# Telecom Customer Churn

# 2 Project Objectives

The objective of the report is to explore the data file Cellphone.csv in R and generate insights about the data set. This exploration report will consists of the following scenarios:

- Exploratory Data Analysis of the data
- Build appropriate models on both the test and train data (CART and Random Forest)
- Interpreting the model outputs and performing the necessary modifications wherever eligible
- Check the performance of all the models that you have built (test and train)

# 3 Exploratory Data Analysis – Step by step approach

A Typical Data exploration activity consists of the following steps:

1. Environment Set up and Data Import
2. Variable Identification
3. Visualisation Plots

We shall follow these steps in exploring the provided dataset.

## 3.1 Environment Set up and Data Import

### 3.1.1 Set up working Directory

Setting a working directory on starting of the R session makes importing and exporting data files and code files easier. Basically, working directory is the location/ folder on the PC where you have the data, codes etc. related to the project.
Please refer Appendix A for Source Code.

### 3.1.2 Import and Read the Dataset

The given dataset is in .csv format. Hence, the command 'read.csv' is used for importing the file.
Please refer Appendix A for Source Code.

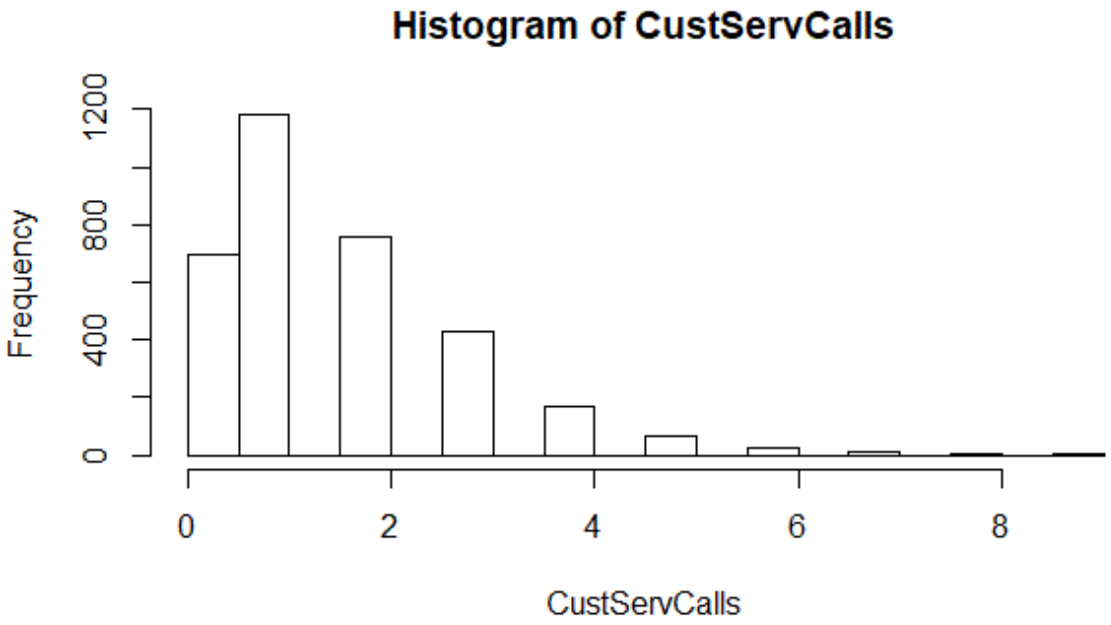
## 3.2 Variable Identification

The length and breadth of the data was examined and the names of the data were pulled from the dataset. The data consisted of 11 variables consisting of 3333 employees determining whether or not they took a loan from the bank. The string type of the data was also verified by using the str() function.

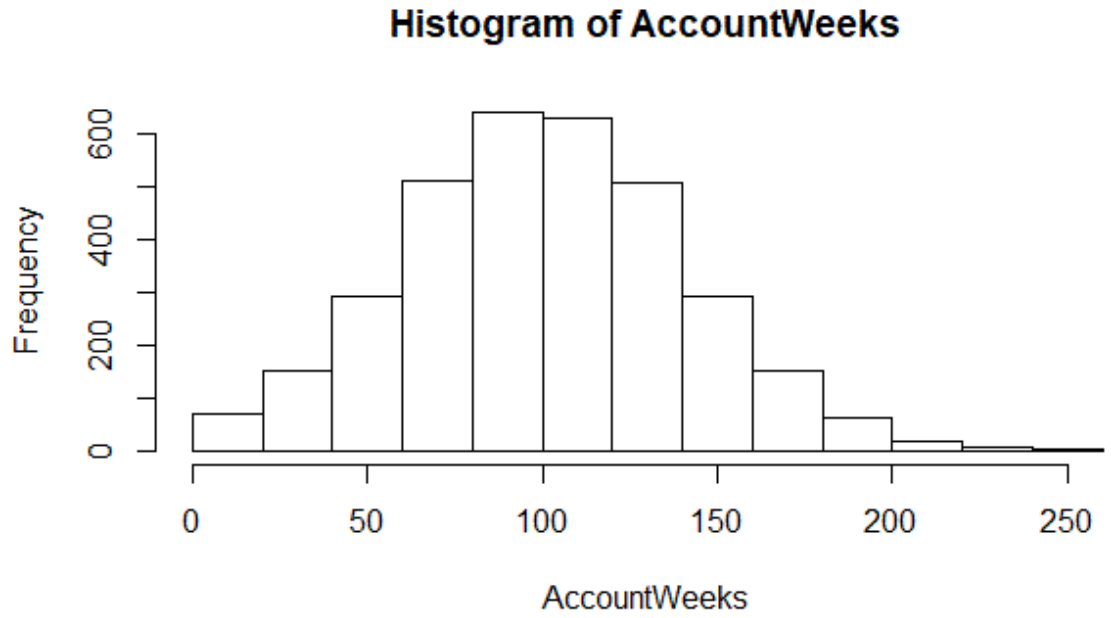
## 3.3 Visualisation Plots

**Histogram Plot**

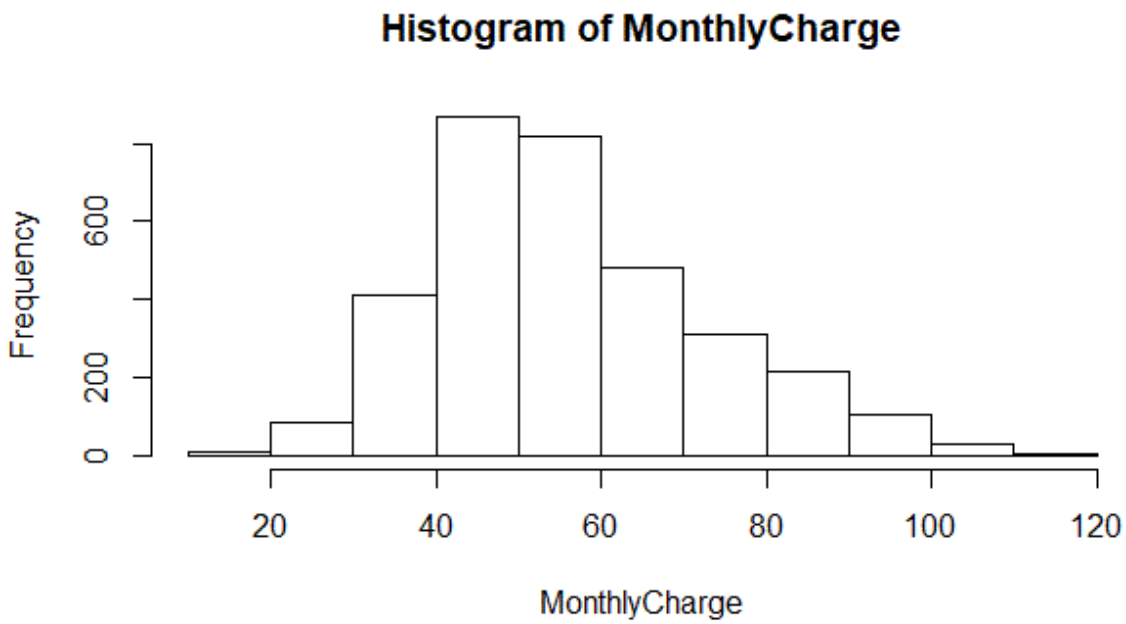
Below are the histogram plots for Customer Service Cell, account weeks, monthly charge and overage fee from the dataset.



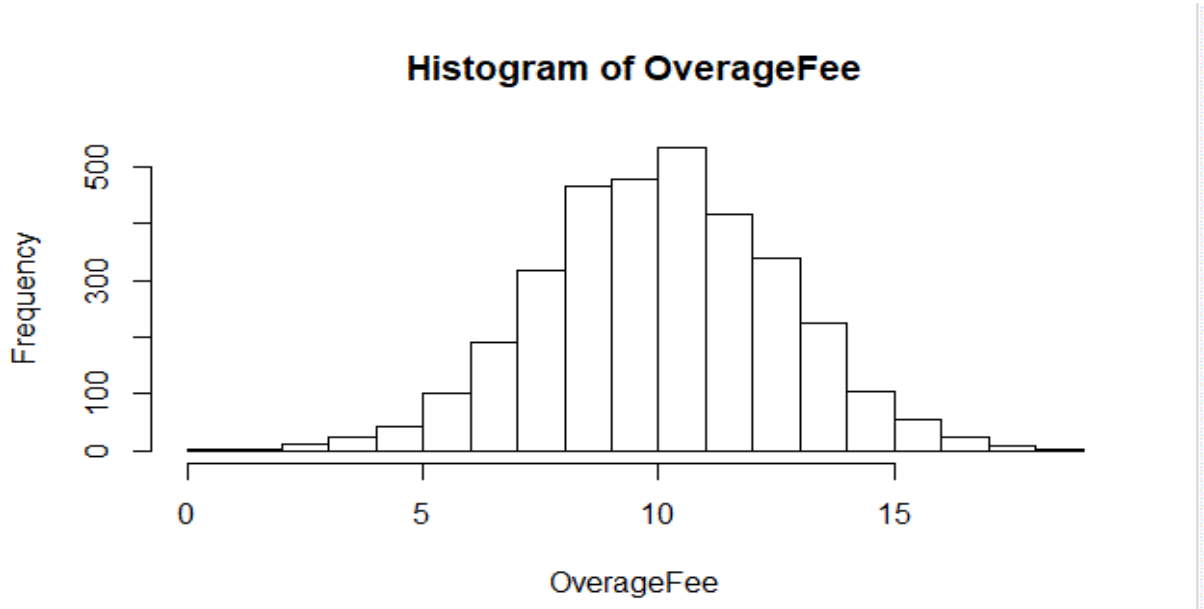

- The above histogram depicts that a minor percentage of the customers were involved with more than 4 Customer Service calls.



- The histogram for account weeks displays a uniform distribution where the highest number of account weeks was averaged at 100




- The mean monthly charge was maximum between 40-60.
- The number of customers decreases with increase in monthly charge.



