# AVIATION MARKETING PROJECT (FINAL REPORT)

By: Omar Ossama

# Table of Contents

# 1. Executive Summary

In an industry that is aimed at consumers, such as the aviation industry, customer satisfaction is a very important factor for the company to be able to understand their customer needs and predict customer satisfaction.

Attracting new customers while retaining the loyal customers should be a priority for Falcon airlines as it translates to more revenue.

As such, it is essential for the company to be able to identify the factors that play a big role in customer satisfaction.

Understanding the customer needs should go a long way in enhancing the customer experience and finding innovative solutions to having a satisfied customer.

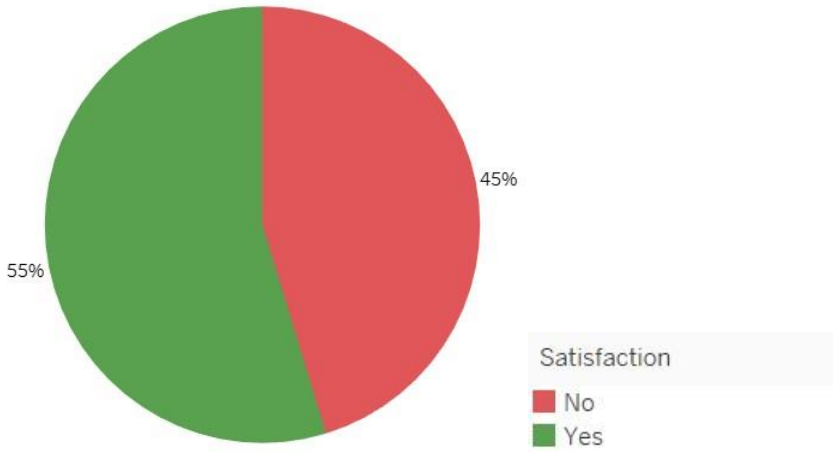
## 1.1. Methods and insights

### 1.1.1. EDA

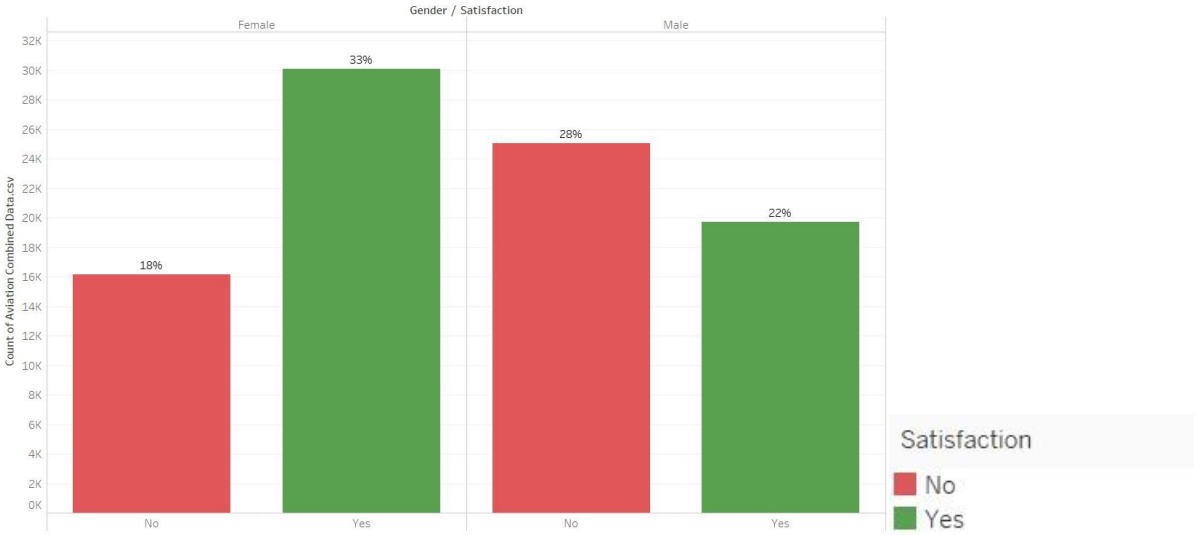
EDA was performed in order to find interesting insights and correlations with the dependent variable.

