CHAPTER I
INTRODUCTION

1.1 PREDICTIVE ANALYTICS

The term predictive analytics refers to the use of statistics and modeling techniques to make predictions about future outcomes and performance. Predictive analytics looks at current and historical data patterns to determine if those patterns are likely to emerge again. This allows businesses and investors to adjust where they use their resources to take advantage of possible future events. Predictive analysis can also be used to improve operational efficiencies and reduce risk.

- Predictive analytics uses statistics and modelling techniques to determine future performance.
- Industries and disciplines, such as insurance and marketing, use predictive techniques to make important decisions.
- Predictive models help make weather forecasts, develop video games, translate voice-to-text messages, customer service decisions, and develop investment portfolios.
- People often confuse predictive analytics with machine learning even though the two are different disciplines.
- Types of predictive models include decision trees, regression, and neural networks.

1.2 OVERVIEW

Airline businesses around the world are decimated by Covid-19 as most international air travel has been grounded. Among the hardest hit might be Singapore Airlines, which operates zero domestic flight in its island home nation. In fact, some airlines such as Thai Airways have already filed for bankruptcy. Nonetheless, once the storm is over, demand for air travel is expected to surge as people rush back for overseas holidays. What factors are highly correlated to a satisfied (or dissatisfied) passenger? Can predict passenger satisfaction? To answer this business problem, a classification model is created from the flight satisfaction survey data to identify the critical factors that lead to customer satisfaction.

Chapter II

Gathering Data

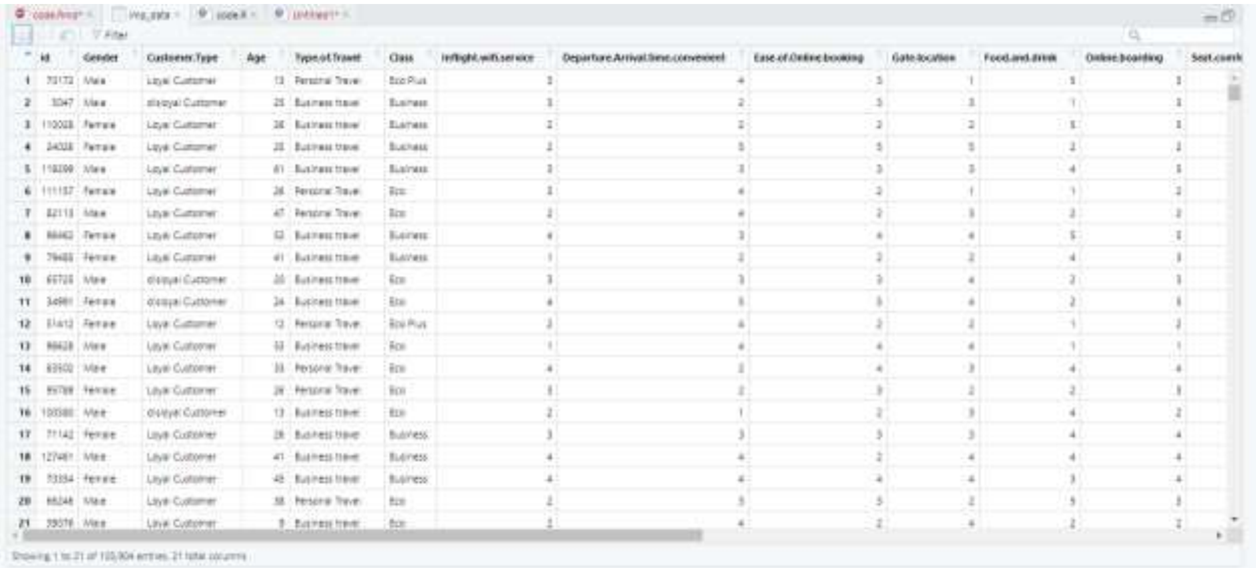
2.1 Data description:

This dataset contains an US airline passenger satisfaction survey.

The following things to be done.

1. Predicting passenger satisfaction
2. Finding factors are highly correlated to a satisfied (or dissatisfied) passenger

This dataset has 103904 rows and 21 columns.



This data frame contains the following columns:

ID

Unique identify number for each passenger

Gender

Gender of the passengers (Female, Male)

Customer Type

The customer type (Loyal customer, disloyal customer)

Age

The actual age of the passengers

Type of Travel

Purpose of the flight of the passengers (Personal Travel, Business Travel)

Class

Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Inflight wifi service

Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)

Departure/Arrival time convenient

Satisfaction level of Departure/Arrival time convenient

Ease of Online booking

Satisfaction level of online booking

Gate location

Satisfaction level of Gate location

Food and drink

Satisfaction level of Food and drink

Online boarding

Satisfaction level of online boarding

Seat comfort

Satisfaction level of Seat comfort

Inflight entertainment

Satisfaction level of inflight entertainment

On-board service

Satisfaction level of On-board service

Leg room service

Satisfaction level of Leg room service

Baggage handling

Satisfaction level of baggage handling

Check-in service

Satisfaction level of Check-in service

Inflight service

Satisfaction level of inflight service

Cleanliness

Satisfaction level of Cleanliness

Satisfaction

Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

The **"Satisfaction"** is the response variable. Other above variables are predictor variables.

2.2 Data understanding

After loading the data, it's a good practice to see if there are any missing values in the data.

```
```{r}
sum(is.na(imp_data))
```
```

```
[1] 0
```

The above output shows that the dataset has no missing values.

This module explains data understanding. This dataset consist of different columns. Each and every columns we should find the summary() function. This function is used to calculate the average value and determine the maximum, minimum of the column in a dataframe.

The following code has been executed in R studio to read the entire dataset named train.csv from the working directory .

```
setwd('C:/Users/PAVITRA/Desktop/PG Project')
getwd()
imp_data<-read.csv("train.csv")
View(imp_data)
dim(imp_data)
summary(imp_data)
summary(imp_data$id)
summary(imp_data$Gender)
summary(imp_data$Customer.Type)
summary(imp_data$Age)
summary(imp_data$Type.of.Travel)
summary(imp_data$Class)
```

ID

The expansion is IDENTIFICATION NUMBER. It is numeric variable. It is string of numerals which is unique for each and every individuals.

```
> summary(imp_data$id)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1   32534   64857   64924   97368  129880 
```

GENDER

It is categorical variable. The values categorized into the values are Male and Female.

```
> summary(imp_data$Gender)
   Length     Class      Mode 
   103904 character character 
> 
```

By using dplyr package, execute the count() command to know how many observations drop in these two ranges.

```
> imp_data %>% count(imp_data$Gender)
  imp_data$Gender     n
1          Female 52727
2            Male 51177
> |
```

CUSTOMER TYPE

It is a categorical variable. The values categorized into the values are Loyal Customer and Disloyal Customer. To market the service, the airlines works on understanding their customer's psyche, demographics and needs. Loyal Customer travel frequently and as they travel frequently with the same airline, the airline offers some benefits to them and also the miles. Disloyal Customer who not travel frequently may be price is the most discriminating factor as they travel frequently with different airline.

```
> summary(imp_data$Customer.Type)
   Length     Class      Mode 
   103904 character character 
> |
```

By using dplyr package, execute the count() command to know how many observations drop in these two ranges.

```
> imp_data %>% count(imp_data$Customer.Type)
  imp_data$Customer.Type     n
1      disloyal Customer 18981
2         Loyal Customer 84923
> |
```

AGE

It is a numeric variable. It is age of the passenger.

```
> summary(imp_data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   7.00   27.00   40.00   39.38   51.00   85.00 
> |
```

From the above output, it has been cleared that the Average age is 39.38. The maximum age is 85.00.The minimum age is 7.00. The below R code explains the range of the column frequency for using count() function.

```
Console   Terminal   Jobs
C:/Users/PAVITRA/Desktop/PG Project/
> imp_data %>% count(imp_data$Age)
   imp_data$Age    n
1             7  562
2             8  640
3             9  692
4            10  683
5            11  678
6            12  635
7            13  633
8            14  707
9            15  818
10           16  899
11           17  984
12           18  978
13           19  904
14           20 1520
15           21 1507
16           22 2351
17           23 2346
18           24 2278
19           25 2798
20           26 2289
21           27 2186
22           28 1707
23           29 1932
24           30 2030
25           31 1607
26           32 1570
27           33 1851
28           34 1753
29           35 1923
30           36 2319
31           37 2316
32           38 2321
33           39 2969
34           40 2574
35           41 2456
36           42 2457
```

The below code presents the result of count command applied on the variable age. Greater than or equal to condition is used for this data to collect the record. Totally 52518 records are observed while the age is greater than 39 .

```
> imp_data %>% count(imp_data$Age>39)
  imp_data$Age > 39     n
1             FALSE 51386
2              TRUE 52518
>
```

TYPE OF TRAVEL

It is a categorical variable. The values are categorized into Personal Travel and Business Travel. It represents the travel type flied by the passenger.

```
> summary(imp_data$Type.of.Travel)
   Length     Class      Mode 
   103904 character character 
>
```

By using dplyr package, execute the count() command to know how many observations drop in these two ranges.

```
> imp_data %>% count(imp_data$Type.of.Travel)
  imp_data$Type.of.Travel     n
1         Business travel 71655
2         Personal Travel 32249
>
```

CLASS

It is a categorical variable. The value are categorized into Eco Plus, Eco and Business. It is the passenger’s choice for which purpose they are travelling.

```
> summary(imp_data$Class)
   Length     Class      Mode 
   103904 character character 
>
```

By using dplyr package, execute the count() command to know how many observations drop in these three ranges.

```
> imp_data %>% count(imp_data$Class)
  imp_data$Class     n
1       Business 49665
2            Eco 46745
3       Eco Plus  7494
>
```

INFLIGHT WIFI SERVICE

It is a categorical variable. The values are categorized within the range 0 to 5. It represents the satisfaction level of passenger about wifi service.

```
> summary(imp_data$Inflight.wifi.service)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    2.00    3.00    2.73    4.00    5.00 
>
```

From the above output, it has been cleared that the Average value is 2.73. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Inflight.wifi.service)
  imp_data$Inflight.wifi.service     n
1                              0  3103
2                              1 17840
3                              2 25830
4                              3 25868
5                              4 19794
6                              5 11469
>
```

DEPARTURE ARRIVAL TIME CONVENIENT

It is a categorical variable. The values are categorized within the range 0 to 5. It represents the satisfaction level of passenger about convenient time of departure and arrival.

