Contents

1. Introduction

1.1 **Problem Statement- :** The objective of this model is to predict the customer behavior. We have public dataset that has customer usage pattern and if the customer has moved or not. We have to build a machine learning model to predict the churn score based on usage pattern.

1.2 **Data:** for the above problem we have data is following way. We will discuss each and every feature in detail.

```
> head(data)
  state account.length area.code phone.number international.plan voice.mail.plan
1    KS            128       415     382-4657                 no             yes
2    OH            107       415     371-7191                 no             yes
3    NJ            137       415     358-1921                 no              no
4    OH             84       408     375-9999                yes              no
5    OK             75       415     330-6626                yes              no
6    AL            118       510     391-8027                yes              no
  number.vmail.messages total.day.minutes total.day.calls total.day.charge total.eve.minutes
1                    25             265.1             110            45.07             197.4
2                    26             161.6             123            27.47             195.5
3                     0             243.4             114            41.38             121.2
4                     0             299.4              71            50.90              61.9
5                     0             166.7             113            28.34             148.3
6                     0             223.4              98            37.98             220.6
  total.eve.calls total.eve.charge total.night.minutes total.night.calls total.night.charge
1              99            16.78               244.7                91              11.01
2             103            16.62               254.4               103              11.45
3             110            10.30               162.6               104               7.32
4              88             5.26               196.9                89               8.86
5             122            12.61               186.9               121               8.41
6             101            18.75               203.9               118               9.18
  total.intl.minutes total.intl.calls total.intl.charge number.customer.service.calls  Churn
1               10.0                3              2.70                             1 False.
2               13.7                3              3.70                             1 False.
3               12.2                5              3.29                             0 False.
4                6.6                7              1.78                             2 False.
5               10.1                3              2.73                             3 False.
6                6.3                6              1.70                             0 False.
```

What type of data we have for see that we have to apply

```
> str(data)
'data.frame':	5000 obs. of  21 variables:
 $ state                        : Factor w/ 51 levels "AK","AL","AR",..: 17 36 32 36 37 2 20 25 19 50
 ...
 $ account.length               : int  128 107 137 84 75 118 121 147 117 141 ...
 $ area.code                    : int  415 415 415 408 415 510 510 415 408 415 ...
 $ phone.number                 : Factor w/ 5000 levels " 327-1058"," 327-1319",..: 1927 1576 1118 17
08 111 2254 1048 81 292 118 ...
 $ international.plan           : Factor w/ 2 levels " no"," yes": 1 1 1 2 2 2 1 2 1 2 ...
 $ voice.mail.plan              : Factor w/ 2 levels " no"," yes": 2 2 1 1 1 1 2 1 1 2 ...
 $ number.vmail.messages        : int  25 26 0 0 0 0 24 0 0 37 ...
 $ total.day.minutes            : num  265 162 243 299 167 ...
 $ total.day.calls              : int  110 123 114 71 113 98 88 79 97 84 ...
 $ total.day.charge             : num  45.1 27.5 41.4 50.9 28.3 ...
 $ total.eve.minutes            : num  197.4 195.5 121.2 61.9 148.3 ...
 $ total.eve.calls              : int  99 103 110 88 122 101 108 94 80 111 ...
 $ total.eve.charge             : num  16.78 16.62 10.3 5.26 12.61 ...
 $ total.night.minutes          : num  245 254 163 197 187 ...
 $ total.night.calls            : int  91 103 104 89 121 118 118 96 90 97 ...
 $ total.night.charge           : num  11.01 11.45 7.32 8.86 8.41 ...
 $ total.intl.minutes           : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
 $ total.intl.calls             : int  3 3 5 7 3 6 7 6 4 5 ...
 $ total.intl.charge            : num  2.7 3.7 3.29 1.78 2.73 1.7 2.03 1.92 2.35 3.02 ...
 $ number.customer.service.calls: int  1 1 0 2 3 0 3 0 1 0 ...
 $ Churn                        : Factor w/ 2 levels " False."," True.": 1 1 1 1 1 1 1 1 1 1 ...
> 
```

2. Methodology

2.1Preprocessing:

2.1.1 Missing Value Treatment:

In data first we need to check we have missing value in data or not because there are so many algorithms which are applicable on missing data.

