To understand the consumer behavior based on churn score

BY: DIPTANSHU GAUTAM

19th January 2019

Table of Content

Chapter 1

Introduction

Problem Statement

Any business grows because of its customers or clients and in a sense, they are the true owner of the business. Therefore, it is truly said that customers are godlike figure for any business. A business can grow if they get new clients continuously, however it is also important that the existing customers must be satisfied with the business and its product and/or service. This analysis is based to understand the consumer behavior.

The major aim of this project is to understand the consumer behavior of a telecommunication company by predicting the churn score of the clients.

Data

The main aim of this project it to create a model which could predict the dependent variable based on the independent variable such as total day charge, international plans, number of voicemail messages, etc. The dependent variable in this case is the "*Churn Score"*.

Churn Score sample data (Column 1 to Column 7)

state	account length	area code	phone number	international plan	voice mail plan	number vmail messages
KS	128	415	382-4657	no	yes	25
OH	107	415	371-7191	no	yes	26
NJ	137	415	358-1921	no	no	0
OH	84	408	375-9999	yes	no	0
OK	75	415	330-6626	yes	no	0
AL	118	510	391-8027	yes	no	0
MA	121	510	355-9993	no	yes	24
MO	147	415	329-9001	yes	no	0
LA	117	408	335-4719	no	no	0

Churn Score sample data continued (Column 8 to Column 14)

total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes
265.1	110	45.07	197.4	99	16.78	244.7
161.6	123	27.47	195.5	103	16.62	254.4
243.4	114	41.38	121.2	110	10.3	162.6
299.4	71	50.9	61.9	88	5.26	196.9
166.7	113	28.34	148.3	122	12.61	186.9
223.4	98	37.98	220.6	101	18.75	203.9
218.2	88	37.09	348.5	108	29.62	212.6
157	79	26.69	103.1	94	8.76	211.8
184.5	97	31.37	351.6	80	29.89	215.8

Churn Score sample data continued (Column 15 to Column 21)

total night calls	total night charge	total intl minutes	total intl calls	total intl charge	customer service calls	Churn
91	11.01	10	3	2.7	1	False.
103	11.45	13.7	3	3.7	1	False.
104	7.32	12.2	5	3.29	0	False.
89	8.86	6.6	7	1.78	2	False.
121	8.41	10.1	3	2.73	3	False.
118	9.18	6.3	6	1.7	0	False.
118	9.57	7.5	7	2.03	3	False.
96	9.53	7.1	6	1.92	0	False.
90	9.71	8.7	4	2.35	1	False.

Variables

There are total of 21 variables which are present in the data. "*Churn*" is the dependent variable and rest 20 variables are independent. The variables are both categorical and numerical type. Names of the dependent variables are as follows:

Dependent Variables - 1
state
account length
area code
phone number
international plan
voice mail plan
number vmail messages
total day minutes
total day calls
total day charge

Dependent Variable - 2
total eve minutes
total eve calls
total eve charge
total night minutes
total night calls
total night charge
total initial minutes
total initial calls
total initial charge
number customer service calls

Chapter 2

Methodology

Pre-Processing

If we need to model a data, it cannot be worked as it is in its raw form. It is because, all the models are based on some assumptions and rules and regulations which cannot be broken if the model is to work. However, in the case of raw data, it is not always the case. The data in its raw form has one of the most common issue, which is empty cell. There is always a chance of human error, due to which there is an empty cell with no data. Model cannot be built upon such samples. It is due to which some manipulation of data up to some level is important to give its final touch so that the model can be built upon. Now manipulation has to be taken care of because, the hidden truth behind the data which were trying to figure out is to be preserved. This method is called **pre-processing** the data for **Exploratory data analysis.**

Thus, we will polish the data so that it may be used for the modelling purpose and inferential data could be extracted from it.

Data Structure

We need to understand the data structure or the types of variables present in the data to broadly classify the data which will help us pre-process the data easily. This is because, the factor type data are processed in a different way than the numeric data. This is because the categorical or factor data are usually nominal or ordinal level of data, whereas the numeric are usually interval scale or ratio scale.

However, if we check the data type of each variable, it turns out that data has assumed that every variable is numeric type which is clearly not the case. Thus, we convert each variable into appropriate data type. Post that we check the data structure of the data.

Missing Value Analysis

After converting the data type of the variable, it is very important to address the missing values. As per the company standards, there is a general rule that if any variable has a missing value of more than 30%, we remove that variable because the imputed value would affect the originality of the data way too much to model using that particular variable.

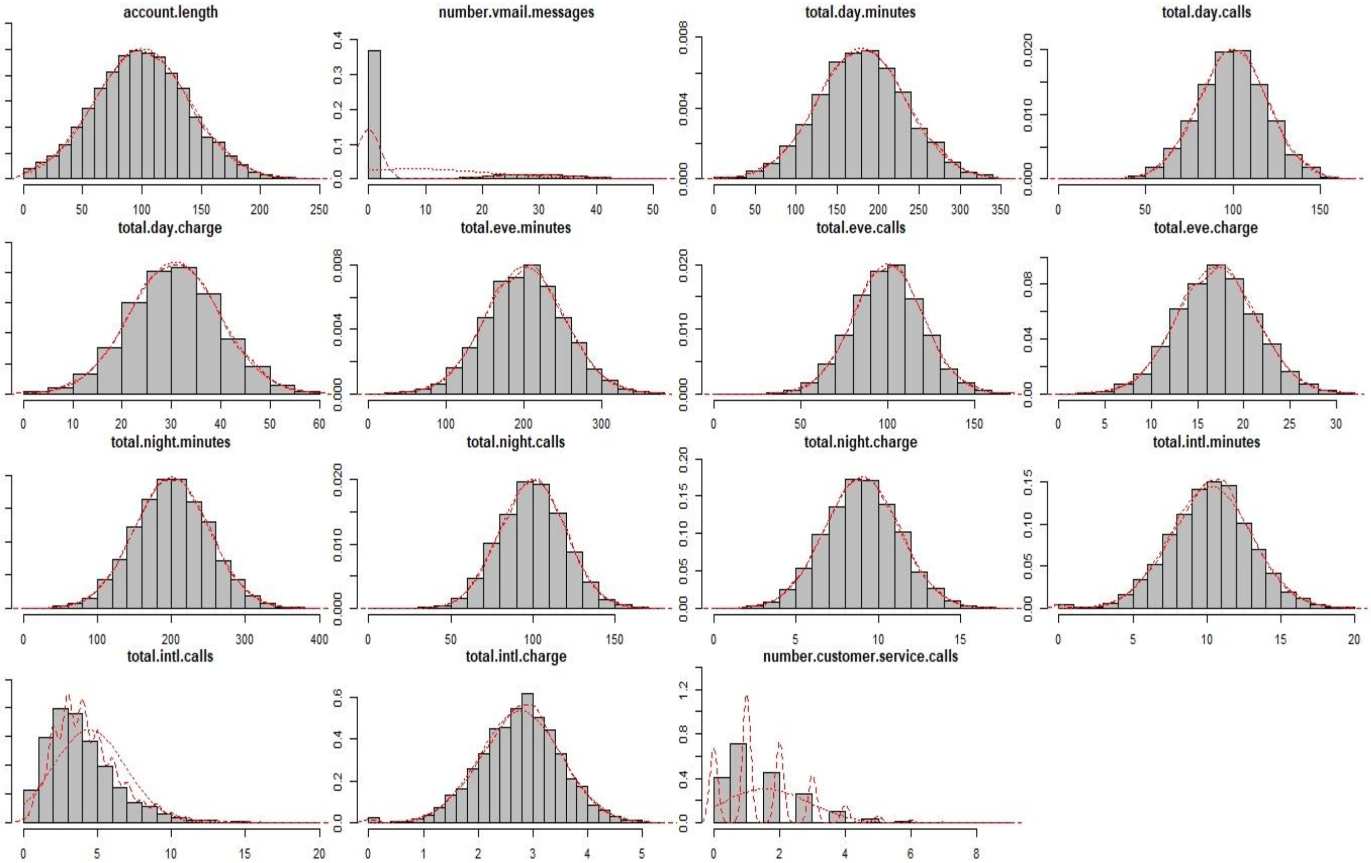
Fortunately, the data which was collected for the modelling purpose does not contain any missing value. Thus, we do not need to impute any missing value here.