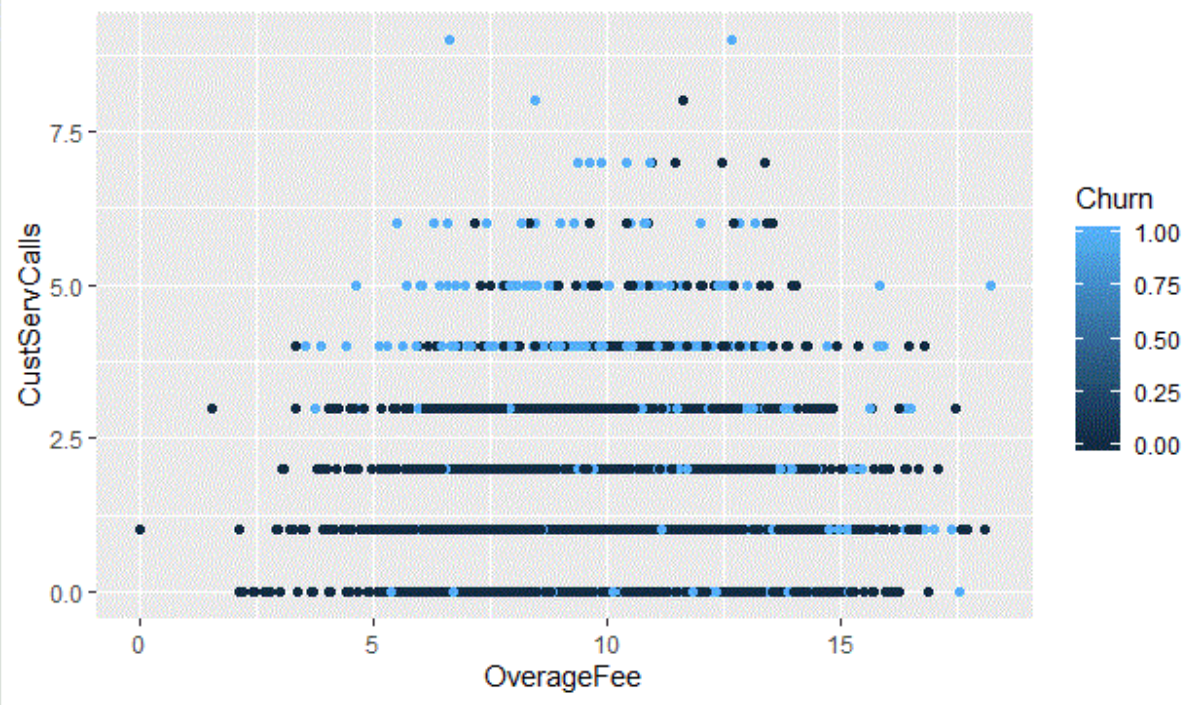

- **The maximum overage fee for customers was averaged at 11.**

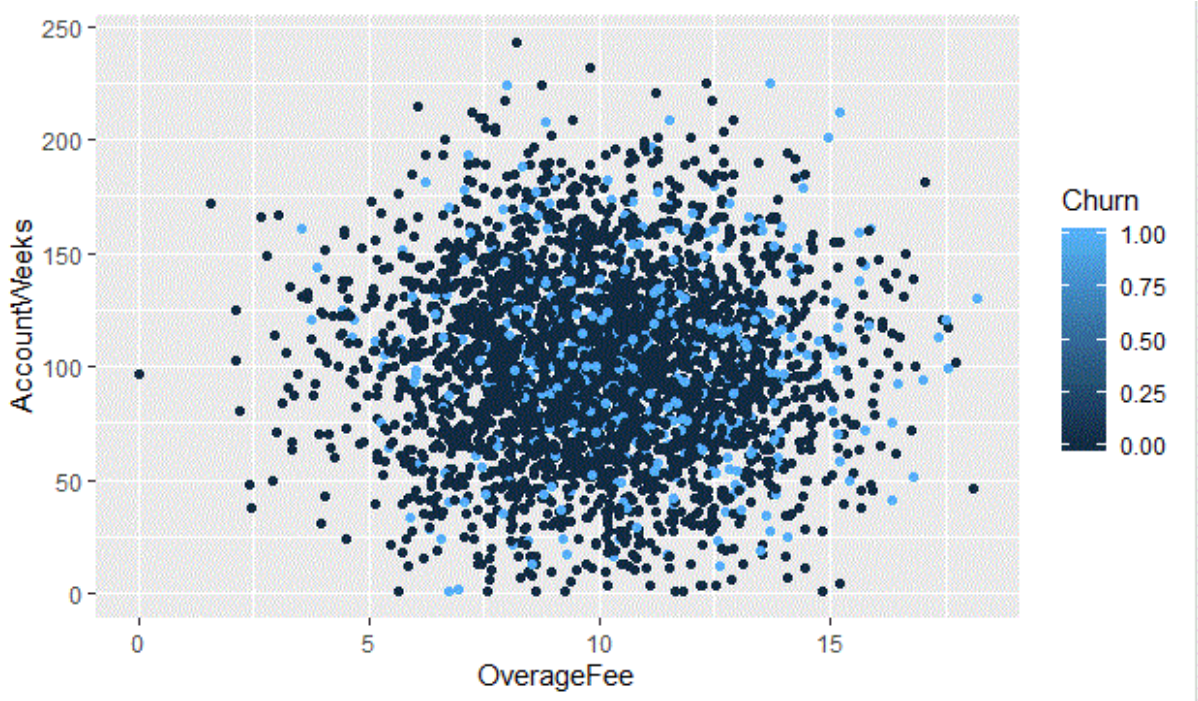
### Missing Values

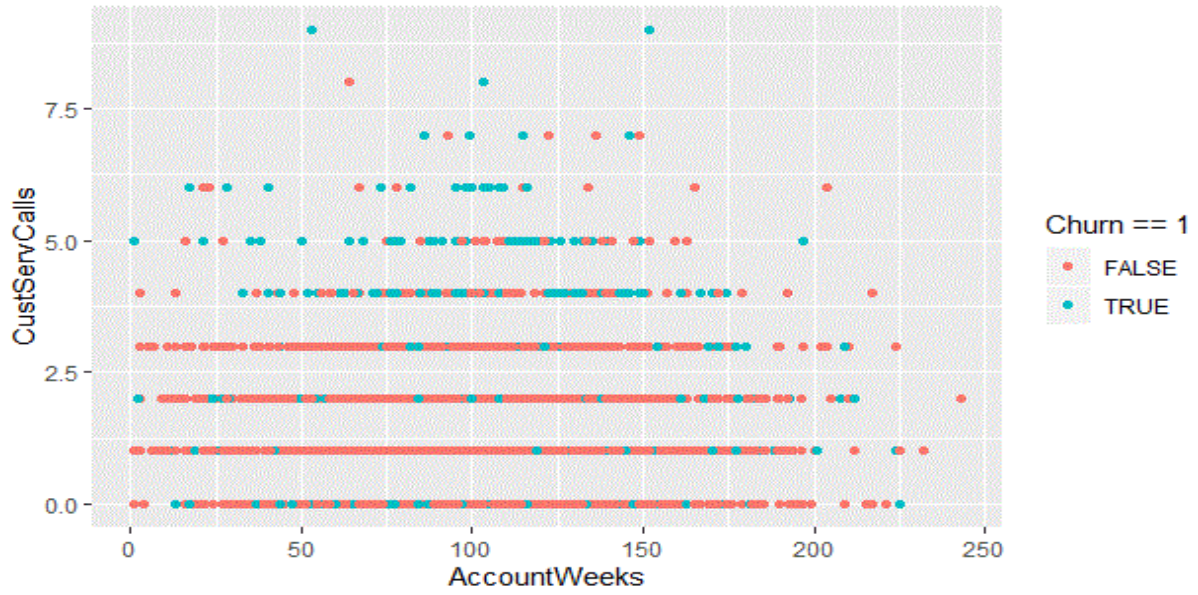
No missing values were found in the data set. The below code was used to determine the missing values

```
sapply(cell, function(x) sum(is.na(x)))
```

**Qplot**

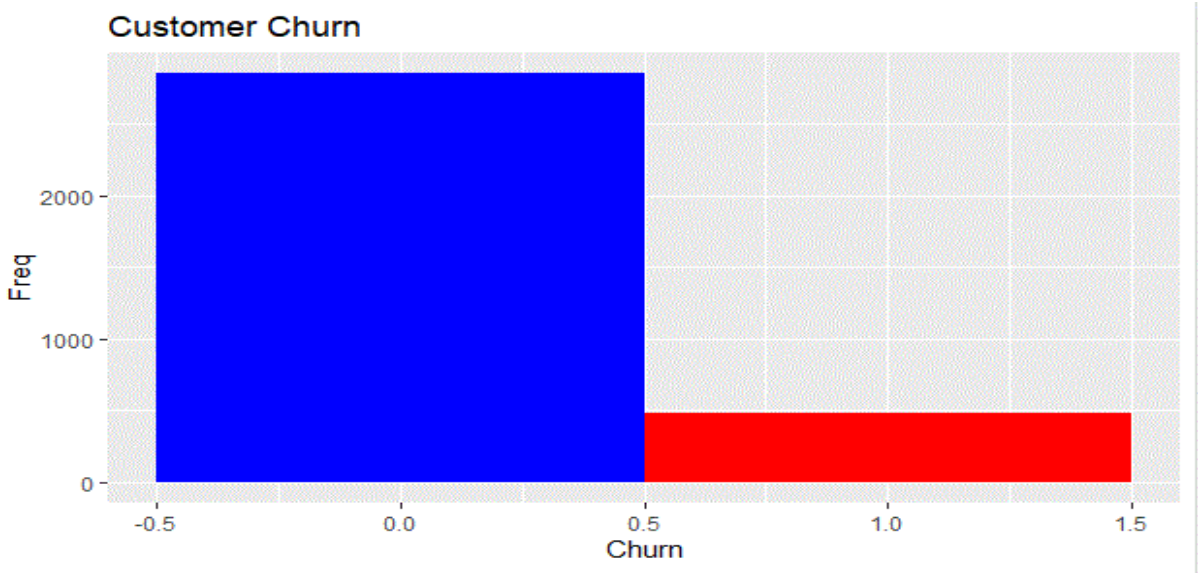

CustServCalls
7.5
5.0
2.5
0.0
0
5
10
15
OverageFee
Churn
1.00
0.75
0.50
0.25
0.00


AccountWeeks
250
200
150
100
50
0
0
5
10
15
OverageFee
Churn
1.00
0.75
0.50
0.25
0.00




- **From the above Qplots plotted we can conclude that as the customer service calls increases, the customers are likely to churn.**

**Ratio of customers who are likely to Churn**

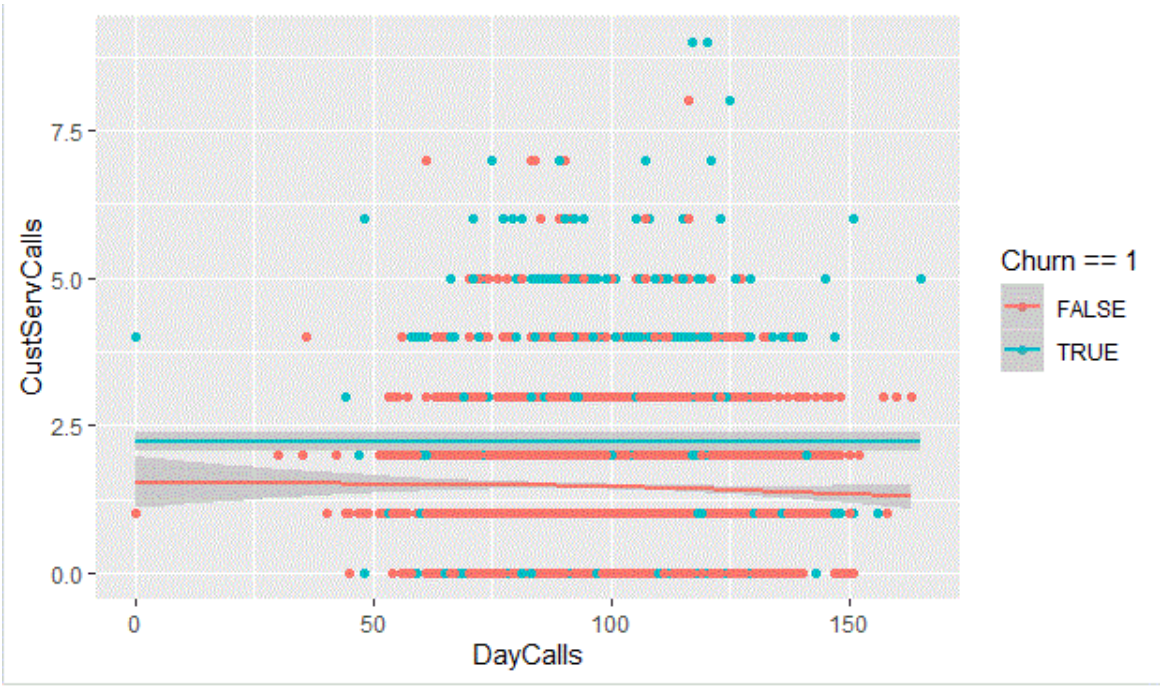



Ratio of customers who are likely to churn

```
> churnrate

   0         1
0.8550855   0.1449145
```

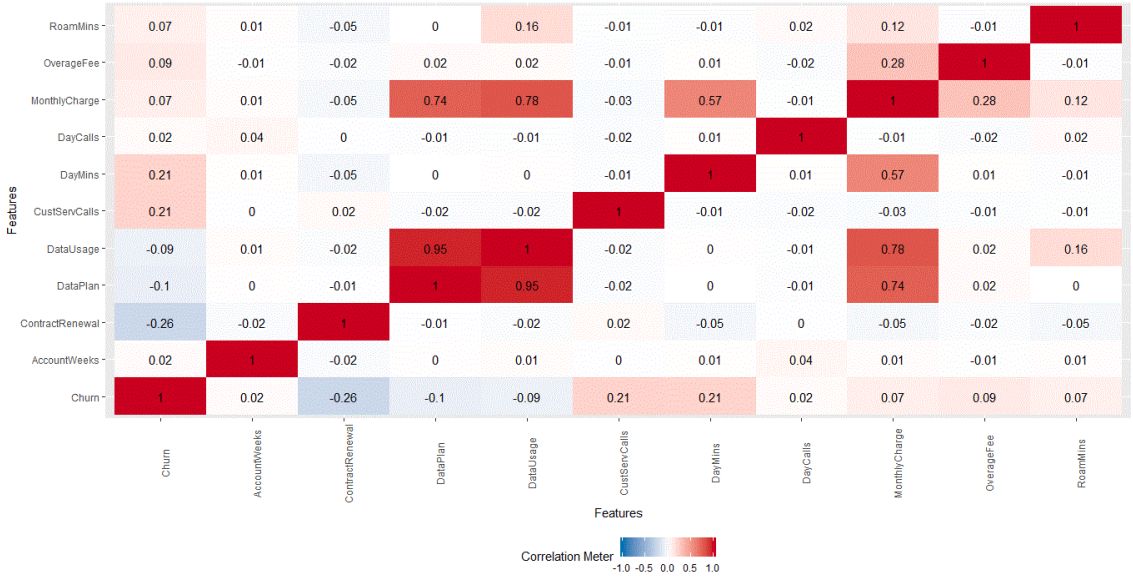
## ggplot




- **The above relativity plot between the 3 variables namely Customer Service calls, Day calls and Churn determines a "smooth line" as the threshold for the customers who are likely to churn or not.**
- **Customers who have CustServCalls more than 2 and engage in more than 50 calls per day are more likely to churn.**

## Corrplot

Corrplot was plotted between all the variables to determine the correlation amongst the variables and the output is displayed in the below figure.

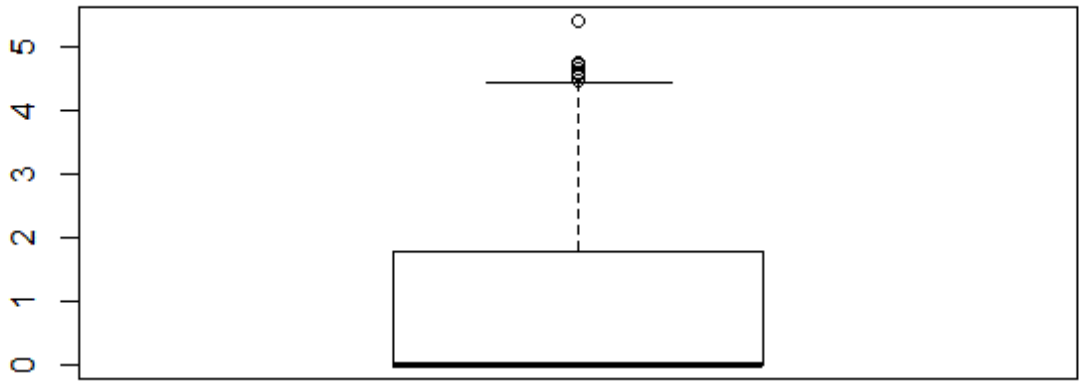


From the above corrplot the below observations can be made:

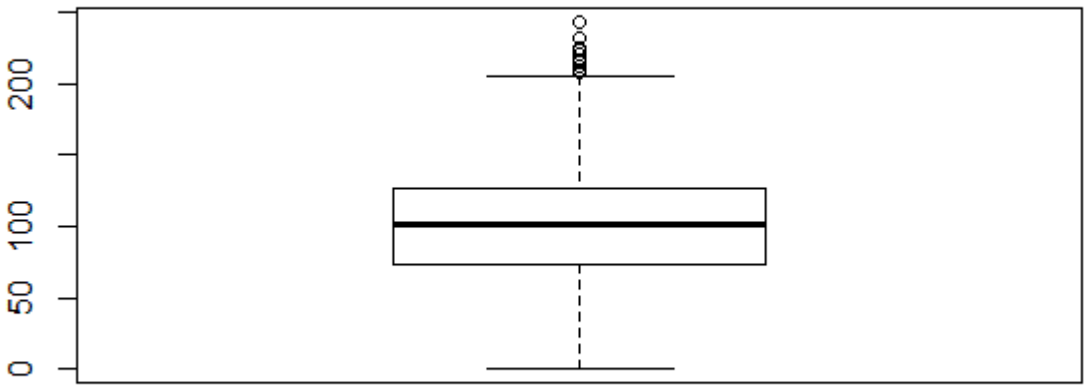
- Data Plan and Data Usage are highly correlated.
- Monthly charge is correlated with Data Plan and Data Usage.