```
> summary(imp_data$Departure.Arrival.time.convenient)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    2.00    3.00    3.06    4.00    5.00 
>
```

From the above output, it has been cleared that the Average value is 3.06. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Departure.Arrival.time.convenient)
  imp_data$Departure.Arrival.time.convenient     n
1                                          0  5300
2                                          1 15498
3                                          2 17191
4                                          3 17966
5                                          4 25546
6                                          5 22403
>
```

EASE OF ONLINE BOOKING

It is a categorical variable. The values are categorized within the range 0 to 5. It represents the satisfaction level of passenger about comfortable in the time of online booking.

```
> summary(imp_data$Ease.of.Online.booking)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   3.000   2.757   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 2.757. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Ease.of.Online.booking)
  imp_data$Ease.of.Online.booking     n
1                               0  4487
2                               1 17525
3                               2 24021
4                               3 24449
5                               4 19571
6                               5 13851
> 
```

GATE LOCATION

It is a categorical variable. The values are categorized within the range 0 to 5. It represents the satisfaction level of passenger about the location where they board to the aircraft. Gates generally have seats, a gate to enter the runway, jet bridge (for passengers to get into the aircraft) and the boarding desk.

```
> summary(imp_data$Gate.location)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   3.000   2.977   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 2.977. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Gate.location)
  imp_data$Gate.location     n
1                      0     1
2                      1 17562
3                      2 19459
4                      3 28577
5                      4 24426
6                      5 13879
> 
```

FOOD AND DRINK

It is a categorical variable. The values are categorized within the range 0 to 5. It represents the satisfaction level of passenger about food and drink facilities. It’s now common in coach and economy classes for flight attendants to offer passengers sealed individual snacks and a limited selection of canned beverages. Instead of multicourse meals, in some cases, offering a pre-packaged boxed meal

```
> summary(imp_data$Food.and.drink)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   3.000   3.202   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.202. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Food.and.drink)
  imp_data$Food.and.drink     n
1                       0   107
2                       1 12837
3                       2 21988
4                       3 22300
5                       4 24359
6                       5 22313
> 
```

ONLINE BOARDING

It is a categorical variable. The values are categorized within the range 0 to 5. It is the entry of passengers onto a vehicle, usually in public transportation. The term is used in rail and air transport. A boarding pass is a document provided by an airline during check-in, giving a passenger permission to board the airplane for a particular flight. It is available in online.

```
> summary(imp_data$Online.boarding)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    2.00    3.00    3.25    4.00    5.00 
> 
```

From the above output, it has been cleared that the Average value is 3.25. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Online.boarding)
  imp_data$Online.boarding     n
1                        0  2428
2                        1 10692
3                        2 17505
4                        3 21804
5                        4 30762
6                        5 20713
> 
```

SEAT COMFORT

It is a categorical variable. The values are categorized within the range 0 to 5. If a piece of furniture or an item of clothing is comfortable, it makes user feel physically relaxed when user use it, for example because it is soft. People probably familiar with the rules that require a passenger who is too large to fit into a standard seat to buy a second seat next to them. What passenger might not know, however is that many airlines allow any passenger to buy an extra seat, called a comfort seat.

```
> summary(imp_data$Seat.comfort)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   4.000   3.439   5.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.439. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Seat.comfort)
  imp_data$Seat.comfort     n
1                     0     1
2                     1 12075
3                     2 14897
4                     3 18696
5                     4 31765
6                     5 26470
> 
```

INFLIGHT ENTERTAINMENT

It is a categorical variable. The values are categorized within the range 0 to 5. In-flight entertainment (IFE) refers to the entertainment available to aircraft passengers during a flight. Moving map systems, Audio entertainment, Video entertainment, Personal televisions, Inflight movies, Closed captioning(for deaf), Inflight games are the varieties of Inflight entertainment. Emirates wins the 2021 award for the World's Best Airline for Inflight Entertainment, ahead of Singapore Airlines in 2nd position and Qatar Airways in 3rd place.

```
> summary(imp_data$Inflight.entertainment)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   4.000   3.358   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.358. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Inflight.entertainment)
  imp_data$Inflight.entertainment     n
1                               0    14
2                               1 12478
3                               2 17637
4                               3 19139
5                               4 29423
6                               5 25213
> 
```

ONBOARD SERVICE

It is a categorical variable. The values are categorized within the range 0 to 5. The word available or situated on a ship, aircraft, or other vehicle.

```
> summary(imp_data$On.board.service)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   4.000   3.382   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.382. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
  0.000   2.000   4.000   3.382   4.000   5.000
> imp_data %>% count(imp_data$On.board.service)
  imp_data$On.board.service     n
1                         0     3
2                         1 11872
3                         2 14681
4                         3 22833
5                         4 30867
6                         5 23648
> 
```

LEG ROOM SERVICE

It is a categorical variable. The values are categorized within the range 0 to 5. It is the distance between a point on one seat and the same point on the seat in front of it.

```
6                         5 23648
> summary(imp_data$Leg.room.service)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   2.000   4.000   3.351   4.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.351. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
  0.000   2.000   4.000   3.351   4.000   5.000
> imp_data %>% count(imp_data$Leg.room.service)
  imp_data$Leg.room.service     n
1                         0   472
2                         1 10353
3                         2 19525
4                         3 20098
5                         4 28789
6                         5 24667
> 
```

BAGGAGE HANDLING

It is a categorical variable. The values are categorized within the range 1 to 5. It is about to provide immediate assistance to customers whose baggage is mishandled by reuniting customers with their belongings.

```
> summary(imp_data$Baggage.handling)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   3.000   4.000   3.632   5.000   5.000 
> 
```

From the above output, it has been cleared that the Average value is 3.632. The maximum value is 5.The minimum value is 1. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Baggage.handling)
  imp_data$Baggage.handling     n
1                         1  7237
2                         2 11521
3                         3 20632
4                         4 37383
5                         5 27131
> |
```

CHECKIN SERVICE

It is a categorical variable. The values are categorized within the range 0 to 5. It is the process in which the passenger, upon arrival at the airport, hands over any baggage that they don't want or are not allowed to carry inside the aircraft's cabin.

```
> summary(imp_data$Checkin.service)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   3.000   3.000   3.304   4.000   5.000 
> |
```

From the above output, it has been cleared that the Average value is 3.304. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
  0.000   3.000   3.000   3.304   4.000   5.000 
> imp_data %>% count(imp_data$Checkin.service)
  imp_data$Checkin.service     n
1                        0     1
2                        1 12890
3                        2 12893
4                        3 28446
5                        4 29055
6                        5 20619
> |
```

INFLIGHT SERVICE

It is a categorical variable. The values are categorized within the range 0 to 5. It includes not only food, beverages and duty free shopping, but also the provision of entertainment services and internet access via Wifi.

```
> summary(imp_data$Inflight.service)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    3.00    4.00    3.64    5.00    5.00 
```

From the above output, it has been cleared that the Average value is 3.64. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Inflight.service)
  imp_data$Inflight.service     n
1                         0     3
2                         1  7084
3                         2 11457
4                         3 20299
5                         4 37945
6                         5 27116
>
```

CLEANLINESS

It is a categorical variable. The values are categorized within the range 0 to 5. It is the quality or state of being clean. The practice of keeping the flights clean which is the necessary thing.

```
> summary(imp_data$Cleanliness)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   3.000   3.286   4.000   5.000
```

From the above output, it has been cleared that the Average value is 3.286. The maximum value is 5.The minimum value is 0. The below R code explains the range of the column frequency for using count() function.

```
> imp_data %>% count(imp_data$Cleanliness)
  imp_data$Cleanliness     n
1                    0    12
2                    1 13318
3                    2 16132
4                    3 24574
5                    4 27179
6                    5 22689
>
```

SATISFACTION

It is a categorical variable. The values are categorized into Satisfied and Neutral or Dissatisfied. It is the response variable. It is the value to be predict with the remaining predictors.

```
> summary(imp_data$satisfaction)
   Length     Class      Mode
   103904 character character
>
```

By using dplyr package, execute the count() command to know how many observations drop in these two ranges.

```
> imp_data %>% count(imp_data$satisfaction)
    imp_data$satisfaction     n
1 neutral or dissatisfied 58879
2               satisfied 45025
>
```

This section examines the nature of all variables available in the given dataset and the values, its count and range in a deep way using R studio.

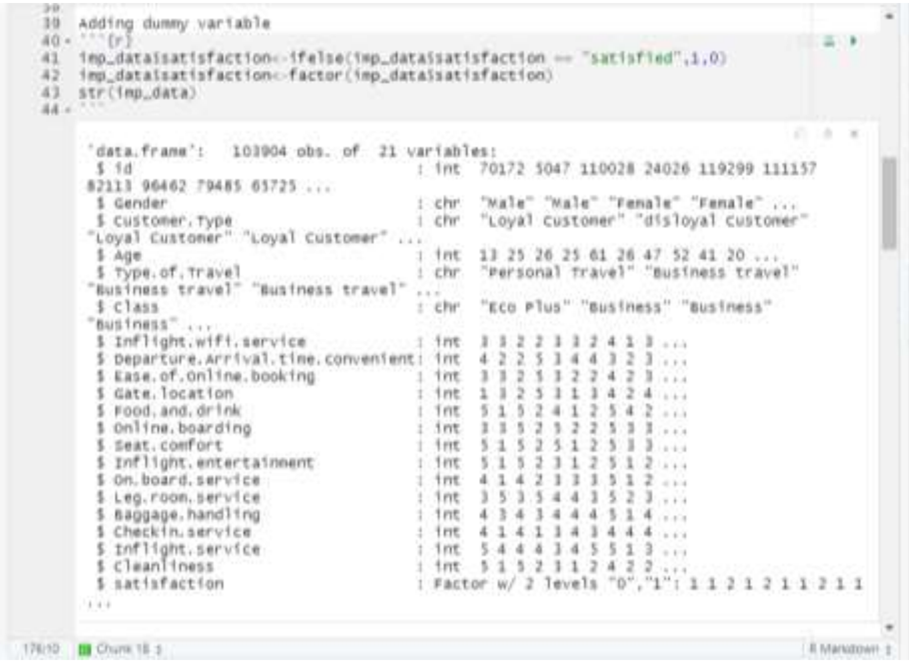
CHAPTER III

Data Preparation

3.1 Adding Dummy Variable

In classification models, encoding all of the independent variables as dummy variables allows easy interpretation and calculation of the odds ratios, and increases the stability and significance of the coefficients. If the response variable have a variable like Yes, No, it obviously doesn't make sense to assign values and interpret that as meaning that a yes is somehow three times as no. The solution is to use dummy variables - variables with only two values, zero and one. It does make sense to create a variable called "Yes" and interpret it as meaning that something assigned a 1 on this variable is Yes and something with an 0 is No.