*Insights from the EDA*

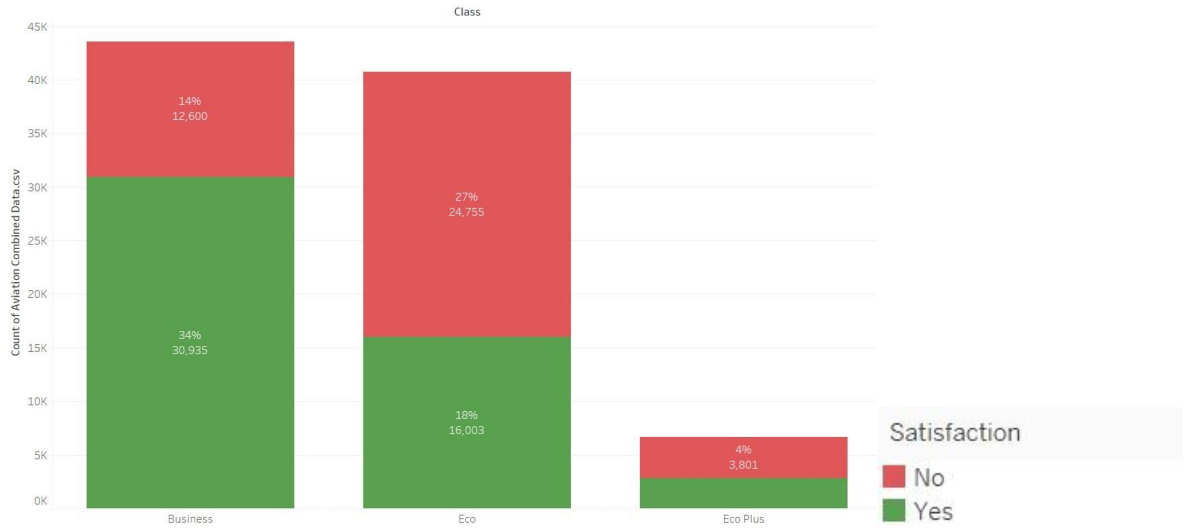
- Falcon airlines has a 55% satisfaction rate. Which when compared with other leading airlines is on the low side[1].



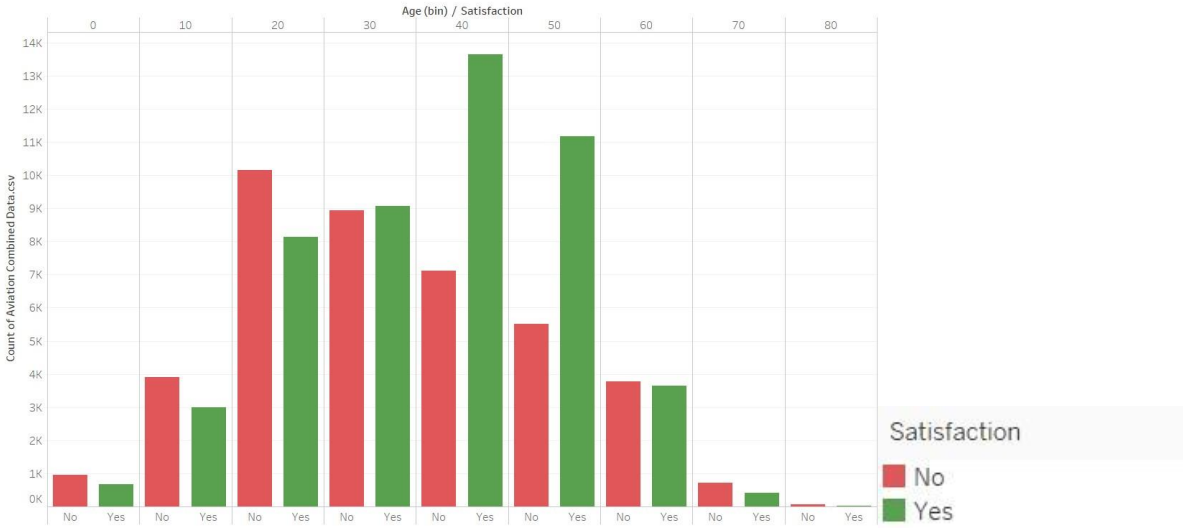
- Males were more likely to not be satisfied by the services provided by the airlines.
  Further investigation was done and it was found that males gave less average scores compared to females.




- Out of the 3 classes of tickets that the company has to offer, the Eco Plus class was significantly less used.
  It was also found that Eco Plus and Eco classes had a relatively low satisfaction rate, with 43% and 39% respectively.



- Customers of the age group 40-60 years old are the only age group that are more likely to be satisfied than not. While all the other age groups are more likely to neutral or not satisfied.

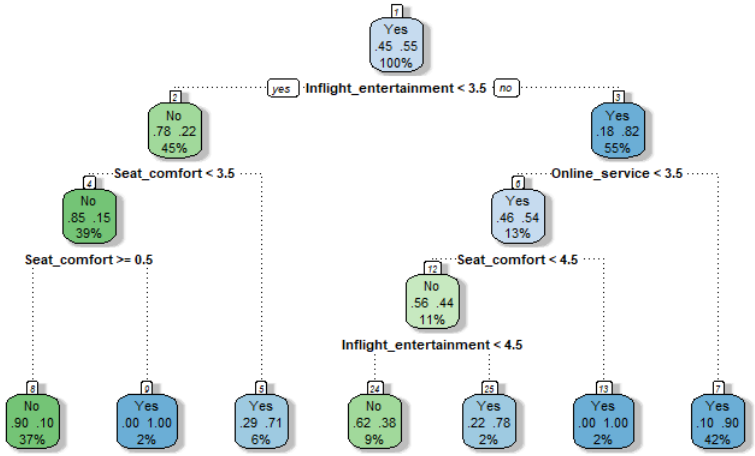



## 1.1.2. Predictive models

CART, Logistic Regression, Random Forest and XGB models were created to accurately predict the customer satisfaction variable. And also to find the variables that were the most impactful on a customer's decision on whether they were satisfied or not.