**Outliers**

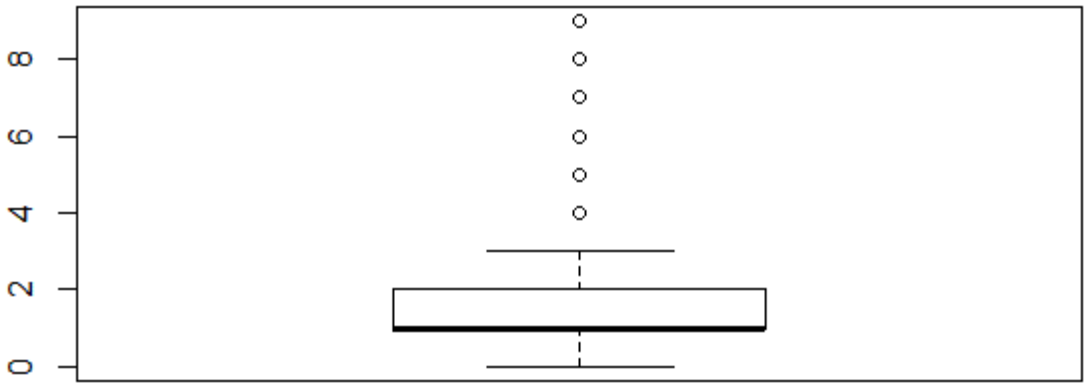
1.Data Usage



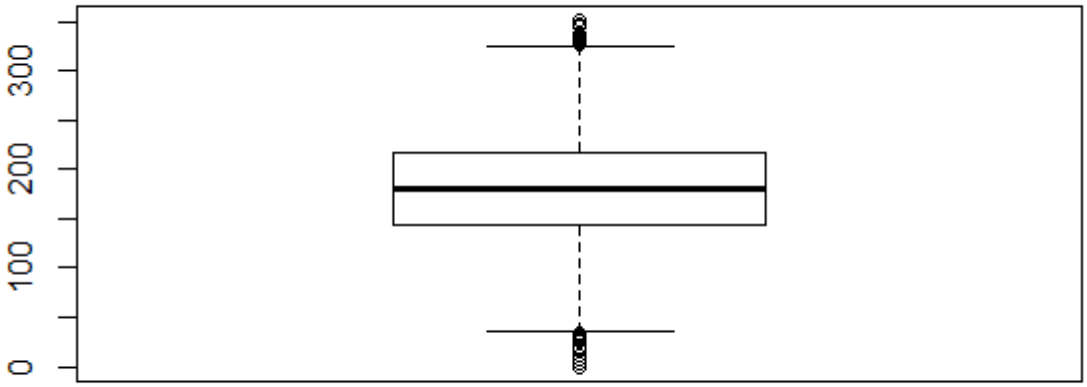

2. Account Week



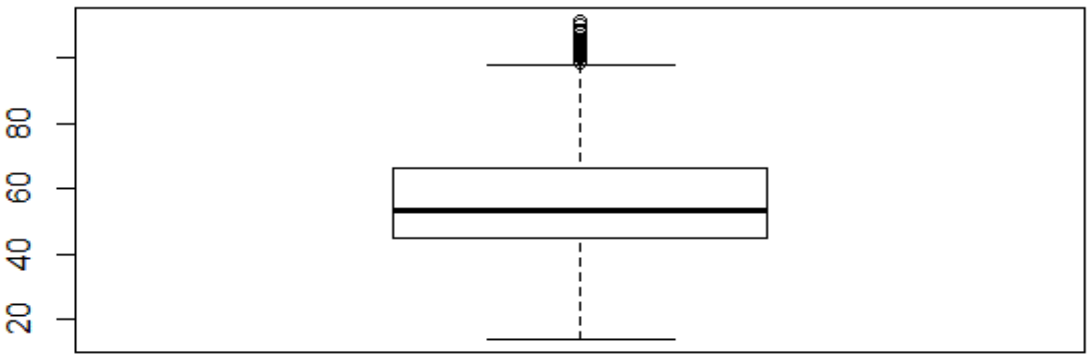
3. Customer Service Calls



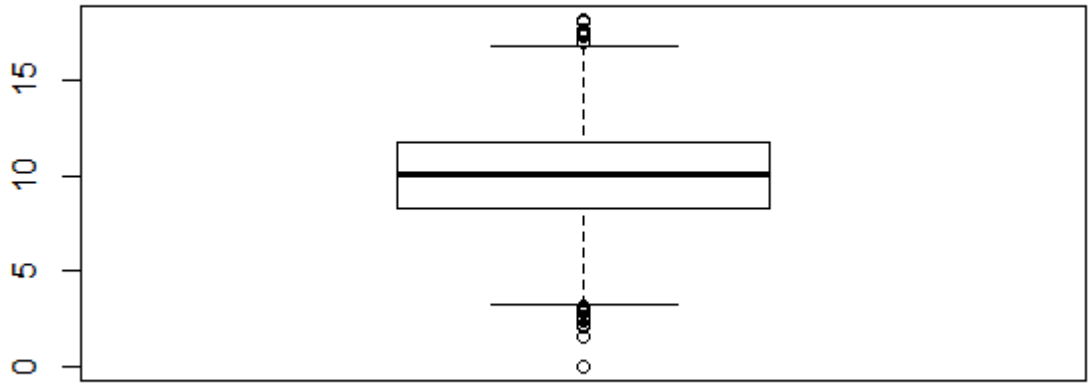

4. Day Mins



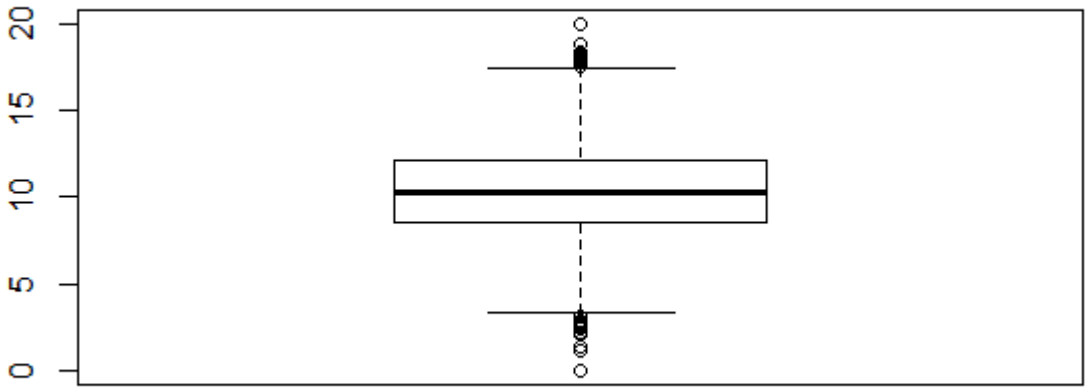

5. Monthly Charge



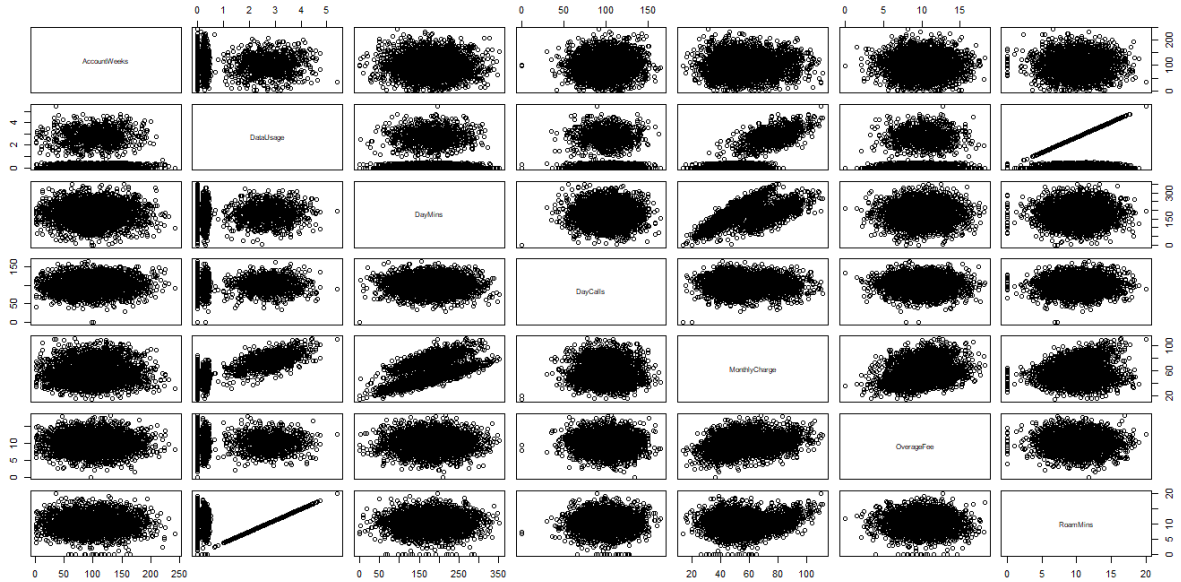
6. Overage Fee




7. Roam Mins

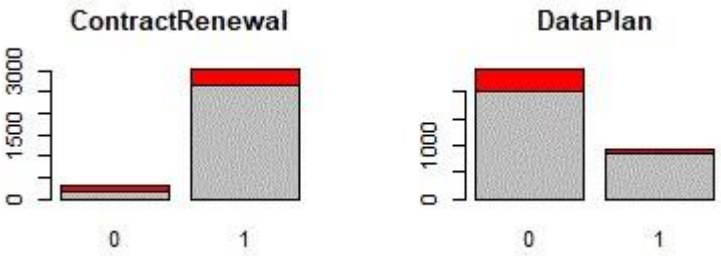


## Visual pattern for continuous variables




- Not much insights could be plotted from the above graph

Ratio of churn between Contact Renewal and Data Plan




## 4. Refining the dataset

Splitting the data into train and test

The data has been splitted in ratio 65:35
The output of sum of Churn in test and train has been mentioned below

```
> sum(cell$Churn)
[1] 483
> sum(val$Churn)
[1] 184
> sum(train$Churn)
[1] 299
```

- Ordinary Least Square was found using lm function
- Dimensionality reduction was performed and DataUsage was removed from the dataset due to multicollinearity

- The VIF values found after dimensionality reduction was found to be good.

```
> vif(OLS.full1)
      Churn    AccountWeeks ContractRenewal   CustServCalls         DayMins
   1.189536        1.002697        1.073598        1.053593        1.036685
    DayCalls      OverageFee        RoamMins        DataPlan
   1.006703        1.014873        1.011628        1.021151
```

## 5. Model algorithms with Performance Measures

### 1. Logistic Regression

- Loading the data
- Found generalised linear model using glm() function
- Using pR2() we found that the independent variable explain 19.38% of dependent variable
- This was determined by checking the McFadden value
- Splitted the categorical and numerical data
- Performed Chi square test on the categorical data
- Built univariate logistic regression models
- Removed account week and day calls as they were not significant
- Final data was prepared, seed set and data splitted into train and test
- % of sample in train,test and full data was checked and found to be similar
- VIF of the final model is mentioned below

```
> vif(LRmodel)
 CustServCalls        DayMins   MonthlyCharge      OverageFee        RoamMins
      1.069130       1.766834       1.849349       1.208662       1.025734
ContractRenewal
      1.047130
```

**Performance Measures**

- **Confusion matrix**

**Train**

```
> conf_mat

   FALSE TRUE
 0  1808   44
 1   253   61
```

**Accuracy**: 0.8628809

**Test**

> conf_mat

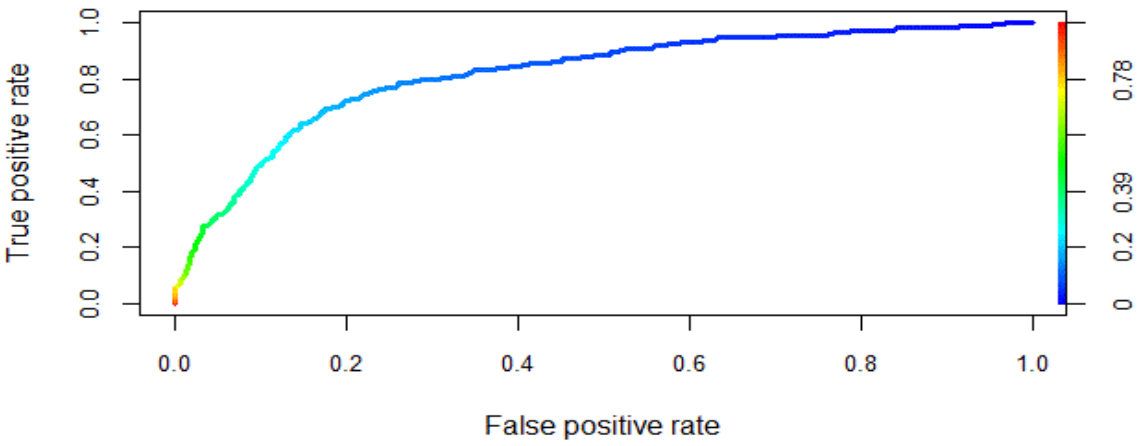
```
  FALSE TRUE
0   979   19
1   144   25
```

**Accuracy**: 0.8603256

- **ROC curve**

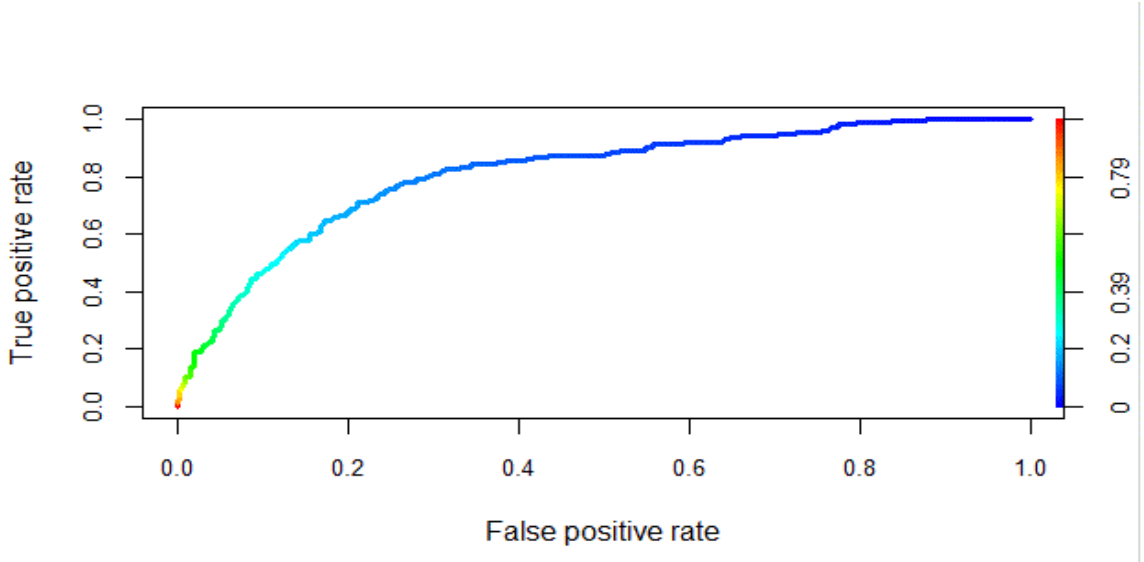
The graphs for all the performance measures have been plotted below.