In the existing dataset, the response variable have the values as “Satisfied” and “Neutral or Dissatisfied”. It can easily interpret if it converts into dummy variables 0 and 1.



```
Adding dummy variable
```{r}
imp_data$satisfaction<-ifelse(imp_data$satisfaction == "satisfied",1,0)
imp_data$satisfaction<-factor(imp_data$satisfaction)
str(imp_data)
```

'data.frame':   103904 obs. of 21 variables:
 $ id   : int 70172 5047 110028 24026 119299 111157
82113 96462 79485 65725 ...
 $ Gender   : chr "Male" "Male" "Female" "Female" ...
 $ Customer.Type   : chr "Loyal Customer" "disloyal Customer"
"Loyal Customer" "Loyal Customer" ...
 $ Age   : int 13 25 26 25 61 26 47 52 41 20 ...
 $ Type.of.Travel   : chr "Personal Travel" "Business travel"
"Business travel" "Business travel" ...
 $ Class   : chr "Eco Plus" "Business" "Business"
"Business" ...
 $ Inflight.wifi.service   : int 3 3 2 2 3 3 2 4 1 3 ...
 $ Departure.Arrival.time.convenient: int 4 2 2 5 3 4 4 3 2 3 ...
 $ Ease.of.Online.booking   : int 3 3 2 5 3 2 2 4 2 3 ...
 $ Gate.location   : int 1 3 2 5 3 1 3 4 2 4 ...
 $ Food.and.drink   : int 5 1 5 2 4 1 2 5 4 2 ...
 $ Online.boarding   : int 3 3 5 2 5 2 2 5 3 3 ...
 $ Seat.comfort   : int 5 1 5 2 5 1 2 5 3 3 ...
 $ Inflight.entertainment   : int 5 1 5 2 3 1 2 5 1 2 ...
 $ On.board.service   : int 4 1 4 2 3 3 3 5 1 2 ...
 $ Leg.room.service   : int 3 5 3 5 4 4 3 5 2 3 ...
 $ Baggage.handling   : int 4 3 4 3 4 4 4 5 1 4 ...
 $ Checkin.service   : int 4 1 4 1 3 4 3 4 4 4 ...
 $ Inflight.service   : int 5 4 4 4 3 4 5 5 1 3 ...
 $ Cleanliness   : int 5 1 5 2 3 1 2 4 2 2 ...
 $ satisfaction   : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 2 1 1
```

The above code assigns 1 for Satisfied and 0 for Neutral or Dissatisfied.

This section prepares the data by identifying and handling the outliers. It helps the dataset for further activity.

CHAPTER IV

EXPLORATORY DATA ANALYSIS

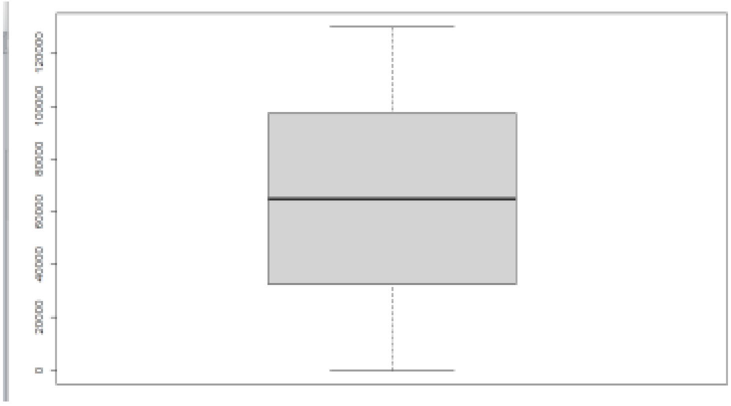
Exploratory data analysis (EDA) is used to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It can also help to determine if the statistical techniques that are considering for data analysis are appropriate. Summary() function helps to see the summary of all the variables and a raw information about the values in a single view.

```
       id             Gender          Customer.Type           Age       
 Min.   :     1   Length:103904      Length:103904      Min.   : 7.00  
 1st Qu.: 32534   Class :character   Class :character   1st Qu.:27.00  
 Median : 64857   Mode  :character   Mode  :character   Median :40.00  
 Mean   : 64924                                         Mean   :39.38  
 3rd Qu.: 97368                                         3rd Qu.:51.00  
 Max.   :129880                                         Max.   :85.00  
 Type.of.Travel        Class           Inflight.wifi.service
 Length:103904      Length:103904      Min.   :0.00         
 Class :character   Class :character   1st Qu.:2.00         
 Mode  :character   Mode  :character   Median :3.00         
                                       Mean   :2.73         
                                       3rd Qu.:4.00         
                                       Max.   :5.00         
 Departure.Arrival.time.convenient Ease.of.Online.booking Gate.location  
 Min.   :0.00                      Min.   :0.000          Min.   :0.000  
 1st Qu.:2.00                      1st Qu.:2.000          1st Qu.:2.000  
 Median :3.00                      Median :3.000          Median :3.000  
 Mean   :3.06                      Mean   :2.757          Mean   :2.977  
 3rd Qu.:4.00                      3rd Qu.:4.000          3rd Qu.:4.000  
 Max.   :5.00                      Max.   :5.000          Max.   :5.000  
 Food.and.drink  Online.boarding  Seat.comfort   Inflight.entertainment
 Min.   :0.000   Min.   :0.00    Min.   :0.000   Min.   :0.000         
 1st Qu.:2.000   1st Qu.:2.00    1st Qu.:2.000   1st Qu.:2.000         
 Median :3.000   Median :3.00    Median :4.000   Median :4.000         
 Mean   :3.202   Mean   :3.25    Mean   :3.439   Mean   :3.358         
 3rd Qu.:4.000   3rd Qu.:4.00    3rd Qu.:5.000   3rd Qu.:4.000         
 Max.   :5.000   Max.   :5.00    Max.   :5.000   Max.   :5.000         
 On.board.service Leg.room.service Baggage.handling Checkin.service
 Min.   :0.000    Min.   :0.000    Min.   :1.000    Min.   :0.000  
 1st Qu.:2.000    1st Qu.:2.000    1st Qu.:3.000    1st Qu.:3.000  
 Median :4.000    Median :4.000    Median :4.000    Median :3.000  
 Mean   :3.382    Mean   :3.351    Mean   :3.632    Mean   :3.304  
 3rd Qu.:4.000    3rd Qu.:4.000    3rd Qu.:5.000    3rd Qu.:4.000  
 Max.   :5.000    Max.   :5.000    Max.   :5.000    Max.   :5.000  
```

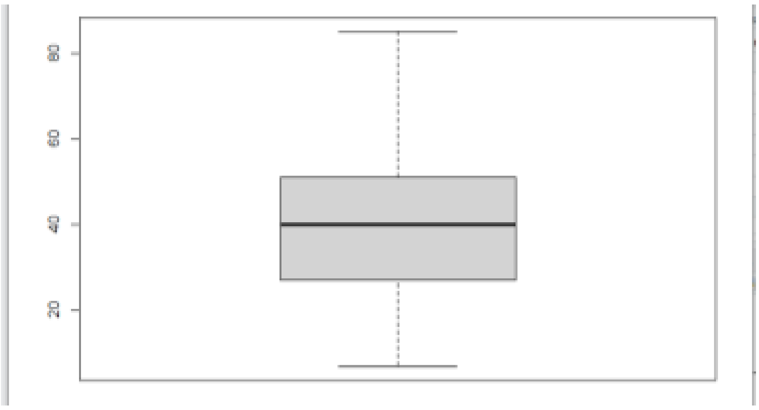
4.1 Outliers

Outliers are extreme values that fall a long way outside of the other observations. No matter how careful during data collection, every data scientists has felt the frustration of finding outliers. It may occur due to the variability in the data, or due to experimental error/human error. Techniques of detecting outliers are Boxplots, Z-score, Inter Quartile Range(IQR). In this dataset, Boxplot technique is used to detect the outliers only the data is numeric.

