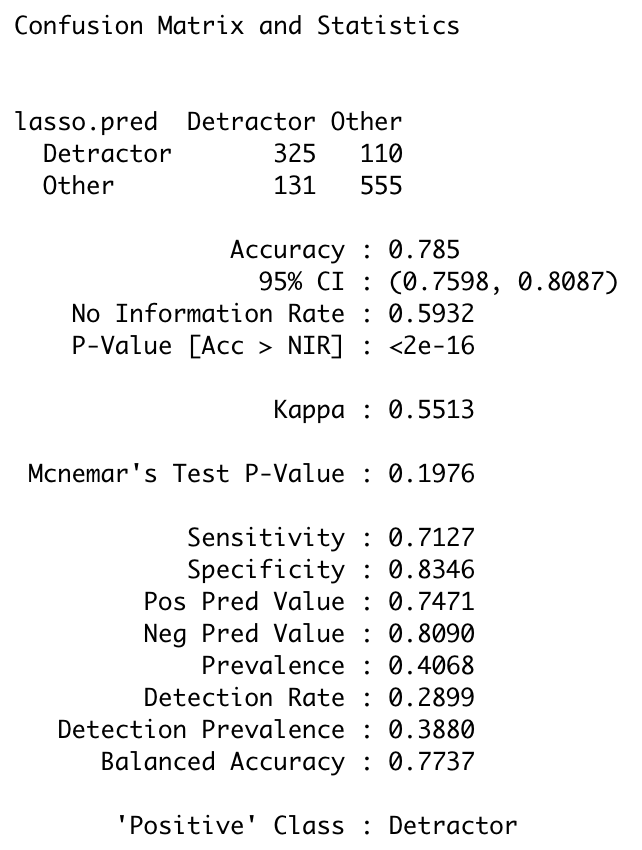
In order to improve our accuracy, we are trying to use some regularization method to improve our logistic regression model. By using LASSO method, we are choosing the lowest lambda 0.01006011 for LASSO.



After running LASSO, we have improved our logistic regression model accuracy from 61.8 percent to 62.09 percent. The accuracy increase is still not so significant.

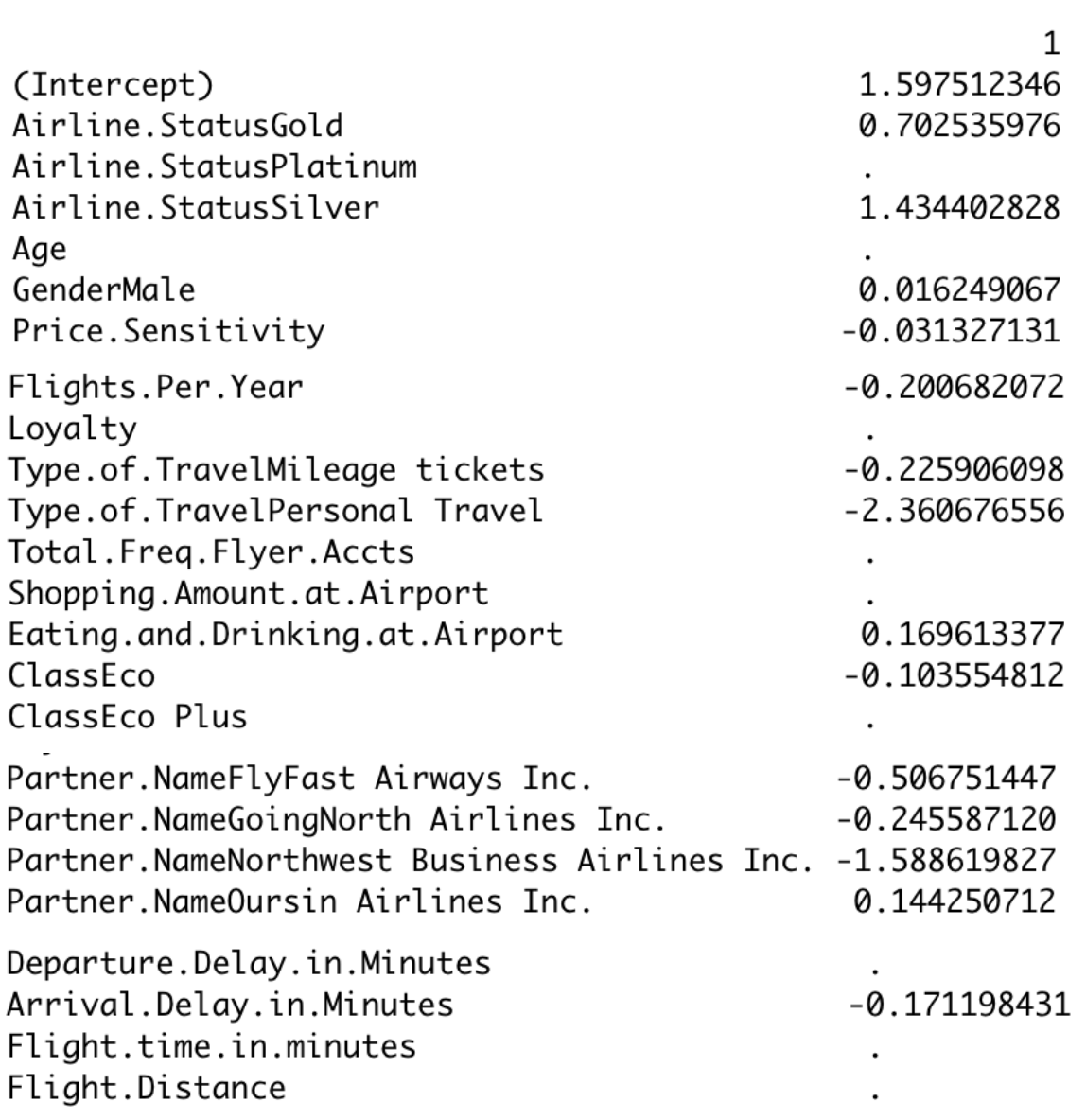


Since our project goal is to analyze how to improve customer’s likelihood of recommend, especially to change people are detractors into higher score class. That means that we are mainly focus on whether a person is detractor or not. It implies that grouping people by detractors or not might be more efficient and accuracy probability increase.



So, we are trying to apply logistic regression for two classes, which are Detractor and Other. After running Logistic Regression and LASSO, we gets 78.5 percent accuracy by model. Now, the accuracy is pretty decent, and it is useful for predicting and useful for analyzing how to improve our recommended grades.

Some Coefficients of this Logistic Regression are present below,



By regulazad coefficient, we can see that some of the variables actually are not important, which help us filter variables.

By the coefficient table, we can see Airline status Gold, Airline Status Silver, Gender Male, Price Sensitivity, Flights per year, Type of travel mileage tickets, Type of Travel Personal Travel, Eating and Drinking at Airports, Class Eco, Arrival Delay in minutes are variables that have significant influence on our likelihood recommend classification.

## Association Rules:

Next, We decided to use Association Rule Mining as our second method for finding any relations between the factors in the data set. This modeling will help us to identify strong rules discovered in the data set using some measures of interestingness. It also generates new rules as it analyses more data.

We removed the irrelevant columns and created a function to convert all the columns into categories. Then all the columns are converted into factors before running the model.