**Train**




**Accuracy**: 0.8156374

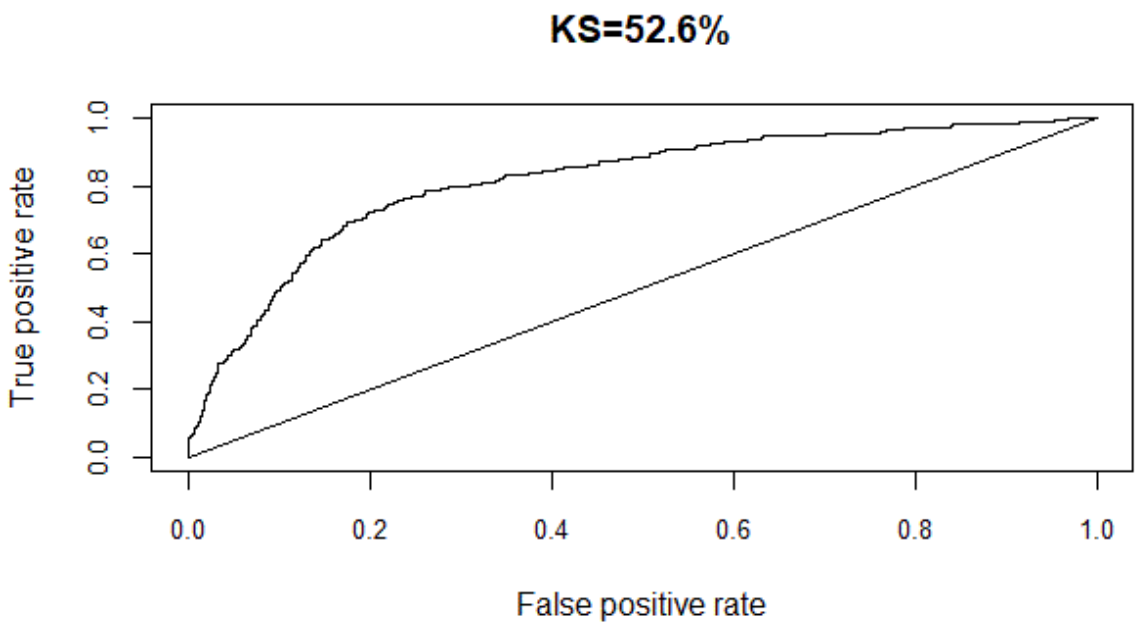
**Test**



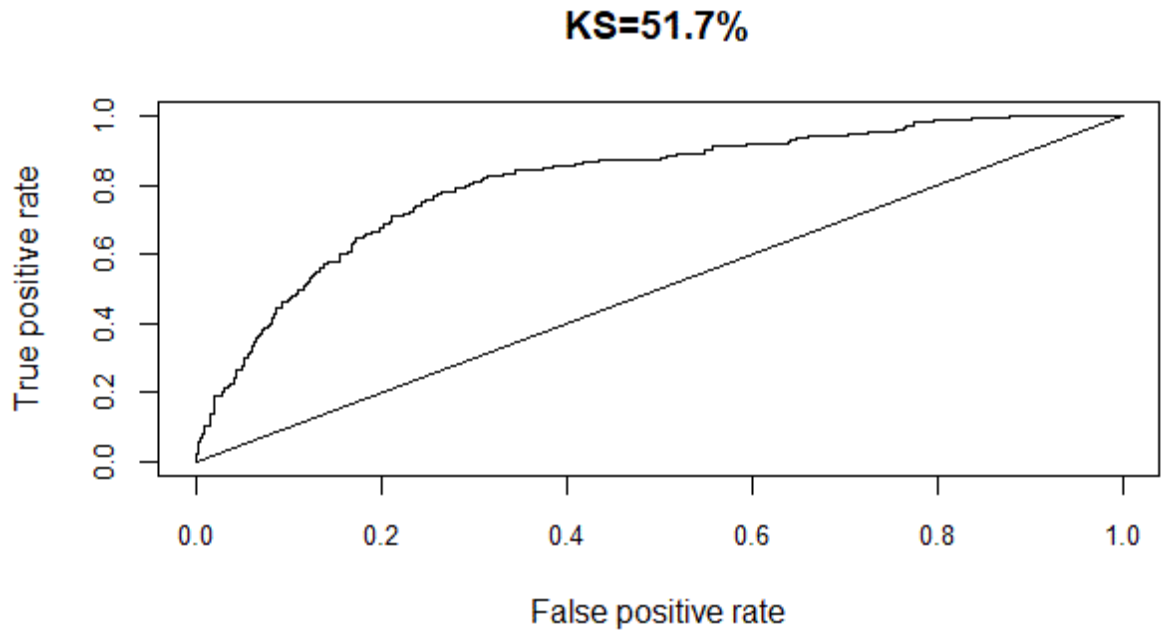

**Accuracy**: 0.8101706

- **Kolmogorov-Smirnov**

**Train**




**Test**



2. **KNN**

- Converted variables to numerical string to be used for KNN
- Converted the data to data frame and then to factor
- Tried different values of k to find the best KNN model which could minimise the False Negatives
- At K=43 we found the best matrix with minimal False negatives

**knn_model**

| | 1 | 2 |
|---|---|---|
| **1** | **1212** | **9** |
| **2** | **141** | **66** |

**Accuracy**: 0.894958
**Total Loss**: 0.1050022

3. **Naïve Bayes**

**pred_nb**

| | 1 | 2 |
|---|---|---|
| **1** | **1124** | **97** |
| **2** | **130** | **77** |

**Accuracy**: 0.8410364
**Total loss**: 0.1263584

Naïve Bayes is not preferred because the algorithm makes a very strong assumption about the data having features independent of each other while in reality, they may be dependent in some way. In other words, it assumes that the presence of one feature in a class is completely unrelated to the presence of all other features. If this assumption of independence holds, Naive Bayes performs extremely well and often better than other models. **Naive Bayes can also be used with continuous features but is more suited to categorical variables. If all the input features are categorical, Naive Bayes is recommended. However, in case of numeric features, it makes another strong assumption which is that the numerical variable is normally distributed which was not in the current case.**

## 6. Conclusion

| | Logistic Regression | KNN | Naïve Bayes |
|---|---|---|---|
| **Confusion** | 86.29 | 89.49 | 84.1 |
| **ROC** | 81.56 | | |
| **KS** | 52.6 | | |

Based on the above data and the plots obtained we can conclude that the model performs better on KNN model as compared to Logistic Regression and Naïve Bayes. With the data splitted at 65:35 ratio we found the K value for the model to be the most significant at K=43. Variables Account Weeks and Day Calls were found to be insignificant and hence removed from the data set. Data Plan and Data Usage induced multicolllinearity with Monthly Charge and hence were removed. The response churn variable is affected by CustServCalls, DayMins, MonthlyCharge, OverageFee, RoamMins and ContractRenewal. The importance of the variable can be identified by the legend of the correlated coefficients.

5 Appendix A – Source Code

*Here is the code produced in RStudio for doing an analysis on Product Service Management.*

```
#loading libraries
>
> library(car)
> library(caret)
> library(class)
> library(devtools)
> library(e1071)
> library(ggord)
Error in library(ggord) : there is no package called ‘ggord’
> library(ggplot2)
> library(Hmisc)
> library(klaR)
> library(MASS)
> library(nnet)
> library(plyr)
> library(pROC)
> library(psych)
> library(scatterplot3d)
> library(SDMTools)
> library(dplyr)
> library(ElemStatLearn)
> library(rpart)
> library(rpart.plot)
> library(randomForest)
> library(neuralnet)
> #Import the data related cellphone
> cell <- read.csv("Cellphone.csv")
> str(cell)
'data.frame':  3333 obs. of  11 variables:
 $ Churn          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ AccountWeeks   : int  128 107 137 84 75 118 121 147 117 141 ...
 $ ContractRenewal: int  1 1 1 0 0 0 1 0 1 0 ...
 $ DataPlan       : int  1 1 0 0 0 0 1 0 0 1 ...
 $ DataUsage      : num  2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
 $ CustServCalls  : int  1 1 0 2 3 0 3 0 1 0 ...
 $ DayMins        : num  265 162 243 299 167 ...
 $ DayCalls       : int  110 123 114 71 113 98 88 79 97 84 ...
 $ MonthlyCharge  : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
 $ OverageFee     : num  9.87 9.78 6.06 3.1 7.42 ...
 $ RoamMins       : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
> #Check for missing values in each column in the data set
> sapply(cell, function(x) sum(is.na(x)))
          Churn    AccountWeeks ContractRenewal        DataPlan       Data
Usage
              0               0               0               0
0
  CustServCalls         DayMins        DayCalls   MonthlyCharge      Overa
geFee
              0               0               0               0
0
       RoamMins
              0
> attach(cell)
The following objects are masked from cell (pos = 10):

    AccountWeeks, Churn, ContractRenewal, CustServCalls, DataPlan,
    DataUsage, DayCalls, DayMins, MonthlyCharge, OverageFee, RoamMins
```

```
> hist(CustServCalls)
> summary(cell)
     Churn          AccountWeeks   ContractRenewal     DataPlan
 Min.   :0.0000   Min.   :  1.0   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.: 74.0   1st Qu.:1.0000   1st Qu.:0.0000
 Median :0.0000   Median :101.0   Median :1.0000   Median :0.0000
 Mean   :0.1449   Mean   :101.1   Mean   :0.9031   Mean   :0.2766
 3rd Qu.:0.0000   3rd Qu.:127.0   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :243.0   Max.   :1.0000   Max.   :1.0000
   DataUsage      CustServCalls      DayMins         DayCalls     MonthlyC
harge
 Min.   :0.0000   Min.   :0.000   Min.   :  0.0   Min.   :  0.0   Min.   :
14.00
 1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:143.7   1st Qu.: 87.0   1st Qu.:
45.00
 Median :0.0000   Median :1.000   Median :179.4   Median :101.0   Median :
53.50
 Mean   :0.8165   Mean   :1.563   Mean   :179.8   Mean   :100.4   Mean   :
56.31
 3rd Qu.:1.7800   3rd Qu.:2.000   3rd Qu.:216.4   3rd Qu.:114.0   3rd Qu.:
66.20
 Max.   :5.4000   Max.   :9.000   Max.   :350.8   Max.   :165.0   Max.   :
111.30
   OverageFee       RoamMins
 Min.   : 0.00   Min.   : 0.00
 1st Qu.: 8.33   1st Qu.: 8.50
 Median :10.07   Median :10.30
 Mean   :10.05   Mean   :10.24
 3rd Qu.:11.77   3rd Qu.:12.10
 Max.   :18.19   Max.   :20.00
> mean(AccountWeeks)
[1] 101.0648
> hist(AccountWeeks)
> hist(MonthlyCharge)
> hist(OverageFee)
> #from the above qplots plotted we can say that as the cust service calls
> #increases the cust. is likely to churn
>
> cell.scatter<-subset(cell[,c(2:11)])
> cell.scatter
    AccountWeeks ContractRenewal DataPlan DataUsage CustServCalls DayMins DayCalls
1            128               1        1      2.70             1   265.1      110
2            107               1        1      3.70             1   161.6      123
3            137               1        0      0.00             0   243.4      114
4             84               0        0      0.00             2   299.4       71
5             75               0        0      0.00             3   166.7      113
6            118               0        0      0.00             0   223.4       98
7            121               1        1      2.03             3   218.2       88
8            147               0        0      0.00             0   157.0       79
9            117               1        0      0.19             1   184.5       97
10           141               0        1      3.02             0   258.6       84
11            65               1        0      0.29             4   129.1      137
12            74               1        0      0.34             0   187.7      127
13           168               1        0      0.00             1   128.8       96
14            95               1        0      0.44             3   156.6       88
15            62               1        0      0.00             4   120.7       70
16           161               1        0      0.00             4   332.9       67
17            85               1        1      3.73             1   196.4      139
18            93               1        0      0.00             3   190.7      114
19            76               1        1      2.70             1   189.7       66
20            73               1        0      0.00             1   224.4       90
21           147               1        0      0.31             0   155.1      117
22            77               1        0      0.00             5    62.4       89
23           130               1        0      0.00             0   183.0      112
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 24 | 111 | 1 | 0 | 0.39 | 2 | 110.4 | 103 |
| 25 | 132 | 1 | 0 | 0.00 | 0 | 81.1 | 86 |
| 26 | 174 | 1 | 0 | 0.00 | 3 | 124.3 | 76 |
| 27 | 57 | 1 | 1 | 2.57 | 0 | 213.0 | 115 |
| 28 | 54 | 1 | 0 | 0.00 | 3 | 134.3 | 73 |
| 29 | 20 | 1 | 0 | 0.32 | 0 | 190.0 | 109 |
| 30 | 49 | 1 | 0 | 0.21 | 1 | 119.3 | 117 |
| 31 | 142 | 1 | 0 | 0.00 | 2 | 84.8 | 95 |
| 32 | 75 | 1 | 0 | 0.00 | 1 | 226.1 | 105 |
| 33 | 172 | 1 | 0 | 0.00 | 3 | 212.0 | 121 |
| 34 | 12 | 1 | 0 | 0.00 | 1 | 249.6 | 118 |
| 35 | 57 | 1 | 1 | 2.24 | 0 | 176.8 | 94 |
| 36 | 72 | 1 | 1 | 3.97 | 3 | 220.0 | 80 |
| 37 | 36 | 1 | 1 | 3.92 | 0 | 146.3 | 128 |
| 38 | 78 | 1 | 0 | 0.00 | 1 | 130.8 | 64 |
| 39 | 136 | 0 | 1 | 2.84 | 3 | 203.9 | 106 |
| 40 | 149 | 1 | 0 | 0.00 | 1 | 140.4 | 94 |
| 41 | 98 | 1 | 0 | 0.30 | 3 | 126.3 | 102 |
| 42 | 135 | 0 | 1 | 3.94 | 0 | 173.1 | 85 |
| 43 | 34 | 1 | 0 | 0.00 | 2 | 124.8 | 82 |
| 44 | 160 | 1 | 0 | 0.38 | 3 | 85.8 | 77 |
| 45 | 64 | 1 | 0 | 0.24 | 1 | 154.0 | 67 |
| 46 | 59 | 1 | 1 | 2.30 | 2 | 120.9 | 97 |
| 47 | 65 | 1 | 0 | 0.00 | 3 | 211.3 | 120 |
| 48 | 142 | 1 | 0 | 0.35 | 2 | 187.0 | 133 |
| 49 | 119 | 1 | 0 | 0.00 | 5 | 159.1 | 114 |
| 50 | 97 | 1 | 1 | 2.97 | 1 | 133.2 | 135 |
| 51 | 52 | 1 | 0 | 0.32 | 3 | 191.9 | 108 |
| 52 | 60 | 1 | 0 | 0.00 | 1 | 220.6 | 57 |
| 53 | 10 | 1 | 0 | 0.00 | 2 | 186.1 | 112 |
| 54 | 96 | 1 | 0 | 0.21 | 2 | 160.2 | 117 |
| 55 | 87 | 1 | 0 | 0.00 | 5 | 151.0 | 83 |
| 56 | 81 | 1 | 0 | 0.11 | 1 | 175.5 | 67 |
| 57 | 141 | 1 | 0 | 0.00 | 1 | 126.9 | 98 |
| 58 | 121 | 1 | 1 | 1.57 | 3 | 198.4 | 129 |
| 59 | 68 | 1 | 0 | 0.00 | 3 | 148.8 | 70 |
| 60 | 125 | 1 | 0 | 0.00 | 1 | 229.3 | 103 |
| 61 | 174 | 1 | 0 | 0.00 | 1 | 192.1 | 97 |
| 62 | 116 | 1 | 1 | 3.13 | 2 | 268.6 | 83 |
| 63 | 74 | 1 | 1 | 3.94 | 2 | 193.7 | 91 |
| 64 | 149 | 1 | 1 | 3.40 | 3 | 180.7 | 92 |
| 65 | 38 | 1 | 0 | 0.00 | 2 | 131.2 | 98 |
| 66 | 40 | 1 | 1 | 1.67 | 2 | 148.1 | 74 |
| 67 | 43 | 0 | 0 | 0.00 | 0 | 251.5 | 105 |
| 68 | 113 | 0 | 0 | 0.00 | 0 | 125.2 | 93 |
| 69 | 126 | 1 | 0 | 0.00 | 1 | 211.6 | 70 |
| 70 | 150 | 1 | 0 | 0.00 | 4 | 178.9 | 101 |
| 71 | 138 | 1 | 0 | 0.00 | 3 | 241.8 | 93 |
| 72 | 162 | 1 | 1 | 3.27 | 0 | 224.9 | 97 |
| 73 | 147 | 1 | 0 | 0.00 | 3 | 248.6 | 83 |
| 74 | 90 | 1 | 0 | 0.00 | 1 | 203.4 | 146 |
| 75 | 85 | 1 | 0 | 0.00 | 0 | 235.8 | 109 |
| 76 | 50 | 1 | 0 | 0.00 | 1 | 157.1 | 90 |
| 77 | 82 | 1 | 0 | 0.21 | 0 | 300.3 | 109 |
| 78 | 144 | 1 | 0 | 0.21 | 4 | 61.6 | 117 |
| 79 | 46 | 1 | 0 | 0.26 | 2 | 214.1 | 72 |
| 80 | 70 | 1 | 0 | 0.30 | 1 | 170.2 | 98 |
| 81 | 144 | 1 | 0 | 0.00 | 1 | 201.1 | 99 |
| 82 | 116 | 0 | 0 | 0.30 | 3 | 215.4 | 104 |
| 83 | 55 | 1 | 1 | 2.97 | 3 | 165.6 | 123 |
| 84 | 70 | 1 | 1 | 2.65 | 1 | 249.5 | 101 |
| 85 | 106 | 1 | 0 | 0.00 | 2 | 210.6 | 96 |
| 86 | 128 | 1 | 1 | 2.32 | 0 | 179.3 | 104 |
| 87 | 94 | 1 | 0 | 0.00 | 4 | 157.9 | 105 |
| 88 | 111 | 1 | 0 | 0.00 | 1 | 214.3 | 118 |
| 89 | 74 | 1 | 1 | 2.94 | 2 | 154.1 | 104 |
| 90 | 128 | 1 | 0 | 0.00 | 1 | 237.9 | 125 |
| 91 | 82 | 1 | 0 | 0.17 | 1 | 143.9 | 61 |
| 92 | 155 | 1 | 0 | 0.00 | 0 | 203.4 | 100 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 93 | 80 | 1 | 0 | 0.00 | 1 | 124.3 | 100 |
| 94 | 78 | 1 | 0 | 0.00 | 3 | 252.9 | 93 |
| 95 | 90 | 1 | 0 | 0.00 | 3 | 179.1 | 71 |
| 96 | 104 | 1 | 0 | 0.30 | 1 | 278.4 | 106 |
| 97 | 73 | 1 | 0 | 0.00 | 0 | 160.1 | 110 |
| 98 | 99 | 1 | 0 | 0.00 | 4 | 198.2 | 87 |
| 99 | 120 | 1 | 0 | 0.00 | 1 | 212.1 | 131 |
| 100 | 77 | 1 | 0 | 0.00 | 2 | 251.8 | 72 |

| | MonthlyCharge | OverageFee | RoamMins |
|---|---|---|---|
| 1 | 89.0 | 9.87 | 10.0 |
| 2 | 82.0 | 9.78 | 13.7 |
| 3 | 52.0 | 6.06 | 12.2 |
| 4 | 57.0 | 3.10 | 6.6 |
| 5 | 41.0 | 7.42 | 10.1 |
| 6 | 57.0 | 11.03 | 6.3 |
| 7 | 87.3 | 17.43 | 7.5 |
| 8 | 36.0 | 5.16 | 7.1 |
| 9 | 63.9 | 17.58 | 8.7 |
| 10 | 93.2 | 11.10 | 11.2 |
| 11 | 44.9 | 11.43 | 12.7 |
| 12 | 49.4 | 8.17 | 9.1 |
| 13 | 31.0 | 5.25 | 11.2 |
| 14 | 52.4 | 12.38 | 12.3 |
| 15 | 47.0 | 15.36 | 13.1 |
| 16 | 84.0 | 15.89 | 5.4 |
| 17 | 95.3 | 14.05 | 13.8 |
| 18 | 51.0 | 10.91 | 8.1 |
| 19 | 78.0 | 10.64 | 10.0 |
| 20 | 52.0 | 7.98 | 13.0 |
| 21 | 50.1 | 11.99 | 10.6 |
| 22 | 26.0 | 8.50 | 5.7 |
| 23 | 38.0 | 3.65 | 9.5 |
| 24 | 34.9 | 6.87 | 7.7 |
| 25 | 35.0 | 12.26 | 10.3 |
| 26 | 45.0 | 13.86 | 15.5 |
| 27 | 78.7 | 9.56 | 9.5 |
| 28 | 37.0 | 7.78 | 14.7 |
| 29 | 58.2 | 12.91 | 6.3 |
| 30 | 41.1 | 10.76 | 11.1 |
| 31 | 27.0 | 6.84 | 14.2 |
| 32 | 56.0 | 10.08 | 10.3 |
| 33 | 39.0 | 1.56 | 12.6 |
| 34 | 64.0 | 12.62 | 11.8 |
| 35 | 69.4 | 9.75 | 8.3 |
| 36 | 95.7 | 10.87 | 14.7 |
| 37 | 78.2 | 8.13 | 14.5 |
| 38 | 42.0 | 11.19 | 10.0 |
| 39 | 79.4 | 9.38 | 10.5 |
| 40 | 47.0 | 13.59 | 11.1 |
| 41 | 39.0 | 8.34 | 9.4 |
| 42 | 86.4 | 10.20 | 14.6 |
| 43 | 46.0 | 14.11 | 10.0 |
| 44 | 32.8 | 8.27 | 9.2 |
| 45 | 48.4 | 11.29 | 3.5 |
| 46 | 62.0 | 10.65 | 8.5 |
| 47 | 50.0 | 8.13 | 13.2 |
| 48 | 47.5 | 6.73 | 7.4 |
| 49 | 47.0 | 11.57 | 8.8 |
| 50 | 71.7 | 10.86 | 11.0 |
| 51 | 59.2 | 13.49 | 7.8 |
| 52 | 56.0 | 10.56 | 6.8 |
| 53 | 48.0 | 9.51 | 11.4 |
| 54 | 52.1 | 13.38 | 9.3 |
| 55 | 45.0 | 10.99 | 9.7 |
| 56 | 53.1 | 12.47 | 10.2 |
| 57 | 37.0 | 9.00 | 8.0 |
| 58 | 56.7 | 3.77 | 5.8 |
| 59 | 47.0 | 12.33 | 12.1 |
| 60 | 55.0 | 8.87 | 12.0 |