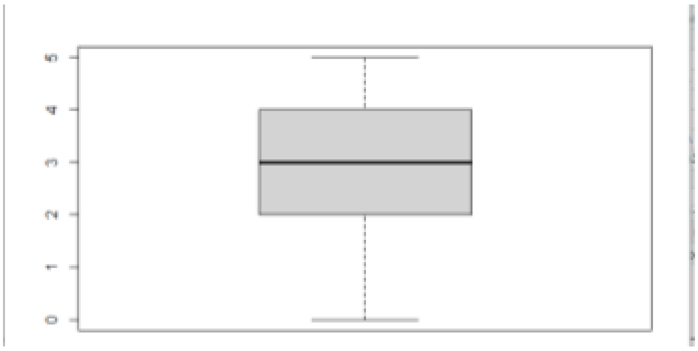
boxplot(imp_data$id)



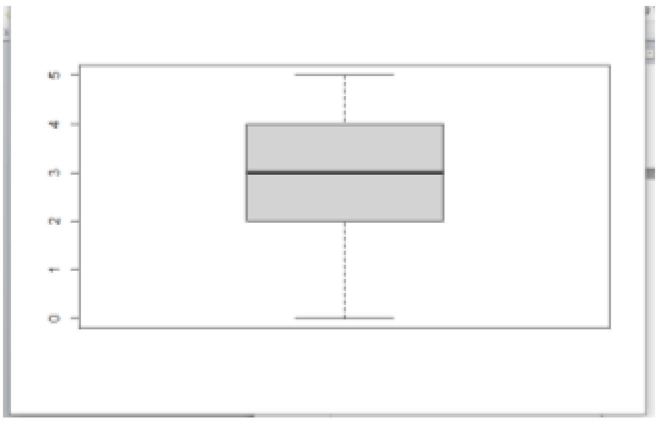
boxplot(imp_data$Age)



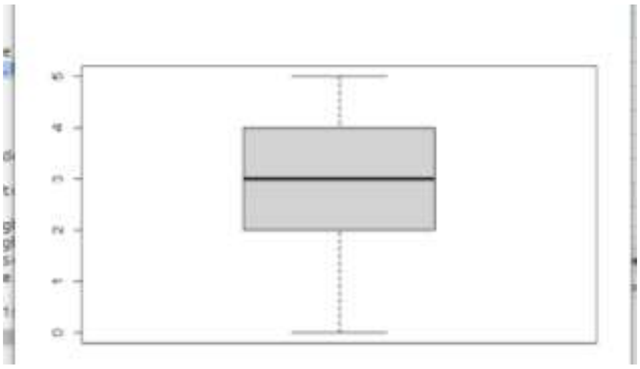
boxplot(imp_data$Inflight.wifi.service)



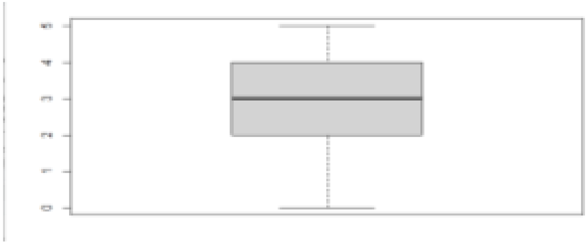
boxplot(imp_data$Departure.Arrival.time.convenient)



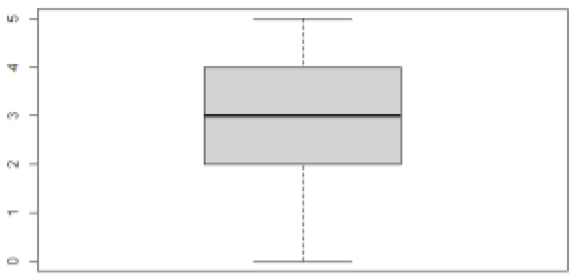
boxplot(imp_data$Ease.of.Online.booking)



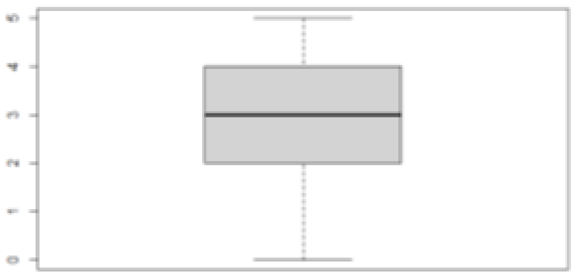
boxplot(imp_data$Gate.location)



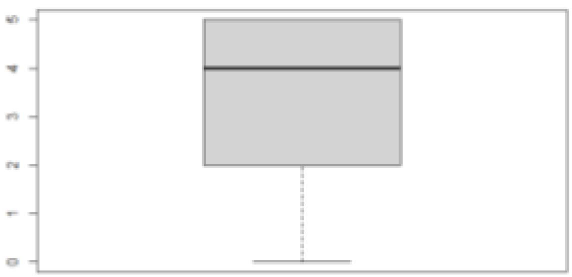
boxplot(imp_data$Food.and.drink)



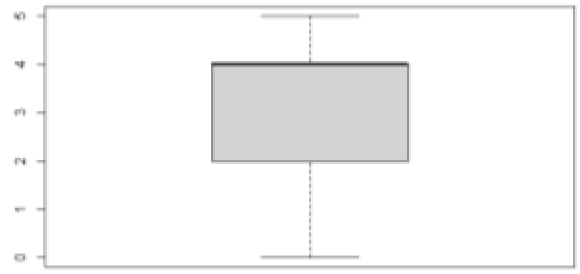
boxplot(imp_data$Online.boarding)



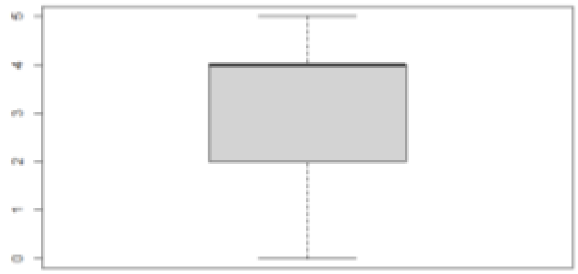
boxplot(imp_data$Seat.comfort)



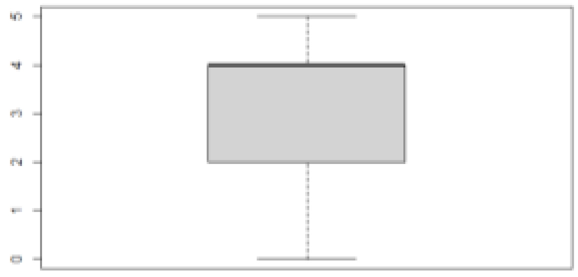
```
boxplot(imp_data$Inflight.entertainment)
```



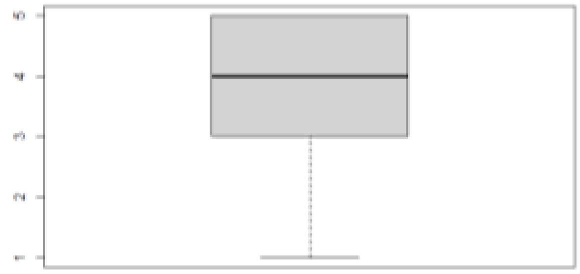
```
boxplot(imp_data$On.board.service)
```



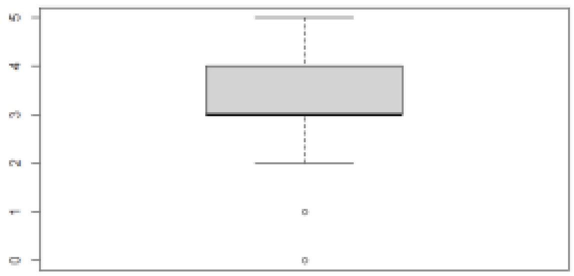
```
boxplot(imp_data$Leg.room.service)
```



```
boxplot(imp_data$Baggage.handling)
```



```
boxplot(imp_data$Checkin.service)
```



The above diagram contains outliers in the values of 0 and 1. Removing outliers is not advisable because it affects the result. So here, replacing the outlier variables by median value.

```
summary(imp_data$Checkin.service)
lowfence<-3.000-1.5*IQR(imp_data$Checkin.service)
lowfence
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   3.000   3.000   3.304   4.000   5.000 
[1] 1.5
```

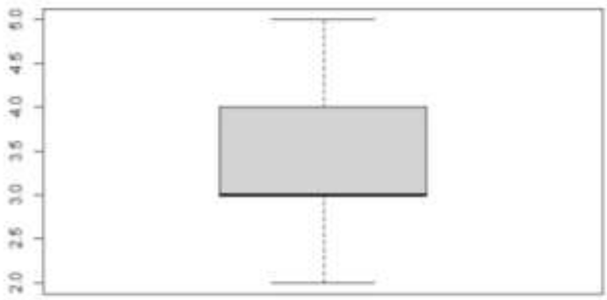
From the above result, 3 is the median value and the value 1.5 to be replace which the value less than 1.5 are considering as outliers.

```
imp_data$Checkin.service<-replace(imp_data$Checkin.service,imp_data$Checkin.service<1.5,median(imp_data$Checkin.service))
summary(imp_data$Checkin.service)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  2.000   3.000   3.000   3.552   4.000   5.000 
```

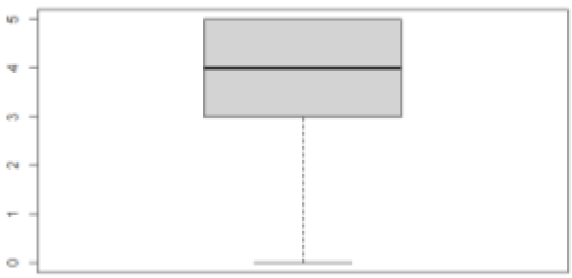
From the above result, the outliers are replaced by median values. The changes can be seen in summary().

boxplot(imp_data$Checkin.service)

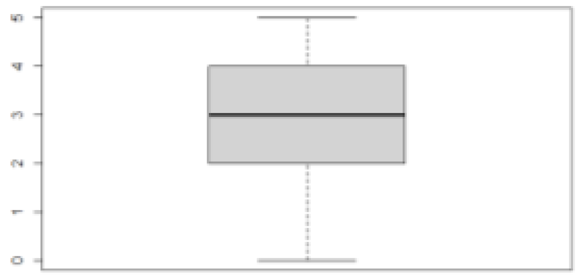


The above plot is the boxplot for checkin service after replacing outliers by median values.

