Project 4 - Airline Passenger Satisfaction

Yohan Chandrasukmana - 01112190011

Contents

Data Source: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

0.1 Libraries and prerequisites

```
library(ggplot2) # ggplot
library(reshape2) # melt()
```

```
## Warning: package 'reshape2' was built under R version 4.1.2
```

```
library(tictoc) # Time: tic(), toc()
```

1 Data Loading

```
train <- read.csv("C:/Users/chand/OneDrive - Universitas Pelita Harapan/Kuliah/Semester 8/Datmin/Projec
test <- read.csv("C:/Users/chand/OneDrive - Universitas Pelita Harapan/Kuliah/Semester 8/Datmin/Project/
```

Since the data has been divided into train and test sets, both datasets will be combined into one data.

```
airline <- rbind(train, test)
# Keeping id column in train and test and which can be used to split the combined data for modelling.
# train = train[2]
# test = test[2]
rm(train, test) # Removing train and test set
head(airline)
```

```
##   X     id Gender     Customer.Type Age  Type.of.Travel    Class
## 1 0  70172   Male    Loyal Customer  13 Personal Travel Eco Plus
## 2 1   5047   Male disloyal Customer  25 Business travel Business
## 3 2 110028 Female    Loyal Customer  26 Business travel Business
## 4 3  24026 Female    Loyal Customer  25 Business travel Business
## 5 4 119299   Male    Loyal Customer  61 Business travel Business
## 6 5 111157 Female    Loyal Customer  26 Personal Travel      Eco
##   Flight.Distance Inflight.wifi.service Departure.Arrival.time.convenient
## 1             460                     3                                 4
## 2             235                     3                                 2
## 3            1142                     2                                 2
## 4             562                     2                                 5
## 5             214                     3                                 3
## 6            1180                     3                                 4
##   Ease.of.Online.booking Gate.location Food.and.drink Online.boarding
## 1                      3             1              5               3
## 2                      3             3              1               3
## 3                      2             2              5               5
## 4                      5             5              2               2
## 5                      3             3              4               5
## 6                      2             1              1               2
##   Seat.comfort Inflight.entertainment On.board.service Leg.room.service
## 1            5                      5                4                3
## 2            1                      1                1                5
## 3            5                      5                4                3
## 4            2                      2                2                5
## 5            5                      3                3                4
## 6            1                      1                3                4
##   Baggage.handling Checkin.service Inflight.service Cleanliness
## 1                4               4                5           5
## 2                3               1                4           1
## 3                4               4                4           5
## 4                3               1                4           2
## 5                4               3                3           3
```

```
## 6                4               4                4           1
##   Departure.Delay.in.Minutes Arrival.Delay.in.Minutes            satisfaction
## 1                         25                       18 neutral or dissatisfied
## 2                          1                        6 neutral or dissatisfied
## 3                          0                        0               satisfied
## 4                         11                        9 neutral or dissatisfied
## 5                          0                        0               satisfied
## 6                          0                        0 neutral or dissatisfied
```

```
dim(airline)
```

```
## [1] 129880     25
```

The whole data set has 129880 observations (rows) with 25 columns including 22 features and 1 outcome.

1.1 Data Description

`satisfaction`: Passenger's overall satisfaction - Dependent Variable `id`: Flight Passenger ID - ID

1. `Gender`: Gender of the passenger's (Female, Male) - Categorical
2. `Customer Type`: The customers' type based on their loyalty (Loyal customer, Disloyal customer) - Categorical
3. `Age`: The actual age of the passengers - Numerical
4. `Type of Travel`: Passenger's purpose of flight (Personal Travel, Business Travel) - Categorical
5. `Class`: Travel class in the plane of the passengers (Business, Eco, Eco Plus) - Categorical
6. `Flight distance`: The flight distance of the journey - Numerical
7. `Inflight wifi service`: Satisfaction level of the in-flight wi-fi service (0:Not Applicable;1-5) - Categorical
8. `Departure/Arrival time convenient`: Satisfaction level of Departure/Arrival time convenience - Categorical
9. `Ease of Online booking`: Satisfaction level of online booking - Categorical
10. `Gate location`: Satisfaction level of gate location - Categorical
11. `Food and drink`: Satisfaction level of food and drinks - Categorical
12. `Online boarding`: Satisfaction level of online boarding - Categorical
13. `Seat comfort`: Satisfaction level of seat comfort - Categorical
14. `Inflight entertainment`: Satisfaction level of in-flight entertainment - Categorical
15. `On-board service`: Satisfaction level of on-board service - Categorical
16. `Leg room service`: Satisfaction level of leg room service - Categorical
17. `Baggage handling`: Satisfaction level of baggage handling - Categorical
18. `Check-in service`: Satisfaction level of check-in service - Categorical
19. `Inflight service`: Satisfaction level of in-flight service - Categorical
20. `Cleanliness`: Satisfaction level of cleanliness - Categorical
21. `Departure Delay in Minutes`: Minutes delayed during departure - Numerical
22. `Arrival Delay in Minutes`: Minutes delayed during Arrival - Numerical
23. `Satisfaction`: Airline satisfaction level (Satisfaction, neutral or dissatisfaction) - Categorical

2 Data Cleaning

```r
# Dropping X (index) variable
airline = airline[-1]
```

```r
# Formatting categorical variables as factors
categorical <- c('Gender',
                 'Customer.Type',
                 'Type.of.Travel',
                 'Class',
                 'Inflight.wifi.service',
                 'Departure.Arrival.time.convenient',
                 'Ease.of.Online.booking',
                 'Gate.location',
                 'Food.and.drink',
                 'Online.boarding',
                 'Seat.comfort',
                 'Inflight.entertainment',
                 'On.board.service',
                 'Leg.room.service',
                 'Baggage.handling',
                 'Checkin.service',
                 'Inflight.service',
                 'Cleanliness',
                 'satisfaction')
airline[, categorical] = lapply(airline[, categorical], factor)
```

```r
# Checking for missing values.
sapply(airline, function(x) sum(is.na(x)))
```

```
##                                 id                             Gender
##                                  0                                  0
##                      Customer.Type                                Age
##                                  0                                  0
##                     Type.of.Travel                              Class
##                                  0                                  0
##                    Flight.Distance              Inflight.wifi.service
##                                  0                                  0
## Departure.Arrival.time.convenient            Ease.of.Online.booking
##                                  0                                  0
##                       Gate.location                     Food.and.drink
##                                  0                                  0
##                    Online.boarding                       Seat.comfort
##                                  0                                  0
##             Inflight.entertainment                    On.board.service
##                                  0                                  0
##                   Leg.room.service                    Baggage.handling
##                                  0                                  0
##                    Checkin.service                    Inflight.service
##                                  0                                  0
##                        Cleanliness         Departure.Delay.in.Minutes
##                                  0                                  0
##          Arrival.Delay.in.Minutes                       satisfaction
##                                393                                  0
```

Since the missing values are in arrival delay, the missing values will be considered.

```
c("Mean" = mean(airline$Arrival.Delay.in.Minutes, na.rm = T),
  "Median" = median(airline$Arrival.Delay.in.Minutes, na.rm = T))
```

```
##     Mean   Median
## 15.09113  0.00000
```