To check missing value in my data

First we apply function in R

- table(is.na(data))

Function we apply in python check missing value

- data. isnull (). sum ()

in our data there is no missing value so we don't need to apply missing value treatment method like we have method

- mean ()- fill NA with the mean of the variable
- median () – fill NA with the median of the variable
- ffill () – fill NA with the value which is after NA
- bfill () – fill NA with the value which is before NA
- knn imputation – its algorithm which fill NA by prediction

2.1.2 Variable Transformation:

In this variable transformation we need to change our categorical data in numeric data because maximum classification algorithm only take numeric data some of the algorithm like decision tree can take categorical data too. In decision tree we don't need to apply variable transformation method

Output

Before :the dataset we have before the variable transformation we can see that we can only transform the categorical variable into the numeric dataset.

```
> head(data)
  state account.length area.code phone.number  international.plan voice.mail.plan
1    KS            128       415     382-4657                  no             yes
2    OH            107       415     371-7191                  no             yes
3    NJ            137       415     358-1921                  no              no
4    OH             84       408     375-9999                 yes              no
5    OK             75       415     330-6626                 yes              no
6    AL            118       510     391-8027                 yes              no
  number.vmail.messages total.day.minutes total.day.calls total.day.charge total.eve.minutes
1                    25             265.1             110            45.07             197.4
2                    26             161.6             123            27.47             195.5
3                     0             243.4             114            41.38             121.2
4                     0             299.4              71            50.90              61.9
5                     0             166.7             113            28.34             148.3
6                     0             223.4              98            37.98             220.6
  total.eve.calls total.eve.charge total.night.minutes total.night.calls total.night.charge
1              99            16.78               244.7                91              11.01
2             103            16.62               254.4               103              11.45
3             110            10.30               162.6               104               7.32
4              88             5.26               196.9                89               8.86
5             122            12.61               186.9               121               8.41
6             101            18.75               203.9               118               9.18
  total.intl.minutes total.intl.calls total.intl.charge number.customer.service.calls  Churn
1               10.0                3              2.70                             1  False.
2               13.7                3              3.70                             1  False.
3               12.2                5              3.29                             0  False.
4                6.6                7              1.78                             2  False.
5               10.1                3              2.73                             3  False.
6                6.3                6              1.70                             0  False.
```

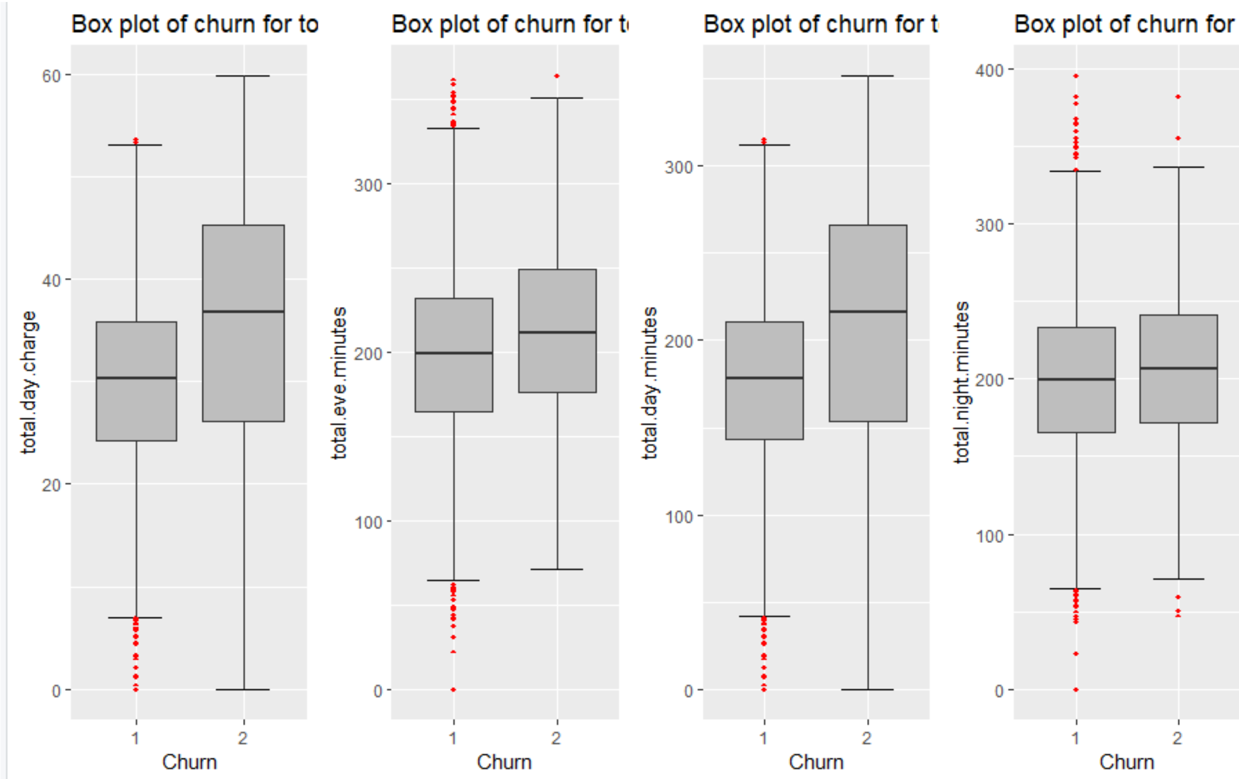
The dataset after variable transformation