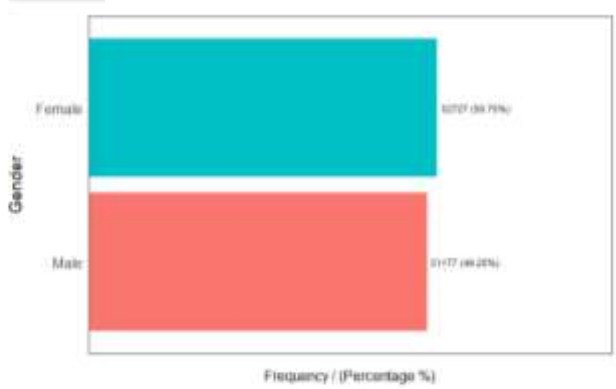
boxplot(imp_data$Inflight.service)

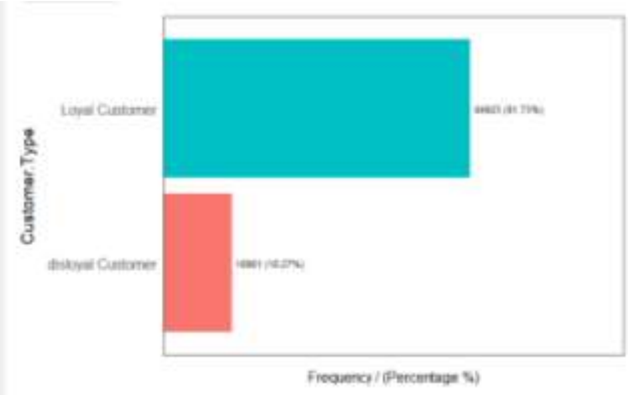


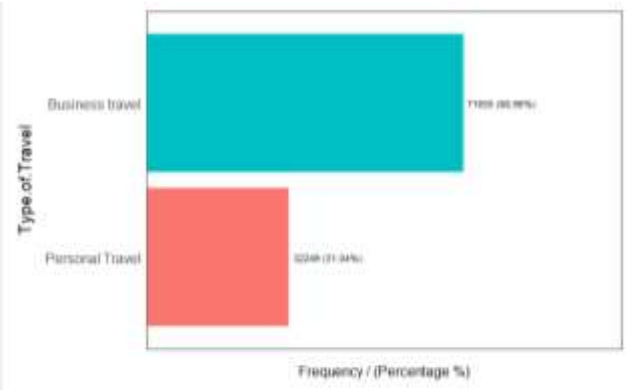
boxplot(imp_data$Cleanliness)

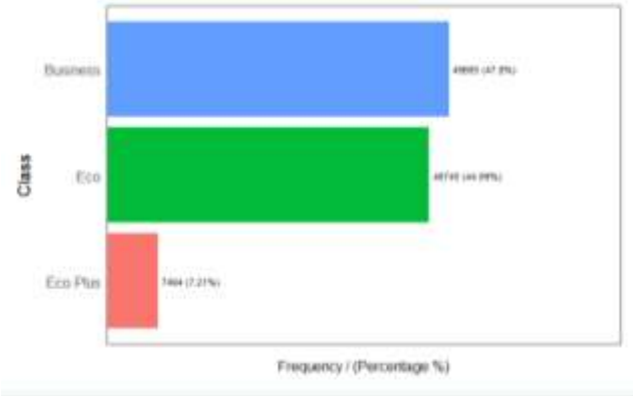


To analyze categorical variables, freq() function to be used in the package of Hmisc and funModeling