From the code above, the mean of the delay is 15 minutes. However, the median is only 0. Since the mean may be subject to outliers, the data will be imputed with the median of the column in the data set.

```
airline$Arrival.Delay.in.Minutes[is.na(airline$Arrival.Delay.in.Minutes)] <- median(airline$Arrival.Dela

sum(is.na(airline))
```

```
## [1] 0
```

```
summary(airline)
```

```
##        id            Gender                Customer.Type         Age
##  Min.   :     1   Female:65899   disloyal Customer: 23780   Min.   : 7.00
##  1st Qu.: 32471   Male  :63981   Loyal Customer   :106100   1st Qu.:27.00
##  Median : 64941                                            Median :40.00
##  Mean   : 64941                                            Mean   :39.43
##  3rd Qu.: 97410                                            3rd Qu.:51.00
##  Max.   :129880                                            Max.   :85.00
##          Type.of.Travel       Class       Flight.Distance Inflight.wifi.service
##  Business travel:89693   Business:62160   Min.   :  31    0: 3916
##  Personal Travel:40187   Eco     :58309   1st Qu.: 414    1:22328
##                          Eco Plus: 9411   Median : 844    2:32320
##                                           Mean   :1190    3:32185
##                                           3rd Qu.:1744    4:24775
##                                           Max.   :4983    5:14356
##  Departure.Arrival.time.convenient Ease.of.Online.booking Gate.location
##  0: 6681                           0: 5682                0:    1
##  1:19409                           1:21886                1:21991
##  2:21534                           2:30051                2:24296
##  3:22378                           3:30393                3:35717
##  4:31880                           4:24444                4:30466
##  5:27998                           5:17424                5:17409
##  Food.and.drink Online.boarding Seat.comfort Inflight.entertainment
##  0:  132        0: 3080         0:    1      0:   18
##  1:16051        1:13261         1:15108      1:15675
##  2:27383        2:21934         2:18529      2:21968
##  3:27794        3:27117         3:23328      3:23884
##  4:30563        4:38468         4:39756      4:36791
##  5:27957        5:26020         5:33158      5:31544
##  On.board.service Leg.room.service Baggage.handling Checkin.service
##  0:    5          0:  598          1: 9028          0:    1
##  1:14787          1:12895          2:14362          1:16108
##  2:18351          2:24540          3:25851          2:16102
##  3:28542          3:25056          4:46761          3:35453
```

```
##  4:38703          4:35886          5:33878          4:36333
##  5:29492          5:30905                           5:25883
##  Inflight.service Cleanliness Departure.Delay.in.Minutes
##  0:    5          0:   14     Min.   :   0.00
##  1: 8862          1:16729     1st Qu.:   0.00
##  2:14308          2:20113     Median :   0.00
##  3:25316          3:30639     Mean   :  14.71
##  4:47323          4:33969     3rd Qu.:  12.00
##  5:34066          5:28416     Max.   :1592.00
##  Arrival.Delay.in.Minutes                  satisfaction
##  Min.   :   0.00          neutral or dissatisfied:73452
##  1st Qu.:   0.00          satisfied              :56428
##  Median :   0.00
##  Mean   :  15.05
##  3rd Qu.:  13.00
##  Max.   :1584.00
```

2.1 Independent and Dependent Variables

For this project, the following variables will be chosen.

- Dependent Variable: `satisfaction`
- Independent Variables:
  - Numerical Variables: `Age`, `Flight.distance`, `Departure.Delay.in.Minutes`, `Arrival.Delay.in.Minutes`
  - Categorical Variables: `Gender`, `Customer.Type`, `Type.of.Travel`, `Class`, `Gate.location`, `Seat.comfort`, `Inflight.service`

```
numerical <- c('Age', 'Flight.Distance', 'Departure.Delay.in.Minutes', 'Arrival.Delay.in.Minutes')
categorical <- c('Gender', 'Customer.Type', 'Type.of.Travel', 'Class', 'Gate.location', 'Seat.comfort',

data = airline[,names(airline) %in% c("id", numerical, categorical, "satisfaction")]
rm(airline)
```

The dependent and independent variables will be renamed.

```
# Renaming Variable Column Names
names(data) <- c("id", "Gender", "Cust_Type", "Age", "Travel_Type", "Class", "Flight_Dist", "Gate_Loc",
numerical <- c('Age', 'Flight_Dist', 'Departure_Delay', 'Arrival_Delay')
categorical <- c('Gender', 'Cust_Type', 'Travel_Type', 'Class', 'Gate_Loc', 'Seat', 'Inflight_Svc')

# Renaming Categ Var Factor Levels
levels(data$Satisfaction) <- c("Neutral or Dissatisfied", "Satisfied")
levels(data$Cust_Type) <- c("Disloyal", "Loyal")
levels(data$Travel_Type) <- c("Business Travel", "Personal Travel")

head(data)
```

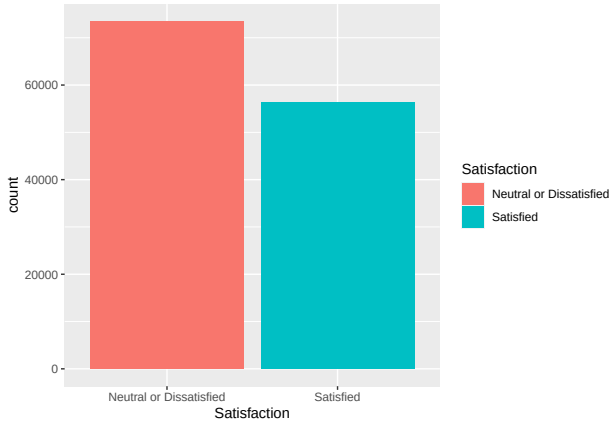
```
##       id Gender Cust_Type Age     Travel_Type    Class Flight_Dist Gate_Loc
## 1  70172   Male     Loyal  13 Personal Travel Eco Plus         460        1
## 2   5047   Male  Disloyal  25 Business Travel Business         235        3
## 3 110028 Female     Loyal  26 Business Travel Business        1142        2
```

```
## 4  24026 Female     Loyal  25 Business Travel Business         562        5
## 5 119299   Male     Loyal  61 Business Travel Business         214        3
## 6 111157 Female     Loyal  26 Personal Travel      Eco        1180        1
##   Seat Inflight_Svc Departure_Delay Arrival_Delay           Satisfaction
## 1    5            5              25            18 Neutral or Dissatisfied
## 2    1            4               1             6 Neutral or Dissatisfied
## 3    5            4               0             0              Satisfied
## 4    2            4              11             9 Neutral or Dissatisfied
## 5    5            3               0             0              Satisfied
## 6    1            4               0             0 Neutral or Dissatisfied
```

3 Exploratory Data Analysis

In this part, the dependent and independent variables in the data will be explored through data visualization.

```
ggplot(data) + geom_bar(aes(x=Satisfaction, fill=Satisfaction))
```