Outlier Analysis

There is a very high possibility that the data which has been taken into consideration has some outlier data. The outlier data is the set of data which is outside the range of the most of the data. It could be on the either side of the range. For e.g., in a cricket match where 8 players score on an average 50 runs, a player hits 150 and one goes to pavilion with nothing but duck! In this case, both of them are an outlier, these data are to be removed because, these data hamper the result in a drastic way. If the statistic is affected by the outlier such as mean, then all the statistic and inference which have mean will be affected by this outlier. Therefore, we tend to find and remove or impute the data.

This is also to be noted that outlier is only possible in numeric data. It can never happen with categorical type of data.

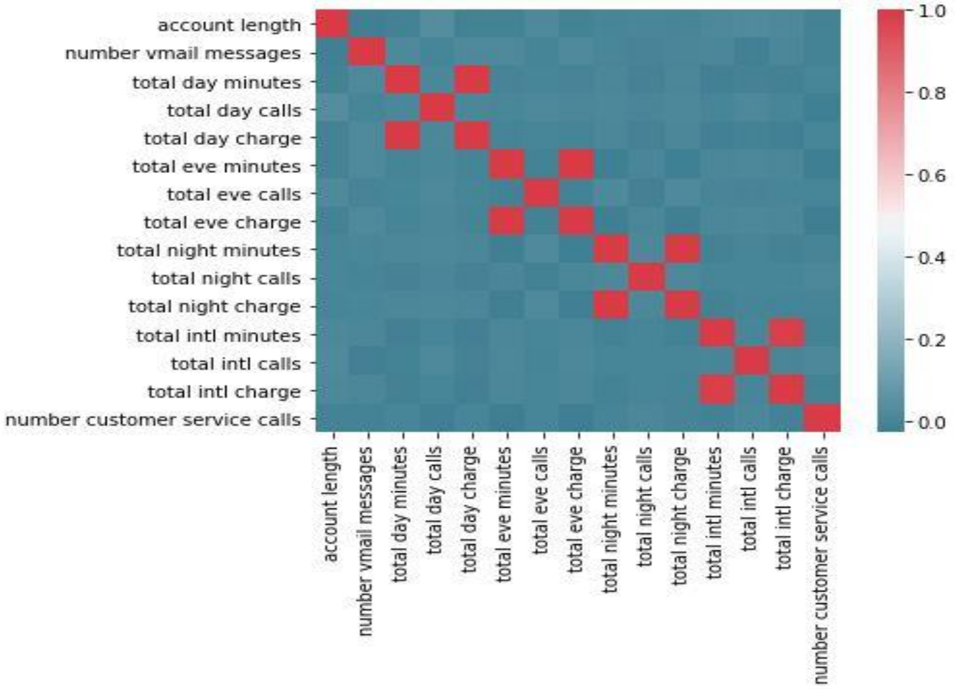
Multiple Histogram

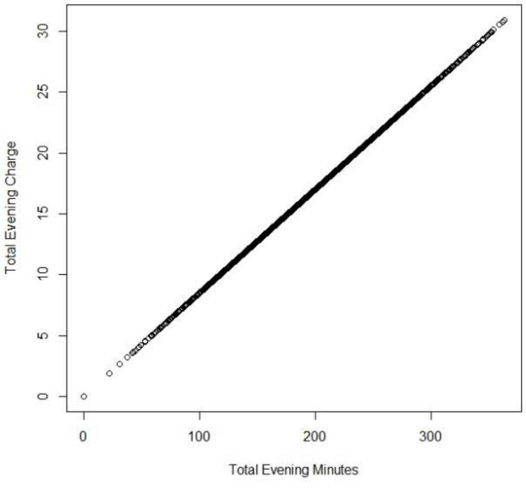


Correlation Matrix and Correlation Plot

One of the major issues in a numeric dataset, is the existence of multicollinearity in the dataset. Multicollinearity is defined when there is a high level of correlation between two independent variables. This is a particular hindrance which should be addressed as soon as possible. Thus, we check the correlation between all the variables and drew a correlation matrix to make it even more clear.

Correlation matrix





We can see that there is specifically high correlation between the variable “total day minutes” and “total day charge”, “total eve minutes” and “total eve charge” and “total night minutes” and “total night charge”. This comes to be more than 99%. This is extremely high, which is a sign of high correlation. We can check by plotting dot plot for the variables. We have plotted one of the highly collinear graphs to show the degree of collinearity. Due to which we will have to remove “total day minutes”, “total eve minutes” and “total night minutes”. We have removed these variables because the other variables can be controlled as they also contain a portion of price of the call.

Chapter 3

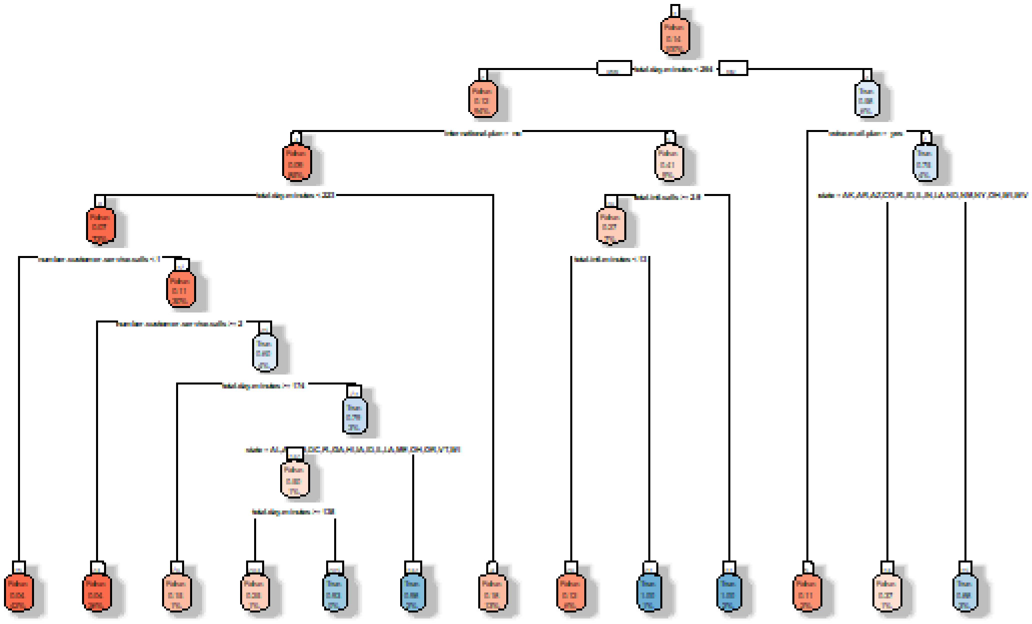
Model Selection

As it is evident, that data contains both numerical and categorical variable, regression analysis or KNN would not be appropriate for this type of data. Therefore, we would be applying random forest, decision tree model and Naïve Bayes algorithm. We will further check the extent of accuracy of the model to check the accuracy of the model.