### *Insights from predictive models*

- Inflight entertainment, Seat comfort and Online services were the most impactful variables.
- Males are more likely to be neutral or not satisfied
- Customers that do not fly Business Class are more likely to be neutral or not satisfied
- Loyal customers usually do not give bad reviews on their trips



# 2. Data and final model selection

## 2.1. Data report

Data was collected using physical pamphlets given to the customers at the end of their flights. Also, using online surveys sent to the customers after arriving to their destinations.

There are two datasets which, when combined, contain 24 unique variables. 20 of which are categorical, while the other 4 are numerical.

All data variables are either a rating that a customer gave to the service they received during or the flight. Or personal information of the customer.

Dataset had a total of 41,791 missing values, which were treated using MICE package in R.

Of all the numerical variables, only the Age variable did not have outliers. Which were also treated by capping the values to the 95$^{th}$ percentile.

## 2.2. Final model selection

After looking at all the model metrics, it was found that the Random Forest was the best performer. So, it would best to use this model for the prediction.

Accuracies against test data for other models were found to be:

CART model: 87%

Logistic regression model: 83%

XGB model: 88%

### Random forest model metrics

ROC = 0.9898

#### Performance against train data

Accuracy = 99.97%
Sensitivity = 99.98%
Specificity = 99.96%

#### Performance against test data

Accuracy = 94.85%
Sensitivity = 95.48%
Specificity = 94.33%

#### Important variables

1. Inflight Entertainment
2. Seat Comfort
3. Online Services

# 3. Conclusion and recommendations

- The company should focus more the groups that were found to be more likely to not be satisfied by the airline. These groups include:
    1) Males
    2) Customers under 40 and over 60
    3) Non-Business Class users

- Maintaining and improving the Inflight and Online services should be an important and achievable goal for the company as it will translate to more satisfied customers and more profits for the company.

- While seat comfort is not a realistic variable to control. It should be taken into consideration if the company ever decides to buy a new airliner.

# 4. References

[1]: https://www.researchgate.net/publication/322913951_AN_ANALYSIS_OF_AIRLINES_CUSTOMER_SATISFACTION_BY_IMPROVING_CUSTOMER_SERVICE_PERFORMANCE

# 5. Appendix

## R Code

## Setting the working directory

```{r}
setwd ("C:/Users/omaro/Documents/DSBA/06-Capstone Project")
```

## Libraries to be used

```{r}
library(esquisse)     #Plotting tool
library(ggplot2)      #Plotting tool
library(gridExtra)    #Plotting tool
library(corrplot)     #Plot corelation between numerical varaibles
library(mice)         #Missing data treatment
library(caTools)      #Splitting the dataset
library(caret)        #Building confusion matrix and comparing models
library(rpart.plot)   #Tunning CART model
library(rpart)        #Evaluating CART model
library(rattle)       #Visulaizing CART model
library(randomForest) #Tunning RF model
```

## Importing the data

```{r}
aviation1 = read.csv("Aviation_Marketing Project-Survey data.csv", header = T, na.strings = c("","  "))
aviation2 = read.csv("Aviation_Marketing Project-Flight data.csv", header = T, na.strings = c("","  "))
```

## Editing column name and performing data join

```{r}
names(aviation1) [names(aviation1) == 'CustomerId'] <- 'CustomerID'

aviation = merge(x = aviation1, y = aviation2, by = "CustomerID", all = TRUE)
```

## Data summary

```{r}
dim(aviation)
summary(aviation)
str(aviation)
sum(is.na(aviation))
```

## Editting variable types

```{r}
colnames(aviation) = make.names(colnames(aviation))

aviation$Satisfaction = as.factor(aviation$Satisfaction)
aviation$Seat_comfort = as.factor(aviation$Seat_comfort)
aviation$Departure.Arrival.time_convenient = as.factor(aviation$Departure.Arrival.time_convenient)
aviation$Food_drink = as.factor(aviation$Food_drink)
```

```
aviation$Gate_location = as.factor(aviation$Gate_location)
aviation$Inflightwifi_service = as.factor(aviation$Inflightwifi_service)
aviation$Inflight_entertainment = as.factor(aviation$Inflight_entertainment)
aviation$Online_support = as.factor(aviation$Online_support)
aviation$Ease_of_Onlinebooking = as.factor(aviation$Ease_of_Onlinebooking)
aviation$Onboard_service = as.factor(aviation$Onboard_service)
aviation$Leg_room_service = as.factor(aviation$Leg_room_service)
aviation$Baggage_handling = as.factor(aviation$Baggage_handling)
aviation$Checkin_service = as.factor(aviation$Checkin_service)
aviation$Cleanliness = as.factor(aviation$Cleanliness)
aviation$Online_boarding = as.factor(aviation$Online_boarding)
aviation$Gender = as.factor(aviation$Gender)
aviation$CustomerType = as.factor(aviation$CustomerType)
aviation$TypeTravel = as.factor(aviation$TypeTravel)
aviation$Class = as.factor(aviation$Class)
aviation$Flight_Distance = as.numeric(aviation$Flight_Distance)
aviation$DepartureDelayin_Mins = as.numeric(aviation$DepartureDelayin_Mins)
aviation$ArrivalDelayin_Mins = as.numeric(aviation$ArrivalDelayin_Mins)
```