```
61           48.0       8.50     11.4
62           92.3       8.91     11.6
63           93.4      12.31     14.6
64           81.0       9.39     12.6
65           37.0       8.15      8.2
66           56.7       8.48      6.2
67           61.0      10.64      9.3
68           39.0      10.32      8.3
69           55.0      10.85      7.8
70           45.0       8.46     13.8
71           56.0       8.53     11.8
72           87.7       9.41     12.1
73           55.0       7.45      8.0
74           54.0      11.34      7.3
75           54.0       7.86     12.0
76           46.0      11.17      6.1
77           69.1       9.05     11.7
78           20.1       3.86      8.2
79           53.6       8.22      8.2
80           46.0       7.76     15.0
81           60.0      15.18     13.2
82           58.0      10.24     12.6
83           69.7       6.81     11.0
84           91.5      12.99      9.8
85           57.0      12.46     12.4
86           73.2      11.30      8.6
87           41.0       7.75      8.0
88           55.0      10.43     12.0
89           66.4       6.17     10.9
90           62.0      12.38     13.9
91           43.7       9.75     11.1
92           51.0       9.55      8.9
93           36.0       8.65      7.9
94           59.0       8.92      9.5
95           47.0       9.53     10.6
96           58.0       4.05      9.8
97           46.0      10.67     13.0
98           52.0      10.37      8.7
99           54.0      10.47      5.3
100          61.0      10.29      9.8
 [ reached 'max' / getOption("max.print") -- omitted 3233 rows ]

> > plot_correlation(cell)
> #dataplan and data usage are related;monthly charge related with data plan and data
usage
> ggplot(cell,aes(x=Churn))+geom_histogram(binwidth = 1,fill=c('Blue','red'))+labs(tit
le='Customer Churn',x='Churn',y='Freq')
> #determines the ratio of customers from the dataset who had churned
> #determines the ratio of customers from the dataset who had churned
> qplot(DayCalls,CustServCalls,data=cell,geom=c('point','smooth'),colour=Churn==1)
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
> #chrnrate
> churnrate<-table(cell$Churn)/nrow(cell)
> #cust serv calls above 2 and day calls above 50 are more likely to churn
> #chrnrate
> churnrate<-table(cell$Churn)/nrow(cell)
> churnrate

        0         1
0.8550855 0.1449145
> #outliers
> options(repr.plot.width=5,repr.plot.height=5)
> boxplot(DataUsage)$out
 [1] 5.40 4.64 4.73 4.46 4.56 4.56 4.56 4.46 4.75 4.59 4.48
> boxplot(AccountWeeks)$out
```

```
 [1] 208 215 209 224 243 217 210 212 232 225 225 224 212 210 217 209 221 209
> boxplot(DataUsage)$out
 [1] 5.40 4.64 4.73 4.46 4.56 4.56 4.56 4.46 4.75 4.59 4.48
> boxplot(AccountWeeks)$out
 [1] 208 215 209 224 243 217 210 212 232 225 225 224 212 210 217 209 221 209
> boxplot(CustServCalls)$out
  [1] 4 4 4 5 5 5 4 4 4 4 4 4 4 4 4 4 5 5 4 5 4 4 5 4 4 4 4 4 5 4 4 7 4 4 4 4 4 5 4
 [40] 4 4 4 4 5 4 7 4 9 5 4 4 5 4 4 5 5 4 6 4 6 5 5 5 6 5 4 4 5 4 4 7 4 6 5 4 4 4 6
 [79] 4 4 5 4 4 4 4 4 4 5 5 6 5 4 4 4 5 4 4 4 4 5 5 4 4 4 4 6 4 5 4 6 4 4 4 4 4 4 4
[118] 4 4 6 4 4 4 4 8 4 4 5 4 4 4 6 5 5 7 4 4 5 4 4 5 4 4 5 7 4 4 5 7 4 4 4 4 8 6 4
[157] 4 5 5 5 4 4 5 4 4 4 4 4 4 4 4 4 4 5 6 4 5 4 4 5 5 4 6 4 4 4 9 6 4 5 5 4 6 4 4
[196] 5 4 4 4 5 5 6 4 5 4 4 4 4 5 4 4 4 5 4 5 6 4 4 5 4 4 4 5 4 4 4 4 4 5 7 6 5 6 7
[235] 5 5 4 6 4 4 4 4 5 6 7 4 4 4 5 5 5 4 4 4 5 6 5 5 4 4 4 4 4 4 4 4 5
> boxplot(DayMins)$out
 [1] 332.9 337.4 326.5 350.8 335.5  30.9  34.0 334.3 346.8  12.5  25.9   0.0   0.0
[14]  19.5 329.8   7.9 328.1  27.0  17.6 326.3 345.3   2.6   7.8  18.9  29.9
> boxplot(MonthlyCharge)$out
 [1] 110.0 104.3 102.9 101.4 101.8 100.3 102.6 108.3 105.6 101.6 110.0 104.7 100.5
[14] 101.2 102.5 102.1 103.9  98.6 108.7 103.5 100.3 108.6 111.3 101.5 102.1 103.8
[27] 101.6 103.1 104.9 105.2 106.9 102.6 100.6 100.0
> boxplot(OverageFee)$out
 [1]  3.10 17.43 17.58  1.56 17.53  2.11 17.37  2.95  2.20  2.65  2.13  3.04  2.93
[14]  2.80  2.41  3.00 17.55  2.46 17.00 18.09 17.71 18.19  0.00 17.07
> boxplot(RoamMins)$out
 [1] 20.0  0.0 17.6  2.7 18.9  0.0 18.0  2.0  0.0 18.2  0.0  0.0  1.3  0.0  0.0  0.0
[17]  2.2 18.0  0.0 17.9  0.0 18.4  2.0 17.8  2.9  3.1 17.6  2.6  0.0  0.0 18.2  0.0
[33] 18.0  1.1  0.0 18.3  0.0  0.0  2.1  2.9  2.1  2.4  2.5  0.0  0.0 17.8
> boxplot(DataPlan~DataUsage)
> set.seed(300)
> pd<-sample(2,nrow(cell),replace=TRUE, prob=c(0.65,0.35))
>
> train<-cell[pd==1,]
> val<-cell[pd==2,]
>
> sum(cell$Churn)
[1] 483
> sum(val$Churn)
[1] 184
> sum(train$Churn)
[1] 299
>
> train.reg<-train[,c(1:11)]
> val.reg<-val[,c(1:11)]
> str(train.reg)
'data.frame':  2086 obs. of  11 variables:
 $ Churn          : int  0 0 0 0 0 0 0 1 0 0 ...
 $ AccountWeeks   : int  118 147 117 74 95 76 73 77 130 111 ...
 $ ContractRenewal: int  0 0 1 1 1 1 1 1 1 1 ...
 $ DataPlan       : int  0 0 0 0 0 1 0 0 0 0 ...
 $ DataUsage      : num  0 0 0.19 0.34 0.44 2.7 0 0 0 0.39 ...
 $ CustServCalls  : int  0 0 1 0 3 1 1 5 0 2 ...
 $ DayMins        : num  223 157 184 188 157 ...
 $ DayCalls       : int  98 79 97 127 88 66 90 89 112 103 ...
 $ MonthlyCharge  : num  57 36 63.9 49.4 52.4 78 52 26 38 34.9 ...
 $ OverageFee     : num  11.03 5.16 17.58 8.17 12.38 ...
 $ RoamMins       : num  6.3 7.1 8.7 9.1 12.3 10 13 5.7 9.5 7.7 ...
> OLS.full<-lm(MonthlyCharge~Churn+AccountWeeks+ContractRenewal+DataUsage+CustServCall
s+DayMins+DayCalls+OverageFee+RoamMins+DataPlan, data=train.reg)
> summary(OLS.full)