First we run the model on the entire data and took the top 5 rules based on lift

For all data:

lhs rhs support confidence lift count

[1] {Price.Sensitivity=Low,

Type.of.Travel=Personal Travel,

Total.Freq.Flyer.Accts=Low} => {Age=High} 0.1024043 0.8008357 3.048605 575

[2] {Airline.Status=Blue,

Age=High,

Total.Freq.Flyer.Accts=Low,

Likelihood.to.recommend=Detractor} => {Type.of.Travel=Personal Travel} 0.1006233 0.8224163 2.722799 565

[3] {Airline.Status=Blue,

Age=High,

Likelihood.to.recommend=Detractor} => {Type.of.Travel=Personal Travel} 0.1079252 0.8222524 2.722257 606

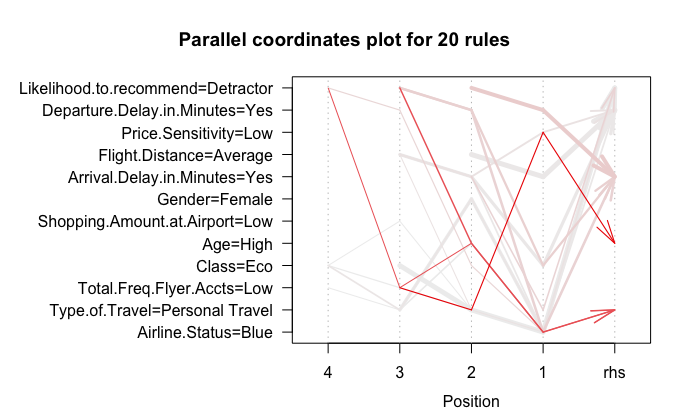
[4] {Departure.Delay.in.Minutes=Yes,

Likelihood.to.recommend=Detractor} => {Arrival.Delay.in.Minutes=Yes} 0.1490650 0.8584615 2.205060 837

[5] {Class=Eco,

Departure.Delay.in.Minutes=Yes,

Likelihood.to.recommend=Detractor} => {Arrival.Delay.in.Minutes=Yes} 0.1214604 0.8514357 2.187013 682



Since it was a bit difficult to get any interpretations/insights from this plot, we fixed the RHS as Promoter, Passive and Detractor and run 3 different models.

For Promoter:

TOP 5 RULES based on LIFT

lhs rhs support confidence lift count

[1] {Price.Sensitivity=Low,

Type.of.Travel=Business travel,

Eating.and.Drinking.at.Airport=High,

Flight.Distance=High} => {Likelihood.to.recommend=Promoter} 0.02030276 0.5968586 2.370128 114

[2] {Type.of.Travel=Business travel,

Eating.and.Drinking.at.Airport=Average,

Arrival.Delay.in.Minutes=No,

Flight.Distance=High} => {Likelihood.to.recommend=Promoter} 0.02154942 0.5873786 2.332483 121

[3] {Age=Average,

Price.Sensitivity=Low,

Type.of.Travel=Business travel,

Eating.and.Drinking.at.Airport=High,

Departure.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Promoter} 0.02030276 0.5786802 2.297942 114

[4] {Age=Average,

Price.Sensitivity=Low,

Type.of.Travel=Business travel,

Arrival.Delay.in.Minutes=No,

Flight.Distance=High} => {Likelihood.to.recommend=Promoter} 0.02172752 0.5781991 2.296031 122

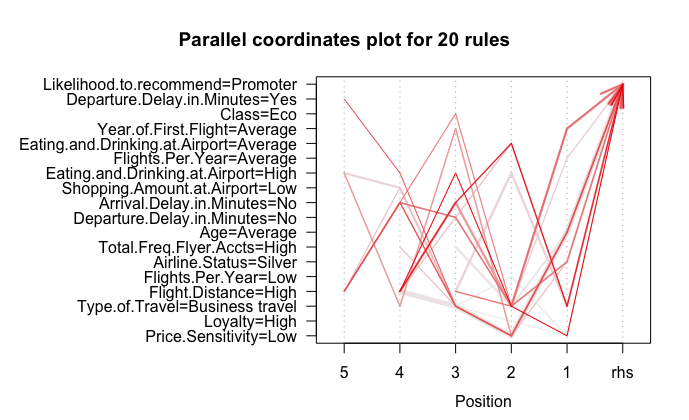
[5] {Year.of.First.Flight=Average,

Type.of.Travel=Business travel,

Departure.Delay.in.Minutes=No,

Arrival.Delay.in.Minutes=No,

Flight.Distance=High} => {Likelihood.to.recommend=Promoter} 0.02154942 0.5734597 2.277211 121



The plot includes the top 20 rules based on highest lift values. We can see that customers whose ticket price is low, spending an average amount of money to eat and drink at the airport, taking long distance business travel are more satisfied.

For Passive:

lhs rhs support confidence lift count

[1] {Age=Average,

Price.Sensitivity=High,

Type.of.Travel=Business travel} => {Likelihood.to.recommend=Passive} 0.04220837 0.4817073 1.410212 237

[2] {Type.of.Travel=Business travel,

Departure.Delay.in.Minutes=No,

Arrival.Delay.in.Minutes=No,

Flight.Distance=Low} => {Likelihood.to.recommend=Passive} 0.04683882 0.4781818 1.399891 263

[3] {Type.of.Travel=Business travel,

Shopping.Amount.at.Airport=Low,

Flight.Distance=Low} => {Likelihood.to.recommend=Passive} 0.04167409 0.4756098 1.392361 234

[4] {Gender=Female,

Loyalty=Average,

Type.of.Travel=Business travel,

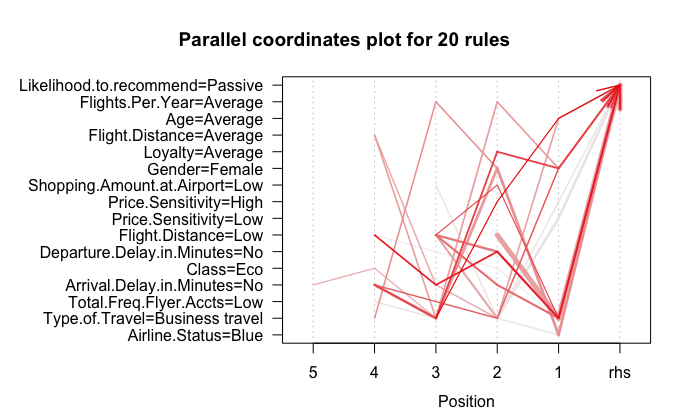
Arrival.Delay.in.Minutes=No} => {Likelihood.to.recommend=Passive} 0.04790739 0.4752650 1.391352 269

[5] {Gender=Female,

Type.of.Travel=Business travel,

Total.Freq.Flyer.Accts=Low,

Arrival.Delay.in.Minutes=No} => {Likelihood.to.recommend=Passive} 0.04327694 0.4736842 1.386724 243



**for Detractor:**

lhs rhs support confidence lift count

[1] {Airline.Status=Blue,

Age=High,

Type.of.Travel=Personal Travel,

Arrival.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Detractor} 0.05325022 0.9966667 2.451285 299

[2] {Airline.Status=Blue,

Age=High,

Type.of.Travel=Personal Travel,

Total.Freq.Flyer.Accts=Low,

Arrival.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Detractor} 0.05022262 0.9964664 2.450792 282

[3] {Airline.Status=Blue,

Age=High,

Type.of.Travel=Personal Travel,

Departure.Delay.in.Minutes=Yes,

Arrival.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Detractor} 0.04238646 0.9958159 2.449192 238