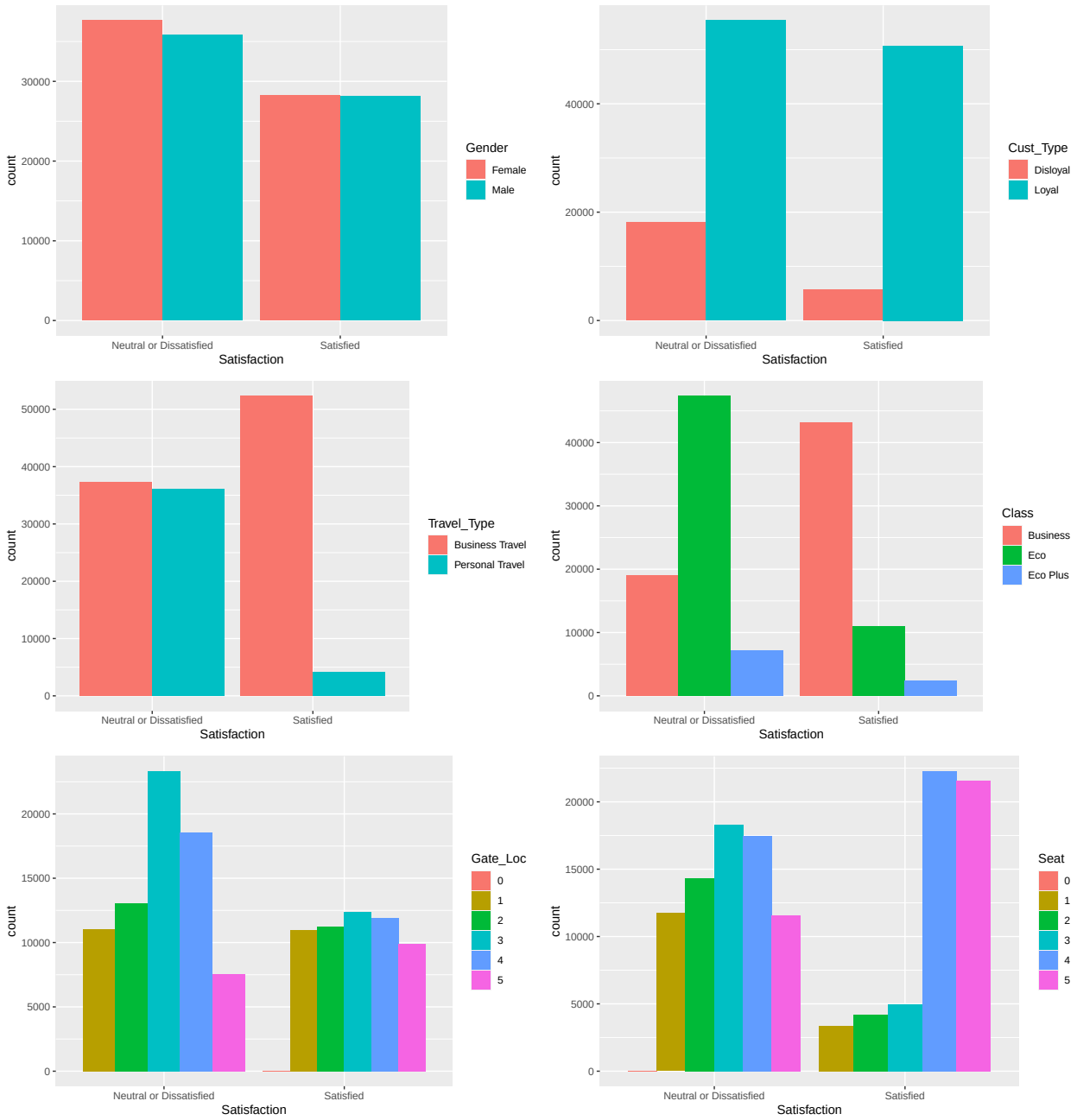
It can be observed that the dependent variable is evenly distributed between neutral or dissatisfied and satisfied.

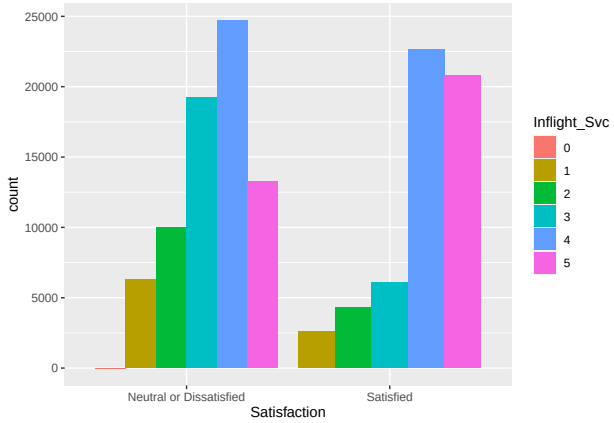
```
lapply(categorical, function(x) ggplot(data, aes(Satisfaction, ..count..)) + geom_bar(aes_string(fill=x)
```

```
## [[1]]

##
## [[2]]

##
## [[3]]

##
## [[4]]

##
## [[5]]
```

```
##
## [[6]]
```

```
##
## [[7]]
```


count
30000
20000
10000
0
Neutral or Dissatisfied
Satisfied
Satisfaction
Gender
Female
Male
40000
Cust_Type
Disloyal
Loyal
50000
Travel_Type
Business Travel
Personal Travel
Class
Business
Eco
Eco Plus
15000
5000
Gate_Loc
0
1
2
3
4
5
Seat