## Univarite analysis

```{r}
prop.table(table(aviation$Satisfaction))

ggplot(aviation) +
  aes(x = Satisfaction) +
```

```
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()



prop.table(table(aviation$Seat_comfort))

ggplot(aviation) +
  aes(x = Seat_comfort) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()



prop.table(table(aviation$Departure.Arrival.time_convenient))

ggplot(aviation) +
  aes(x = Departure.Arrival.time_convenient) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()



prop.table(table(aviation$Food_drink))

ggplot(aviation) +
  aes(x = Food_drink) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()



prop.table(table(aviation$Gate_location))
```

```
ggplot(aviation) +
  aes(x = Gate_location) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Inflightwifi_service))

ggplot(aviation) +
  aes(x = Inflightwifi_service) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Inflight_entertainment))

ggplot(aviation) +
  aes(x = Inflight_entertainment) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Online_support))

ggplot(aviation) +
  aes(x = Online_support) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()
```

```
prop.table(table(aviation$Ease_of_Onlinebooking))

ggplot(aviation) +
  aes(x = Ease_of_Onlinebooking) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Onboard_service))

ggplot(aviation) +
  aes(x = Onboard_service) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Leg_room_service))

ggplot(aviation) +
  aes(x = Leg_room_service) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Baggage_handling))

ggplot(aviation) +
```

```
  aes(x = Baggage_handling) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Checkin_service))

ggplot(aviation) +
  aes(x = Checkin_service) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Cleanliness))

ggplot(aviation) +
  aes(x = Cleanliness) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Online_boarding))

ggplot(aviation) +
  aes(x = Online_boarding) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()
```

```
prop.table(table(aviation$Gender))

ggplot(aviation) +
  aes(x = Gender) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$CustomerType))

ggplot(aviation) +
  aes(x = CustomerType) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$TypeTravel))

ggplot(aviation) +
  aes(x = TypeTravel) +
  geom_bar(fill = "#0c4c8a") +
  theme_minimal()


prop.table(table(aviation$Class))

ggplot(aviation) +
  aes(x = Class) +
  geom_bar(fill = "#0c4c8a") +
```

```
  theme_minimal()
```

```
## Creating function to combine histograms and boxplots
```{r}
plot_histogram_n_boxplot = function(variable, variableNameString, binw){
  h = ggplot(data = aviation, aes(x= variable))+
    labs(x = variableNameString,y ='Count')+
    geom_histogram(fill = 'green',col = 'red',binwidth = binw)+
    geom_vline(aes(xintercept=mean(variable)),
               color="black", linetype="dashed", size=0.5)
  b = ggplot(data = aviation, aes('',variable))+
    geom_boxplot(outlier.colour = 'red',col = 'red',outlier.shape = 19)+
    labs(x = '',y = variableNameString)+ coord_flip()
  grid.arrange(h,b,ncol = 2)
}
```

## Creating histograms and boxplots for the numerical variables
```{r}
plot_histogram_n_boxplot(aviation$Age, 'Age', 1)

plot_histogram_n_boxplot(aviation$Flight_Distance, 'Flight Distance', 1)

plot_histogram_n_boxplot(aviation$DepartureDelayin_Mins, 'Departure Delay in Mins', 1)

plot_histogram_n_boxplot(aviation$ArrivalDelayin_Mins, 'Arrival Delay in Mins', 1)
```
```

```

## Bivariate analysis for the dependent variable