Call:
```

```
lm(formula = MonthlyCharge ~ Churn + AccountWeeks + ContractRenewal +
    DataUsage + CustServCalls + DayMins + DayCalls + OverageFee +
    RoamMins + DataPlan, data = train.reg)

Residuals:
     Min       1Q   Median       3Q      Max
-0.53998 -0.24526 -0.00055  0.24490  0.56626

Coefficients:
                 Estimate Std. Error  t value Pr(>|t|)
(Intercept)      4.330e-01  5.912e-02    7.325  3.4e-13 ***
Churn           -1.891e-02  1.966e-02   -0.962    0.336
AccountWeeks    -7.551e-05  1.580e-04   -0.478    0.633
ContractRenewal  3.322e-02  2.201e-02    1.510    0.131
DataUsage        1.002e+01  1.736e-02  577.269  < 2e-16 ***
CustServCalls   -9.965e-04  4.882e-03   -0.204    0.838
DayMins          1.700e-01  1.186e-04 1433.815  < 2e-16 ***
DayCalls         1.794e-04  3.160e-04    0.568    0.570
OverageFee       1.705e+00  2.526e-03  674.729  < 2e-16 ***
RoamMins        -2.432e-03  2.650e-03   -0.918    0.359
DataPlan        -6.396e-02  4.860e-02   -1.316    0.188
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2884 on 2075 degrees of freedom
Multiple R-squared:  0.9997,   Adjusted R-squared:  0.9997
F-statistic: 6.751e+05 on 10 and 2075 DF,  p-value: < 2.2e-16

> vif(OLS.full)
          Churn    AccountWeeks ContractRenewal       DataUsage   CustServCalls
       1.189908        1.003706        1.076146       12.141706        1.053594
        DayMins         DayCalls      OverageFee        RoamMins        DataPlan
       1.037198        1.006778        1.014882        1.391767       11.815011
> #removing data usage from the dataset as its showing multicollinearity with dataplan
and monthlycharge using corrplot
> #final_data=MonthlyCharge~Churn+AccountWeeks+ContractRenewal+CustServCalls+DayMins+D
ayCalls+OverageFee+RoamMins+DataPlan
>
> OLS.full1<-lm(MonthlyCharge~Churn+AccountWeeks+ContractRenewal+CustServCalls+DayMins
+DayCalls+OverageFee+RoamMins+DataPlan, data=train.reg)
> summary(OLS.full1)

Call:
lm(formula = MonthlyCharge ~ Churn + AccountWeeks + ContractRenewal +
    CustServCalls + DayMins + DayCalls + OverageFee + RoamMins +
    DataPlan, data = train.reg)

Residuals:
     Min       1Q   Median       3Q      Max
-19.5556  -2.1219  -0.1293   2.1120  18.9545

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -7.124450   0.732684  -9.724   <2e-16 ***
Churn           -0.219505   0.249793  -0.879   0.3796
AccountWeeks     0.002817   0.002007   1.403   0.1607
ContractRenewal -0.584968   0.279344  -2.094   0.0364 *
CustServCalls    0.001197   0.062049   0.019   0.9846
DayMins          0.171541   0.001507 113.856   <2e-16 ***
DayCalls         0.001753   0.004016   0.437   0.6625
OverageFee       1.700348   0.032106  52.960   <2e-16 ***
```

```
RoamMins         0.797009   0.028712  27.759   <2e-16 ***
DataPlan        26.753121   0.181594 147.324   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.665 on 2076 degrees of freedom
Multiple R-squared:  0.9503,   Adjusted R-squared:  0.9501
F-statistic:  4415 on 9 and 2076 DF,  p-value: < 2.2e-16

> vif(OLS.full1)
          Churn    AccountWeeks ContractRenewal   CustServCalls         DayMins
       1.189536        1.002697        1.073598        1.053593        1.036685
       DayCalls      OverageFee        RoamMins        DataPlan
       1.006703        1.014873        1.011628        1.021151
> pred.reg<-predict(OLS.full1,newdata=val.reg, interval="predict")
> pred.reg
           fit      lwr       upr
1     89.82636 82.62543  97.02729
2     74.83141 67.63095  82.03186
3     54.65706 47.45411  61.86001
4     55.12977 47.89071  62.36883
5     42.55079 35.33446  49.76713
7     92.58730 85.37118  99.80342
10    92.33398 85.11399  99.55398
11    44.20211 36.98607  51.41815
13    32.88111 25.67715  40.08507
15    49.85591 42.64101  57.07080
16    81.07522 73.83578  88.31465
17    88.10732 80.89757  95.31707
18    50.47546 43.27955  57.67136
21    48.35126 41.15289  55.54963
28    40.55697 33.35141  47.76254
29    52.10350 44.89690  59.31010
32    56.82127 49.62746  64.01508
33    42.05235 34.83110  49.27360
38    42.05825 34.85896  49.25754
40    48.91516 41.71603  56.11429
43    45.90285 38.69908  53.10662
44    28.99242 21.78706  36.19777
46    65.00496 57.80542  72.20449
50    69.63697 62.43469  76.83925
52    53.77802 46.57307  60.98296
54    50.41224 43.21566  57.60881
55    44.78815 37.57558  52.00072
58    64.46150 57.22903  71.69397
59    48.74285 41.54139  55.94431
69    53.73294 46.53413  60.93176
70    48.74781 41.53671  55.95891
72    83.89374 76.69021  91.09727
78    16.35235  9.11859  23.58610
80    47.00690 39.80792  54.20589
81    63.69966 56.49551  70.90382
84    92.11687 84.91534  99.31839
87    39.16484 31.95468  46.37499
88    56.87131 49.67674  64.06589
89    65.04991 57.84925  72.25056
91    42.73965 35.54024  49.93906
93    35.01938 27.82428  42.21449
95    48.04778 40.85017  55.24539
97    48.65663 41.45979  55.85346
101   72.12506 64.91335  79.33677
```

| | | | |
|---|---|---|---|
| 103 | 37.32304 | 30.12176 | 44.52433 |
| 104 | 41.46279 | 34.26892 | 48.65666 |
| 106 | 58.84081 | 51.63370 | 66.04793 |
| 118 | 67.13666 | 59.92746 | 74.34586 |
| 123 | 48.94835 | 41.75384 | 56.14286 |
| 125 | 36.02082 | 28.82149 | 43.22015 |
| 126 | 56.23784 | 49.03366 | 63.44202 |
| 128 | 68.94659 | 61.72280 | 76.17037 |
| 129 | 32.24908 | 25.05279 | 39.44537 |
| 132 | 39.56099 | 32.36147 | 46.76052 |
| 133 | 59.49870 | 52.30314 | 66.69426 |
| 134 | 49.81860 | 42.61805 | 57.01916 |
| 138 | 55.01110 | 47.81565 | 62.20655 |
| 142 | 80.96703 | 73.76446 | 88.16960 |
| 148 | 67.55925 | 60.34215 | 74.77634 |
| 150 | 86.46734 | 79.26443 | 93.67025 |
| 151 | 53.29945 | 46.10704 | 60.49186 |
| 153 | 44.28713 | 37.09062 | 51.48365 |
| 156 | 56.22644 | 49.02311 | 63.42977 |
| 160 | 67.20569 | 60.00664 | 74.40475 |
| 161 | 36.38552 | 29.18813 | 43.58290 |
| 164 | 30.84729 | 23.64708 | 38.04750 |
| 166 | 88.98870 | 81.78927 | 96.18814 |
| 168 | 56.90241 | 49.70631 | 64.09850 |
| 170 | 53.09552 | 45.89136 | 60.29967 |
| 171 | 62.87022 | 55.67253 | 70.06791 |
| 173 | 39.34026 | 32.13832 | 46.54221 |
| 176 | 59.28611 | 52.08998 | 66.48224 |
| 180 | 57.37903 | 50.15178 | 64.60628 |
| 182 | 23.86528 | 16.63940 | 31.09116 |
| 183 | 67.36711 | 60.16680 | 74.56743 |
| 185 | 65.58485 | 58.35406 | 72.81565 |
| 186 | 60.95626 | 53.75092 | 68.16160 |
| 189 | 41.54574 | 34.34831 | 48.74318 |
| 191 | 38.14071 | 30.94431 | 45.33712 |
| 192 | 38.48584 | 31.28536 | 45.68632 |
| 195 | 90.86506 | 83.65057 | 98.07956 |
| 197 | 41.61979 | 34.42357 | 48.81601 |
| 198 | 72.31002 | 65.07856 | 79.54149 |
| 199 | 98.18960 | 90.95971 | 105.41949 |
| 200 | 32.22735 | 25.01932 | 39.43538 |
| 201 | 73.64177 | 66.42931 | 80.85422 |
| 202 | 60.74585 | 53.54043 | 67.95128 |
| 204 | 37.50228 | 30.30702 | 44.69755 |
| 208 | 50.40256 | 43.20432 | 57.60080 |
| 210 | 63.60316 | 56.39847 | 70.80784 |
| 213 | 75.45199 | 68.25035 | 82.65362 |
| 214 | 67.61161 | 60.40330 | 74.81993 |
| 216 | 47.82621 | 40.63158 | 55.02084 |
| 219 | 68.66484 | 61.45418 | 75.87549 |
| 220 | 46.08265 | 38.87910 | 53.28620 |
| 222 | 71.05184 | 63.85451 | 78.24918 |
| 223 | 42.73457 | 35.52888 | 49.94027 |
| 225 | 29.22835 | 22.02617 | 36.43053 |
| 229 | 97.00075 | 89.78372 | 104.21778 |
| 232 | 87.67803 | 80.45795 | 94.89812 |
| 234 | 48.99131 | 41.79478 | 56.18784 |
| 237 | 71.56389 | 64.36319 | 78.76460 |
| 241 | 48.40562 | 41.19869 | 55.61255 |
| 248 | 50.31117 | 43.11238 | 57.50997 |
| 252 | 54.51680 | 47.31297 | 61.72063 |

| | | | |
|---|---|---|---|
| 255 | 49.90151 | 42.68483 | 57.11819 |
| 256 | 77.67924 | 70.47059 | 84.88789 |
| 259 | 27.49250 | 20.26637 | 34.71863 |
| 265 | 60.99792 | 53.78765 | 68.20820 |
| 266 | 86.06671 | 78.86619 | 93.26722 |
| 273 | 82.47265 | 75.26826 | 89.67703 |
| 274 | 56.23035 | 49.03165 | 63.42905 |
| 275 | 69.91317 | 62.71712 | 77.10921 |
| 279 | 80.52140 | 73.32332 | 87.71949 |
| 280 | 55.59448 | 48.37251 | 62.81645 |
| 281 | 47.57650 | 40.38063 | 54.77237 |
| 282 | 52.94550 | 45.75277 | 60.13823 |
| 283 | 79.02328 | 71.82130 | 86.22526 |
| 284 | 89.12248 | 81.91789 | 96.32707 |
| 285 | 47.41062 | 40.21290 | 54.60835 |
| 287 | 46.19475 | 38.99585 | 53.39365 |
| 292 | 28.63882 | 21.43693 | 35.84070 |
| 294 | 67.31707 | 60.09378 | 74.54036 |
| 297 | 36.46511 | 29.26530 | 43.66491 |
| 298 | 44.79362 | 37.59333 | 51.99391 |
| 301 | 40.94589 | 33.74553 | 48.14625 |
| 302 | 65.56366 | 58.34977 | 72.77754 |
| 303 | 60.56545 | 53.34398 | 67.78692 |
| 304 | 52.15502 | 44.95305 | 59.35699 |
| 305 | 38.43137 | 31.23091 | 45.63182 |
| 307 | 68.87176 | 61.66159 | 76.08192 |
| 309 | 51.40393 | 44.20969 | 58.59818 |
| 310 | 61.14328 | 53.92887 | 68.35770 |
| 311 | 64.32464 | 57.11651 | 71.53277 |
| 312 | 84.46313 | 77.26577 | 91.66049 |
| 313 | 50.83183 | 43.62524 | 58.03842 |
| 315 | 46.82767 | 39.61505 | 54.04030 |
| 317 | 69.40246 | 62.20058 | 76.60433 |
| 319 | 81.57777 | 74.36668 | 88.78886 |
| 320 | 81.79671 | 74.57608 | 89.01734 |
| 321 | 49.89199 | 42.69036 | 57.09361 |
| 322 | 66.37896 | 59.17803 | 73.57989 |
| 324 | 37.71907 | 30.51367 | 44.92448 |
| 331 | 58.65259 | 51.44779 | 65.85739 |
| 333 | 41.03151 | 33.80175 | 48.26126 |
| 341 | 51.23852 | 44.02611 | 58.45093 |
| 349 | 70.48588 | 63.28274 | 77.68903 |
| 351 | 50.36792 | 43.16785 | 57.56798 |
| 354 | 53.08540 | 45.88850 | 60.28229 |
| 357 | 60.39849 | 53.18457 | 67.61242 |
| 364 | 49.89685 | 42.69564 | 57.09805 |
| 369 | 32.96128 | 25.76076 | 40.16179 |
| 377 | 41.05752 | 33.85895 | 48.25609 |
| 379 | 54.95973 | 47.74151 | 62.17796 |
| 380 | 49.82706 | 42.63013 | 57.02399 |
| 381 | 62.37140 | 55.16730 | 69.57550 |
| 387 | 48.23845 | 41.03446 | 55.44245 |
| 388 | 52.90126 | 45.70618 | 60.09634 |
| 389 | 41.69465 | 34.49583 | 48.89347 |
| 390 | 42.91906 | 35.72006 | 50.11807 |
| 391 | 51.09363 | 43.89410 | 58.29316 |
| 392 | 39.61556 | 32.40731 | 46.82382 |
| 399 | 41.78412 | 34.58537 | 48.98288 |
| 401 | 68.44442 | 61.24621 | 75.64263 |
| 407 | 32.28607 | 25.07521 | 39.49693 |
| 417 | 40.46886 | 33.22958 | 47.70814 |

| | | | |
|---|---|---|---|
| 425 | 77.64220 | 70.44159 | 84.84282 |
| 427 | 33.57728 | 26.37099 | 40.78357 |
| 428 | 63.39129 | 56.18691 | 70.59567 |
| 432 | 56.57806 | 49.37852 | 63.77760 |
| 433 | 60.18723 | 52.98805 | 67.38641 |
| 435 | 93.09735 | 85.89108 | 100.30362 |
| 436 | 79.63058 | 72.42446 | 86.83671 |
| 438 | 61.82067 | 54.60440 | 69.03694 |
| 442 | 51.85336 | 44.65495 | 59.05176 |
| 465 | 78.17709 | 70.97816 | 85.37602 |
| 467 | 81.99052 | 74.77902 | 89.20203 |
| 474 | 67.46204 | 60.22541 | 74.69867 |
| 476 | 38.52502 | 31.32821 | 45.72183 |
| 481 | 54.88897 | 47.68986 | 62.08808 |
| 482 | 38.93297 | 31.73306 | 46.13288 |
| 483 | 40.64315 | 33.44619 | 47.84011 |
| 484 | 50.88477 | 43.68765 | 58.08189 |
| 488 | 60.95336 | 53.75251 | 68.15421 |
| 489 | 52.48143 | 45.25773 | 59.70513 |
| 490 | 48.07459 | 40.87863 | 55.27055 |
| 491 | 60.21397 | 53.01426 | 67.41368 |
| 492 | 59.06913 | 51.85021 | 66.28806 |
| 496 | 44.49280 | 37.27736 | 51.70824 |
| 497 | 71.75757 | 64.55571 | 78.95943 |
| 498 | 57.22858 | 50.02802 | 64.42913 |
| 499 | 45.93449 | 38.70276 | 53.16622 |
| 501 | 65.41780 | 58.21602 | 72.61958 |
| 503 | 61.81367 | 54.58463 | 69.04270 |
| 509 | 79.69739 | 72.49508 | 86.89969 |
| 510 | 73.28240 | 66.06390 | 80.50091 |
| 511 | 68.10018 | 60.89308 | 75.30727 |
| 512 | 62.23079 | 55.02581 | 69.43578 |
| 514 | 79.98797 | 72.77954 | 87.19640 |
| 515 | 94.89395 | 87.66685 | 102.12105 |
| 516 | 66.18224 | 58.96306 | 73.40142 |
| 517 | 41.75866 | 34.55381 | 48.96351 |
| 523 | 89.88143 | 82.64785 | 97.11502 |
| 528 | 47.37396 | 40.15800 | 54.58993 |
| 529 | 34.30009 | 27.09744 | 41.50273 |
| 530 | 43.41232 | 36.18011 | 50.64453 |
| 531 | 70.85266 | 63.65142 | 78.05391 |
| 545 | 67.49623 | 60.29289 | 74.69957 |
| 547 | 68.85499 | 61.63558 | 76.07440 |
| 552 | 57.67788 | 50.48039 | 64.87537 |
| 557 | 55.52523 | 48.31991 | 62.73055 |
| 560 | 33.19159 | 25.99443 | 40.38875 |
| 565 | 87.02542 | 79.82412 | 94.22673 |
| 566 | 48.71004 | 41.50669 | 55.91339 |
| 567 | 72.34959 | 65.14820 | 79.55097 |
| 570 | 92.75766 | 85.52796 | 99.98736 |
| 571 | 48.71712 | 41.52274 | 55.91150 |
| 578 | 43.12417 | 35.93093 | 50.31741 |
| 581 | 64.00893 | 56.77971 | 71.23815 |
| 584 | 43.33950 | 36.13985 | 50.53915 |
| 590 | 58.74715 | 51.54387 | 65.95042 |
| 595 | 51.80987 | 44.60983 | 59.00992 |
| 598 | 36.01281 | 28.80656 | 43.21907 |
| 601 | 38.50350 | 31.30648 | 45.70051 |
| 602 | 48.25444 | 41.03332 | 55.47556 |
| 603 | 45.43633 | 38.23480 | 52.63786 |
| 608 | 59.18058 | 51.97526 | 66.38591 |

| | | | |
|---|---|---|---|
| 610 | 46.95708 | 39.75081 | 54.16336 |
| 612 | 49.15001 | 41.95392 | 56.34610 |
| 614 | 86.74631 | 79.52607 | 93.96655 |
| 616 | 72.27635 | 65.07661 | 79.47609 |
| 621 | 52.41783 | 45.21943 | 59.61623 |
| 623 | 76.36193 | 69.15942 | 83.56444 |
| 627 | 76.67118 | 69.43864 | 83.90371 |
| 629 | 35.80367 | 28.60484 | 43.00250 |
| 631 | 51.90512 | 44.70610 | 59.10413 |
| 632 | 37.37989 | 30.17654 | 44.58324 |
| 642 | 45.76150 | 38.56752 | 52.95549 |
| 650 | 60.04628 | 52.83604 | 67.25653 |
| 651 | 64.25932 | 57.05853 | 71.46012 |
| 652 | 54.36988 | 47.17734 | 61.56243 |
| 653 | 73.13969 | 65.92733 | 80.35204 |
| 654 | 80.22422 | 73.00963 | 87.43882 |
| 655 | 75.99486 | 68.79663 | 83.19308 |
| 657 | 43.99386 | 36.78777 | 51.19995 |
| 661 | 71.40908 | 64.19202 | 78.62615 |
| 662 | 90.26352 | 83.05384 | 97.47320 |
| 671 | 97.82545 | 90.61072 | 105.04018 |
| 672 | 67.84296 | 60.63284 | 75.05309 |
| 673 | 44.65412 | 37.45532 | 51.85291 |
| 675 | 48.20270 | 41.00823 | 55.39717 |
| 680 | 65.89066 | 58.66501 | 73.11630 |
| 681 | 51.86046 | 44.66590 | 59.05502 |
| 682 | 34.76294 | 27.56631 | 41.95958 |
| 683 | 85.67558 | 78.46311 | 92.88804 |
| 684 | 45.81599 | 38.61928 | 53.01269 |
| 685 | 38.62292 | 31.42447 | 45.82138 |
| 687 | 57.48864 | 50.28356 | 64.69372 |
| 691 | 53.95183 | 46.75444 | 61.14922 |
| 692 | 42.65800 | 35.46323 | 49.85277 |
| 693 | 52.52055 | 45.29671 | 59.74439 |
| 694 | 68.56713 | 61.36784 | 75.76643 |
| 699 | 50.39643 | 43.19683 | 57.59603 |
| 703 | 50.74283 | 43.54666 | 57.93900 |
| 706 | 29.91617 | 22.71413 | 37.11822 |
| 710 | 48.38830 | 41.19124 | 55.58535 |
| 711 | 54.50761 | 47.30700 | 61.70823 |
| 712 | 51.91313 | 44.70531 | 59.12096 |
| 715 | 73.64094 | 66.43489 | 80.84700 |
| 716 | 68.83530 | 61.61646 | 76.05413 |
| 724 | 47.63299 | 40.43320 | 54.83278 |
| 726 | 44.74484 | 37.55026 | 51.93942 |
| 729 | 92.64151 | 85.43789 | 99.84514 |
| 730 | 78.97464 | 71.77175 | 86.17753 |
| 736 | 25.48900 | 18.25940 | 32.71861 |
| 737 | 42.17099 | 34.95658 | 49.38540 |
| 739 | 49.61787 | 42.42117 | 56.81458 |
| 742 | 31.39353 | 24.17640 | 38.61065 |
| 749 | 60.56554 | 53.36710 | 67.76399 |
| 751 | 68.33081 | 61.12821 | 75.53342 |
| 754 | 31.96716 | 24.76984 | 39.16449 |
| 762 | 53.11836 | 45.91593 | 60.32079 |
| 763 | 49.76094 | 42.52404 | 56.99783 |
| 767 | 62.28329 | 55.06273 | 69.50384 |
| 769 | 58.96437 | 51.76715 | 66.16159 |
| 770 | 54.90648 | 47.71265 | 62.10031 |
| 772 | 62.98505 | 55.74961 | 70.22048 |
| 773 | 50.14196 | 42.94699 | 57.33694 |