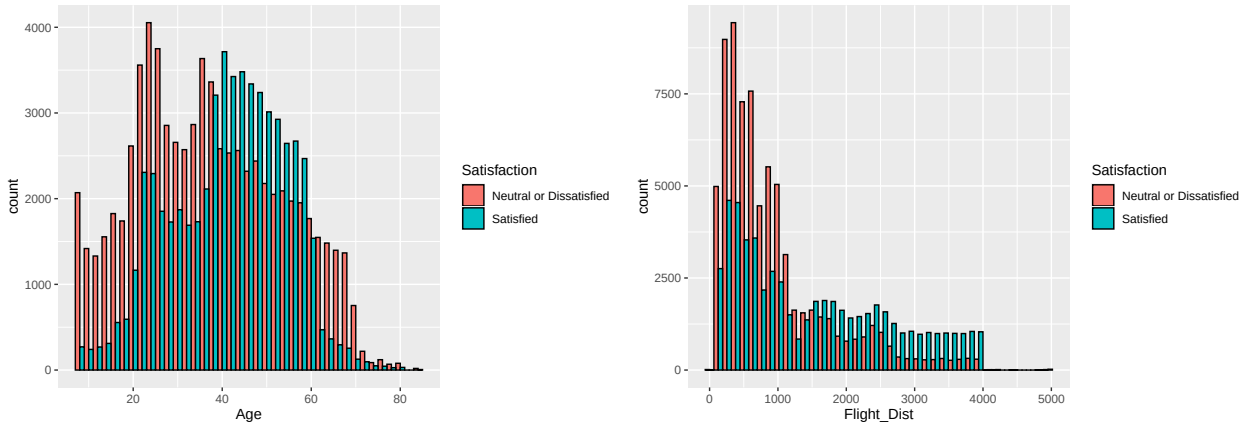


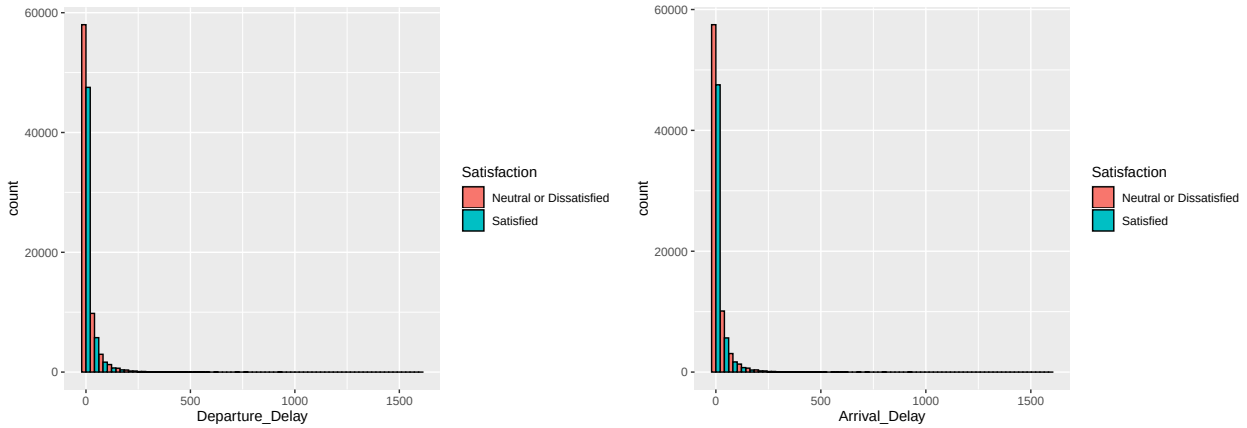
From the bar plots above, the general distribution of each independent variable in the data with respect to the customer's overall satisfaction. There are more loyal customers than disloyal customers with less customers are satisfied in both types. Customers who travel for business seemed to have a much more overall satisfaction in contrast to customers who travel for personal means who tend to be less satisfied with the flight. The same also holds for customers in Business Class in comparison to Eco and Eco Plus Classes.

Meanwhile, satisfied customers rated higher in the satisfaction of seat comfort and in-flight service while their perspectives vary with respect to the gate location.

```
lapply(numerical, function(x) ggplot(data, aes_string(x=x, fill="Satisfaction"))
       + geom_histogram(color="black", bins=40, position = "dodge"))
```

```
## [[1]]

##
## [[2]]

##
## [[3]]

##
## [[4]]
```

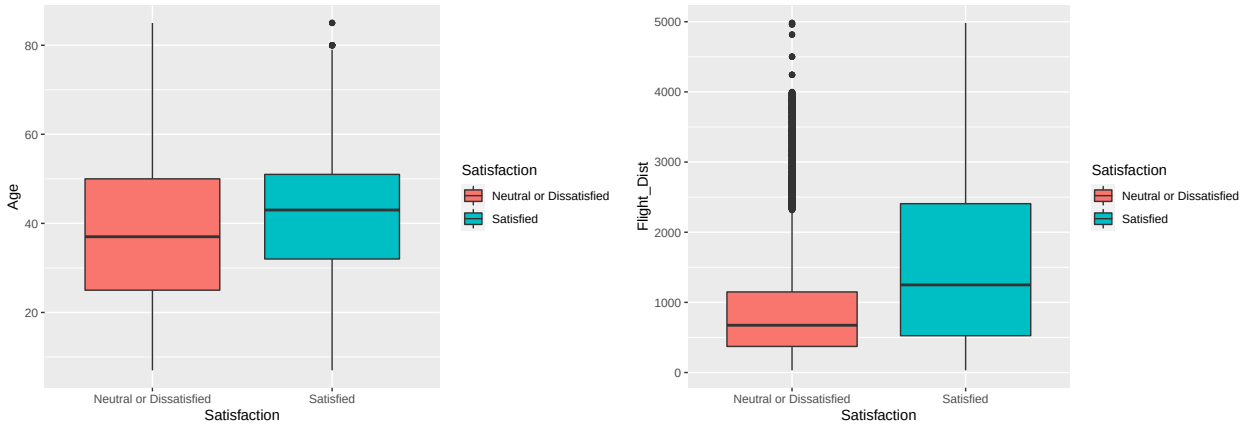


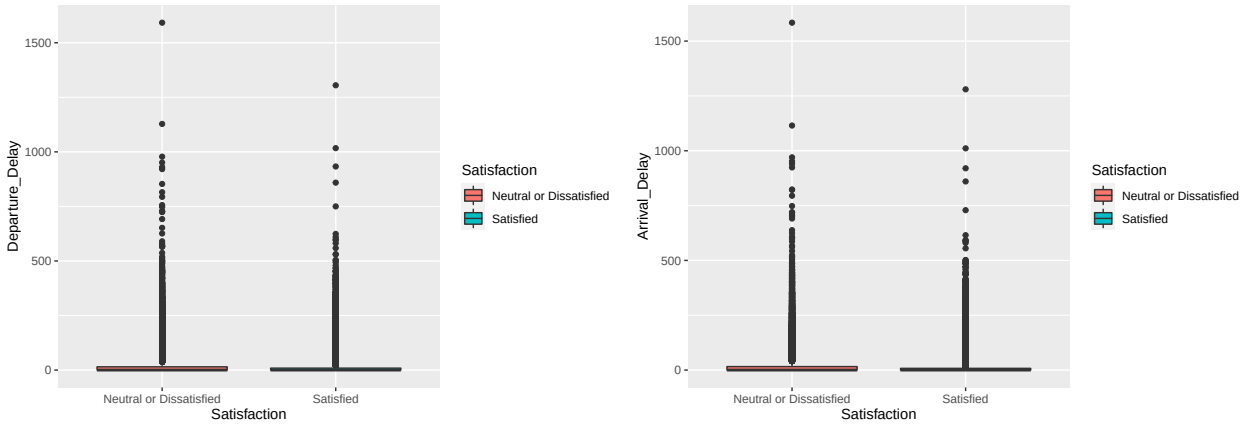


In the numerical variables, it can be observed that their distributions are skewed. This is increasingly apparent especially in the departure and arrival delay of the flights with only a small number of customers experiencing delays. There might be outliers in the data that can be removed in the future. Additionally, the flight distance is right-skewed while the age of the passengers resembles a normal distribution with people aged 20-40 are more likely to be unsatisfied, and people aged 40-60 are more likely to be satisfied.

```
lapply(numerical, function(y) ggplot(data, aes_string(x="Satisfaction", y=y, fill="Satisfaction"))
       + geom_boxplot())
```

```
## [[1]]

##
## [[2]]

##
## [[3]]

##
## [[4]]
```





The histograms further show the existence of outliers in the data. As previously mentioned, Departure and Arrival Delay might have a number of outliers. A removal of these outliers can be considered in future modelling.

4 Train-Test and CV Split

```
library(caret) # createDataPartition() and createFolds()
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
## Loading required package: lattice
```

Stratified sampling will be applied in the train-test set split with `createDataPartition()` with a split ratio of 80:20. Folds for cross validation in the train set will be done with `createFolds()` with an amount of 5 folds.

```
set.seed(1)

train_idx <- createDataPartition(y=data$Satisfaction, p=0.8, list=F)
train <- data[train_idx, ]
test <- data[-train_idx, ]

n.folds <- 5
folds <- createFolds(y=train$Satisfaction, k=n.folds, list=T, returnTrain=F)
```

5 Modelling - Random Forest

Random Forest model will be created using the `randomForest()` function.

```
library(randomForest) # randomForest() function
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

5.1 Cross Validation

Parameter optimization will be done with 5-Fold Cross Validation. The parameters to be tuned include `n.trees` and `mtry`. The n.trees and mtry choices are based on risk of overfitting and computation and memory limitations.