```{r}

chisq.test(aviation$Satisfaction, aviation$Seat_comfort)
ggplot(aviation, aes(fill = Satisfaction, x = Seat_comfort)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Departure.Arrival.time_convenient)
ggplot(aviation, aes(fill = Satisfaction, x = Departure.Arrival.time_convenient)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Food_drink)
ggplot(aviation, aes(fill = Satisfaction, x = Food_drink)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Gate_location)
ggplot(aviation, aes(fill = Satisfaction, x = Gate_location)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Inflightwifi_service)
ggplot(aviation, aes(fill = Satisfaction, x = Inflightwifi_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Inflight_entertainment)
ggplot(aviation, aes(fill = Satisfaction, x = Inflight_entertainment)) +
  geom_bar(position="fill")

```
chisq.test(aviation$Satisfaction, aviation$Online_support)
ggplot(aviation, aes(fill = Satisfaction, x = Online_support)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Ease_of_Onlinebooking)
ggplot(aviation, aes(fill = Satisfaction, x = Ease_of_Onlinebooking)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Onboard_service)
ggplot(aviation, aes(fill = Satisfaction, x = Onboard_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Leg_room_service)
ggplot(aviation, aes(fill = Satisfaction, x = Leg_room_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Baggage_handling)
ggplot(aviation, aes(fill = Satisfaction, x = Baggage_handling)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Checkin_service)
ggplot(aviation, aes(fill = Satisfaction, x = Checkin_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Cleanliness)
ggplot(aviation, aes(fill = Satisfaction, x = Cleanliness)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Online_boarding)
```

```
ggplot(aviation, aes(fill = Satisfaction, x = Online_boarding)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Gender)
ggplot(aviation, aes(fill = Satisfaction, x = Gender)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$CustomerType)
ggplot(aviation, aes(fill = Satisfaction, x = CustomerType)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$TypeTravel)
ggplot(aviation, aes(fill = Satisfaction, x = TypeTravel)) +
  geom_bar(position="fill")

chisq.test(aviation$Satisfaction, aviation$Class)
ggplot(aviation, aes(fill = Satisfaction, x = Class)) +
  geom_bar(position="fill")



ggplot(aviation, aes(fill = Satisfaction, x = cut(Age, 5))) +
  geom_bar(position="fill") + labs(x = 'Age',y ='Count')

ggplot(aviation, aes(fill = Satisfaction, x = cut(Flight_Distance, 5))) +
  geom_bar(position="fill") + labs(x = 'Flight Distance',y ='Count')

ggplot(aviation, aes(fill = Satisfaction, x = cut(DepartureDelayin_Mins, 5))) +
  geom_bar(position="fill") + labs(x = 'Deparute Delay in Mins',y ='Count')
```

```
ggplot(aviation, aes(fill = Satisfaction, x = cut(ArrivalDelayin_Mins, 5))) +
  geom_bar(position="fill") + labs(x = 'Arrival Delay in Mins',y ='Count')
```

```
## Multivariate analysis for the numerical variables
```{r}
corrplot(cor(aviation[c(19,22:24)]),type="lower",method="number")
```

## Bivariate analysis for random variables
```{r}

chisq.test(aviation$Seat_comfort, aviation$Leg_room_service)
ggplot(aviation, aes(fill = Seat_comfort, x = Leg_room_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Food_drink, aviation$Onboard_service)
ggplot(aviation, aes(fill = Food_drink, x = Onboard_service)) +
  geom_bar(position="fill")

chisq.test(aviation$Ease_of_Onlinebooking, aviation$Online_support)
ggplot(aviation, aes(fill = Ease_of_Onlinebooking, x = Online_support)) +
  geom_bar(position="fill")

chisq.test(aviation$Checkin_service, aviation$Baggage_handling)
ggplot(aviation, aes(fill = Checkin_service, x = Baggage_handling)) +
  geom_bar(position="fill")
```
```

```

##Removing Customer ID variable

```{r}
aviation$CustomerID <- NULL
```

## Missing data treatment

```{r}
init.impute = mice(aviation, m=2, method = "pmm", seed = 1000)
aviation = complete(init.impute, 2)
sum(is.na(aviation))
```

##Outlier treatment

```{r}

quantile(aviation$Flight_Distance, probs = seq(0,1,0.05))
aviation$Flight_Distance[which(aviation$Flight_Distance > 3833)] <- 3833

quantile(aviation$DepartureDelayin_Mins, probs = seq(0,1,0.05))
aviation$DepartureDelayin_Mins[which(aviation$DepartureDelayin_Mins > 76)] <- 76

quantile(aviation$ArrivalDelayin_Mins, probs = seq(0,1,0.05))
aviation$ArrivalDelayin_Mins[which(aviation$ArrivalDelayin_Mins > 78)] <- 78

```
```