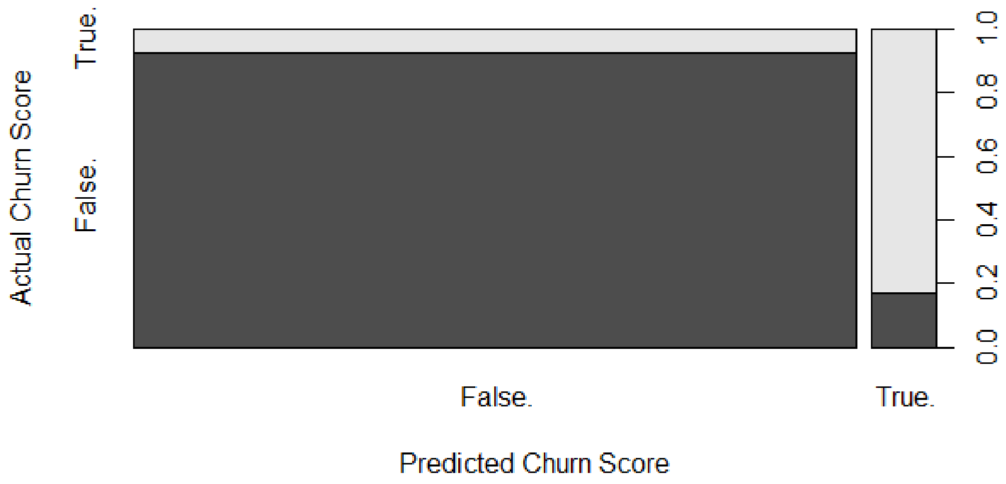
Decision Tree Modelling

We use the decision tree analysis, to try and analyze the optimum rules which would be applicable for the model. Following is the decision tree which was extracted from the model.

Decision Tree



Churn Score plot



From the Churn score plot it is quite clear that the true negative is more than 90% and true positive is more than 80%. We will analyze it deeply using confusion matrix in the model evaluation section.

Understanding Decision Trees

- Let's see some of the rules which were pulled by decision tree forest, to understand it in a better way.
  - Rule 0/1: (1034/58, lift 1.1)
    - International plan = no
    - Total day minutes <= 264.4
    - Number customer service calls > 1.935249
    - class False. [0.943]
  - It means that, if the client has no international plan, total minutes he spends during day time is less at most 264.4 and receives at least 2 customer service calls, there is a 94.3% chance that he will not stay with the company.

  - Rule 0/2: (1645/102, lift 1.1)
    - International plan = no
    - Total day minutes <= 264.4
    - Number customer service calls <= 1.013625
    - False. [0.937]
  - It means that, if the client has no international plan, total minutes he spends during day time is less at most 264.4 and receives at most 1 customer service call, there is a 93.7% chance that he will not stay with the company.

- Rule 0/6: (62, lift 6.8)
  - International plan = yes
  - Total initial calls <= 2.922947
  - class True. [0.984]
- It means that if the client has an international plan, total initial calls are at most 3, there is more than 98% chance that he will stay with the company.

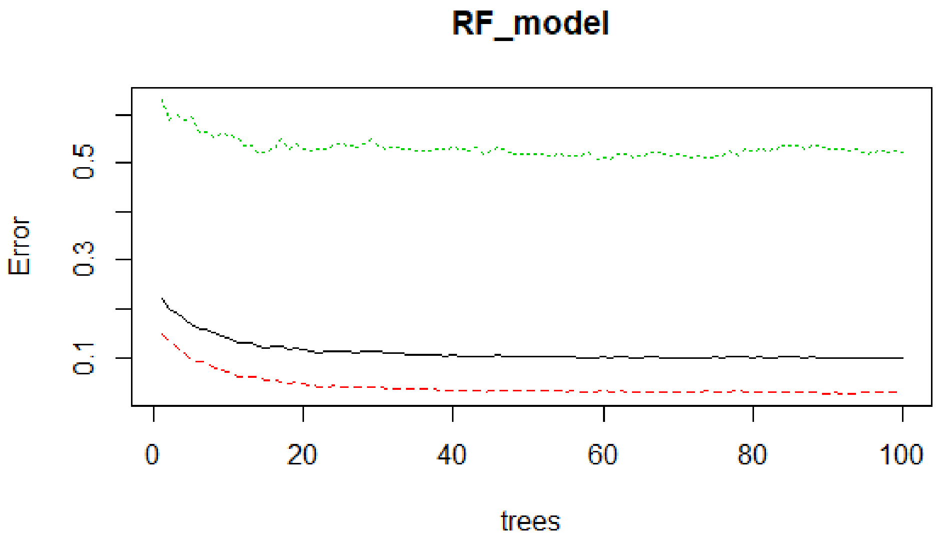
- We can see from the some of the rules that international plan plays a key role in retention of the client. Company should focus on some of the key areas from such analysis to increasing Churn Score.

Random Forest Modelling

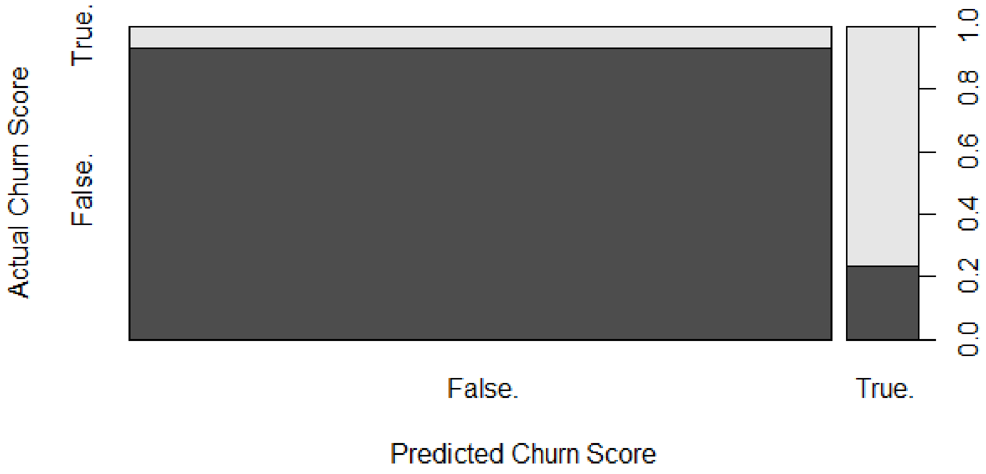
Unlike the decision tree model, random forest model tend to build multiple number of decision tree model, and tries to maximize the possible number of results with greater accuracy. This increases the accuracy up to extent. If we plot the curve of the random forest, model, we can see an "***Elbow Point"*** which is supposed to the optimum number of point post which the accuracy won't differ much. Below are the plots of the random forest model along with the predicted plots.

Random Forest Plot

We can see that almost after 20 random forest plots, the results do not vary much. We can safely assume 20 trees. Increasing the number of trees will only increase the size of the code which will result in lagging and much time taking.



Churn Score Plot



In the Churn Score plot predicted by the random forest model, we can see that the true FALSE is more than 90% and true positive is close to 80%. Thus, it is a good model. However, we will analyze it deeply using confusion matrix in the model evaluation section.

Understanding tha Random Forest Rules

So, lets see some of the top rules of Random Forest Model.