```
try_mtry_rf <- c(3, 5, 7, 9)
try_ntrees_rf <- c(50, 100, 200, 500)

cv_acc_rf <- NULL
cv_acc_rf <- matrix(nrow = length(try_ntrees_rf), ncol = length(try_mtry_rf))
rownames(cv_acc_rf) = try_ntrees_rf
colnames(cv_acc_rf) = try_mtry_rf
```

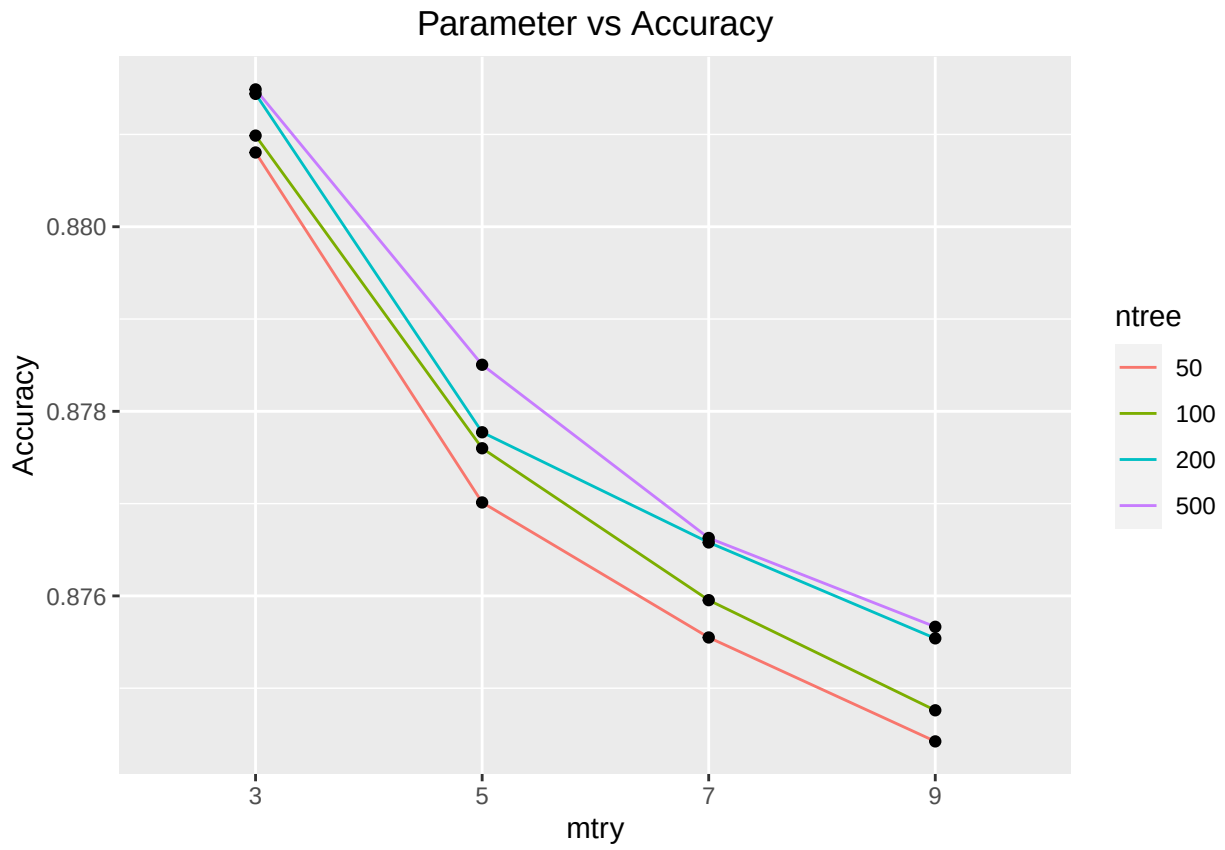
```
# Note: this code will not be run for knitting purposes
# Results are presented in the next chunk
tic("RF CV")
for (n in try_ntrees_rf){
  acc.ave <- NULL; print(n)
  for (m in try_mtry_rf){
    acc <- NULL; print(m); i = 1
    for(fold in folds){
      print(i); i=i+1
      ## Random Forest
      mod_rf = randomForest(Satisfaction~.-id, data=train[-fold, ],
                            ntree=n, mtry=m)
      ## Accuracy in the validation set
      pred = factor(predict(mod_rf, newdata=train[fold, ], type="response"))
      acc = c(acc, confusionMatrix(pred, train[fold, ]$Satisfaction,
                                   positive="Satisfied")$overall["Accuracy"])

      ## Freeing Memory
      rm(mod_rf); gc()
    }
  acc.ave = c(acc.ave, mean(acc))
  }
  cv_acc_rf[paste(n), ] = acc.ave
}
toc()
```

```
# RF Cross Validation Results ~ time elapsed: 46.23533 mins
cv_acc_rf["50", ] <- c(0.8808046,0.8770127,0.8755498,0.8744238)
cv_acc_rf["100", ] <- c(0.8809874,0.8775998,0.8759540,0.8747606)
```

```
cv_acc_rf["200", ] <- c(0.8814398,0.8777730,0.8765796,0.8755402)
cv_acc_rf["500", ] <- c(0.8814879,0.8785044,0.8766277,0.8756653)

cv_acc_rf = melt(cv_acc_rf)
cv_acc_rf$Var1 = as.factor(cv_acc_rf$Var1)
cv_acc_rf$Var2 = as.factor(cv_acc_rf$Var2)
```

```
ggplot(melt(cv_acc_rf), aes(x = Var2, y = value)) +
  geom_line(aes(color = Var1, group = Var1)) +
  geom_point()+
  ggtitle("Parameter vs Accuracy") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x = "mtry", y = "Accuracy", color = "ntree")
```

```
## Using Var1, Var2 as id variables
```



From the graph above, it can be observed that there is a pattern in which as mtry increases, the accuracy decreases, and vice versa for the ntree parameter. Additionally, the increase from 200 to 500 number of trees does not change much in the accuracy of the model. Theoretically, the ideal amount of variable in each tree split will be around the square root of total independent variables. In this case, it would be the square root of 12 would be around 3.46, which is close to the ideal mtry from the cross validation.

Next, the risk of overfitting should be taken into context when choosing the optimum number of trees for the random forest model. Therefore, since the ntree=50 model's accuracy decrease only around 1.5% in comparison to the ntree=500 model, a better choice for the ntree would be 50 to avoid overfitting from creating too many trees.

Therefore, parameters mtry=3 and ntree=200 will be chosen as the optimal parameters.