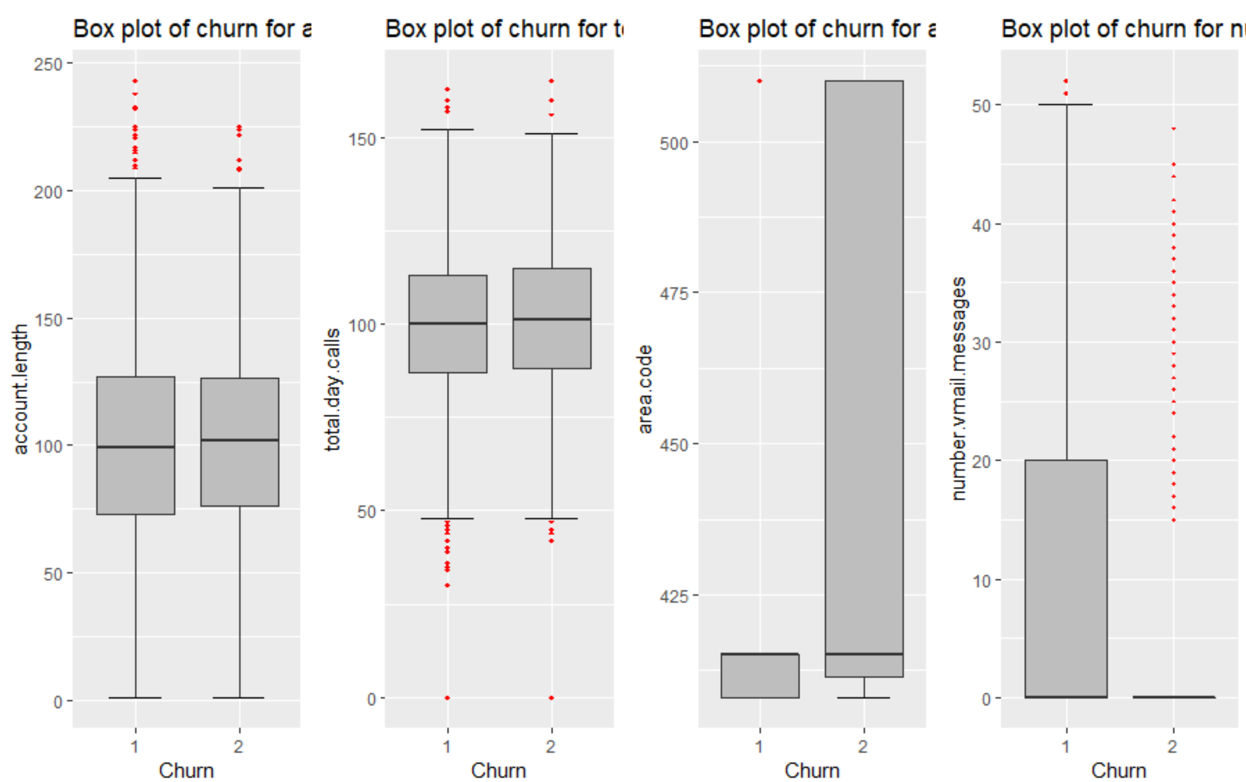
```
> for(i in 1:ncol(data)){
+
+   if(class(data[,i]) == 'factor'){
+
+     data[,i] = factor(data[,i],
+
+   labels=(1:length(levels(factor(data[,i])))))
+
+   }
+ }
> head(data)
  state account.length area.code phone.number international.plan voice.mail.plan
1    17            128       415         1927                  1               2
2    36            107       415         1576                  1               2
3    32            137       415         1118                  1               1
4    36             84       408         1708                  2               1
5    37             75       415          111                  2               1
6     2            118       510         2254                  2               1
  number.vmail.messages total.day.minutes total.day.calls total.day.charge total.eve.minutes
1                    25             265.1             110            45.07             197.4
2                    26             161.6             123            27.47             195.5
3                     0             243.4             114            41.38             121.2
4                     0             299.4              71            50.90              61.9
5                     0             166.7             113            28.34             148.3
6                     0             223.4              98            37.98             220.6
  total.eve.calls total.eve.charge total.night.minutes total.night.calls total.night.charge
1              99            16.78               244.7                91              11.01
2             103            16.62               254.4               103              11.45
3             110            10.30               162.6               104               7.32
4              88             5.26               196.9                89               8.86
5             122            12.61               186.9               121               8.41
6             101            18.75               203.9               118               9.18
  total.intl.minutes total.intl.calls total.intl.charge number.customer.service.calls Churn
1               10.0                3              2.70                             1     1
2               13.7                3              3.70                             1     1
3               12.2                5              3.29                             0     1
```