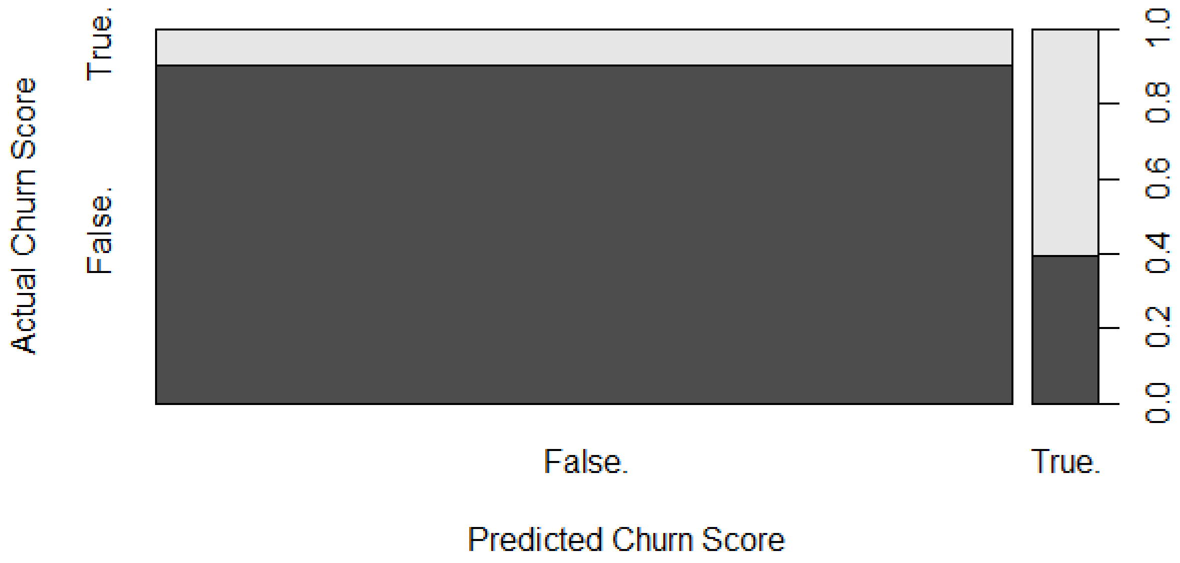
- If the client is from New Jersey, the total day minutes lie between 177.05 and 227.65, total day charge is less than 17.495 and total initial calls are less 3, then the client will not stay with the company.
- If the client is from New Jersey, and the total day minutes spent lie between 177.05 and 227.65, total initial call is less than 3 and number of customer service call is at most 1, the client will not stay with the company.
- If the client is from New Jersey, number of voicemail message is less than 40, total day minutes spend is less than 227.65, total day calls are less than 115 and total initial calls are less than 3, the client will stay with the company.

Naïve Bayes Model

Naïve Bayes model is unlike the other two model is a probabilistic model. It is that classifier which is able to predict, given an observation of an input, a probability distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to.

Instead, of giving an exact answer, it gives probability of the event occurring over the range of the predictors based on the **Bayes Theorem.** The foremost important point is that it assumes that all the predictor variables are pairwise independent. From the probability we try to estimate the possibility of the even happening and then we extrapolate the classifier result. For e.g., if the probability is more than 0.5, we could say the result is **TRUE** else, **FALSE.**

Churn Score Plot



In this graph, we can see that the true false is close to 0.9, false negative is close to 0.1 and false positive is close 0.4. We will analyze it deeply using confusion matrix in the model evaluation section.

Model Evaluation

Confusion Matrix

- It is a type of model evaluation technique which is actually suitable for classification problems. It draws a contingency table between actual and predicted values so that we can understand the accuracy and percentage of type of errors in the model

Decision Tree Model:

The confusion matrix shows the following result:

Description	False	True
False	1358	58
True	73	151

- Accuracy: 92.14%
- FNR: 32.58%

From this, we can say that the decision tree model, fitted very nicely and gave a result which is accurate to 94%. This is very positive sign which means that the rules which were detected by the model are quite accurate.

Random Forest Model:

The confusion matrix shows the following result:

Description	False	True
False	1437	6
True	66	158

- Accuracy: 95.68%
- False Negative Rate: 29.46%

From this, we can say that the random forest model, fitted very nicely and gave a result which is accurate to 92%. This is very positive sign which means that the rules which were detected by the model are quite accurate. Although, we can say that if seen closely, random forest model was not as good as decision tree model.

Naïve Bayes Model:

The confusion matrix shows the following result:

Description	False	True
False	1342	101
True	135	89

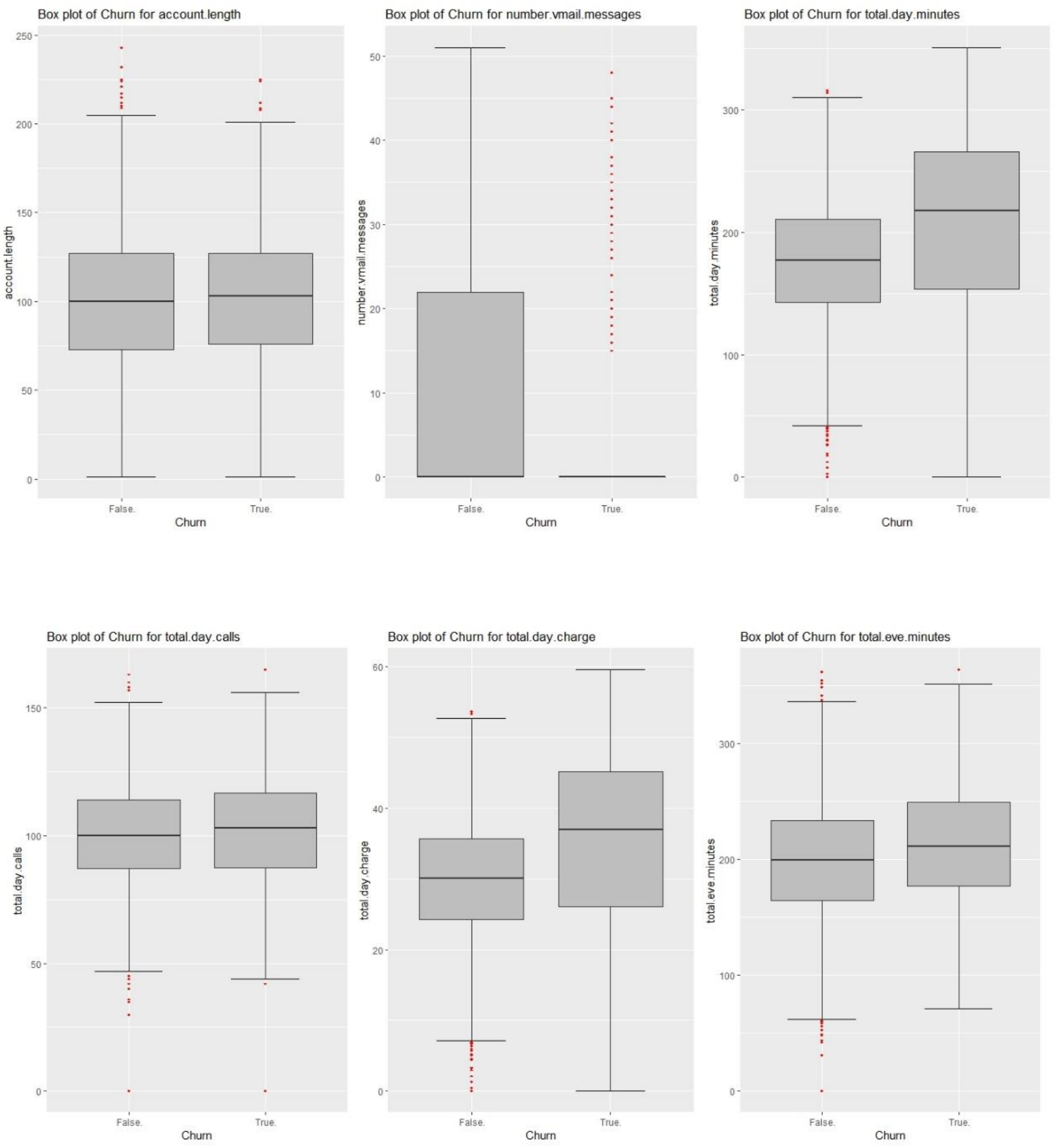
- Accuracy: 85.84%
- Specificity: 60.26%