```
best_mtry_rf = 3
best_ntree_rf = 50
```

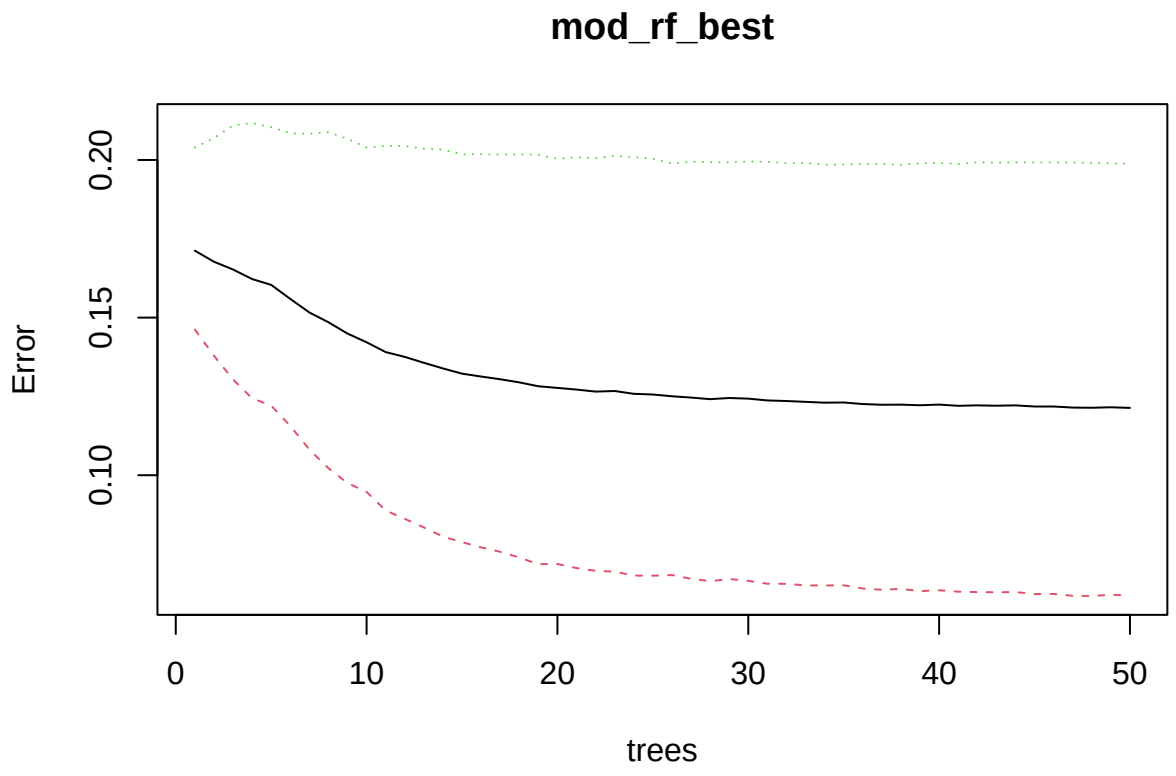
5.2 Modelling with Tuned Parameters

A model will now be created with optimized parameters obtained from the cross validation.

```
tic("RF Best")
mod_rf_best = randomForest(Satisfaction~.-id, data=train,
                           ntree=best_ntree_rf, mtry=best_mtry_rf)
toc()
```

```
## RF Best: 16.23 sec elapsed
```

```
plot(mod_rf_best)
```



From the plot above, with more ntrees as previously mentioned, the amount error would not be much less. On the other hand, a simpler model is obtained.

Model's prediction on the train set is as follows.

```
yhat = factor(predict(mod_rf_best, newdata=train, type="response"))
cf_rf = confusionMatrix(yhat, train$Satisfaction, positive="Satisfied")
cf_rf$overall["Accuracy"]
```

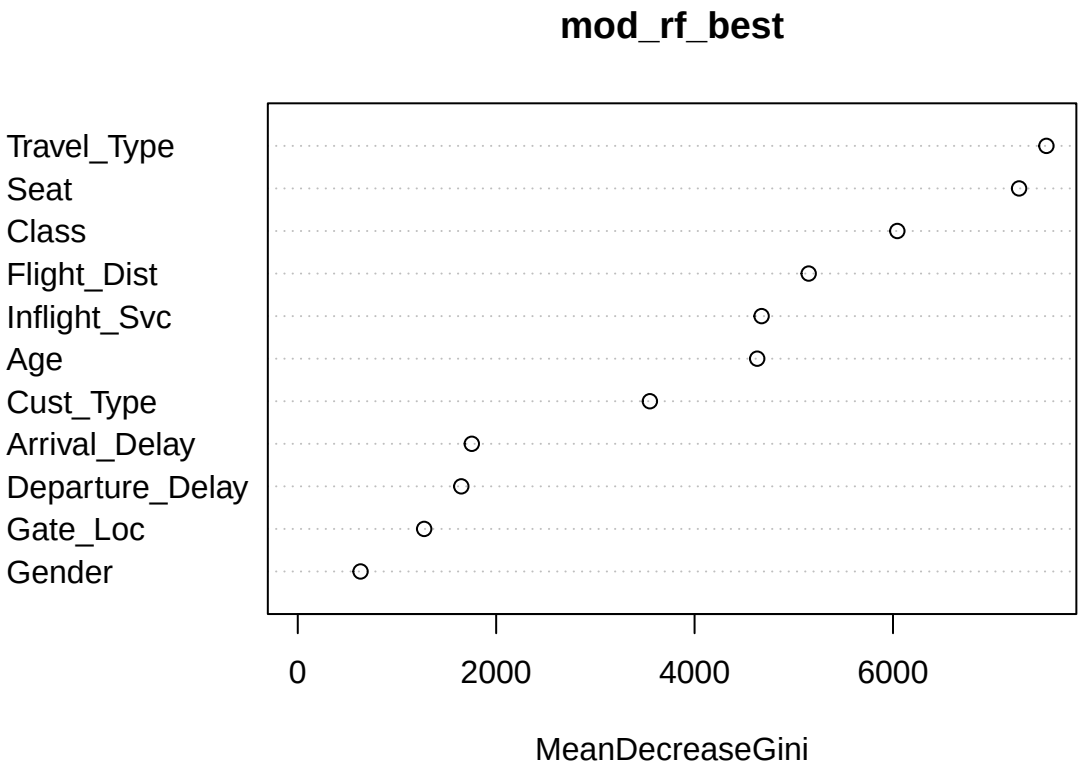
```
##  Accuracy
## 0.9490015
```

5.3 Prediction and Model Evaluation

```
## Predicting the test set
pred = factor(predict(mod_rf_best, newdata=test, type="response"))
cf_rf = confusionMatrix(pred, test$Satisfaction, positive="Satisfied")
cf_rf$overall["Accuracy"]
```

```
##  Accuracy
## 0.8811935
```

```
(imp_rf <- varImpPlot(mod_rf_best))
```




```
##                 MeanDecreaseGini
## Gender                  632.8495
```

```
## Cust_Type              3549.2908
## Age                    4631.2600
## Travel_Type            7546.4581
## Class                  6043.4645
## Flight_Dist            5150.1806
## Gate_Loc               1274.2154
## Seat                   7271.0254
## Inflight_Svc           4675.3204
## Departure_Delay        1648.2911
## Arrival_Delay          1754.2483
```

It can be seen from the plot above that the three main variables that have a high importance towards the dependent variable include Seat, Travel_Type, and Class. Additionally, arrival and departure Delay, gate location satisfaction, and gender received the lowest variable importance.

6 Modelling - Gradient Boosting Method

Gradient Boosting Method (GBM) model will be created using the `gbm()` function.