##Creating Inflight COmfort variable

```{r}
#Editing Seat Comfort
aviation$Seat_comfort <- as.character(aviation$Seat_comfort)

aviation$Seat_comfort[aviation$Seat_comfort=="excellent"] <- "5"
aviation$Seat_comfort[aviation$Seat_comfort=="good"] <- "4"
aviation$Seat_comfort[aviation$Seat_comfort=="acceptable"] <- "3"
aviation$Seat_comfort[aviation$Seat_comfort=="need improvement"] <- "2"
aviation$Seat_comfort[aviation$Seat_comfort=="poor"] <- "1"
aviation$Seat_comfort[aviation$Seat_comfort=="extremely poor"] <- "0"

aviation$Seat_comfort <- as.numeric(aviation$Seat_comfort)

#Editing Leg room
aviation$Leg_room_service <- as.character(aviation$Leg_room_service)

aviation$Leg_room_service[aviation$Leg_room_service=="excellent"] <- "5"
aviation$Leg_room_service[aviation$Leg_room_service=="good"] <- "4"
aviation$Leg_room_service[aviation$Leg_room_service=="acceptable"] <- "3"
aviation$Leg_room_service[aviation$Leg_room_service=="need improvement"] <- "2"
aviation$Leg_room_service[aviation$Leg_room_service=="poor"] <- "1"
aviation$Leg_room_service[aviation$Leg_room_service=="extremely poor"] <- "0"

aviation$Leg_room_service <- as.numeric(aviation$Leg_room_service)

#Creating new variable
aviation$Inflight_comfort = ((aviation$Seat_comfort)+(aviation$Leg_room_service))/2
```

##Creating Inflight Services variable

```{r}

#Editing Food & drink

aviation$Food_drink <- as.character(aviation$Food_drink)

aviation$Food_drink[aviation$Food_drink=="excellent"] <- "5"

aviation$Food_drink[aviation$Food_drink=="good"] <- "4"

aviation$Food_drink[aviation$Food_drink=="acceptable"] <- "3"

aviation$Food_drink[aviation$Food_drink=="need improvement"] <- "2"

aviation$Food_drink[aviation$Food_drink=="poor"] <- "1"

aviation$Food_drink[aviation$Food_drink=="extremely poor"] <- "0"

aviation$Food_drink <- as.numeric(aviation$Food_drink)

#Editing Inflight WiFi

aviation$Inflightwifi_service <- as.character(aviation$Inflightwifi_service)

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="excellent"] <- "5"

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="good"] <- "4"

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="acceptable"] <- "3"

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="need improvement"] <- "2"

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="poor"] <- "1"

aviation$Inflightwifi_service[aviation$Inflightwifi_service=="extremely poor"] <- "0"

aviation$Inflightwifi_service <- as.numeric(aviation$Inflightwifi_service)

#Editing Inflight Entertainment

aviation$Inflight_entertainment <- as.character(aviation$Inflight_entertainment)
```

```
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="excellent"] <- "5"
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="good"] <- "4"
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="acceptable"] <- "3"
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="need improvement"] <- "2"
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="poor"] <- "1"
aviation$Inflight_entertainment[aviation$Inflight_entertainment=="extremely poor"] <- "0"

aviation$Inflight_entertainment <- as.numeric(aviation$Inflight_entertainment)

#Editing Onboard Service
aviation$Onboard_service <- as.character(aviation$Onboard_service)

aviation$Onboard_service[aviation$Onboard_service=="excellent"] <- "5"
aviation$Onboard_service[aviation$Onboard_service=="good"] <- "4"
aviation$Onboard_service[aviation$Onboard_service=="acceptable"] <- "3"
aviation$Onboard_service[aviation$Onboard_service=="need improvement"] <- "2"
aviation$Onboard_service[aviation$Onboard_service=="poor"] <- "1"
aviation$Onboard_service[aviation$Onboard_service=="extremely poor"] <- "0"

aviation$Onboard_service <- as.numeric(aviation$Onboard_service)

#Editing Cleanliness
aviation$Cleanliness <- as.character(aviation$Cleanliness)

aviation$Cleanliness[aviation$Cleanliness=="excellent"] <- "5"
aviation$Cleanliness[aviation$Cleanliness=="good"] <- "4"
aviation$Cleanliness[aviation$Cleanliness=="acceptable"] <- "3"
aviation$Cleanliness[aviation$Cleanliness=="need improvement"] <- "2"
```

```
aviation$Cleanliness[aviation$Cleanliness=="poor"] <- "1"
aviation$Cleanliness[aviation$Cleanliness=="extremely poor"] <- "0"

aviation$Cleanliness <- as.numeric(aviation$Cleanliness)

#Creating new variable
aviation$Inflight_services = ((aviation$Food_drink)
                              +(aviation$Inflightwifi_service)
                              +(aviation$Inflight_entertainment)
                              +(aviation$Onboard_service)
                              +(aviation$Cleanliness))/5
```