From this, we can say that the all the model fitted well giving an accuracy of over 85%. Although, we can say that if seen closely, Naïve Bayes model, significantly underperformed when compared to random forest model or decision tree model.

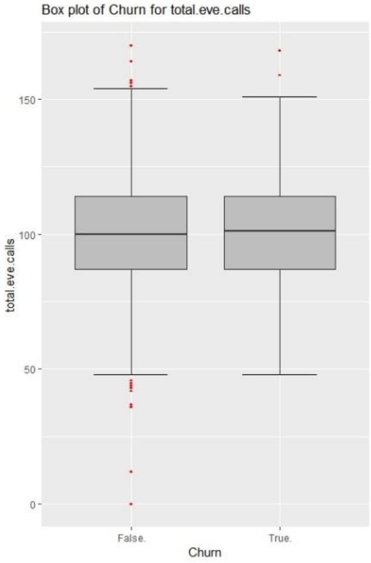
Appendix

Extra Figures

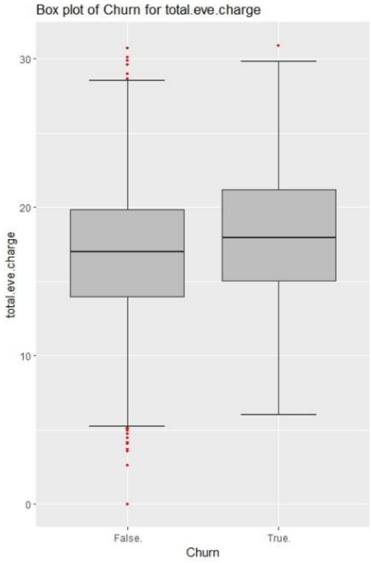
Outlier Box Plot



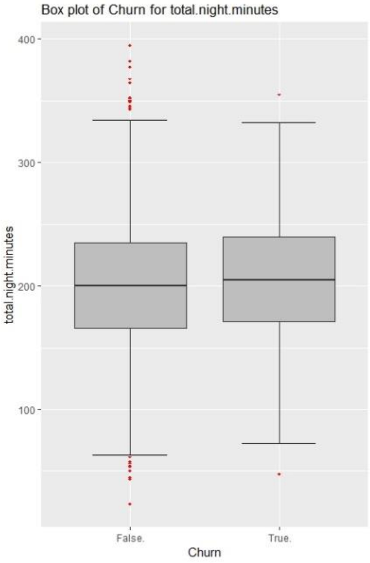
Box plot outlier continued



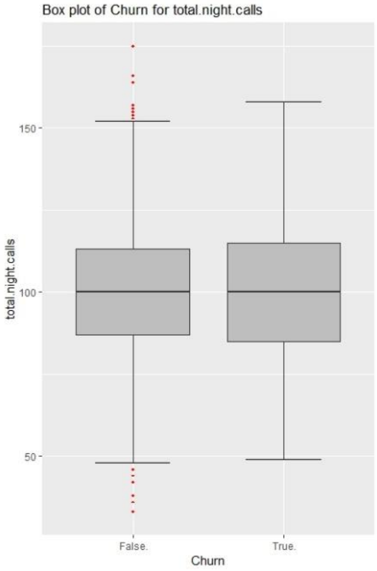
Box plot of Churn for total.eve.calls



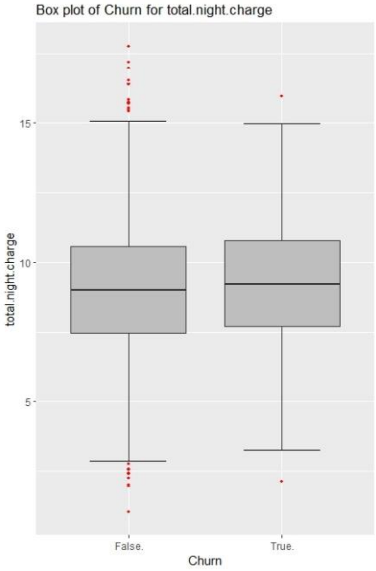
Box plot of Churn for total.eve.charge



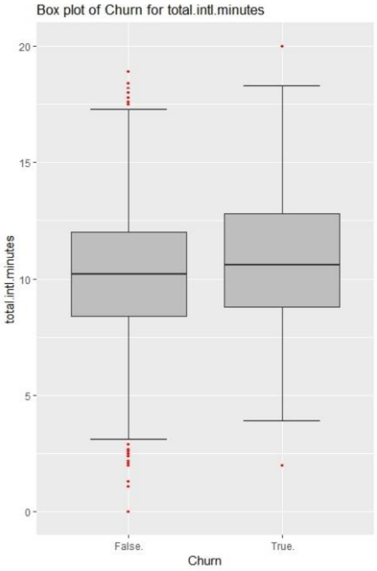
Box plot of Churn for total.night.minutes



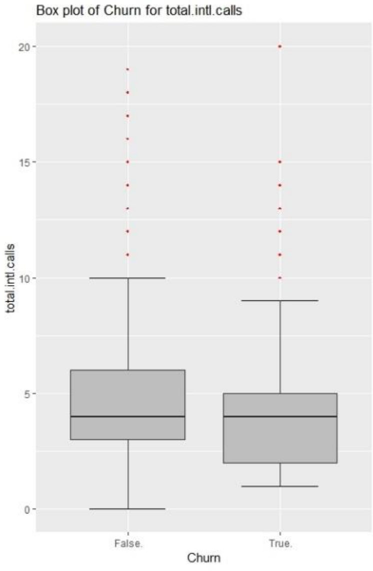
Box plot of Churn for total.night.calls



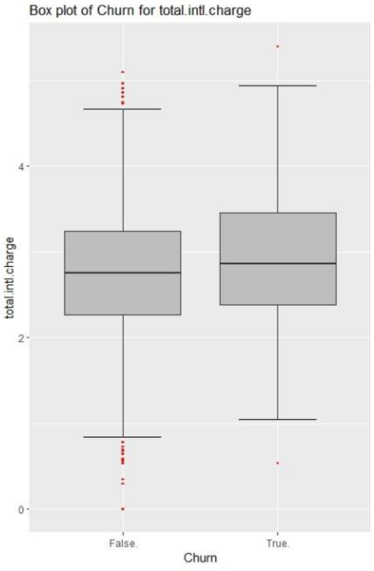
Box plot of Churn for total.night.charge