[4] {Airline.Status=Blue,

Age=High,

Type.of.Travel=Personal Travel,

Class=Eco,

Arrival.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Detractor} 0.04238646 0.9958159 2.449192 238

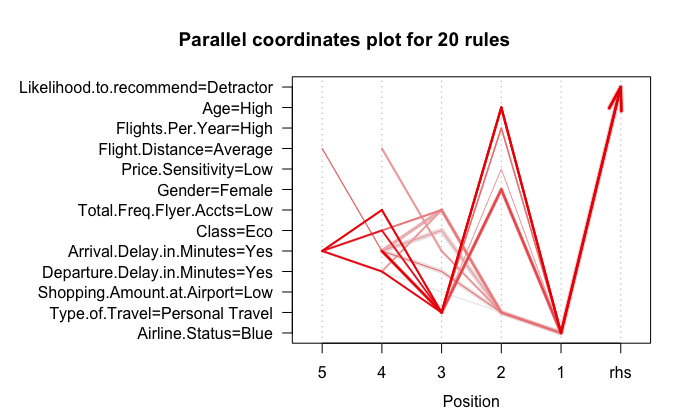
[5] {Airline.Status=Blue,

Gender=Female,

Type.of.Travel=Personal Travel,

Departure.Delay.in.Minutes=Yes,

Arrival.Delay.in.Minutes=Yes} => {Likelihood.to.recommend=Detractor} 0.05093500 0.9896194 2.433952 286



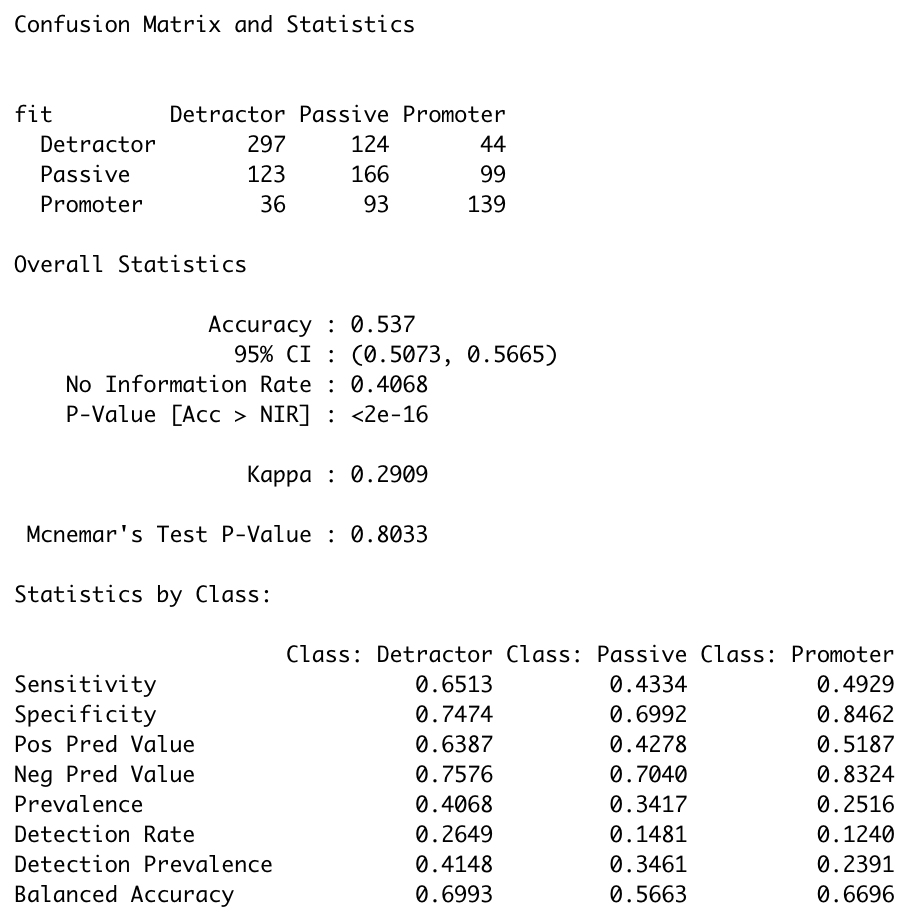
The plot includes the top 20 rules based on highest lift values. We can see that customers who are elder, travelling by blue status airlines for personal travel and has arrival delay tend to be not satisfied and likely to affect the nps.

## 

## 

## KNN Classification Model

We are using a KNN classification model for three recommend class to see if there is a difference in accuracy rate. By applied KNN model, our accuracy only 53.7 percent, which is lower than three class logistic regression. And KNN Model is hard to interpret and analyze how to improve our recommend grade. To sum up, it is better to use Logistic model than Knn model.