```
774   43.97333 36.75542  51.19124
777   85.21318 78.00534  92.42103
780   59.81732 52.59919  67.03546
782   46.82317 39.60717  54.03918
788   64.45161 57.23966  71.66356
792   51.77597 44.56971  58.98223
793   94.42883 87.20481 101.65285
797   68.16136 60.96124  75.36149
801   92.10235 84.90225  99.30246
803   57.72993 50.52253  64.93733
804   60.15473 52.95647  67.35298
805   62.48478 55.28305  69.68650
809   74.18131 66.96280  81.39982
810   76.77921 69.57334  83.98508
811   56.34211 49.14638  63.53784
814   58.46277 51.25448  65.67106
815   69.08796 61.88853  76.28739
816   65.40930 58.19218  72.62642
819   54.45041 47.24783  61.65300
821   67.65971 60.45221  74.86722
823   45.79707 38.60227  52.99186
825   49.08958 41.89175  56.28740
830   54.29292 47.09081  61.49502
831   40.58725 33.36243  47.81207
832   55.21371 48.01955  62.40787
833   72.71282 65.50139  79.92425
835   23.21613 16.01126  30.42099
836   30.59539 23.38928  37.80150
841   40.01601 32.81419  47.21784
843   42.97266 35.75138  50.19394
844   54.86037 47.65496  62.06578
845   75.38174 68.17811  82.58536
847   47.93181 40.72670  55.13693
848   65.92279 58.72496  73.12062
849   57.21706 50.01602  64.41810
851   53.19805 45.97978  60.41631
855   85.63034 78.42848  92.83220
857   44.70497 37.50393  51.90600
861   46.48920 39.29182  53.68659
867   89.00797 81.80306  96.21289
868   36.56116 29.36313  43.75920
869   49.98400 42.77675  57.19124
873   78.59206 71.39104  85.79309
876   74.29443 67.09282  81.49604
882   36.53291 29.33386  43.73197
 [ reached getOption("max.print") -- omitted 914 rows ]
> View(val.reg)
> #mean sq error
> mse1<-mean((val.reg$MonthlyCharge-pred.reg)^2)
> mse1
[1] 46.42296
> actual_pred_OLS.full1=data.frame(val.reg$MonthlyCharge,pred.reg)



>logit<-glm(Churn~MonthlyCharge+AccountWeeks+ContractRenewal+CustServCalls
+            +DayMins+DayCalls+OverageFee+RoamMins+DataPlan, data=train.logit,
+            family=binomial())
> summary(logit)

Call:
```

```
glm(formula = Churn ~ MonthlyCharge + AccountWeeks + ContractRenewal +
    CustServCalls + DayMins + DayCalls + OverageFee + RoamMins +
    DataPlan, family = binomial(), data = train.logit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8109  -0.5280  -0.3583  -0.2034   3.0319

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     -6.1463920  0.6796920  -9.043  < 2e-16 ***
MonthlyCharge    0.0238460  0.0234461   1.017 0.309127
AccountWeeks     0.0006455  0.0017355   0.372 0.709962
ContractRenewal -1.9238236  0.1833008 -10.495  < 2e-16 ***
CustServCalls    0.4813163  0.0488911   9.845  < 2e-16 ***
DayMins          0.0070472  0.0041930   1.681 0.092817 .
DayCalls         0.0039802  0.0034749   1.145 0.252033
OverageFee       0.1219375  0.0487607   2.501 0.012394 *
RoamMins         0.1034138  0.0280041   3.693 0.000222 ***
DataPlan        -1.8199445  0.7118565  -2.557 0.010570 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1714.6  on 2085  degrees of freedom
Residual deviance: 1378.4  on 2076  degrees of freedom
AIC: 1398.4

Number of Fisher Scoring iterations: 6

> glm_pred=fitted(logit)
> glmreg_actual_pred<-data.frame(Actual=train.logit$Churn,Predicted=glm_pred)
> View(glmreg_actual_pred)
> #check goodness of fir using Pseudo r2
> library(pscl)
> pR2(logit)
         llh      llhNull           G2     McFadden         r2ML         r2CU
-689.1762979 -857.2924437  336.2322916    0.1961013    0.1488656    0.2656294
> #the independent var explain 19.38% of dependent var(McFadden value)
> logit<-glm(Churn~MonthlyCharge+AccountWeeks+ContractRenewal+CustServCalls
+            +DayMins+DayCalls+OverageFee+RoamMins+DataPlan, data=train.logit,
+            family=binomial())
> #visual pattern for cont. var
> pairs(cell[,c(2,5,7,8,9,10,11)])
> ct.data<-subset(cell,select=c(ContractRenewal,DataPlan))
> num.data<-subset(cell,select=c(AccountWeeks,DataUsage,CustServCalls,DayMins,DayCalls
,MonthlyCharge,OverageFee,RoamMins))
> par(mfrow=c(2,3))
> for(i in names(ct.data)){
+   print(i)
+   print(table(cell$Churn,ct.data[[i]]))
+   barplot(table(cell$Churn,ct.data[[i]]),
+           col=c('grey','red'),main=names(ct.data[i]))
+ }
[1] "ContractRenewal"

       0    1
  0  186 2664
  1  137  346
[1] "DataPlan"

       0    1
  0 2008  842
  1  403   80

> > #perform chi sq
> ChiSqStat<-NA
```

```
> for(i in 1 :(ncol(fact_ct.data2))){
+   Statistic<-data.frame(
+     'Row'=colnames(fact_ct.data2[1]),
+     'Column'=colnames(fact_ct.data2[i]),
+     'Chi Square'=chisq.test(fact_ct.data2[[1]],fact_ct.data2[[i]])$statistic,
+     'df'=chisq.test(fact_ct.data2[[1]],fact_ct.data2[[i]])$parameter,
+     'p.value'=chisq.test(fact_ct.data2[[1]],fact_ct.data2[[i]])$p.value)
+   ChiSqStat<-rbind(ChiSqStat,Statistic)
+ }
> ChiSqStat<-data.table::data.table(ChiSqStat)
> ChiSqStat
     Row          Column Chi.Square df      p.value
1:  <NA>            <NA>         NA NA           NA
2: Churn           Churn 3324.93479  1 0.000000e+00
3: Churn ContractRenewal  222.56576  1 2.493108e-50
4: Churn        DataPlan   34.13166  1 5.150640e-09
> #running a sample model with factor var alone
> fact.model<-glm(Churn~.,data=fact_ct.data2,family = binomial)
> summary(fact.model)

Call:
glm(formula = Churn ~ ., family = binomial, data = fact_ct.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1402  -0.5369  -0.5369  -0.3648   2.3423

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.08805    0.11897  -0.740    0.459
ContractRenewal1 -1.77617    0.12839 -13.835  < 2e-16 ***
DataPlan1        -0.81253    0.13366  -6.079 1.21e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2546.0  on 3330  degrees of freedom
AIC: 2552