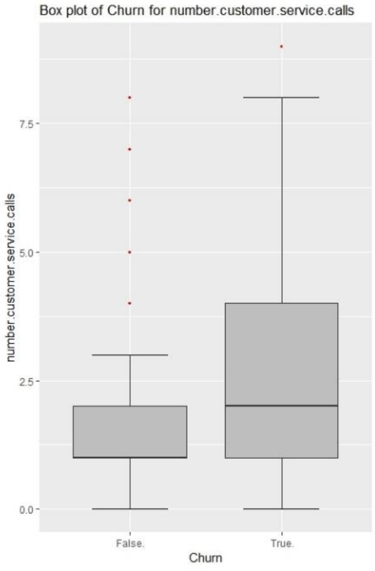
Box plot of Churn for total.intl.minutes



Box plot of Churn for total.intl.calls

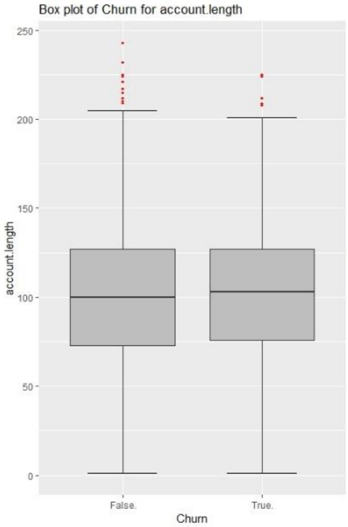


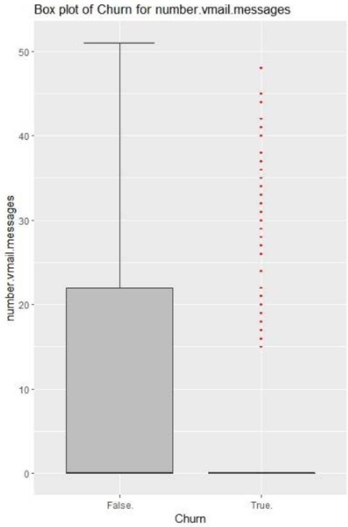
Box plot of Churn for total.intl.charge

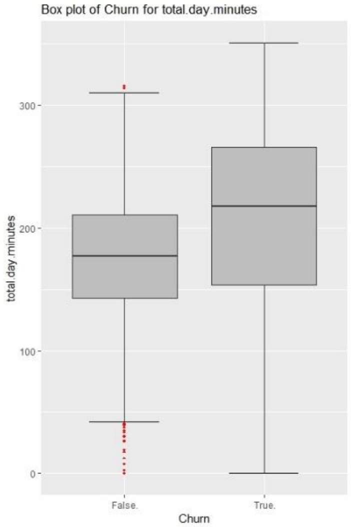


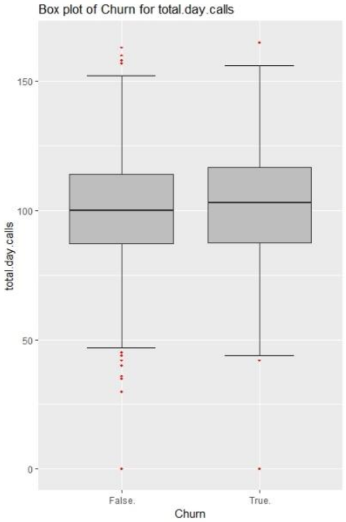
Box plot of Churn for number.customer.service.calls