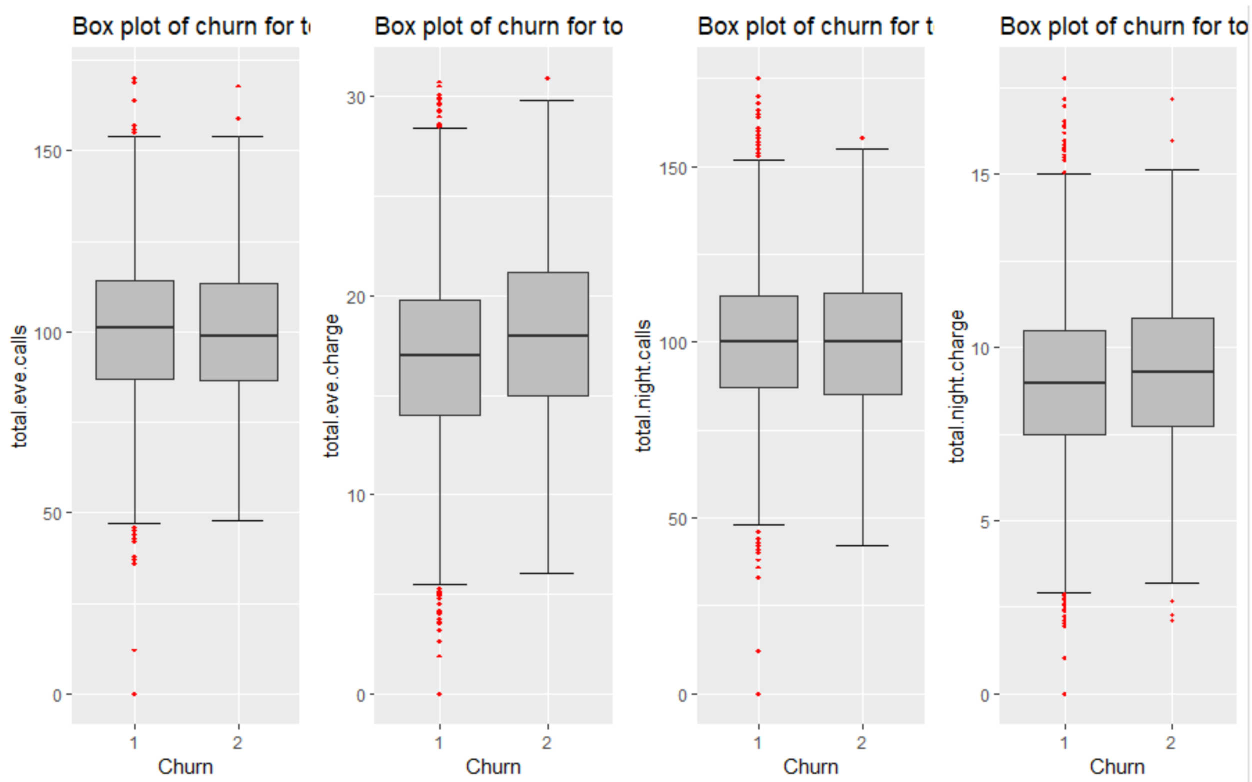
2.1.3 Outlier analysis:

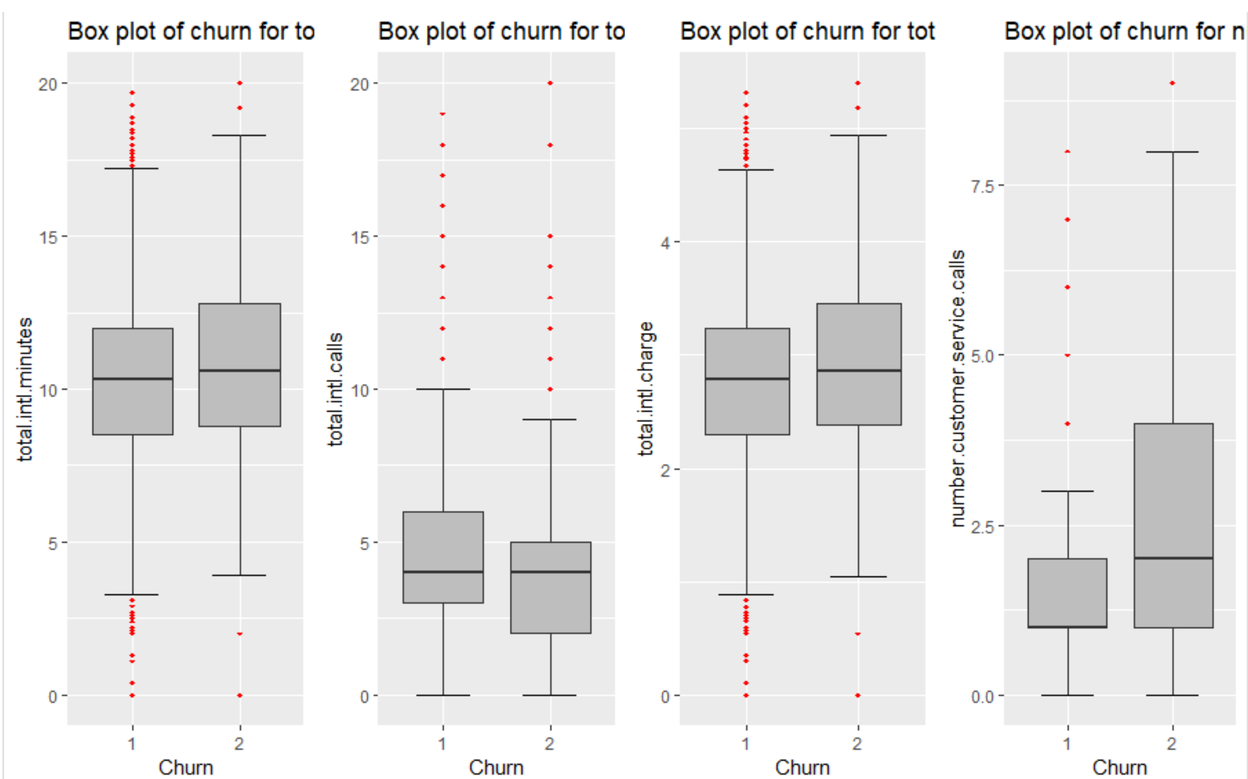
Outliers are points in a dataset that lie far away from the estimated value of the centre of tha dataset. This estimated centre could be either the mean, or median,or percentile. Outlier detection is an important aspect of machine learning algorithms of any sophistication. Because of the fact that outliers can throw off a machine learning algorithm or deflate an assumption about the dataset.

Here we remove the ouliers which are in the red circles using boxplot we remove them.