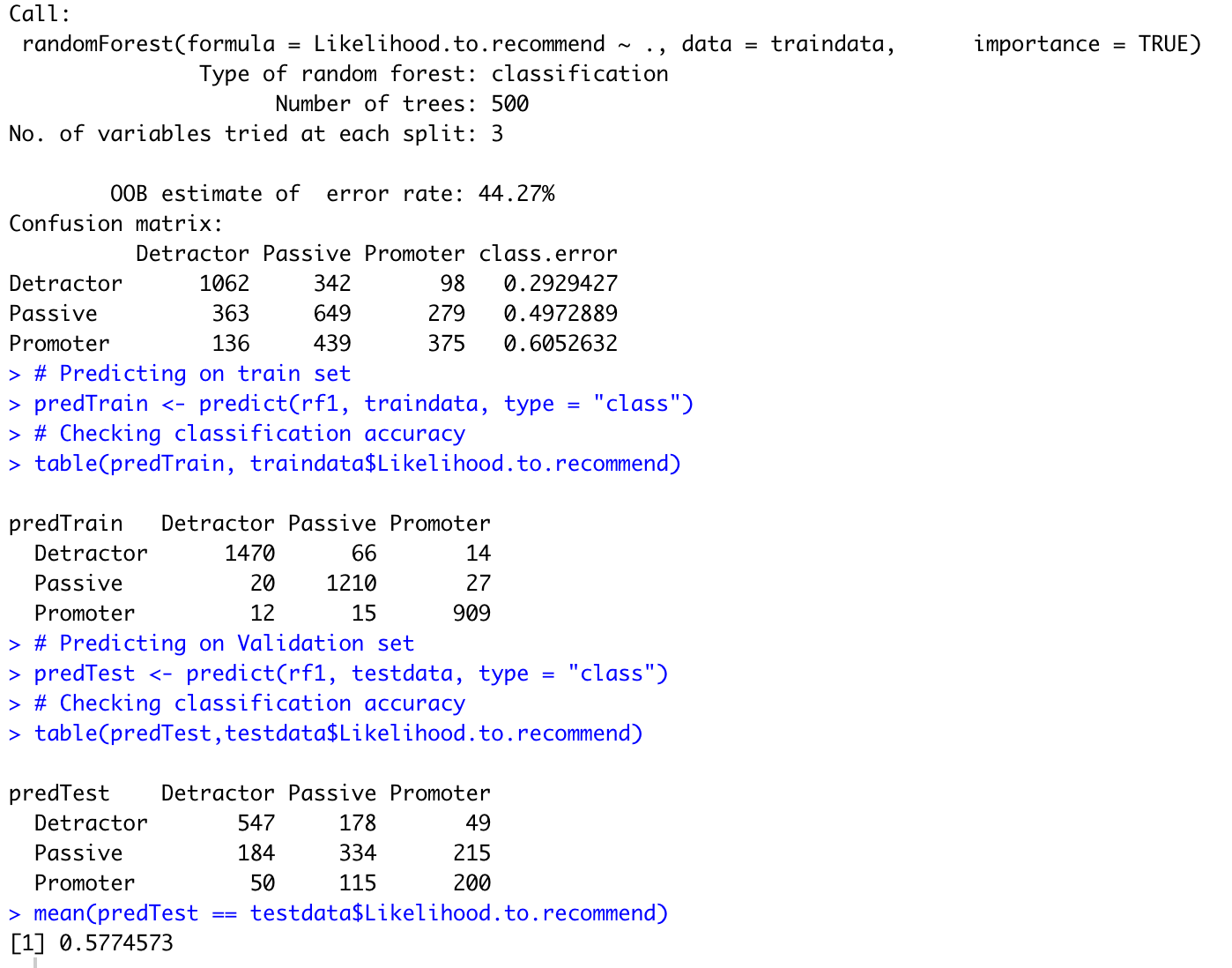


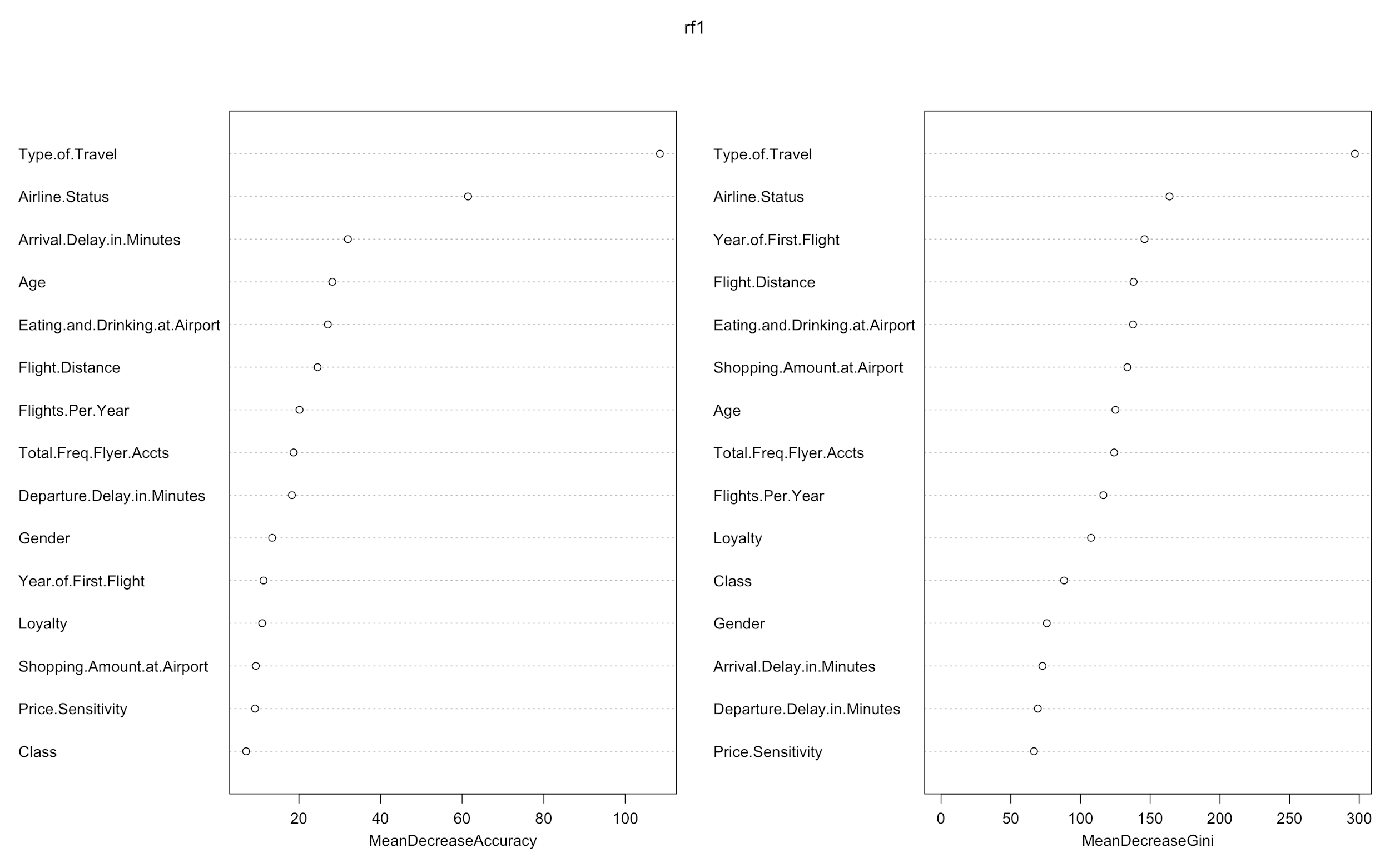
RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. Here, we have used the random forest model to build trees which can validate our results which were found out using other models such as the Linear Model or the Apriori algorithm.

For Random Forest, We performed the first model with default parameters and for second model we wrote a function to change parameters and run the model.

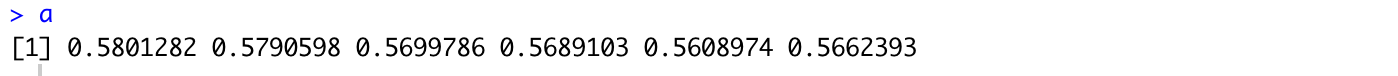
With Default parameters:

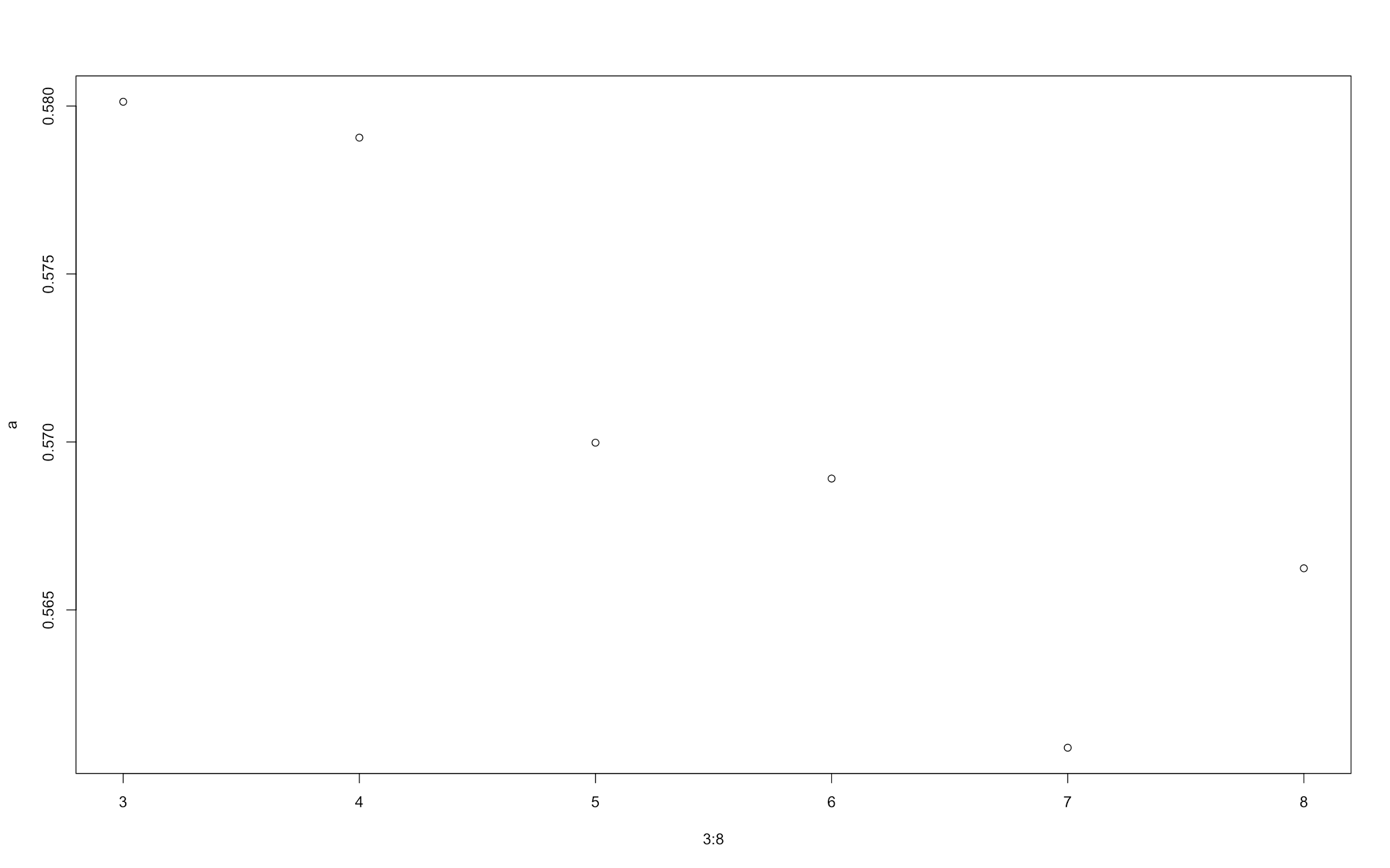




With varying mtry values:

Here the parameter mtry’s values is replaced with numbers from 3 to 8. We obtained the following accuracy values





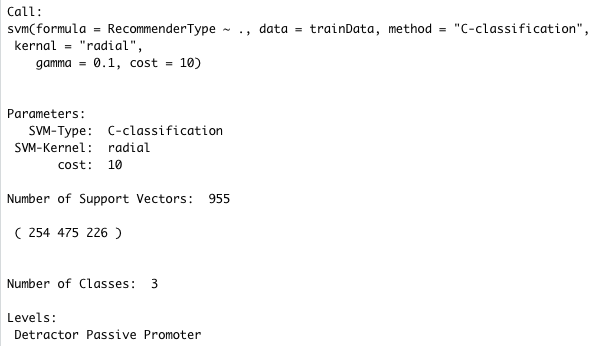
## Support Vector Machine (SVM)

An SVM performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. For our model, we have used the *svm()* function in the *e1071* package that implements the SVM supervised learning algorithm.

We use an SVM “Hyperplane Classifier” as we have two possible classes into which we group our customers. This class is calculated based on if they are default customers or not. An optimal hyperplane is the one with the largest margin between classes. For our SVM, we divide the data into training and test datasets. Two-thirds of the data is used to train the model, and the remaining one-third is used to test the model.

Furthermore, to boost the model’s performance, only the most significant variables are used in our analysis.

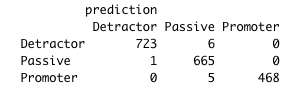
Our model:



As seen above, we use *type=”C-Classification”* for our classification problem. A *“radial” kernel* is chosen as we have multiple classes (more than two). We tune the operation of *svm()* with two additional arguments: *gamma* and *cost*, where *gamma* is the argument for use by the kernel function, and *cost* allows us to specify the cost of a violation to the margin. The “support vectors” are the data points closest to the separating hyperplane. They support the largest margin hyperplane in that if these points moved slightly, the hyperplane would also move.

We test the above model on our test dataset and create a confusion matrix. This matrix allows us to view the model’s performance in terms of how many predictions did it make correctly.

Confusion Matrix:



Our model produces remarkable results, making only 12 incorrect predictions over the entire test data. Finally, we calculate its accuracy as:

(total number of correct predictions / total number of observations)

**Accuracy = 0.993576**

# Interpretation/ Insights

1. Personal and Mileage ticket customers are always less likely to recommend the airlines. Airline should provide benefits to them to bridge the gap between the services offered compared to business tickets.
2. Age is negatively impacting NPS. Making the boarding process faster and providing baggage assistance for elderly people, Adding wheelchairs and personnel to assist them might get a better feedback.
3. Also customers whose flights have arrival delays tend to give a negative rating. So airlines must be able to run the flights on time to decrease the negative impact on nps.
4. High Price Sensitivity is also a factor for Low Nps score. Providing discount coupons or offers on ticket price gives a way to boost nps.
5. Customers travelling Long distance are contributing for better nps score. Airlines where experience of long distance travel is mediocre should introduce new services to improve the experience of long distance travel. Providing good inflight entertainment and services should do the trick to boost nps.
6. While the visualization of gender shows that we have more female customers than male customers, our model indicates that female customers tend to be detractors. It is necessary to increase female customers’ satisfaction. We can improve this by giving priorities/services to female customers. For example, we can give priorities on selecting seats for female customers first. Also, we can provide more facilities including rest lounge and lactation room, etc,.
7. From Modelling, we found that silver status customers have highest contribution in giving good feedback, we can relax the requirement of customer level up, so that it could let more people become at least silver customer, it eventually boost the likelihood to recommend.
8. People who take flights more per year, will tend to lead recommend score lower. For this point, airlines should give some rewards to people whose number of flights exceed 10, 20 per year.
9. Since the average price sensitivity GoingNorth Airlines is the highest among five airlines, Going North Airline should focus on price discount provides to customers.

# 

# 

# 

# Appendix

Getting Data

cat('\014')

install.packages("corrplot")

library(jsonlite)

library(ggplot2)

library(arules)

library(arulesViz)

library(corrplot)

library(tidyverse)

library(PerformanceAnalytics)

library(caret)

library(questionr)

library(nnet)

library(glmnet)

library(class)

data<-fromJSON("fall2019-survey-M04.json")

View(data)

str(data)

summary(data)

table(data$Partner.Name)

#no. of rows and columns

ncol(data)

nrow(data)

ndata<-data %>% filter(Partner.Name =="Oursin Airlines Inc." | Partner.Name=="Cheapseats Airlines Inc." | Partner.Name=="Northwest Business Airlines Inc." | Partner.Name=="FlyFast Airways Inc." | Partner.Name=="GoingNorth Airlines Inc.")