Number of Fisher Scoring iterations: 5

> ##working on numerical variables
> ##boxplots of all numerical variables
> library(RColorBrewer)
> boxplot(num.data,las=1,horizontal=TRUE,cex=0.8,par(cex.axis=0.8),col=brewer.pal(8,'S
et1'),main='boxplot of cont var')
> boxplot(num.data,las=1,horizontal=TRUE,cex=0.8,par(cex.axis=0.8),col=brewer.pal(8,'S
et1'),main='boxplot of cont var')
> boxplot(num.data,las=1,horizontal=TRUE,cex=0.8,par(cex.axis=0.8),col=brewer.pal(8,'S
et1'),main='boxplot of continuous variables')
> #add churn to the dataset
> num.data2=cbind(cell$Churn,num.data)
> colnames(num.data2)[1]<-'Churn'
> num.data2$Churn=as.factor(num.data2$Churn)
> #build univariate logistic reg. models
> ##AccountWeeks,DataUsage,CustServCalls,DayMins,DayCalls,MonthlyCharge,OverageFee,Roa
mMins
> mod.num<-glm(Churn~AccountWeeks,data=num.data2,family=binomial)
> summary(mod.num)
```

```
Call:
glm(formula = Churn ~ AccountWeeks, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6041  -0.5658  -0.5566  -0.5452   2.0169

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.894953   0.135634 -13.971   <2e-16 ***
AccountWeeks  0.001179   0.001234   0.955     0.34
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2757.4  on 3331  degrees of freedom
AIC: 2761.4

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~DataUsage,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ DataUsage, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6012  -0.6012  -0.5853  -0.4422   2.4047

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.61888    0.05594 -28.941  < 2e-16 ***
DataUsage   -0.22506    0.04531  -4.967  6.8e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2730.5  on 3331  degrees of freedom
AIC: 2734.5

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~CustServCalls,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ CustServCalls, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4760  -0.5799  -0.4820  -0.3991   2.2671

Coefficients:
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.49016    0.08631  -28.85   <2e-16 ***
CustServCalls  0.39617    0.03456   11.46   <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: 2758.3  on 3332  degrees of freedom
Residual deviance: 2627.2  on 3331  degrees of freedom
AIC: 2631.2

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~DayMins,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ DayMins, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0241  -0.6001  -0.4902  -0.3738   2.8102

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.929289   0.202823  -19.37   <2e-16 ***
DayMins      0.011272   0.000975   11.56   <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: 2758.3  on 3332  degrees of freedom
Residual deviance: 2614.3  on 3331  degrees of freedom
AIC: 2618.3

Number of Fisher Scoring iterations: 5

>
> mod.num<-glm(Churn~DayCalls,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ DayCalls, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6031  -0.5665  -0.5563  -0.5443   2.0792

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.039138   0.253579  -8.041 8.88e-16 ***
DayCalls     0.002620   0.002458   1.066    0.287
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
```

```
Residual deviance: 2757.2  on 3331  degrees of freedom
AIC: 2761.2

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~MonthlyCharge,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ MonthlyCharge, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7498  -0.5707  -0.5366  -0.5043   2.1888

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.468836   0.177192  -13.93  < 2e-16 ***
MonthlyCharge  0.012072   0.002902    4.16 3.19e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2741.3  on 3331  degrees of freedom
AIC: 2745.3

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~OverageFee,data=num.data2,family=binomial)
> summary(mod.num)

Call:
glm(formula = Churn ~ OverageFee, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8069  -0.5874  -0.5366  -0.4781   2.2644

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.85680    0.21318 -13.401  < 2e-16 ***
OverageFee   0.10513    0.01971   5.335 9.56e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2729.4  on 3331  degrees of freedom
AIC: 2733.4

Number of Fisher Scoring iterations: 4

>
> mod.num<-glm(Churn~RoamMins,data=num.data2,family=binomial)
> summary(mod.num)
```

```
Call:
glm(formula = Churn ~ RoamMins, family = binomial, data = num.data2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7338  -0.5814  -0.5463  -0.4995   2.2190

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.51472    0.19778 -12.715  < 2e-16 ***
RoamMins     0.07091    0.01803   3.932 8.41e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2742.6  on 3331  degrees of freedom
AIC: 2746.6

Number of Fisher Scoring iterations: 4

>> #removing account week and day calls freom the data set
> names(num.data2)
[1] "Churn"         "AccountWeeks"  "DataUsage"     "CustServCalls" "DayMins"
[6] "DayCalls"      "MonthlyCharge" "OverageFee"    "RoamMins"
> num.data2<-subset(num.data2,select= -c(AccountWeeks,DayCalls,Churn))#Churn is remove
d as it is coming twice from fact_ct.data2 and num.data2
> full.data<-cbind(num.data2,fact_ct.data2)
> names(full.data)
[1] "DataUsage"       "CustServCalls"   "DayMins"         "MonthlyCharge"
[5] "OverageFee"      "RoamMins"        "Churn"           "ContractRenewal"
[9] "DataPlan"
> full.data<-full.data[,c(7,1,2,3,4,5,6,8,9)]
> names(full.data)
[1] "Churn"           "DataUsage"       "CustServCalls"   "DayMins"
[5] "MonthlyCharge"   "OverageFee"      "RoamMins"        "ContractRenewal"
[9] "DataPlan"
> library(caTools)
> spl=sample.split(full.data$Churn,SplitRatio = 0.65)
> train=subset(full.data,spl==T)
> test=subset(full.data,spl==F)
> #chk % of the sample
> sum(as.integer(as.character(train$Churn)))/nrow(train)
[1] 0.1449677
> sum(as.integer(as.character(test$Churn)))/nrow(test)
[1] 0.1448158
> sum(as.integer(as.character(full.data$Churn)))/nrow(full.data)
[1] 0.1449145
> #build model
> LRmodel=glm(Churn~.,data=train,family = binomial)
> summary(LRmodel)

Call:
glm(formula = Churn ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0057  -0.5197  -0.3491  -0.2091   2.9887

Coefficients:
```

```
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.64850    0.56439 -10.008  < 2e-16 ***
DataUsage         1.89784    2.39209   0.793 0.427555
CustServCalls     0.50141    0.04922  10.187  < 2e-16 ***
DayMins           0.04425    0.04036   1.096 0.272921
MonthlyCharge    -0.18361    0.23715  -0.774 0.438788
OverageFee        0.45455    0.40480   1.123 0.261483
RoamMins          0.09477    0.02767   3.425 0.000614 ***
ContractRenewal1 -2.00850    0.17866 -11.242  < 2e-16 ***
DataPlan1        -1.06576    0.67205  -1.586 0.112778
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1792.9  on 2165  degrees of freedom
Residual deviance: 1426.8  on 2157  degrees of freedom
AIC: 1444.8

Number of Fisher Scoring iterations: 6

>
> #check for multicollinearity
> vif(LRmodel)
      DataUsage   CustServCalls         DayMins   MonthlyCharge      OverageFee
    1597.630063        1.102608      929.049524     2736.123290      216.751789
       RoamMins ContractRenewal        DataPlan
       1.206056        1.059049       14.206694
> # there is high multicollinearity between the variables, hence removing dataplan and
data usage as 74 and 78% of data s explained by monthly charge
>
> full.data1<-subset(full.data,select=-c(DataUsage,DataPlan))
> names(full.data1)
[1] "Churn"           "CustServCalls"   "DayMins"         "MonthlyCharge"
[5] "OverageFee"      "RoamMins"        "ContractRenewal"
> #checking again after removng data usage
>
> set.seed(300)
> library(caTools)
> spl=sample.split(full.data1$Churn,SplitRatio = 0.65)
> train=subset(full.data1,spl==T)
> test=subset(full.data1,spl==F)
> #chk % of the sample
> sum(as.integer(as.character(train$Churn)))/nrow(train)
[1] 0.1449677
> sum(as.integer(as.character(test$Churn)))/nrow(test)
[1] 0.1448158
> sum(as.integer(as.character(full.data1$Churn)))/nrow(full.data1)
[1] 0.1449145
> #build model
> LRmodel=glm(Churn~.,data=train,family = binomial)
> summary(LRmodel)

Call:
glm(formula = Churn ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9037  -0.5161  -0.3466  -0.2060   2.9953

Coefficients:
```

```
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.259208   0.552217 -11.335  < 2e-16 ***
CustServCalls     0.504590   0.049233  10.249  < 2e-16 ***
DayMins           0.018310   0.001716  10.670  < 2e-16 ***
MonthlyCharge    -0.030590   0.006093  -5.020 5.16e-07 ***
OverageFee        0.215258   0.030730   7.005 2.47e-12 ***
RoamMins          0.110687   0.025397   4.358 1.31e-05 ***
ContractRenewal1 -1.867982   0.177711 -10.511  < 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: 1792.9  on 2165  degrees of freedom
Residual deviance: 1420.8  on 2159  degrees of freedom
AIC: 1434.8

Number of Fisher Scoring iterations: 5

> #check for multicollinearity
> vif(LRmodel)
  CustServCalls         DayMins    MonthlyCharge      OverageFee         RoamMins
       1.069130        1.766834         1.849349         1.208662         1.025734
ContractRenewal
       1.047130
>
> #get the confidence intervals
> confint(LRmodel)
Waiting for profiling to be done...
                      2.5 %      97.5 %
(Intercept)      -7.36069831 -5.19457864
CustServCalls     0.40879920  0.60199522
DayMins           0.01500242  0.02173499
MonthlyCharge    -0.04287020 -0.01894046
OverageFee        0.15560450  0.27615284
RoamMins          0.06129911  0.16092041
ContractRenewal1 -2.21717659 -1.51973549
> #############train###
>
>
> ###predict the response of the model using the test data
>
> predTrain=predict(LRmodel,newdata=train,type='response')
>
> #build confusion matrix; >0.5=true else false
> conf_mat=table(train$Churn,predTrain>0.5)
> conf_mat

    FALSE TRUE
  0  1808   44
  1   253   61
> #get accuracy by using the right classifiers
> (conf_mat[1,1]+conf_mat[2,2])/nrow(na.omit(train))
[1] 0.8628809
> #plot the ROC curve for calculating AUC
> library(ROCR)
> ROCRpred=prediction(predTrain,train$Churn)
> as.numeric(performance(ROCRpred,'auc')@y.values)
[1] 0.8156374
> perf_train=performance(ROCRpred,'tpr','fpr')
> plot(perf_train,col='black',lty=2,lwd=2)
```

```
> plot(perf_train,lwd=3,colorize=TRUE)
> ks_train <- max(perf_train@y.values[[1]]- perf_train@x.values[[1]])
> plot(perf_train,main=paste0('KS=',round(ks_train*100,1),'%'))
> lines(x = c(0,1),y=c(0,1))
> ############################
>
> #predict the response of the model using the test data
> predTest=predict(LRmodel,newdata=test,type='response')
>
> #build confusion matrix; >0.5=true else false
> conf_mat=table(test$Churn,predTest>0.5)
> conf_mat

    FALSE TRUE
  0   979   19
  1   144   25
> #get accuracy by using the right classifiers
> (conf_mat[1,1]+conf_mat[2,2])/nrow(na.omit(test))
[1] 0.8603256
> #plot the ROC curve for calculating AUC
> library(ROCR)
> ROCRpred=prediction(predTest,test$Churn)
> as.numeric(performance(ROCRpred,'auc')@y.values)
[1] 0.8101706
> perf_test=performance(ROCRpred,'tpr','fpr')
> plot(perf_test,col='black',lty=2,lwd=2)
> plot(perf_test,lwd=3,colorize=TRUE)
> ks_test <- max(perf_test@y.values[[1]]- perf_test@x.values[[1]])
> plot(perf_test,main=paste0('KS=',round(ks_test*100,1),'%'))
> lines(x = c(0,1),y=c(0,1))
>> ######################################################
> #knn and nb#
> str(full.data1)
'data.frame':  3333 obs. of  7 variables:
 $ Churn          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ CustServCalls  : int  1 1 0 2 3 0 3 0 1 0 ...
 $ DayMins        : num  265 162 243 299 167 ...
 $ MonthlyCharge  : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
 $ OverageFee     : num  9.87 9.78 6.06 3.1 7.42 ...
 $ RoamMins       : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
 $ ContractRenewal: Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 1 ...
> #convert variables to num in order to use knn
> full.data1$CustServCalls<-as.numeric(full.data1$CustServCalls)
> full.data1$ContractRenewal<-as.numeric(full.data1$ContractRenewal)
> str(full.data1)
'data.frame':  3333 obs. of  7 variables:
 $ Churn          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ CustServCalls  : num  1 1 0 2 3 0 3 0 1 0 ...
 $ DayMins        : num  265 162 243 299 167 ...
 $ MonthlyCharge  : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
 $ OverageFee     : num  9.87 9.78 6.06 3.1 7.42 ...
 $ RoamMins       : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
 $ ContractRenewal: num  2 2 2 1 1 1 2 1 2 1 ...
> View(full.data1)
> tcnorm<-scale(full.data1[,-1])
> tcnorm<-cbind(full.data1[,1],tcnorm)
> colnames(tcnorm)[1]<-'CHURN'
> View(tcnorm)
> #convert to a data frame
>
> df_tcnorm<-as.data.frame(tcnorm)
```

```
> df_tcnorm$CHURN<-as.factor(df_tcnorm$CHURN)
> #check number values currently1=benign;2=malignant
> table(df_tcnorm$CHURN)

   1    2
2850  483
> #partition the data
> library(caTools)
> spl=sample.split(df_tcnorm,SplitRatio = 0.65)
> train=subset(df_tcnorm, spl==T)
> test=subset(df_tcnorm, spl==F)
> #train using knn
>
> library(class)
> sqrt(nrow(train))
[1] 43.64631
> knn_model<- knn(train[-1],test[-1],train[,1],k=10)##works for test
>
> #check confusion matrix
> table.knn=table(test[,1],knn_model)
> table.knn
   knn_model
       1    2
  1 1192   29
  2  101  106
> #check accuracy
> sum(diag(table.knn)/sum(table.knn))
[1] 0.9089636
> #ch loss
> loss.knn<-table.knn[2,1]/(table.knn[2,1]+table.knn[1,1])
> loss.knn
[1] 0.07811292
> opp.loss.knn<-table.knn[1,2]/(table.knn[1,2]+table.knn[2,2])
> opp.loss.knn
[1] 0.2148148
> tot.loss.knn<-0.95*loss.knn+0.05*opp.loss.knn
> tot.loss.knn
[1] 0.08494801
> library(e1071)
> nb_model=naiveBayes(CHURN~.,data=train1)
> #apply predict function to see the performance
> pred_nb=predict(nb_model,test,type='class')
>
> #confusion matrix
> tab.NB=table(test[,1],pred_nb)
> tab.NB
   pred_nb
       1    2
  1 1124   97
  2  130   77
> #check accuracy
> sum(diag(tab.NB)/sum(tab.NB))
[1] 0.8410364
> #check loss
> loss.NB<-tab.NB[2,1]/(tab.NB[2,1]+tab.NB[1,1])
> loss.NB
[1] 0.1036683
> opp.loss.NB<-tab.NB[1,2]/(tab.NB[1,2]+tab.NB[2,2])
> opp.loss.NB
[1] 0.5574713
> tot.loss.NB<-0.95*loss.NB+0.05*opp.loss.NB
```

```
> tot.loss.NB
[1] 0.1263584
```