Box plot of churn for to
Box plot of churn for t
Box plot of churn for t
Box plot of churn for t
total.day.charge
total.eve.minutes
total.day.minutes
total.night.minutes
Churn
Churn
Churn
Churn


Box plot of churn for a
Box plot of churn for t
Box plot of churn for a
Box plot of churn for nu
account.length
total.day.calls
area.code
number.vmail.messages
Churn
Churn
Churn
Churn





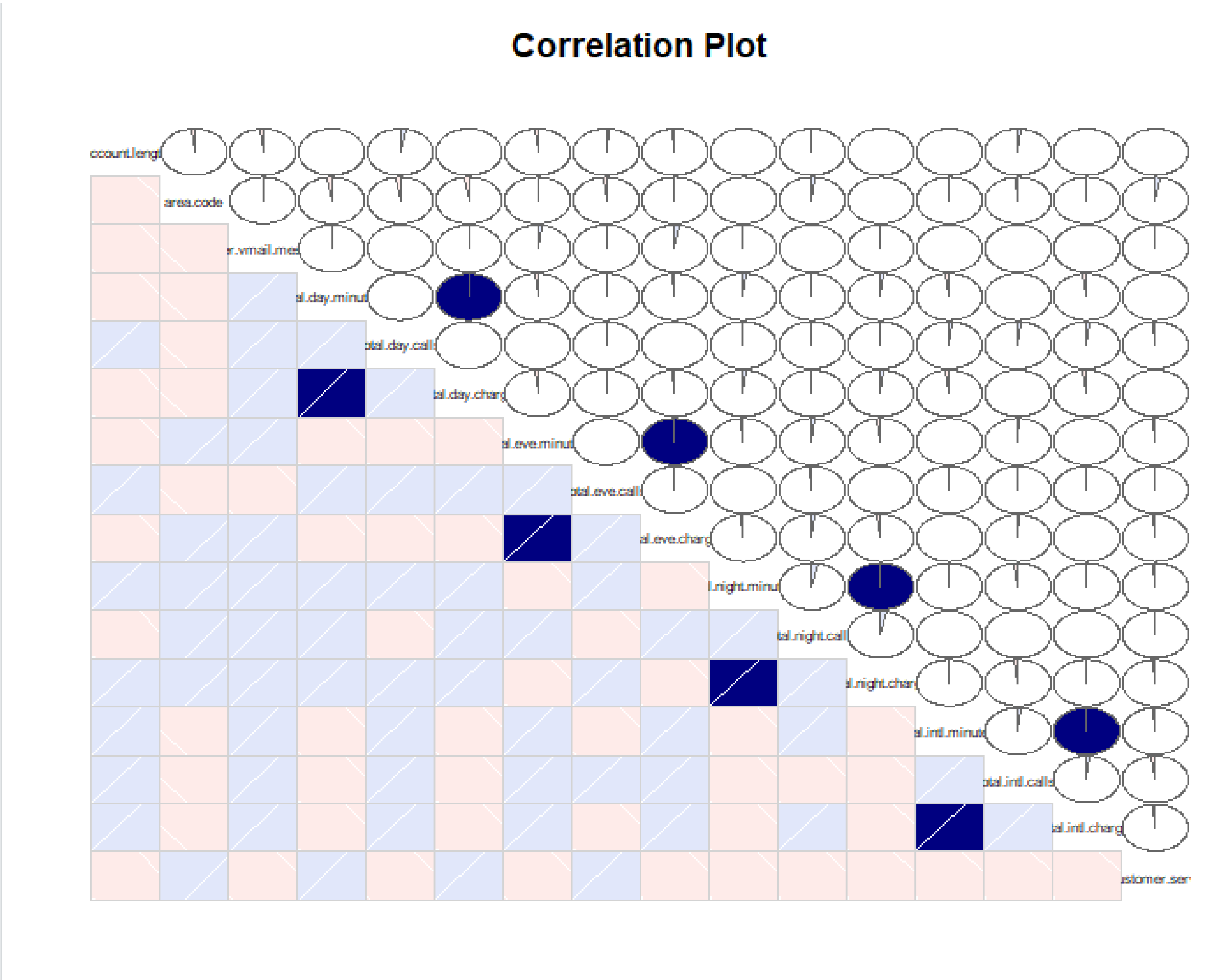
Above which are in the red color circles are the outliers we have to detect them .

2.1.4 Feature Selection:

For feature selection we perform two task

a. Correlation plot
b. Chi-square test

a.Cerrelation plot : correlation plot for categorical data we find out the correlation betwwen dependent variable and categorical variable.