View(ndata)

str(ndata)

#no. of rows and columns

ncol(ndata)

nrow(ndata)

## PREPROCESSING OF DATA

#converting integer columns to numeric

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

ndata[6:8]<-lapply(ndata[6:8],as.numeric)

ndata[11:13]<-lapply(ndata[11:13],as.numeric)

ndata$Day.of.Month<-as.numeric(ndata$Day.of.Month)

ndata[21:23]<-lapply(ndata[21:23],as.numeric)

ndata[25:27]<-lapply(ndata[25:27],as.numeric)

str(ndata)

table(ndata$Flight.cancelled,ndata$Likelihood.to.recommend)

#taking data where flight is not cancelled

fdata<-ndata[ndata$Flight.cancelled=="No",]

View(fdata)

#no.of rows in fdata

nrow(fdata)

str(fdata)

summary(fdata)

sapply(fdata,function(x)sum(is.na(x)))

#checking for any incomplete values

sum(!complete.cases(fdata))

## Data Exploration

library(ggplot2)

library(dplyr)

# Create early visualization to explore the dataset

# Count the number of female and male

female.count <- ndata %>% filter(Gender=="Female") %>% nrow()

male.count <- ndata %>% filter(Gender=="Male") %>% nrow()

# Create dataframe for gender

data <- data.frame(gender=c("Female", "Male"),value=c(female.count, male.count))

# Piechart for gender

ggplot(data, aes(x="", y=value, fill=gender)) +

geom\_bar(stat="identity", width=1, color="white") +

coord\_polar("y", start=0) +

theme\_void() +

scale\_fill\_manual(values = c("Female" = "#F6D55C",

"Male" = "#20639B"))

# bar chart: gender vs type of customer

ggplot(ndata, aes(score)) +

geom\_bar(aes(fill = Gender), color="white") +

ggtitle("Recommend barchart by Gender") +

scale\_fill\_manual(values = c("Female" = "#F6D55C",

"Male" = "#20639B"))

# bar chart: Airline status vs type of customer

ggplot(ndata, aes(score)) +

geom\_bar(aes(fill = Airline.Status)) +

ggtitle("Recommend barchart by Airline Status") +

scale\_fill\_manual(values = c("Blue" = "#20639B",

"Silver" = "#ED553B",

"Gold"= "#3CAEA3",

"Platinum" = "#F6D55C"))

#bar chart: type of travel vs type of customer

ggplot(ndata, aes(score)) +

geom\_bar(aes(fill = Type.of.Travel), color="white") +

ggtitle("Recommend barchart by Type of travel") +

scale\_fill\_manual(values = c("Mileage tickets" = "#F6D55C",

"Business travel"= "#20639B",

"Personal Travel" = "#ED553B"))

# bar chart: class vs type of customer

ggplot(ndata, aes(score)) +

geom\_bar(aes(fill = Class), color="white") +

ggtitle("Recommend barchart by Class") +

scale\_fill\_manual(values = c("Eco" = "#F6D55C",

"Eco Plus"= "#20639B",

"Business" = "#ED553B"))

# Histogram: Likelihood to recommend

ggplot(ndata, aes(Likelihood.to.recommend)) +

geom\_histogram(binwidth = 1, color="white", fill="#20639B") +

ggtitle("Histogram of Likelohood to recommend")

# Histogram: flight distance with type of customer

ggplot(ndata, aes(Flight.Distance)) +

geom\_histogram(aes(fill = score), color="white", binwidth=200) +

ggtitle("Histogram of flight distance with type of customer") +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B"))

# Bar chart: Day of Month vs Type of Customer

ggplot(ndata, aes(Day.of.Month)) +

geom\_bar(aes(fill = score), color="white") +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B")) +

ggtitle("Bar chart of Day of Month vs Type of Customer")

#relationship between EAD and likelihood to recommend

g <- ggplot(ndata, aes(x=Eating.and.Drinking.at.Airport, y=Likelihood.to.recommend))

g <- g + geom\_point(aes(color=Age))+ggtitle("Scatter graph of Eating and Drinking")

g

#relationship between Age and likelihood

Hist\_Age <- ggplot(ndata, aes(x = Age,fill=factor(score))) + geom\_histogram(binwidth = 10, color = "black") +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B")) +coord\_polar(theta = "x")

Hist\_Age <- Hist\_Age+ggtitle("Histogram of Age Filled with score")

Hist\_Age

#relationship between flight per year and likelihood

Hist\_year <- ggplot(ndata, aes(x =Flights.Per.Year,fill=factor(score))) + geom\_histogram(binwidth = 10, color = "black") +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B"))

Hist\_year <- Hist\_year+ggtitle("Histogram of flights per year Filled with score")

Hist\_year

#relationship between Price sensitivity and likelihood

Sati\_Prsy <- ggplot(ndata,aes(x=Price.Sensitivity))

Sati\_Prsy <- Sati\_Prsy + geom\_histogram(aes(fill=factor(score)),position = "dodge") +scale\_fill\_manual(values = c("Passive" = "#F6D55C", "Detractor"= "#20639B", "Promoter" = "#ED553B"))

Sati\_Prsy <- Sati\_Prsy+ ggtitle("score versus Price Sensitivity")

Sati\_Prsy

#loyalty

ggplot(ndata, aes(Loyalty)) +

geom\_histogram(aes(fill = score), color="white", binwidth=0.05) +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B")) + ggtitle("Score versus Loyalty")

#flight delay

ggplot(ndata, aes(Arrival.Delay.in.Minutes)) +

geom\_histogram(aes(fill = score), color="white", binwidth=40) +

scale\_fill\_manual(values = c("Passive" = "#F6D55C",

"Detractor"= "#20639B",

"Promoter" = "#ED553B"))+ggtitle("Score versus Arrival Delay")

#Likelihood to Recommend Map

dataset <- getURL("https://s3.us-east-1.amazonaws.com/blackboard.learn.xythos.prod/5956621d575cd/8614412?response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27fall2019-survey-M04.json&response-content-type=application%2Fjson&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20191211T021550Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=AKIAIL7WQYDOOHAZJGWQ%2F20191211%2Fus-east-1%2Fs3%2Faws4\_request&X-Amz-Signature=f4c7dcd84dd2945e532a3d4c053f3f8816d3680124e98641864ebb16c5da1231")

Df <- jsonlite::fromJSON(dataset)

colnames(Df)

sub <- Df[, -c(3,4,5,6,7, 8,9,10,11,12, 13, 14,15, 16, 17, 18, 21,22,23, 24, 25,26, 32)]

colnames(sub)

colnames(sub)[3] <- "region"

sub$region<- tolower(sub$region)

sub2<-sub%>%

group\_by(region)%>%

summarise(average=mean(Likelihood.to.recommend))

us <- map\_data("state")

merged <-merge(sub2, us, by="region")

map1 <- ggplot(merged, aes(map\_id=region)) #using "merged" to create visualization

map1 <- map1 + geom\_map(map=us, aes(fill=average)) #creating map of US, coloring states by "average" score

map1 <- map1+ expand\_limits(x= us$long, y=us$lat) #setting x-y axes as longitude and latitude

map1 <- map1 + coord\_map()

map1 <- map1 + ggtitle("US States: Mean Likelihood to Recommend") #setting title

map1

**Correlation**

cdata <- fdata

str(cdata)

cdata<-cdata[,-c(1,2,15,16,17,18,19,20,21,24,25,28,29,30,31,32)]

str(cdata)

cdata$Gender <- 0

cdata$Gender[fdata$Gender == "Male"] <- 1

cdata$Airline.Status <- as.character(fdata$Airline.Status)

cdata$Airline.Status <- 0

cdata$Airline.Status[fdata$Airline.Status == "Silver"] <- 1

cdata$Airline.Status[fdata$Airline.Status == "Gold"] <- 2

cdata$Airline.Status[fdata$Airline.Status == "Platinum"] <- 3

cdata$Class <- as.character(fdata$Class)

cdata$Class <- 0

cdata$Class[fdata$Class == "Eco Plus"] <- 1

cdata$Class[fdata$Class == "business"] <- 2

cdata$Type.of.Travel <- as.character(fdata$Type.of.Travel)

cdata$Type.of.Travel<- 0

cdata$Type.of.Travel[fdata$Type.of.Travel == "Mileage tickets"] <- 1

cdata$Type.of.Travel[fdata$Type.of.Travel == "Business travel"] <- 2

cdata$Likelihood.to.recommend<-as.numeric(fdata$Likelihood.to.recommend)

str(cdata)