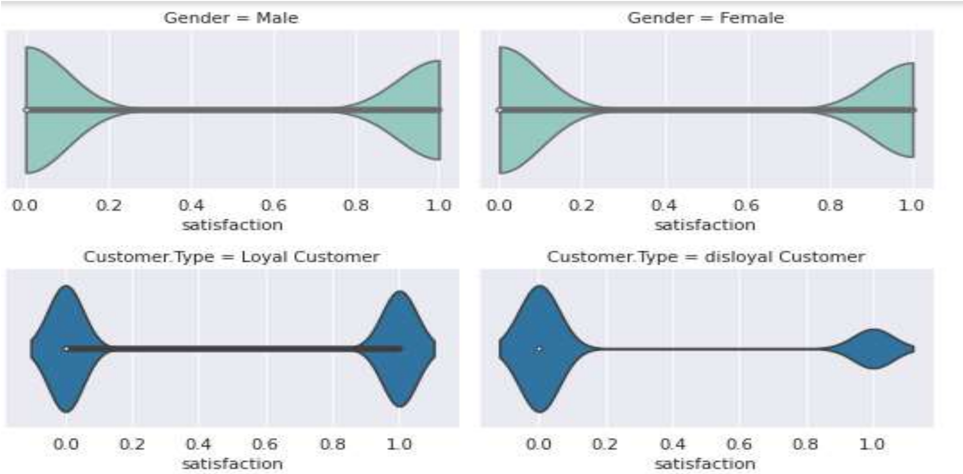


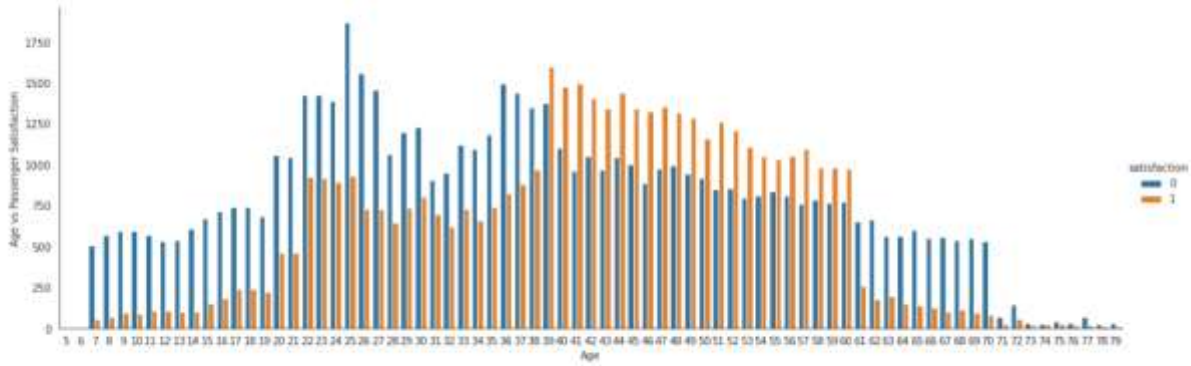


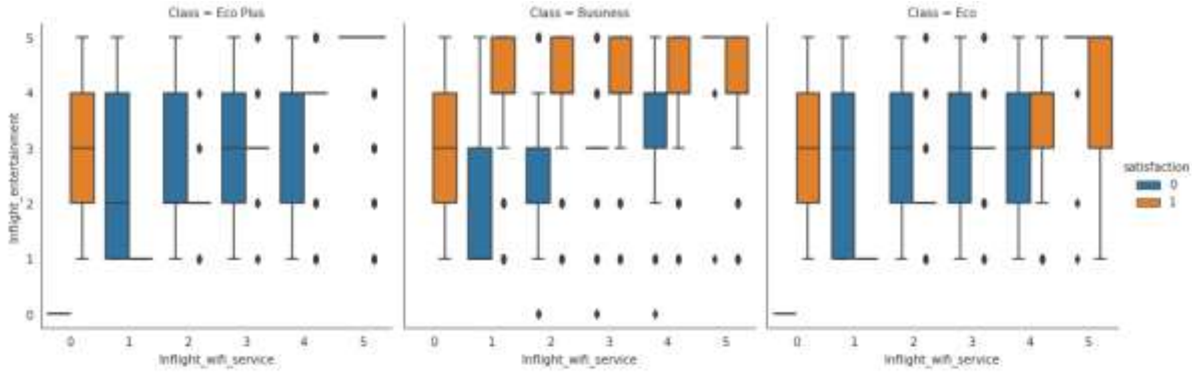


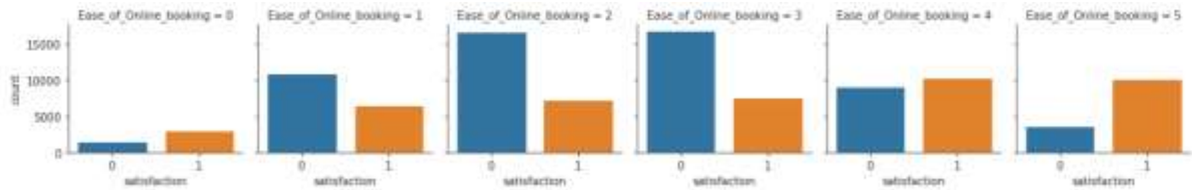


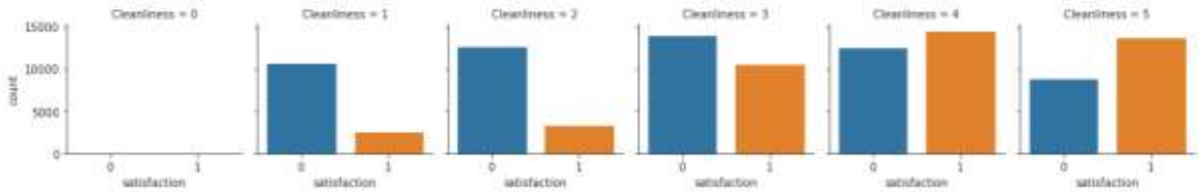
4.2 Bivariate and Multivariate Analysis

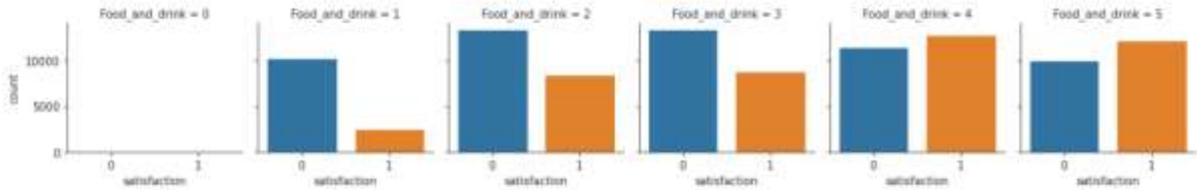


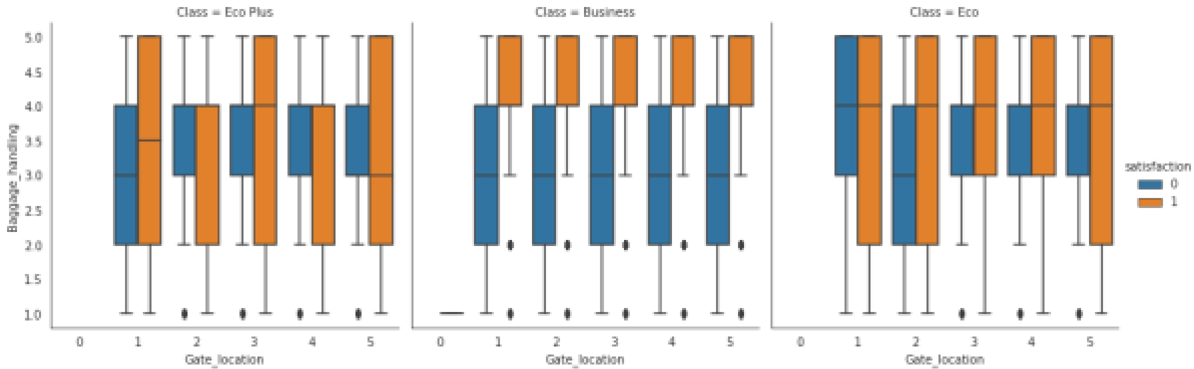


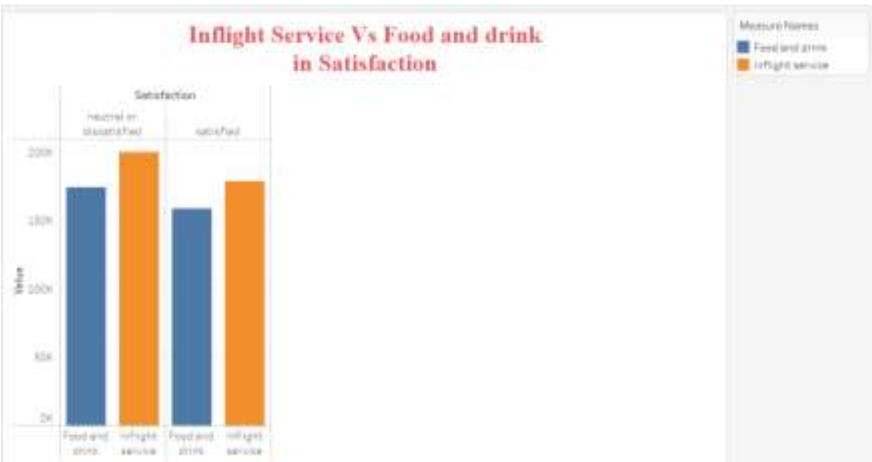


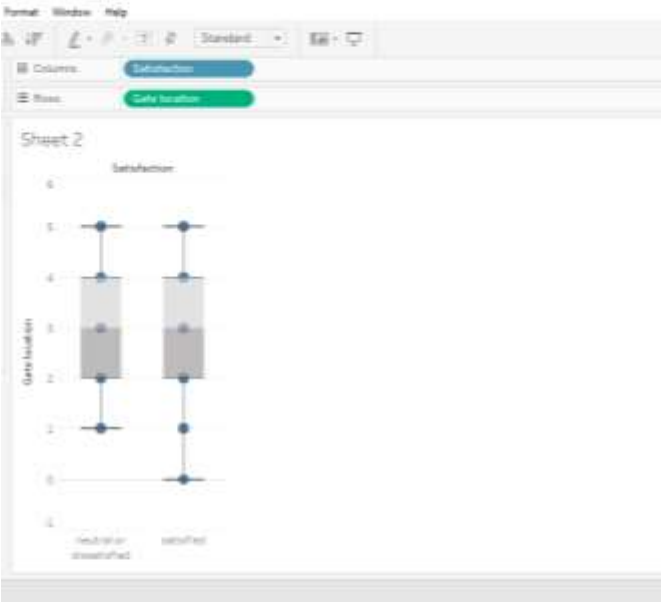

Ease_of_Online_booking = 0
Ease_of_Online_booking = 1
Ease_of_Online_booking = 2
Ease_of_Online_booking = 3
Ease_of_Online_booking = 4
Ease_of_Online_booking = 5
count
satisfaction


Cleanliness = 0
Cleanliness = 1
Cleanliness = 2
Cleanliness = 3
Cleanliness = 4
Cleanliness = 5
count
satisfaction

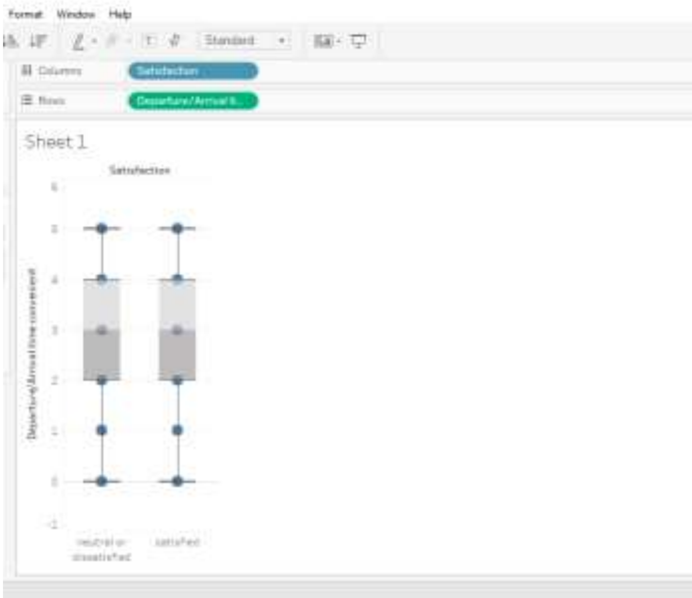

Food_and_drink = 0
Food_and_drink = 1
Food_and_drink = 2
Food_and_drink = 3
Food_and_drink = 4
Food_and_drink = 5
count
satisfaction


Class = Eco Plus
Class = Business
Class = Eco
Baggage_handling
Gate_location
satisfaction
0
1


Inflight Service Vs Food and drink
in Satisfaction
Satisfaction
Value
Measure Names



Format
Window
Help
Standard
Columns
Satisfaction
Rows
Gate location
Sheet 2
Satisfaction
Gate location



Format
Window
Help
Standard
Columns
Satisfaction
Rows
Sheet 1
Satisfaction
satisfied

The above plot is the correlation matrix which shows the variables correlated with each other. Here, values highlighted with light orange color are considered as highly correlated.

CHAPTER V
MODEL BUILDING

5.1 Algorithm

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. Logistic regression is easier to implement, interpret, and very efficient to train.

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line. Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. It uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes. It is a generative model. It does quite well when the training data doesn't contain all possibilities so it can be very good with low amounts of data. It handles both continuous and discrete data. It is highly scalable with the number of predictors and data points. It is fast and can be used to make real-time predictions.

5.2 Training and test dataset

This project with the passenger satisfaction dataset. The goal of the dataset is to classify whether the passenger satisfied or not on different independent variables. Split the dataset into training set and testing set before the model building. The 75% data will be split into training set and 25% data will be split into testing set.

````
Training and Testing data
```{r}
set.seed(1)
train_data<-sample(1:nrow(imp_data),nrow(imp_data)*0.75)
test_data<-imp_data[-train_data,]
names(test_data)
dim(test_data)
test_data1<-test_data[,-c(21)]
names(test_data1)
```
````

```
 [1] "id"                          "Gender"
 [3] "Customer.Type"               "Age"
 [5] "Type.of.Travel"              "Class"
 [7] "Inflight.wifi.service"       "Departure.Arrival.time.convenient"
 [9] "Ease.of.Online.booking"      "Gate.location"
[11] "Food.and.drink"              "Online.boarding"
[13] "Seat.comfort"                "Inflight.entertainment"
[15] "On.board.service"            "Leg.room.service"
[17] "Baggage.handling"            "Checkin.service"
[19] "Inflight.service"            "Cleanliness"
[21] "satisfaction"
[1] 25976    21
```

5.3 Model

Once the dataset was split into training and test dataset, build a model with training dataset. The following R code has been implemented the different models to classify the target variable Satisfaction. The following models built only with the variables which highly correlated.

Logistic Regression

```
set.seed(1)
model1 <- glm(satisfaction~-id+Inflight.wifi.service+Ease.of.Online.booking+Food.and.drink+Seat.comfort+
  Inflight.entertainment+Cleanliness+Baggage.handling+Inflight.service,data = imp_data,subset =
train_data,family="binomial")
summary(model1)
```

```
Call:
glm(formula = satisfaction ~ -id + Inflight.wifi.service + Ease.of.Online.booking +
    Food.and.drink + Seat.comfort + Inflight.entertainment +
    Cleanliness + Baggage.handling + Inflight.service, family = "binomial",
    data = imp_data, subset = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2051  -0.8823  -0.4310   0.9112   3.1259

Coefficients:
                        Estimate Std. Error  z value Pr(>|z|)
(Intercept)            -5.268712   0.049447 -106.553   <2e-16 ***
Inflight.wifi.service   0.412812   0.009698   42.566   <2e-16 ***
Ease.of.Online.booking -0.002741   0.009074   -0.302    0.763
Food.and.drink         -0.200537   0.009403  -21.326   <2e-16 ***
Seat.comfort            0.426136   0.009247   46.085   <2e-16 ***
Inflight.entertainment  0.346604   0.011370   30.483   <2e-16 ***
Cleanliness             0.119716   0.010510   11.391   <2e-16 ***
Baggage.handling        0.205616   0.009376   21.929   <2e-16 ***
Inflight.service        0.175649   0.009679   18.148   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

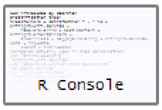
(Dispersion parameter for binomial family taken to be 1)

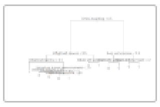
    Null deviance: 106660  on 77927  degrees of freedom
Residual deviance:  85039  on 77919  degrees of freedom
AIC: 85057
```

From the above output, relationship between the variables can be seen. The variables which having three stars are highly correlated. Except ease of online booking, all other variables are correlated. So for the below models the variable “ease of online booking” not used to build.