In above find out which are blue circle or rectangular shape are highly correlated if we find out the red circle or rectangular we can say that negative correlated and neglect therm.

b.Chi-square test:

in this we find out the p value of the numeric dataset if p-value is greater then >0.05 we neglect that independent variable from the dataset.

```
[1] "state"

        Pearson's Chi-squared test

data:  table(factor_data$Churn, factor_data[, i])
X-squared = 96.899, df = 50, p-value = 7.851e-05

[1] "phone.number"

        Pearson's Chi-squared test

data:  table(factor_data$Churn, factor_data[, i])
X-squared = 5000, df = 4999, p-value = 0.4934

[1] "international.plan"

        Pearson's Chi-squared test with Yates' continuity correction

data:  table(factor_data$Churn, factor_data[, i])
X-squared = 333.19, df = 1, p-value < 2.2e-16

[1] "voice.mail.plan"

        Pearson's Chi-squared test with Yates' continuity correction

data:  table(factor_data$Churn, factor_data[, i])
X-squared = 60.552, df = 1, p-value = 7.165e-15
```

In above we can say that independent variable “phone number” have p- value >0.05

We easily remove them from the dataset.

2.1.5 **Feature Scalling** : in feature scalling we can perform two method.

a. Normalization
b. Standardization

   a. Normalization:
      In this we normalize the datset we impute the numeric dataset between 0 and 1.
   b. Standardization : standardization refers to the shifting the distribution of each attribute to have a mean of zero and a standard deviation of 1.

It is hard to know whether rescalling your data will improve the performance of your algorithms before you apply them. If often can, but not always.

```r
#Normalisation
cnames = c("account.length","area.code","number.vmail.messages",
           "total.day.minutes","total.day.calls","total.day.charge",
           "total.eve.minutes","total.eve.calls","total.eve.charge",
           "total.night.minutes","total.night.calls","total.night.charge",
           "total.intl.minutes","total.intl.calls","total.intl.charge",
           "number.customer.service.calls")

#for(i in cnames){
 # print(i)
  #data[,i] = (data[,i] - min(data[,i]))/(max(data[,i] - min(data[,i])))
#}
#Standardization
 for(i in cnames){
   print(i)
   data[,i] = (data[,i] - mean(data[,i]))/sd(data[,i])
 }
View(data)
```

2.1.6 **Feature Sampling** : in this we select the dataset from the whole dataset which have impact on the whole dataset to analyze the dataset we make a subset of the whole data which is called the data sampling.

2.1.7 **Feature Splitting** :

In this we divide the dataset in two part like train and test dataset its ratio like 70:30
70%train and 30% test dataset

2.2Modeling

2.2.1 Model Selection :

According to the dataset we select our model if we have categorical dependent variable like yes or no,

We perform classification that dataset if we have continuous dependent variable like in numeric we perform regression on that dataset.

Here we have to predict value is yes or no so we have classification model on that we perform classification analysis.

2.2.2 **Classification** :

We have to predict churn variable which is in the form of yes or no

To apply algorithm for that we have used multiple algorithm one at a time but we find the maximum accuracy using the random forest algorithm.

```
> confusionMatrix(ConfMatrix_RF)
Confusion Matrix and Statistics

   RF_Predictions
      1   2
  1 554   2
  2  24  41

               Accuracy : 0.9581
                 95% CI : (0.9393, 0.9725)
    No Information Rate : 0.9308
    P-Value [Acc > NIR] : 0.002802

                  Kappa : 0.7374
 Mcnemar's Test P-Value : 3.814e-05

            Sensitivity : 0.9585
            Specificity : 0.9535
         Pos Pred Value : 0.9964
         Neg Pred Value : 0.6308
             Prevalence : 0.9308
         Detection Rate : 0.8921
   Detection Prevalence : 0.8953
      Balanced Accuracy : 0.9560

       'Positive' Class : 1

>
```