#correlation

corr <- cor(cdata)

corr

#visualize it

corrplot(corr,type="upper",order = "hclust", tl.col = "black", tl.srt = 45, tl.cex = 0.5)

corrplot(corr,method = "pie")

Feature Scaling

scale\_data <- fdata

scale\_data$Age <- scale(scale\_data$Age)

scale\_data$Flights.Per.Year <- scale(scale\_data$Flights.Per.Year)

scale\_data$Shopping.Amount.at.Airport <- scale(scale\_data$Shopping.Amount.at.Airport)

scale\_data$Eating.and.Drinking.at.Airport <- scale(scale\_data$Eating.and.Drinking.at.Airport)

scale\_data$Departure.Delay.in.Minutes <- scale(scale\_data$Departure.Delay.in.Minutes)

scale\_data$Arrival.Delay.in.Minutes <- scale(scale\_data$Arrival.Delay.in.Minutes)

scale\_data$Flight.time.in.minutes <- scale(scale\_data$Flight.time.in.minutes)

scale\_data$Flight.Distance <- scale(scale\_data$Flight.Distance)

## Logistic Regression

set.seed(111)

trainIndex <- createDataPartition(scale\_data$recommend.class, p = .8,

list = FALSE,

times = 1)

Train\_data <- scale\_data[trainIndex,]

Test\_Data <- scale\_data[-trainIndex,]

Train\_data <- subset(Train\_data, select = -Likelihood.to.recommend)

Test\_Data <- subset(Test\_Data, select = -Likelihood.to.recommend)

Logistic regression for three class

log\_fit <- multinom(recommend.class~ Airline.Status + Age + Gender + Price.Sensitivity + Year.of.First.Flight + Flights.Per.Year + Loyalty + Type.of.Travel + Total.Freq.Flyer.Accts + Shopping.Amount.at.Airport + Eating.and.Drinking.at.Airport + Class + Day.of.Month + Partner.Name + Origin.State + Destination.State + Scheduled.Departure.Hour + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes + Flight.time.in.minutes + Flight.Distance,data = Train\_data)

summary(log\_fit)

Log\_pred <- predict(log\_fit, Test\_Data)

mean(Log\_pred == Test\_Data$recommend.class)

confusionMatrix(Log\_pred, Test\_Data$recommend.class)

Logistic regression for three class with LASSO

train\_y <- Train\_data$recommend.class

train\_x <- model.matrix(recommend.class~.,Train\_data)[,-1]

test\_x <- model.matrix(recommend.class~.,Test\_Data)[,-1]

test\_y <- Test\_Data$recommend.class

lasso.log <- glmnet(train\_x,train\_y,family = "multinomial",type.multinomial = "grouped")

plot(lasso.log, xvar = "lambda", label = TRUE, type.coef = "2norm")

cvfit=cv.glmnet(train\_x,train\_y, family="multinomial", type.multinomial = "grouped", parallel = TRUE)

plot(cvfit)

bestlam <- cvfit$lambda.min

lasso.pred <- predict(lasso.log,s = bestlam, newx = test\_x,type = "class")

lasso\_table <- table(lasso.pred,Test\_Data$recommend.class)

confusionMatrix(lasso\_table)

## KNN with three class

library(kknn)

knn\_model <- kknn(recommend.class~.,Train\_data,Test\_Data,distance = 1, k = 3)

summary(knn\_model)

fit <- fitted(knn\_model)

CM <- table(fit,Test\_Data$recommend.class)

confusionMatrix(CM)

Logistic Regression with 2 class and LASSO

Train\_data$recommend.class.2[Train\_data$Likelihood.to.recommend>6]<-"Other"

Train\_data$recommend.class.2[Train\_data$Likelihood.to.recommend<7]<-"Detractor"

Test\_Data$recommend.class.2[Test\_Data$Likelihood.to.recommend>6]<-"Other"

Test\_Data$recommend.class.2[Test\_Data$Likelihood.to.recommend<7]<-"Detractor"

Train\_data <- subset(Train\_data, select = -recommend.class)

Test\_Data <- subset(Test\_Data, select = -recommend.class)

Train\_data$recommend.class.2 <- factor(Train\_data$recommend.class.2)

Test\_Data$recommend.class.2 <- factor(Test\_Data$recommend.class.2)

train\_y <- Train\_data$recommend.class.2

train\_x <- model.matrix(recommend.class.2~.,Train\_data)[,-1]

test\_x <- model.matrix(recommend.class.2~.,Test\_Data)[,-1]

test\_y <- Test\_Data$recommend.class.2

lasso.log <- glmnet(train\_x,train\_y,family = "binomial",type.multinomial = "grouped")

plot(lasso.log, xvar = "lambda", label = TRUE, type.coef = "2norm")

cvfit=cv.glmnet(train\_x,train\_y, family="binomial", type.multinomial = "grouped", parallel = TRUE)

plot(cvfit)

bestlam <- cvfit$lambda.min

lasso.pred <- predict(lasso.log,s = bestlam, newx = test\_x,type = "class")

lasso\_table <- table(lasso.pred,Test\_Data$recommend.class.2)

confusionMatrix(lasso\_table)

## ARULES

ar<-fdata

str(ar)

ar<-ar[-c(1,2,15,16,21,24,25,28,29,30,31,32)]

str(ar)

breaks<-function(variable)

{

v<-variable

b<-quantile(variable,c(0.25,0.75))

v[variable<=b[1]]<-"Low"

v[variable>b[1] & variable<b[2]]<-"Average"

v[variable>=b[2]]<-"High"

v<-factor(v)

return(v)

}

ar$Likelihood.to.recommend<-"Passive"

ar$Likelihood.to.recommend[fdata$Likelihood.to.recommend>8]<-"Promoter"

ar$Likelihood.to.recommend[fdata$Likelihood.to.recommend<7]<-"Detractor"

View(ar)

sum(!complete.cases(ar))

str(ar)

for(i in c(2,4,5,6,7,9,10,11,17,18,19))

{

ar[,i]<-breaks(ar[,i])

}

str(ar)

ar$Departure.Delay.in.Minutes<-"No"

ar$Departure.Delay.in.Minutes[fdata$Departure.Delay.in.Minutes>5]<-"Yes"

ar$Arrival.Delay.in.Minutes<-"No"

ar$Arrival.Delay.in.Minutes[fdata$Arrival.Delay.in.Minutes>5]<-"Yes"

for(i in c(1,3,8,12,13,14,15,16,17,18,20))

{

ar[,i]<-factor(ar[,i])

}

str(ar)

ar<-ar[,-c(13,14,15,16)]

View(ar)

str(ar)

arx<-as(ar,"transactions")

arx

inspect(arx)

itemFrequency(arx)

itemFrequencyPlot(arx,cex.names = 0.3)

#for entire data

rule<-apriori(arx,parameter = list(support=0.1,confidence=0.8))

inspect(head(rule,5))