```
library(gbm) # gbm() function
```

```
## Warning: package 'gbm' was built under R version 4.1.3
```

```
## Loaded gbm 2.1.8
```

6.1 Cross Validation

Parameter optimization will be done with 5-Fold Cross Validation. The parameters to be tuned include `n.trees` and `shrinkage` or learning rate of the GBM model. The n.trees and shrinkage choices are based on risk of overfitting and computation and memory limitations. Furthermore, the GBM model will be created using bernoulli distribution as the dependent variable is either neutral or dissatisfied (0), or satisfied (1). The dependent variable in the train and test data will be converted to 0 and 1 for the modelling. The threshold of 0.5 will also be used for classification.

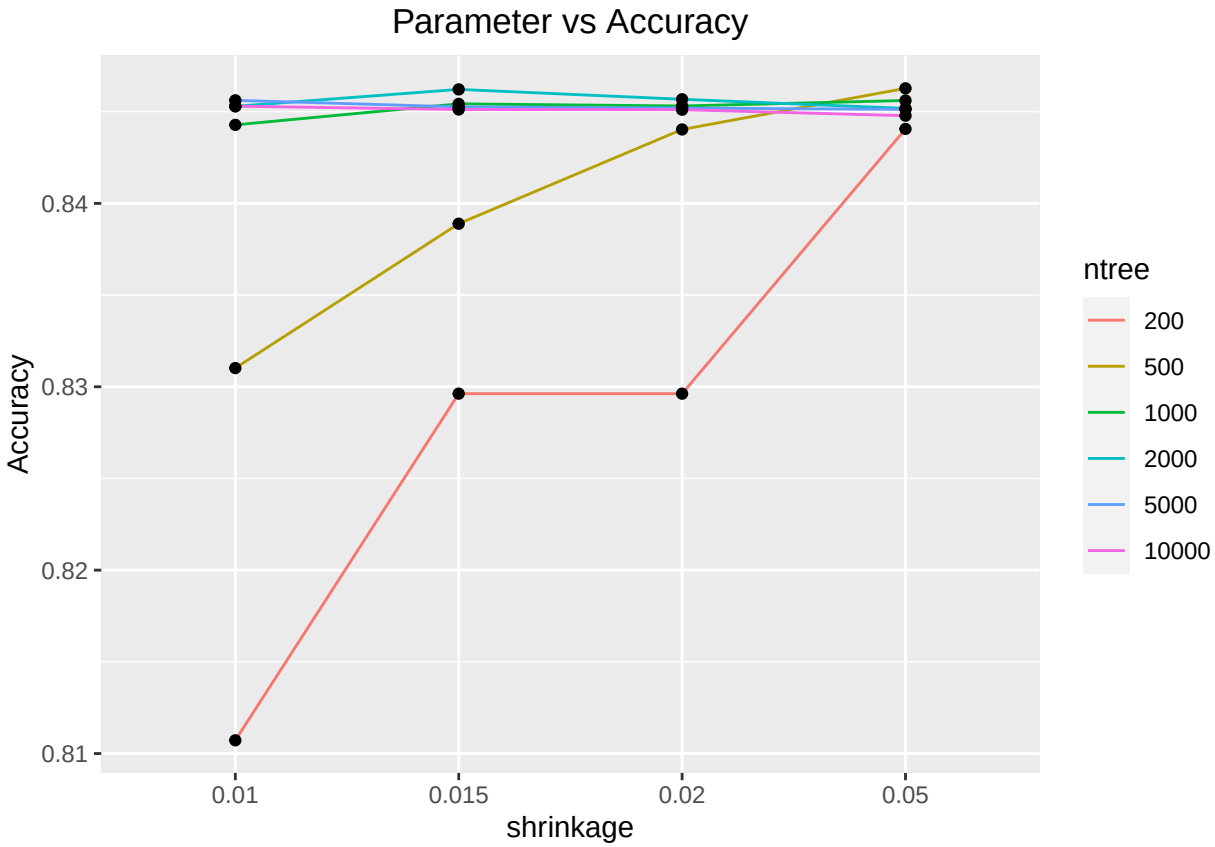
```
train_dep_temp <- train$Satisfaction
test_dep_temp <- test$Satisfaction

# Changing Factor Levels to Numeric for Bernoulli GBM
train$Satisfaction <- as.numeric(train_dep_temp)-1
test$Satisfaction <- as.numeric(test_dep_temp)-1
```

```
try_shrinkage_gbm = c(0.01, 0.015, 0.02, 0.05)
try_ntrees_gbm <- c(200, 500, 1000, 2000, 5000, 10000)

cv_acc_gbm <- NULL
cv_acc_gbm <- matrix(nrow = length(try_ntrees_gbm), ncol = length(try_shrinkage_gbm))
rownames(cv_acc_gbm) = try_ntrees_gbm
colnames(cv_acc_gbm) = try_shrinkage_gbm
```

```
# Note: this code will not be run for knitting purposes
# Results are presented in the next chunk
tic("GBM CV")
for (n in try_ntrees_gbm){
  acc.ave <- NULL; print(n)
    for (s in try_shrinkage_gbm){
    acc <- NULL; print(s);i=1
    for(fold in folds){
      print(i); i=i+1
      ## GBM
      mod_gbm = gbm(Satisfaction~.-id, data=train[-fold, names(train)!="Sat_temp"],
                    n.trees=n, shrinkage=s,
                    distribution="bernoulli", verbose=F)

      ## Error in the validation set
      pred = predict(mod_gbm, newdata=train[fold, ], type="response")
      # print(head(pred))
      pred = factor(ifelse(pred>0.5, 1, 0))
      # print(head(pred))

      acc = c(acc, confusionMatrix(pred, as.factor(train[fold, ]$Satisfaction),
                                   positive="1")$overall["Accuracy"])

      ## Freeing Memory
      rm(mod_gbm); gc()
    }
    acc.ave = c(acc.ave, mean(acc));
  }
  cv_acc_gbm[paste(n), ] = acc.ave
}
toc()
```

```
# GBM Cross Validation Results ~ time elapsed: 117.1442 mins
cv_acc_gbm["200", ] <- c(0.8107214,0.8296232,0.8296232,0.8440594)
cv_acc_gbm["500", ] <- c(0.8310187,0.8388913,0.8440306,0.8462730)
cv_acc_gbm["1000", ] <- c(0.8442808,0.8454261,0.8453106,0.8456090)
cv_acc_gbm["2000", ] <- c(0.8452914,0.8462153,0.8456763,0.8451759)
cv_acc_gbm["5000", ] <- c(0.8456186,0.8452625,0.8451951,0.8451278)
cv_acc_gbm["10000", ] <- c(0.845301,0.8451181,0.8451085,0.8447813)


cv_acc_gbm = melt(cv_acc_gbm)
cv_acc_gbm$Var1 = as.factor(cv_acc_gbm$Var1)
cv_acc_gbm$Var2 = as.factor(cv_acc_gbm$Var2)
```

```
ggplot(cv_acc_gbm, aes(x = Var2, y = value)) +
  geom_line(aes(color = Var1, group = Var1)) +
  geom_point()+
  ggtitle("Parameter vs Accuracy") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x = "shrinkage", y = "Accuracy", color = "ntree")
```



From the graph above, it is apparent that there does not exist the same pattern of parameters just as in the case of parameters in the random forest model. It can be observed that the accuracy converges when shrinkage=0.05 starting from ntree=500. At ntree = 500, the accuracy even exceeds that of models with greater ntree. The plot also shows that a lower learning rate value does not guarantee an increase in accuracy.

Since an ideal model would be a model with less complexity and high accuracy, the parameters used will be shrinkage=0.05 and ntree=500.