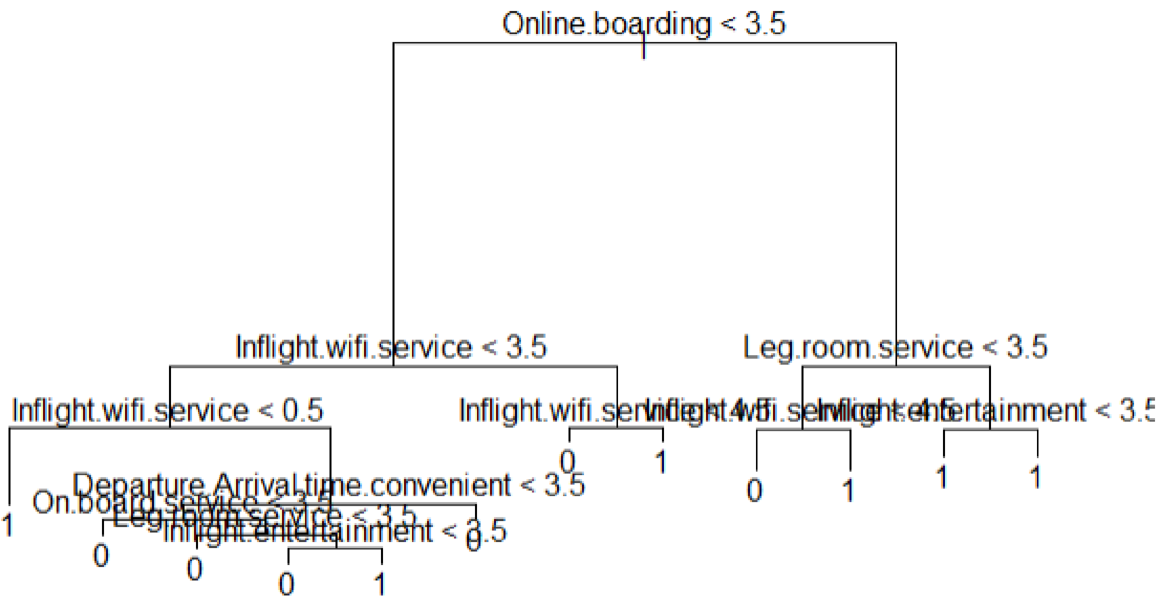
Decision tree

~~~~
Tree
```{r}
library(tree)
model2 <- tree(satisfaction~.-id+Inflight.wifi.service+Food.and.drink+Seat.comfort+
Inflight.entertainment+Cleanliness+Baggage.handling+Inflight.service,data = imp_data,subset = train_data)
summary(model2)
plot(model2)
text(model2,pretty=0)
```
~~~~



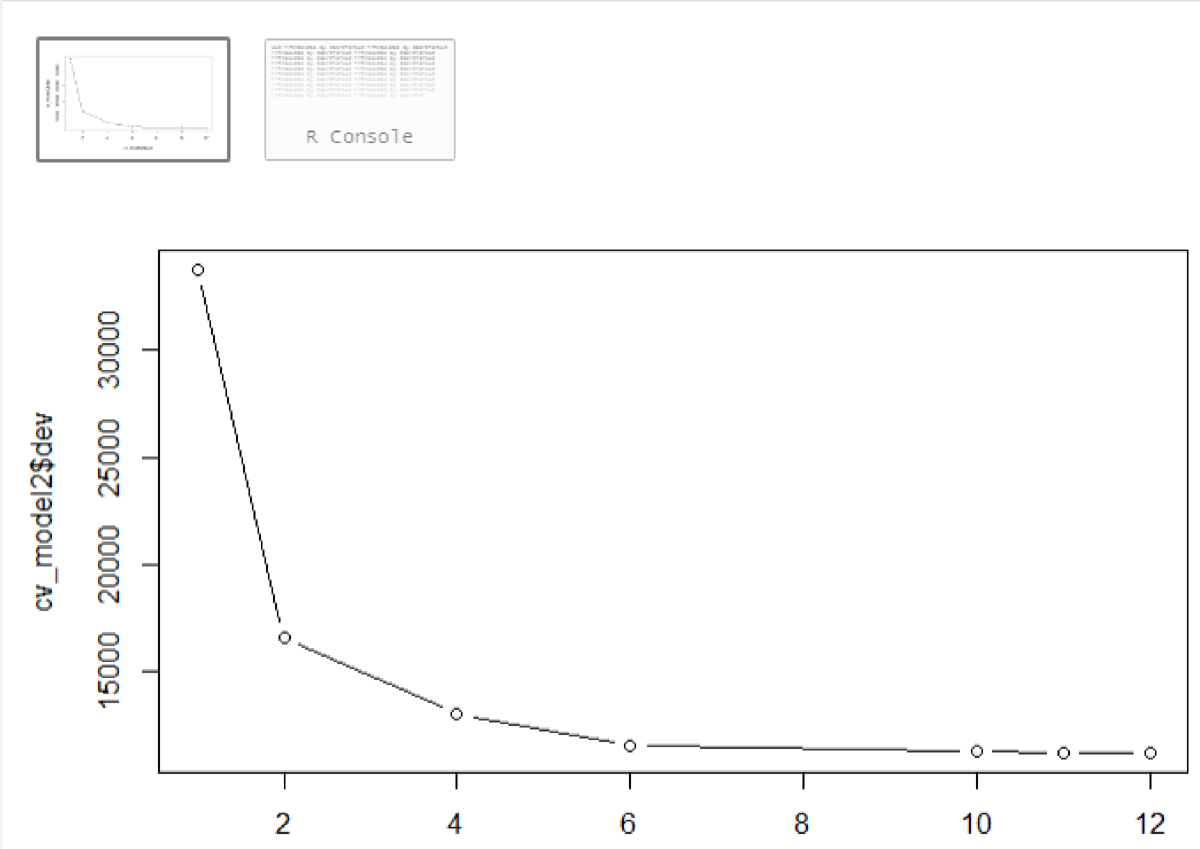


```
NAs introduced by coercion
Classification tree:
tree(formula = satisfaction ~ . - id + Inflight.wifi.service +
    Food.and.drink + Seat.comfort + Inflight.entertainment +
    Cleanliness + Baggage.handling + Inflight.service, data = imp_data,
    subset = train_data)
Variables actually used in tree construction:
[1] "Online.boarding"                  "Inflight.wifi.service"
[3] "Departure.Arrival.time.convenient" "On.board.service"
[5] "Leg.room.service"                 "Inflight.entertainment"
Number of terminal nodes:  12
Residual mean deviance:  0.632 = 49250 / 77920
Misclassification error rate: 0.152 = 11843 / 77928
```



The above is the classification tree. Internal nodes are the features of the dataset and terminal nodes is the response variable.

```
set.seed(1)
cv_model2<-cv.tree(model2,FUN=prune.misclass)
plot(cv_model2$size,cv_model2$dev,type="b")
```



The above output tells about the tree need to be predict the prune or not. Here after pruning of tree, the tree size is same. So no need to be prune of the tree.

SVM

```
SVM
```{r}
library(e1071)
train_data1<-scale(train_data)
model4<-svm(satisfaction~.-id+Inflight.wifi.service+Ease.of.Online.booking+Food.and.drink+Seat.comfort+
 Inflight.entertainment+Cleanliness+Baggage.handling+Inflight.service,data=imp_data,type='C-classification')
summary(model4)
```
```

```
package �e1071� was built under R version 4.0.5
Call:
svm(formula = satisfaction ~ . - id + Inflight.wifi.service + Ease.of.Online.booking +
    Food.and.drink + Seat.comfort + Inflight.entertainment + Cleanliness + Baggage.handling +
    Inflight.service, data = imp_data, type = "C-classification")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  16270

 ( 8229 8041 )


Number of Classes:  2

Levels:
 0 1
```

Naïve Bayes

```
Naive Bayes
```{r}
library(e1071)
library(caTools)
library(caret)
set.seed(1)
model5<-naiveBayes(satisfaction~-id+Inflight.wifi.service+Ease.of.Online.booking+Food.and.drink+Seat.comfort
+ Inflight.entertainment+Cleanliness+Baggage.handling+Inflight.service,data=imp_data,subset=train_data)
model5
```
```

```
package �caTools� was built under R version 4.0.5package �caret� was built under R version 4.0.5Loading
required package: ggplot2
Loading required package: lattice

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
       0        1
0.566215 0.433785

Conditional probabilities:
   Inflight.wifi.service
Y        [,1]      [,2]
  0 2.400689 0.9648518
  1 3.166460 1.5882405

   Ease.of.Online.booking
Y        [,1]     [,2]
  0 2.547842 1.206187
  1 3.036623 1.573911
```

CHAPTER VI

Evaluation of Model

6.1 Model Evaluation

Evaluating algorithm is an essential part of any project. The model may give satisfying results when evaluated using a metric accuracy score but may give poor results when evaluated against the model which is not suited for the data. The performance measure is the way to evaluate a solution to the problem. It is the measurement that will make of the predictions made by a trained model on the test model. Performance measures are typically specialized to the class of problem that are working with, for example classification, regression and clustering. Many standard performance measures will give a score that is meaningful to the problem domain.

There are different metrics for the classification performance. Accuracy, confusion matrix, log-loss and AUC-ROC are some of the most popular metrics. Precision-recall is a widely used metrics for classification problems. Accuracy simply measures how often the classifier correctly predicts. But it is good choice for the balanced data, not for unbalanced data. Confusion matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.