# for Promoter

rule1<-apriori(arx,parameter = list(support=0.02,confidence=0.4),

appearance = list(default="lhs",rhs="Likelihood.to.recommend=Promoter"))

inspect(head(rule1))

inspectDT(rule1)

top.lift.rule1 <- sort(rule1, decreasing = TRUE, na.last = NA, by = "lift")

inspect(head(top.lift.rule1,10))

inspectDT(head(top.lift.rule1,10))

plot(head(top.lift.rule1,20), method = "paracoord" ,measure = "support", shading = "lift", jitter=0)

# for Passive

rule2<-apriori(arx,parameter = list(support=0.04,confidence=0.4),

appearance = list(default="lhs",rhs="Likelihood.to.recommend=Passive"))

inspect(head(rule2))

inspectDT(rule2)

top.lift.rule2 <- sort(rule2, decreasing = TRUE, na.last = NA, by = "lift")

inspect( head(top.lift.rule2,10))

plot(head(top.lift.rule2,20), method = "paracoord" ,measure = "support", shading = "lift", jitter=0)

# for Detractor

rule3<-apriori(arx,parameter = list(support=0.04,confidence=0.7),

appearance = list(default="lhs",rhs="Likelihood.to.recommend=Detractor"))

inspect(head(rule3))

inspectDT(rule3)

top.lift.rule3 <- sort(rule3, decreasing = TRUE, na.last = NA, by = "lift")

inspect(head(top.lift.rule3,10))

plot(head(top.lift.rule3,20), method = "paracoord" ,measure = "support", shading = "lift", jitter=0)

## Support Vector Machine (SVM)

library(jsonlite)

library(RCurl)

library(tidyverse)

library(dplyr)

library(stringr)

library(kernlab)

library(e1071)

dataset <- getURL("https://s3.us-east-1.amazonaws.com/blackboard.learn.xythos.prod/5956621d575cd/8614412?response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27fall2019-survey-M04.json&response-content-type=application%2Fjson&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20191210T200836Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=AKIAIL7WQYDOOHAZJGWQ%2F20191210%2Fus-east-1%2Fs3%2Faws4\_request&X-Amz-Signature=775be2b4b77944378f9a17025ab9aa325ab64cb884f7175092fd35d5d09812e9")

Df <- jsonlite::fromJSON(dataset)

#Creating subset

Df <- Df%>%

filter(Partner.Name=="Oursin Airlines Inc."|Partner.Name=="Cheapseats Airlines Inc."|Partner.Name=="Northwest Business Airlines Inc."|Partner.Name=="FlyFast Airways Inc."|Partner.Name=="GoingNorth Airlines Inc.")

Df <- Df[, -c(1,2,7, 12, 13, 15, 16, 17, 19, 20, 21, 24, 25, 28, 29, 30, 31, 32)] #removing unnecessary columns

colnames(Df)

#removing data with NA values (especially in Arrival Delay and Departure Delay)

na\_rows<- c(which(is.na(Df$Departure.Delay.in.Minutes),)) #rows with NA values

Df <- Df[-na\_rows,] #removing those rows

na\_rows2<- c(which(is.na(Df$Arrival.Delay.in.Minutes),)) #rows with NA values

Df <- Df[-na\_rows2,] #removing those rows

#converting columns to numeric type

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

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

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

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

Df$Total.Freq.Flyer.Accts <- as.numeric(Df$Total.Freq.Flyer.Accts)

Df$Departure.Delay.in.Minutes <- as.numeric(Df$Departure.Delay.in.Minutes)

Df$Arrival.Delay.in.Minutes <- as.numeric(Df$Arrival.Delay.in.Minutes)

Df$Flight.Distance <- as.numeric(Df$Flight.Distance)

Df$Likelihood.to.recommend <- as.numeric(Df$Likelihood.to.recommend)

#converting columns to factors

Df$Airline.Status <- as.factor(Df$Airline.Status)

Df$Gender <- as.factor(Df$Gender)

Df$Type.of.Travel <- as.factor(Df$Type.of.Travel)

Df$Class <- as.factor(Df$Class)

Df$Partner.Name <- as.factor(Df$Partner.Name)

# creating column that classifies customer as Promoter, Passive or Detractor based on “Likelihood to Recommend” score

Df$RecommenderType <- "Passive"

Df$RecommenderType[Df$Likelihood.to.recommend>8] <- "Promoter"

Df$RecommenderType[Df$Likelihood.to.recommend<7] <- "Detractor"

Df$RecommenderType <- as.factor(Df$RecommenderType)

#Scaling variables

Df$Age <- scale(Df$Age)

Df$Price.Sensitivity <- scale(Df$Price.Sensitivity)

Df$Flights.Per.Year <- scale(Df$Flights.Per.Year)

Df$Loyalty <- scale(Df$Loyalty)

Df$Total.Freq.Flyer.Accts <- scale(Df$Total.Freq.Flyer.Accts)

Df$Departure.Delay.in.Minutes <- scale(Df$Departure.Delay.in.Minutes)

Df$Arrival.Delay.in.Minutes <- scale(Df$Arrival.Delay.in.Minutes)

Df$Flight.Distance <- scale(Df$Flight.Distance)

Df$Likelihood.to.recommend <- scale(Df$Likelihood.to.recommend)

#generating random indices

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

#Dividing data into test and train sets for SVM

# cut off point :2/3 -->Train, 1/3 -->Test

cutPoint2\_3 <- floor(2\*dim(Df)[1]/3)

cutPoint2\_3

trainData <- Df[randIndex[1:cutPoint2\_3],] #create train set by using first 2/3rd of data

testData <- Df[randIndex[(cutPoint2\_3+1):dim(Df)[1]],] #create test set by using the remaining 1/3rd part of data

#creating svm model

svm1 <- svm(RecommenderType~., data=trainData,

method="C-classification", kernel="radial",

gamma=0.1, cost=10)

summary(svm1) #viewing our svm model

#Make predictions using our model for the test dataset

prediction <- predict(svm1, testData)

#Creating confusion matrix

tab <- table(testData$RecommenderType, prediction)

tab

#Computing accuracy

accuracy=(723+665+468)/nrow(testData)

accuracy

RANDOM FOREST

library(randomForest)

random <- sample(1:dim(ar)[1])

summary(random)

length(random)

# Build a training dataset and test dataset. The training datatset should be two-thirds of the data, and the test dataset should be one third of the data.

cutpoint <- floor(2\*dim(ar)[1]/3)

cutpoint

traindata <- ar[random[1:cutpoint],]

testdata <- ar[random[(cutpoint+1):dim(ar)[1]],]

# Use the dim() command to demonstrate that the resulting training dataset and test data contain the appropriate number of cases.

dim(traindata)

dim(testdata)

# model 1

rf1 <- randomForest(Likelihood.to.recommend ~ .,data = traindata, importance = TRUE)

rf1

# Predicting on train set

predTrain <- predict(rf1, traindata, type = "class")

# Checking classification accuracy

table(predTrain, traindata$Likelihood.to.recommend)

# Predicting on Validation set

predTest <- predict(rf1, testdata, type = "class")

# Checking classification accuracy

mean(predTest == testdata$Likelihood.to.recommend)

table(predTest,testdata$Likelihood.to.recommend)

varImpPlot(rf1)

#model 2

# running the model with mtry values from 3 to 8

a=c()

i=5

for (i in 3:8) {

rf2 <- randomForest(Likelihood.to.recommend ~ ., data = traindata, ntree = 300, mtry = i, importance = TRUE)

predTest2 <- predict(rf2, testdata, type = "class")

a[i-2] = mean(predTest2 == testdata$Likelihood.to.recommend)

}

a

plot(3:8,a)