```
best_shrinkage_gbm = 0.05
best_ntree_gbm = 500
```

6.2 Modelling with Tuned Parameters

A model will now be created with optimized parameters obtained from the cross validation.

```
tic("GBM Best")
mod_gbm_best = gbm(Satisfaction~.-id, data=train,
                   n.trees=best_ntree_gbm, shrinkage=best_shrinkage_gbm,
                   distribution="bernoulli", verbose=F)
toc()
```

```
## GBM Best: 32.48 sec elapsed
```

Model's prediction on the train set is as follows.

```
yhat = predict(mod_gbm_best, newdata=train, type="response")
```

```
## Using 500 trees...
```

```
yhat = factor(ifelse(yhat>0.5, "Satisfied", "Neutral or Dissatisfied"))
cf_gbm = confusionMatrix(as.factor(yhat), train_dep_temp, positive="Satisfied")
cf_gbm$overall["Accuracy"]
```

```
##  Accuracy
## 0.8464174
```

6.3 Prediction and Model Evaluation

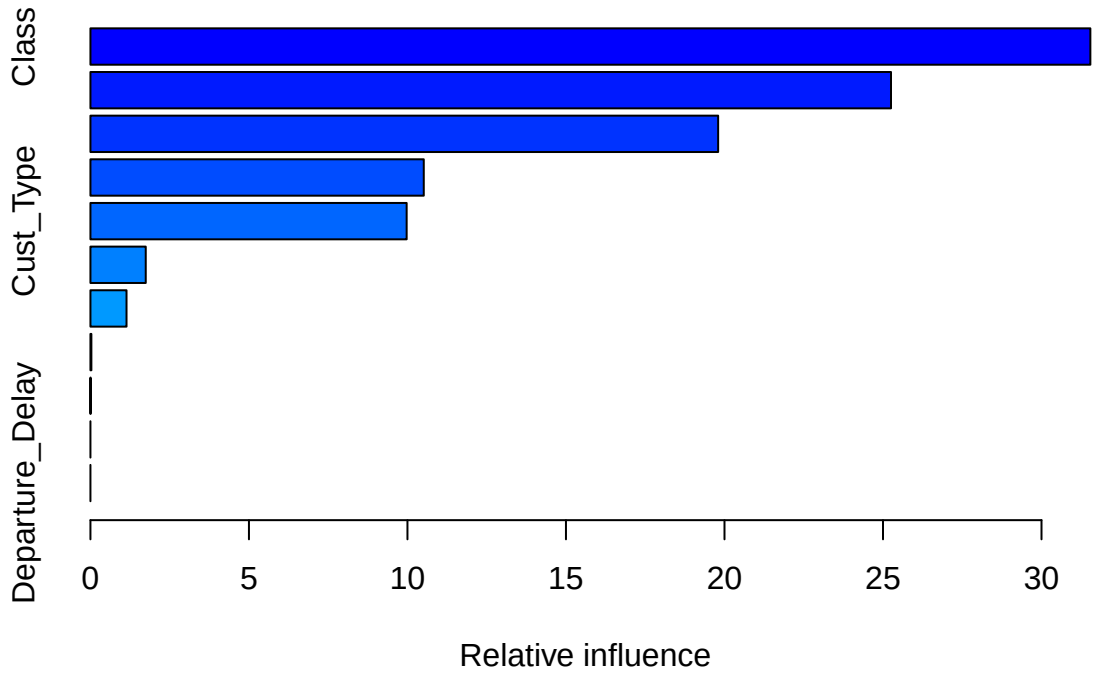
```
## Predicting the test set
pred = predict(mod_gbm_best, newdata=test, type="response")
```

```
## Using 500 trees...
```

```
pred = factor(ifelse(pred>0.5, "Satisfied", "Neutral or Dissatisfied"))
cf_gbm = confusionMatrix(as.factor(pred), test_dep_temp, positive="Satisfied")
cf_gbm$overall["Accuracy"]
```

```
##  Accuracy
## 0.8478537
```

```
(imp_gbm <- summary(mod_gbm_best))
```




```
##                              var     rel.inf
## Class                      Class 31.54110659
## Travel_Type          Travel_Type 25.25250227
## Seat                        Seat 19.79972127
## Inflight_Svc        Inflight_Svc 10.51458514
## Cust_Type              Cust_Type  9.97456587
## Arrival_Delay      Arrival_Delay  1.74460390
## Age                          Age  1.13769425
## Gate_Loc                Gate_Loc  0.02980892
## Flight_Dist          Flight_Dist  0.00541179
## Gender                    Gender  0.00000000
## Departure_Delay Departure_Delay  0.00000000
```

From the variable importance plot, the three main variables that have a high importance towards the dependent variable include Class, Travel_Type, and Seat. Additionally, flight distance, gender, and departure delay received the lowest variable importance. GBM's top variable importance results are similar to that of random forest's. However, the least important variables differ.

7 Conclusion

The metrics of the model will be compared in the conclusion.

```
eval_mod_rf <- c(cf_rf$overall["Accuracy"], cf_rf$byClass["Precision"],
                 cf_rf$byClass["Recall"], cf_rf$byClass["F1"])
eval_mod_gbm <- c(cf_gbm$overall["Accuracy"], cf_gbm$byClass["Precision"],
                  cf_gbm$byClass["Recall"], cf_gbm$byClass["F1"])
eval_df <- data.frame(Mod_RF = eval_mod_rf, Mod_GBM = eval_mod_gbm)
eval_df
```

```
##              Mod_RF   Mod_GBM
## Accuracy  0.8811935 0.8478537
## Precision 0.9142164 0.8454074
## Recall    0.8017723 0.7952149
## F1        0.8543103 0.8195434
```

From the results above, the random forest model received a higher score in both accuracy and F1 score in comparison to the gradient boosted model. Therefore, it can be inferred that the random forest model is more accurate than the GBM model and has better ability in detecting true positives in matters of precision and recall.

7.1 Insights

The two models created have performed quite well. It has also been shown the importance of the independent variables towards a customer's satisfaction. The flight seat and the passenger's class and means of travel highly impact the passenger's overall satisfaction, which are also according to the results in the data exploration. Therefore, airline companies should consider optimizing their flight seats for the passengers. Furthermore, this certain airline company should prioritize their target market on company executives, employees, and government officials who most likely travel abroad for business purposes by offering business class flights.

For further studies, modelling can be done with more computation power and RAM of more than 8 gigabytes to cross validate the model with more parameters such as depth of trees and more options in the number of tree generated. The outliers in the delay variable can be considered to be removed in future models.