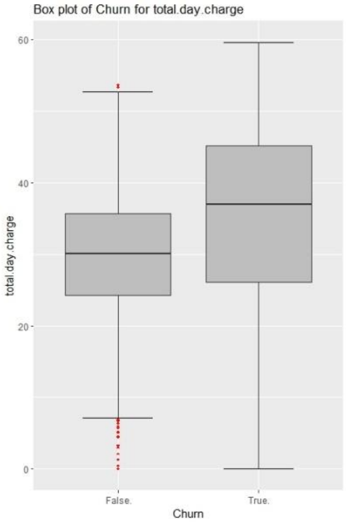
Box plot outlier continued

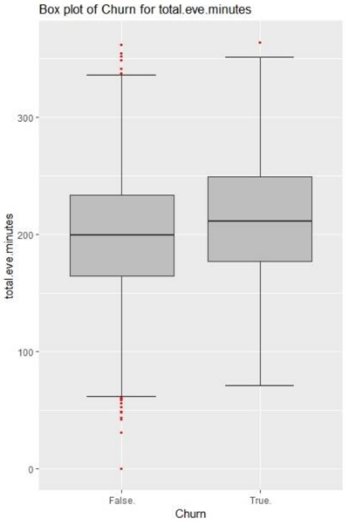


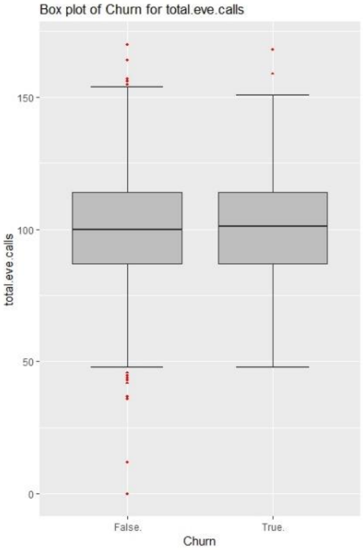


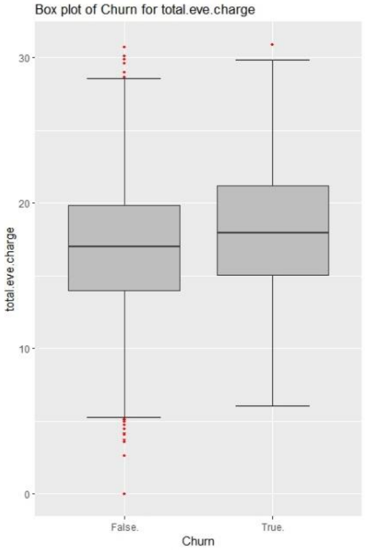


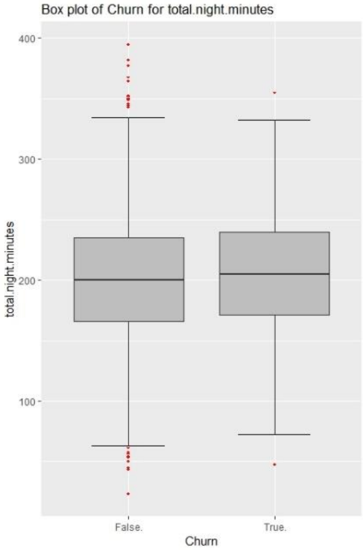




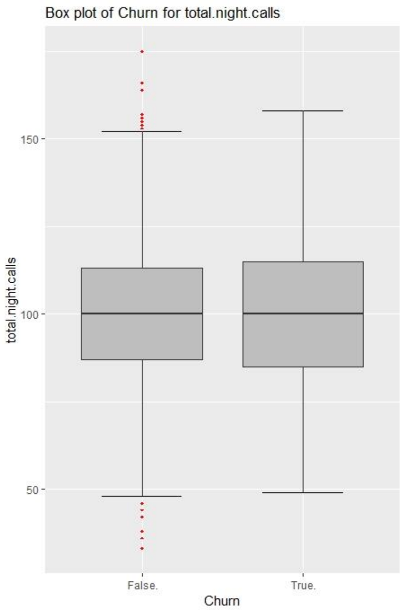


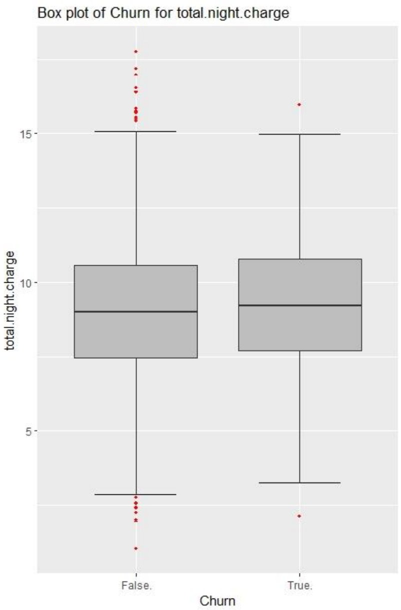


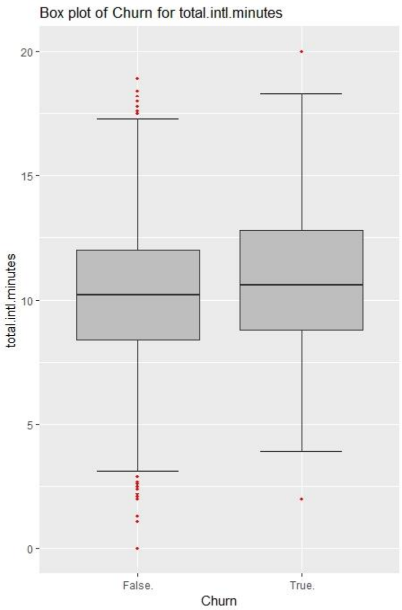


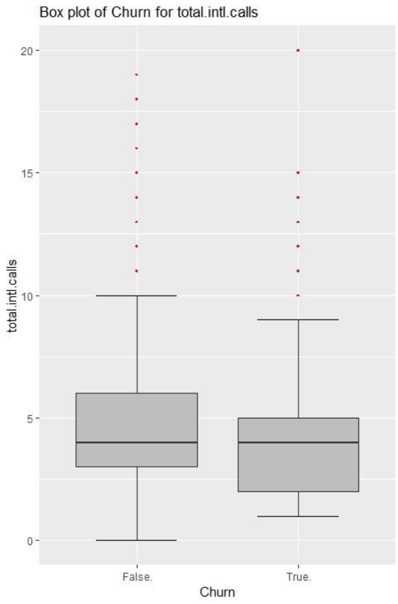


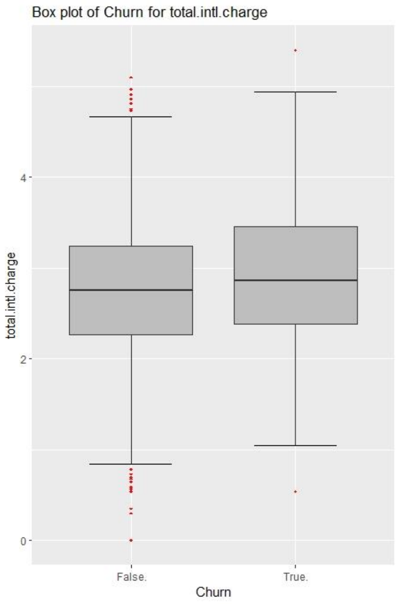
Box plot outlier continued

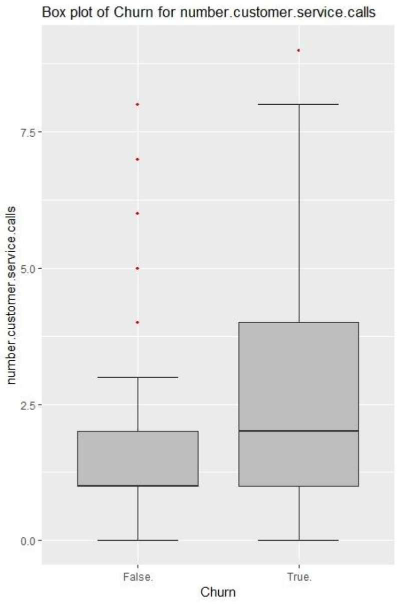




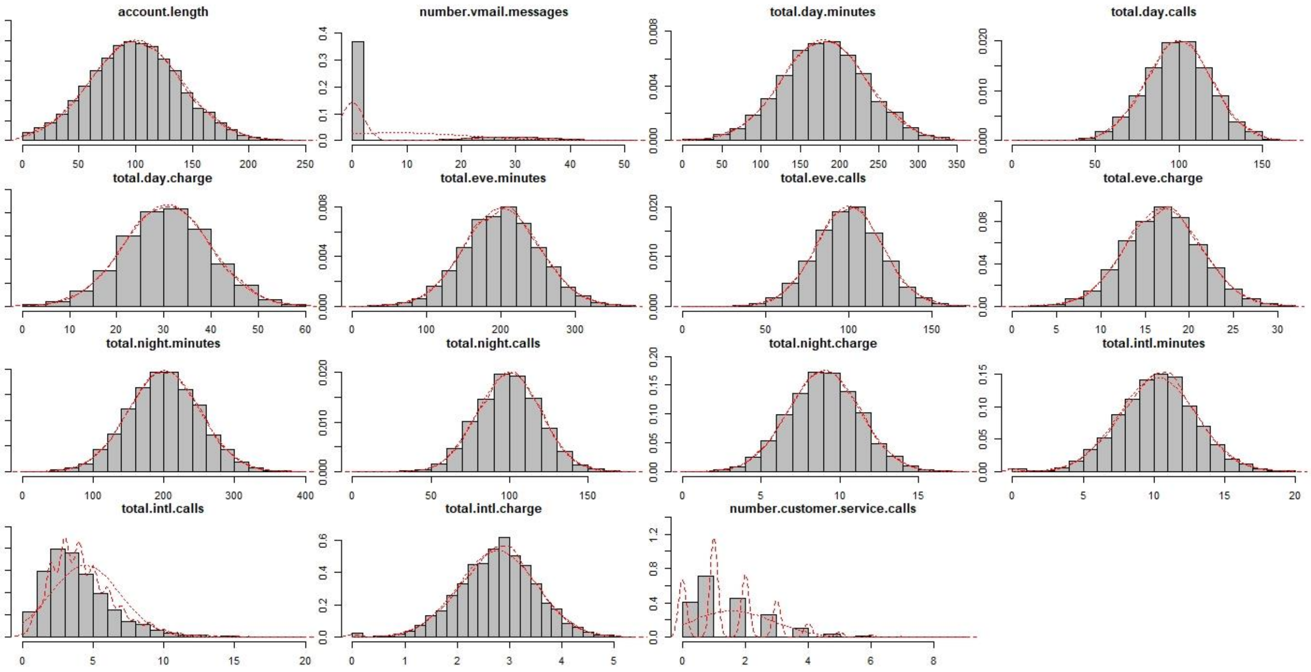




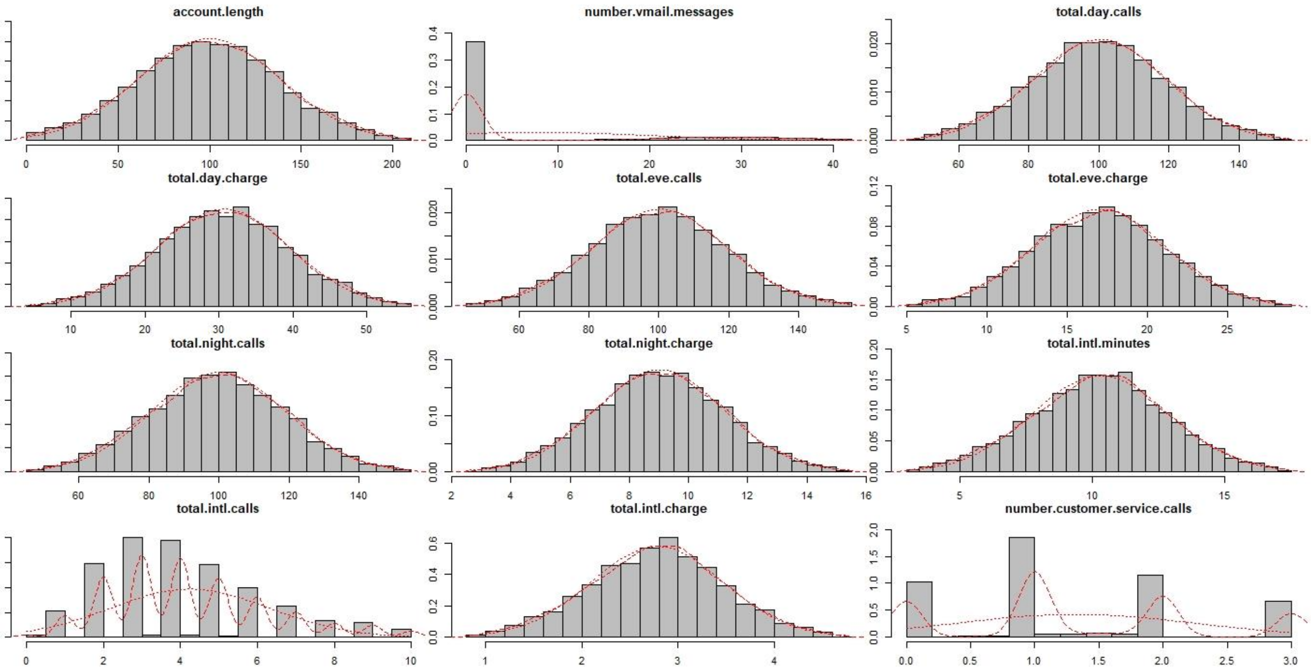




Multiple Histograms before Outlier Analysis



Multiple Histograms after Outlier Analysis



R Code

```
#Setting Working Directory
setwd("E:/Notes and ppts/Edwisor/Projects/Project 3")

#Loading train and test data
train = read.csv("Train_data.csv")
test = read.csv("Test_data.csv")


#converting the data type
train$area.code = as.factor(train$area.code)
test$area.code = as.factor(test$area.code)

#Combining Train and test data
data = rbind(train, test)

#Exploring the data
str(data)

#Missing Value Analysis
sum(is.na(data))

#splitting data into factor and numeric type
factor.index = sapply(data, is.factor)
numeric.index = sapply(data, is.numeric)
factor.data = data[,factor.index]
numeric.data = data[,numeric.index]

#Outlier Analysis
cnames = colnames(numeric.data)
for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i]), x = "Churn"), data = subset(data))+
           stat_boxplot(geom = "errorbar", width = 0.5) +
           geom_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,
                        outlier.size=1, notch=FALSE) +
           theme(legend.position="bottom")+
           labs(y=cnames[i],x="Churn")+
           ggtitle(paste("Box plot of Churn for",cnames[i])))
}

# Plotting plots together
gridExtra::grid.arrange(gn1,gn2,gn3,ncol=3)
gridExtra::grid.arrange(gn4,gn5,gn6,ncol=3)
gridExtra::grid.arrange(gn7,gn8,gn9,ncol=3)
gridExtra::grid.arrange(gn10,gn11,gn12,ncol=3)
gridExtra::grid.arrange(gn13,gn14,gn15,ncol=3)
```

```
#replacing all the outliers with NA and imputing them using knn method

columns = colnames(data)
for(i in columns){
    val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
    print(length(val))
    data[,i][data[,i] %in% val] = NA
  }
sum((is.na(data)))

#number of outliers detected and removed = 1080
library(DMwR)
data = knnImputation(data, k = 5)

#Detecting correlation and dependence between variables

#plotting correlogram
corrgram(numeric.data, order = FALSE, upper.panel = panel.pie, text.panel = panel.txt, main = "Correlation
Plot")
plot(data$total.day.minutes, data$total.day.charge)

corre_ana = cor(numeric.data, numeric.data)