3. R CODE

```
#Load Libraries
x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest",
      "unbalanced", "C50", "dummies", "e1071", "Information",
      "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')

#install.packages(x)
lapply(x, require, character.only = TRUE)
rm(x)
train=read.csv(file.choose(),sep=',',header = T)

test=read.csv(file.choose(),sep=',',header = T)
data=rbind(train,test)
View(data)
head(data)
str(data)
#Missing Value Trweatment
table(is.na(data))
```

```r
#**************************Variable transformation****************
table(is.na(data))
for(i in 1:ncol(data)){

  if(class(data[,i]) == 'factor'){

    data[,i] = factor(data[,i],

  labels=(1:length(levels(factor(data[,i])))))

  }
}

View(data)
#************************Outlier********************************************

# BoxPlots - Distribution and Outlier Check
numeric_index = sapply(data,is.numeric) #selecting only numeric

numeric_data = data[,numeric_index]

cnames = colnames(numeric_data)
cnames
library(ggplot2)
```

```r
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,gn5,gn2,gn3,ncol=4)
gridExtra::grid.arrange(gn6,gn7,gn4,gn10,ncol=4)
gridExtra::grid.arrange(gn8,gn9,gn11,gn12,ncol=4)
gridExtra::grid.arrange(gn13,gn14,gn15,gn16,ncol=4)


# # #loop to remove from all variables
 for(i in cnames){
   print(i)
   val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
   print(length(val))
   data = data[which(!data[,i] %in% val),]
 }
```

```r
 for(i in cnames){
  val = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
  print(length(val))
   data[,i][data[,i] %in% val] = NA
 }

data = knnImputation(data, k = 3)
table(is.na(data))
library(corrgram)
###################################Feature Selection#######################
## Correlation Plot
corrgram(data[,numeric_index], order = F,upper.panel=panel.pie,
         text.panel=panel.txt, main = "Correlation Plot")

## Chi-squared Test of Independence
factor_index = sapply(data,is.factor)
factor_index
factor_data = data[,factor_index]
factor_data
names(factor_data)

for (i in 1:4)
{
  print(names(factor_data)[i])
  print(chisq.test(table(factor_data$Churn,factor_data[,i])),
        simulate.p.value = TRUE)
}
```

```r
## Dimension Reduction
data = subset(data,select = -c(phone.number))

#################################Feature Scaling##########################
#Normality check
qqnorm(data$account.length)
hist(data$number.vmail.messages)

#Normalisation
cnames = c("account.length","area.code","number.vmail.messages",
           "total.day.minutes","total.day.calls","total.day.charge",
           "total.eve.minutes","total.eve.calls","total.eve.charge",
           "total.night.minutes","total.night.calls","total.night.charge",
           "total.intl.minutes","total.intl.calls","total.intl.charge",
           "number.customer.service.calls")

#for(i in cnames){
 # print(i)
#   data[,i] = (data[,i] - min(data[,i]))/(max(data[,i] - min(data[,i])))
#}

# #Standardisation
 for(i in cnames){
   print(i)
   data[,i] = (data[,i] - mean(data[,i]))/sd(data[,i])
 }
View(data)
```

```
View(data)


#########################################Sampling#########################
# ##Simple Random Sampling
 data_sample = data[sample(nrow(data), 1000, replace = F), ]


#Divide data into train and test using stratified sampling method
set.seed(1234)
library(caret)
sampl = createDataPartition(data$Churn, p = .80, list = FALSE)
train = data[ sampl,]
test  = data[-sampl,]
View(test)
# Random Forest
RF_model = randomForest(Churn ~ ., train, importance = TRUE, ntree = 500)

#Presdict test data using random forest model
RF_Predictions = predict(RF_model, test[,-20])

##Evaluate the performance of classification model
ConfMatrix_RF = table(test$Churn, RF_Predictions)
confusionMatrix(ConfMatrix_RF)
#Accuracy
#95.81
```

```
> confusionMatrix(ConfMatrix_RF)
Confusion Matrix and Statistics

   RF_Predictions
     1   2
  1 554   2
  2  24  41

               Accuracy : 0.9581
                 95% CI : (0.9393, 0.9725)
    No Information Rate : 0.9308
    P-Value [Acc > NIR] : 0.002802

                  Kappa : 0.7374
 Mcnemar's Test P-Value : 3.814e-05

            Sensitivity : 0.9585
            Specificity : 0.9535
         Pos Pred Value : 0.9964
         Neg Pred Value : 0.6308
             Prevalence : 0.9308
         Detection Rate : 0.8921
   Detection Prevalence : 0.8953
      Balanced Accuracy : 0.9560

       'Positive' Class : 1

>
```