##Creating online services variable

```{r}
#Editing online support
aviation$Online_support <- as.character(aviation$Online_support)

aviation$Online_support[aviation$Online_support=="excellent"] <- "5"
aviation$Online_support[aviation$Online_support=="good"] <- "4"
aviation$Online_support[aviation$Online_support=="acceptable"] <- "3"
aviation$Online_support[aviation$Online_support=="need improvement"] <- "2"
aviation$Online_support[aviation$Online_support=="poor"] <- "1"
aviation$Online_support[aviation$Online_support=="extremely poor"] <- "0"

aviation$Online_support <- as.numeric(aviation$Online_support)
```

```
#Editing Ease of online booking
aviation$Ease_of_Onlinebooking <- as.character(aviation$Ease_of_Onlinebooking)

aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="excellent"] <- "5"
aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="good"] <- "4"
aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="acceptable"] <- "3"
aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="need improvement"] <- "2"
aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="poor"] <- "1"
aviation$Ease_of_Onlinebooking[aviation$Ease_of_Onlinebooking=="extremely poor"] <- "0"

aviation$Ease_of_Onlinebooking <- as.numeric(aviation$Ease_of_Onlinebooking)

#Editing online boarding
aviation$Online_boarding <- as.character(aviation$Online_boarding)

aviation$Online_boarding[aviation$Online_boarding=="excellent"] <- "5"
aviation$Online_boarding[aviation$Online_boarding=="good"] <- "4"
aviation$Online_boarding[aviation$Online_boarding=="acceptable"] <- "3"
aviation$Online_boarding[aviation$Online_boarding=="need improvement"] <- "2"
aviation$Online_boarding[aviation$Online_boarding=="poor"] <- "1"
aviation$Online_boarding[aviation$Online_boarding=="extremely poor"] <- "0"

aviation$Online_boarding <- as.numeric(aviation$Online_boarding)

#Creating new variable
aviation$Online_service = ((aviation$Online_support)
                +(aviation$Online_boarding)
                +(aviation$Ease_of_Onlinebooking))/3
```

##Creating Pre/post flight services variable

```{r}

#Editing Departure and arrival time convinience

aviation$Departure.Arrival.time_convenient <-
as.character(aviation$Departure.Arrival.time_convenient)


aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="excellent"]
<- "5"

aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="good"] <-
"4"

aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="acceptable"
] <- "3"

aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="need
improvement"] <- "2"

aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="poor"] <-
"1"

aviation$Departure.Arrival.time_convenient[aviation$Departure.Arrival.time_convenient=="extremely
poor"] <- "0"


aviation$Departure.Arrival.time_convenient <- as.numeric(aviation$Departure.Arrival.time_convenient)


#Editing Gate location

aviation$Gate_location <- as.character(aviation$Gate_location)


aviation$Gate_location[aviation$Gate_location=="very convinient"] <- "5"

aviation$Gate_location[aviation$Gate_location=="Convinient"] <- "4"

aviation$Gate_location[aviation$Gate_location=="manageable"] <- "3"

aviation$Gate_location[aviation$Gate_location=="need improvement"] <- "2"

aviation$Gate_location[aviation$Gate_location=="Inconvinient"] <- "1"

aviation$Gate_location[aviation$Gate_location=="very inconvinient"] <- "0"
```

```
aviation$Gate_location <- as.numeric(aviation$Gate_location)

#Editing Baggage handling
aviation$Baggage_handling <- as.character(aviation$Baggage_handling)

aviation$Baggage_handling[aviation$Baggage_handling=="excellent"] <- "5"
aviation$Baggage_handling[aviation$Baggage_handling=="good"] <- "4"
aviation$Baggage_handling[aviation$Baggage_handling=="acceptable"] <- "3"
aviation$Baggage_handling[aviation$Baggage_handling=="need improvement"] <- "2"
aviation$Baggage_handling[aviation$Baggage_handling=="poor"] <- "1"
aviation$Baggage_handling[aviation$Baggage_handling=="extremely poor"] <- "0"

aviation$Baggage_handling <- as.numeric(aviation$Baggage_handling)

#Editing Check-in service
aviation$Checkin_service <- as.character(aviation$Checkin_service)

aviation$Checkin_service[aviation$Checkin_service=="excellent"] <- "5"
aviation$Checkin_service[aviation$Checkin_service=="good"] <- "4"
aviation$Checkin_service[aviation$Checkin_service=="acceptable"] <- "3"
aviation$Checkin_service[aviation$Checkin_service=="need improvement"] <- "2"
aviation$Checkin_service[aviation$Checkin_service=="poor"] <- "1"
aviation$Checkin_service[aviation$Checkin_service=="extremely poor"] <- "0"

aviation$Checkin_service <- as.numeric(aviation$Checkin_service)

#Creating new variable
aviation$pre.post.flight = ((aviation$Departure.Arrival.time_convenient)
```

```
                +(aviation$Gate_location)
                +(aviation$Baggage_handling)
                +(aviation$Checkin_service))/4

```

##Editiing the dependent variable levels

```{r}
aviation$Satisfaction <- as.character(aviation$Satisfaction)

aviation$Satisfaction[aviation$Satisfaction=="neutral or dissatisfied"] <- "No"
aviation$Satisfaction[aviation$Satisfaction=="satisfied"] <- "Yes"

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

##Checking for multicollinerity

```{r}

corrplot(cor(aviation[c(3,5:10)]),type="lower",method="number")

chisq.test(aviation$Gender, aviation$CustomerType)

chisq.test(aviation$Gender, aviation$TypeTravel)

chisq.test(aviation$Gender, aviation$Class)

chisq.test(aviation$CustomerType, aviation$TypeTravel)


```

```
chisq.test(aviation$CustomerType, aviation$Class)

chisq.test(aviation$TypeTravel, aviation$Class)

```

##Splitting data into train and test sets