Prediction code for logistic regression

```
predict_model1 <- predict(model1,test_data1)
predict_factor<-ifelse(predict_model1>0.5,1,0)
predict_factor
```

```
   2    4    9   13   16   19   21   22   23   29   31   32   39   47   49   50   53   56   59   64 
   0    0    0    0    0    1    0    1    0    0    1    1    1    0    0    0    1    0    0    1 
  68   71   72   76   81   83   86   87   88   92   95  100  106  107  114  120  129  131  133  138 
   1    1    0    0    0    1    0    1    0    0    0    0    0    0    0    1    0    0    0    0 
 139  141  142  151  153  158  159  161  162  164  174  177  179  183  186  195  201  202  203  205 
   0    0    0    0    0    1    0    0    0    0    1    0    0    0    0    0    1    1    0    0 
 210  217  218  220  223  230  252  258  259  260  267  269  270  281  283  284  287  290  291  292 
   0    0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    0 
 296  301  302  309  310  312  319  321  322  328  331  337  338  344  351  354  361  366  374  376 
   1    0    0    0    0    1    0    0    1    1    0    1    0    1    0    1    0    0    0    1 
 379  382  384  389  390  394  400  405  411  415  419  420  427  429  441  443  444  446  451  455 
   0    0    1    1    1    0    0    0    0    0    1    0    0    1    1    0    0    0    0    1 
 456  457  459  461  465  467  471  477  478  487  491  493  515  517  518  521  523  525  528  531 
   1    0    1    0    0    0    0    0    1    0    0    1    0    0    1    0    0    0    0    0 
 532  545  551  552  554  565  566  574  579  586  599  604  605  615  616  623  625  636  637  642 
   1    1    0    0    0    1    0    0    0    1    0    0    1    0    0    0    0    0    0    1 
 648  652  653  656  659  662  664  665  669  671  674  676  679  682  684  686  687  692  696  699 
   1    0    0    1    0    0    0    0    0    0    0    1    0    1    1    0    0    1    1    0 
 702  703  705  706  714  715  727  737  739  742  747  750  751  758  763  764  767  769  771  775 
   0    0    0    0    0    0    0    0    0    0    0    1    0    1    0    0    1    0    0    0 
 778  781  786  791  793  805  808  815  819  824  825  828  831  842  843  854  855  859  869  871 
   1    0    0    1    0    0    1    1    1    0    0    0    1    0    0    1    1    0    0    0 
 873  874  880  881  886  888  890  893  898  906  913  914  920  924  925  926  929  933  938  943 
   0    1    1    0    0    0    1    0    0    0    1    1    1    0    0    1    0    0    0    0 
 951  957  962  963  968  969  971  972  973  978  979  981  984  990  997  999 1002 1006 1007 1016 
   0    1    0    1    1    0    0    1    0    1    0    0    0    0    0    0    0    0    0    1 
1018 1020 1021 1027 1033 1040 1042 1043 1045 1051 1055 1056 1061 1067 1072 1075 1076 1078 1083 1088 
   0    0    0    0    0    0    0    0    0    1    0    0    0    0    1    0    0    0    1    0 
1089 1100 1106 1109 1111 1113 1118 1121 1123 1134 1135 1138 1142 1145 1152 1159 1168 1171 1173 1177 
   0    1    0    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    0    1 
```

The predicted value converted into factor.

Prediction code for decision tree

```{r}
predict_model2 <- predict(model2,test_data1,type="class")
predict_model2
#predict_factor2<-ifelse(predict_model2>0.5,1,0)
#predict_factor2
```

```
NAs introduced by coercion   [1] 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 
1 0 0 0 0 1 1 1 1 1 1
   [49] 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1 0 1
   [97] 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0
  [145] 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1
  [193] 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 0
  [241] 0 1 0 1 1 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0
  [289] 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0
  [337] 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0
  [385] 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 1 0 1 0
  [433] 0 0 1 0 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 1
  [481] 0 0 1 0 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1
  [529] 1 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
  [577] 0 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0 0 0 1 0 0
  [625] 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 1 0 1 0
  [673] 1 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 1 0
  [721] 0 0 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1
  [769] 0 1 1 0 1 0 1 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 1
  [817] 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0
  [865] 1 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0
  [913] 0 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0
  [961] 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0
  [ reached getOption("max.print") -- omitted 24976 entries ]
```

Prediction code for svm

```{r}
predict_model4<-predict(model4,test_data1)
predict_model4
```

```
   2    4    9   13   16   19   21   22   23   29   31   32   39   47   49   50   53   56   59   64 
   0    0    0    0    0    1    0    0    0    0    1    0    0    0    1    0    1    1    0    1 
  68   71   72   76   81   83   86   87   88   92   95  100  106  107  114  120  129  131  133  138 
   1    1    0    1    1    0    1    0    0    1    0    0    0    0    0    1    0    1    0    0 
 139  141  142  151  153  158  159  161  162  164  174  177  179  183  186  195  201  202  203  205 
   0    0    0    1    0    0    1    1    1    0    1    1    1    0    0    0    1    1    0    0 
 210  217  218  220  223  230  252  258  259  260  267  269  270  281  283  284  287  290  291  292 
   1    0    0    0    1    0    0    0    0    1    0    1    1    0    0    0    0    1    0    0 
 296  301  302  309  310  312  319  321  322  328  331  337  338  344  351  354  361  366  374  376 
   1    1    1    1    1    1    0    0    1    1    0    1    0    0    0    1    0    1    1    1 
 379  382  384  389  390  394  400  405  411  415  419  420  427  429  441  443  444  446  451  455 
   0    0    1    1    1    0    0    0    0    0    0    0    0    1    1    0    0    0    1    1 
 456  457  459  461  465  467  471  477  478  487  491  493  515  517  518  521  523  525  528  531 
   1    0    1    1    0    0    0    1    1    0    0    1    1    0    1    0    1    1    0    1 
 532  545  551  552  554  565  566  574  579  586  599  604  605  615  616  623  625  636  637  642 
   1    1    1    0    1    1    1    0    0    0    0    0    1    0    1    0    0    1    1    1 
 648  652  653  656  659  662  664  665  669  671  674  676  679  682  684  686  687  692  696  699 
   0    1    1    1    0    0    0    1    1    0    0    1    0    1    1    0    1    0    0    0 
 702  703  705  706  714  715  727  737  739  742  747  750  751  758  763  764  767  769  771  775 
   0    0    0    1    1    0    0    0    0    1    0    1    0    0    0    1    0    0    0    0 
 778  781  786  791  793  805  808  815  819  824  825  828  831  842  843  854  855  859  869  871 
   1    0    0    1    0    0    1    1    1    1    0    1    1    1    1    1    1    0    0    0 
 873  874  880  881  886  888  890  893  898  906  913  914  920  924  925  926  929  933  938  943 
   0    1    1    0    0    0    1    0    1    0    1    1    0    1    1    1    1    0    0    0 
 951  957  962  963  968  969  971  972  973  978  979  981  984  990  997  999 1002 1006 1007 1016 
   0    1    0    0    1    0    1    1    0    1    1    0    1    1    0    1    0    1    0    0 
```

Prediction code for naïve bayes

```{r}
predict_model5<-predict(model5,test_data1)
predict_model5
```

```
   [1] 0 0 0 0 0 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1
  [49] 0 1 1 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1
  [97] 0 1 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 1 1 0 0 1 1 1 0
 [145] 0 1 0 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 0 0 0 1 0 1
 [193] 0 1 0 0 1 0 1 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0
 [241] 0 1 0 1 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 1 0 0
 [289] 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 0
 [337] 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 0 0 1 0 0
 [385] 1 1 0 0 1 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0 0 1 0 0 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 0
 [433] 0 0 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 0 1 0 0 0 1
 [481] 0 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 0 0 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1
 [529] 1 0 0 1 1 0 1 0 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0
 [577] 1 0 1 1 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1
 [625] 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1
 [673] 1 1 1 1 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 1 0
 [721] 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1
 [769] 1 0 1 0 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 1 1 1 0 1 1 1 0 0 0 0 1
 [817] 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1 0 1 0
 [865] 1 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 0
 [913] 0 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0
```

The above four outputs shows the prediction of the four built models.

Confusion Matrix & Accuracy

Confusion matrix is a table like structure where can see the true and false positive and negative rates comparing with prediction and original values. Accuracy can be calculated by using the positive and negative rates in the confusion matrix.

```{r}
table(predict_factor,newdata=test_data$satisfaction)
accuracy_glm<-(13161+5387)/(13161+5834+1594+5387)
accuracy_glm
```

```
                newdata
predict_factor neutral or dissatisfied satisfied
             0                   13161      5834
             1                    1594      5387
[1] 0.7140437
```

The accuracy score of the logistic regression is 0.71

```{r}
table(predict_model2,newdata=test_data$satisfaction)
accuracy_tree<-(13428+8682)/(13428+2539+1327+8682)
accuracy_tree
```

```
                newdata
predict_model2 neutral or dissatisfied satisfied
             0                   13428      2539
             1                    1327      8682
[1] 0.8511703
```

The accuracy score of the tree is 0.85

```{r}
table(predict_model4,newdata=test_data$satisfaction)
accuracy_svm<-(14292+10479)/(14292+742+463+10479)
accuracy_svm
```

```
                newdata
predict_model4 neutral or dissatisfied satisfied
             0                   14292       742
             1                     463     10479
[1] 0.953611
```

The accuracy score of the svm is 0.95

```{r}
table(predict_model5,newdata=test_data$satisfaction)
accuracy_nb = (10407+8845)/(10407+2376+4348+8845)
accuracy_nb
```

```
                newdata
predict_model5 neutral or dissatisfied satisfied
             0                   10407      2376
             1                    4348      8845
[1] 0.7411457
```

The accuracy score of the naïve bayes is 0.74

The above results shows that the confusion matrix and accuracy of the models.

This section tells about the prediction and accuracy of the built models.

CHAPTER VII
CONCLUSION

The above model predicts the satisfaction of the passenger with the conclusions below in this section.

- In Bivariate and Multivariate analysis, feature selection was done using correlation matrix.
- Highly correlated variables were used to built models.
- Even the variable “ease of online booking” has 0.72 in correlation, but not linearly significant with the response variable.
- Compared to other models, accuracy score of SVM model is high (0.95).
- Compared to other models, accuracy score of logistic regression is low (0.71).
- So, SVM may better to classify the passenger satisfaction.