write.csv(corre_ana, "Correlation Analysis.csv")

plot(data2$total.eve.minutes,data2$total.eve.charge, xlab = "Total Evening Minutes",ylab = "Total Evening
Charge")

#Chi Square Test of Independence
for (i in 1:21)
{
  print(names(data)[i])
  print(chisq.test(table(factor.data$Churn,data[,i])))
}

#Dimensionality Reduction
data = subset(data, select = -
c(area.code,phone.number,total.eve.minutes,total.eve.charge,total.night.minutes,total.eve.calls,total.night.ca
lls,total.night.charge))

#recreating original train and test data partition

train = data[1:3333,]
test = data[3334:5000,]
```

```r
#Creating Model

#Decision Tree Model
library(C50)
library(rpart)
classifier = C5.0(Churn~., train, trials = 100, rules = TRUE)
summary(classifier)
write(capture.output(summary(classifier)), "C50 Output.txt")
regressor = rpart(Churn~., data = train)
C50_predict = predict(classifier, test[,-13], type = "class")
table(C50_predict)
table(test$Churn)

#Plotting Decision Tree and result
regressor = rpart(Churn~., data = train)
rpart.plot::rpart.plot(regressor, box.palette = "RdBu", shadow.col = "gray", nn=TRUE)
plot(C50_predict, test[,13], xlab = "Predicted Churn Score", ylab = "Actual Churn Score")


#Evaluate the performance of classification model
ConfMatrix_C50 = table(test$Churn, C50_predict)
library(caret)
confusionMatrix(ConfMatrix_C50)

#Random Forest Model
library(randomForest)
library(rpart)
RF_model = randomForest(Churn~.,data = train, ntree = 100)
pred = predict(RF_model, test[,-13], type = "class")
table(pred)

#Extract rules fromn random forest
#transform rf object to an inTrees' format
library(inTrees)
treeList = RF2List(RF_model)

#Extract rules
exec = extractRules(treeList, train[,-13])

#Make rules more readable:
readableRules = presentRules(exec, colnames(train))

#Get rule metrics
ruleMetric = getRuleMetric(exec, train[,-17], train$responded)  # get rule metrics

#Writing rules to a text file
write(capture.output(readableRules), "Random Forest Rules.txt")
```

```
#Get rule metrics
ruleMetric = getRuleMetric(exec, train[,-13], train$Churn)
write(capture.output(ruleMetric), "Random Forest Rule metric.txt")

#Plotting Random Forest and result
plot(RF_model)
plot(pred, test[,13], xlab = "Predicted Churn Score", ylab = "Actual Churn Score")

#Evaluate the performance of classification model
ConfMatrix_RF = table(test$Churn, pred)
confusionMatrix(ConfMatrix_RF)

#Naive Bayes
library(e1071)
class_nb = naiveBayes(Churn~., data = train)
predict_nb = predict(class_nb, test[,-13], type = 'class')

#Plotting Naive Bayes result
plot(predict_nb, test[,13], xlab = "Predicted Churn Score", ylab = "Actual Churn Score")

#Evaluating Naive Bayes by confusion matrix
ConfMatrix_nb = table(test$Churn, predict_nb)
confusionMatrix(ConfMatrix_nb)
```