```{r}
set.seed(1000)

sample = sample.split(aviation$Satisfaction,SplitRatio = 0.7)
train = subset(aviation,sample == TRUE)
test = subset(aviation,sample == FALSE)

dim(train)
dim(test)

prop.table(table(train$Satisfaction))
prop.table(table(test$Satisfaction))
```

##CART model

```{r}
r.ctrl = rpart.control(minsplit = 10, minbucket = 3, xval = 10)

cart_model <- rpart(formula = Satisfaction~., data = train, method = "class", control = r.ctrl)
cart_model
fancyRpartPlot(cart_model)
prp(cart_model)
```

```

##Pruning

```{r}
cart_model$cptable
```

No pruning required

##Variable importance

```{r}
cart_model$variable.importance
```

##CART model performance

```{r}
cart_train_pred <- predict(cart_model, train, type="class")

caret::confusionMatrix(cart_train_pred, train$Satisfaction)

cart_test_pred <- predict(cart_model, test, type = "class")

caret::confusionMatrix(cart_test_pred, test$Satisfaction)
```

Train:  Accuracy = 86.76%

Sensitivity = 86.39%

Specificity = 87.06%

Test:  Accuracy = 87.04%

Sensitivity = 86.94%
```

Specificity = 87.12%

##Training model parameters

```{r}
fitControl <- trainControl(
        method = 'repeatedcv',
        number = 4,
        repeats = 1,
        allowParallel = TRUE,
        classProbs = TRUE,
        summaryFunction=twoClassSummary
        )
```

##Building logistic regression model

```{r}
lr_model <- train(Satisfaction ~.
            -Inflight_comfort -Inflight_services -Online_service -pre.post.flight,
            data = train,
            method = "glm",
            family = "binomial",
            trControl = fitControl)

lr_model
summary(lr_model)
```

ROC = 0.9094

##Logistic regression model performance

```{r}
lr_train_pred <- predict(lr_model, newdata = train, type = "raw")

caret::confusionMatrix(lr_train_pred, train$Satisfaction)

lr_test_pred <- predict(lr_model, newdata = test, type = "raw")

caret::confusionMatrix(lr_test_pred, test$Satisfaction)
```

Train:  Accuracy = 83.59%

Sensitivity = 81.67%

Specificity = 85.19%

Test:  Accuracy = 83.43%

Sensitivity = 81.65%

Specificity = 84.90%

##Building random forest model

```{r}
rf_model <- train(Satisfaction ~ ., data = train,
                  method = "rf",
                  ntree = 30,
                  maxdepth = 5,
                  tuneLength = 10,
                  trControl = fitControl)

rf_model
```

ROC = 0.9898

##Random forest model performance

```{r}
rf_train_pred <- predict(rf_model, newdata = train, type = "raw")

caret::confusionMatrix(rf_train_pred, train$Satisfaction)

rf_test_pred <- predict(rf_model, newdata = test, type = "raw")

caret::confusionMatrix(rf_test_pred, test$Satisfaction)
```

Train:  Accuracy = 99.97%

Sensitivity = 99.98%

Specificity = 99.96%

Test:  Accuracy = 94.85%

Sensitivity = 95.48%

Specificity = 94.33%

##Variable importance

```{r}
varImp(rf_model, scale=FALSE)
```

##Extreme Gradient Boosting model

```{r}
cv_ctrl <- trainControl(method = "repeatedcv", repeats = 1,number = 3,
                        summaryFunction = twoClassSummary,
                        classProbs = TRUE,
```

```
                allowParallel=T)

  xgb_grid <- expand.grid(nrounds = 100,
                          eta = c(0.01),
                          max_depth = 4,
                          gamma = 0,
                          colsample_bytree = 1,
                          min_child_weight = 1,
                          subsample = 1
  )

  xgb_model <-train(Satisfaction~.,
                    data=train,
                    method="xgbTree",
                    trControl=cv_ctrl,
                    tuneGrid=xgb_grid,
                    verbose=T,
                    nthread = 2
  )

xgb_model
```
ROC = 0.9529

##XGB model performance

```{r}
xgb_train_pred <- predict(xgb_model, newdata = train, type = "raw")

caret::confusionMatrix(xgb_train_pred, train$Satisfaction)
```

```
xgb_test_pred <- predict(xgb_model, newdata = test, type = "raw")


caret::confusionMatrix(xgb_test_pred, test$Satisfaction)
```

Train:  Accuracy = 87.97%

Sensitivity = 87.21%

Specificity = 88.60%

Test:  Accuracy = 88.07%

Sensitivity = 87.77%

Specificity = 88.31%

##Variable importance

```{r}
varImp(xgb_model, scale=FALSE)
```