Churn Reduction

MD SAMEER KHALEEL M

20 February 2019

Contents

CHAPTER 1

1. Introduction

1.1.Problem Statement:

The aim of the Project is to find out how many customers churn out of the network, the churn rate is the percentage of subscribers who discontinue their subscriptions to the service within a given time period. For a company to expand its growth rate, the number of new customers must exceed its churn rate. We can utilize insights obtained from our analysis to predict customers who are likely to churn. We can identify the causes for churn as a customer may churn for many reasons so we should work to resolve those issues like engaging with customers to foster relationships.

1.2.Data:

Given below is the sample of the dataset

Sample Data (Columns: 1-10)

state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge
KS	128	415	382-4657	no	yes	25	265.1	110	45.07
OH	107	415	371-7191	no	yes	26	161.6	123	27.47
NJ	137	415	358-1921	no	no	0	243.4	114	41.38
OH	84	408	375-9999	yes	no	0	299.4	71	50.9
OK	75	415	330-6626	yes	no	0	166.7	113	28.34
AL	118	510	391-8027	yes	no	0	223.4	98	37.98
MA	121	510	355-9993	no	yes	24	218.2	88	37.09

Sample Data (Columns: 10-20)

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	Churn
197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	False.
195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	False.
121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	False.
61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	False.
148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	False.
220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	False.
348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	False.

Predictor Values:

State
account length
area code
phone number
international plan
voice mail plan
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

Dependent Values:

Churn

Chapter 2

2. Methodology

2.1.PreProcessing

First, we look at the data, looking at the data refers to exploring the data, cleaning the data checking for missing values and visualizing the data through graphs and plots. First we look at the size of the data and what variables contribute to our 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 1708 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 ...
```

We infer from above, the shape of the data set and the variables which are contributing to our data, Now, we transform the dataset in the following ways so that we can get started up with our EDA. We notice that that the variables state, phone number, international plan, voice mail plan and Churn is in factor, we convert it to categorical for easy analysis converting the

factors to categories by replacing yes with 1 and no with 0, also the names of the states are replaced by the corresponding numbers.

Now, we check for missing values, since there are no missing values in the dataset we can move on with EDA.

```
table(is.na(df))
FALSE
105000
```

First we will analyze all the probability distributions of the variables, ,we see from the histogram from Churn the distribution that most of the data is distributed towards False meaning few customers have churned. Similarly other inferences can be made from probability distribution diagrams, For Classification ideally it is preferred to have the data normally distributed. In Fig 2.1, we can see the probability density functions for all Independent integer variables. We did not perform this on the variable Phone numbers as it is not necessary and we use histogram to infer information from them. The blue lines indicate Kernel Density Estimations (KDE) of the variable and the red lines represent the normal distribution, we can infer from the plot most variables resemble normal distribution

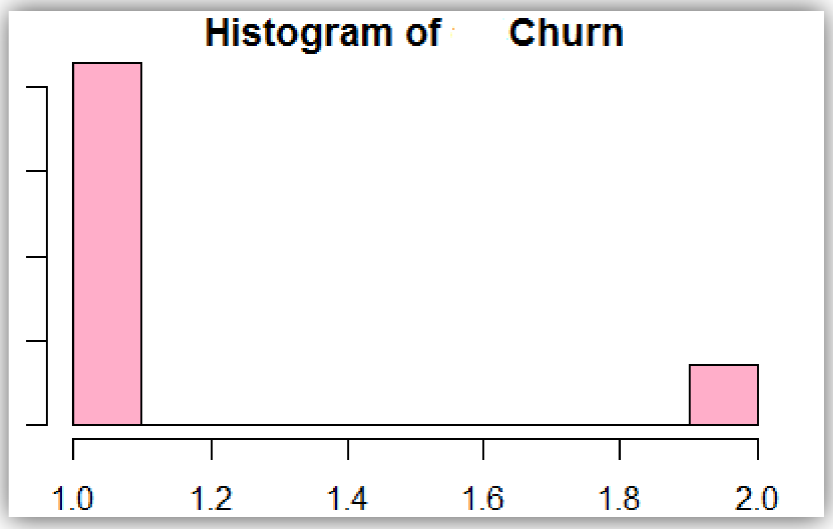


Fig 2.0 – Histogram of Churn

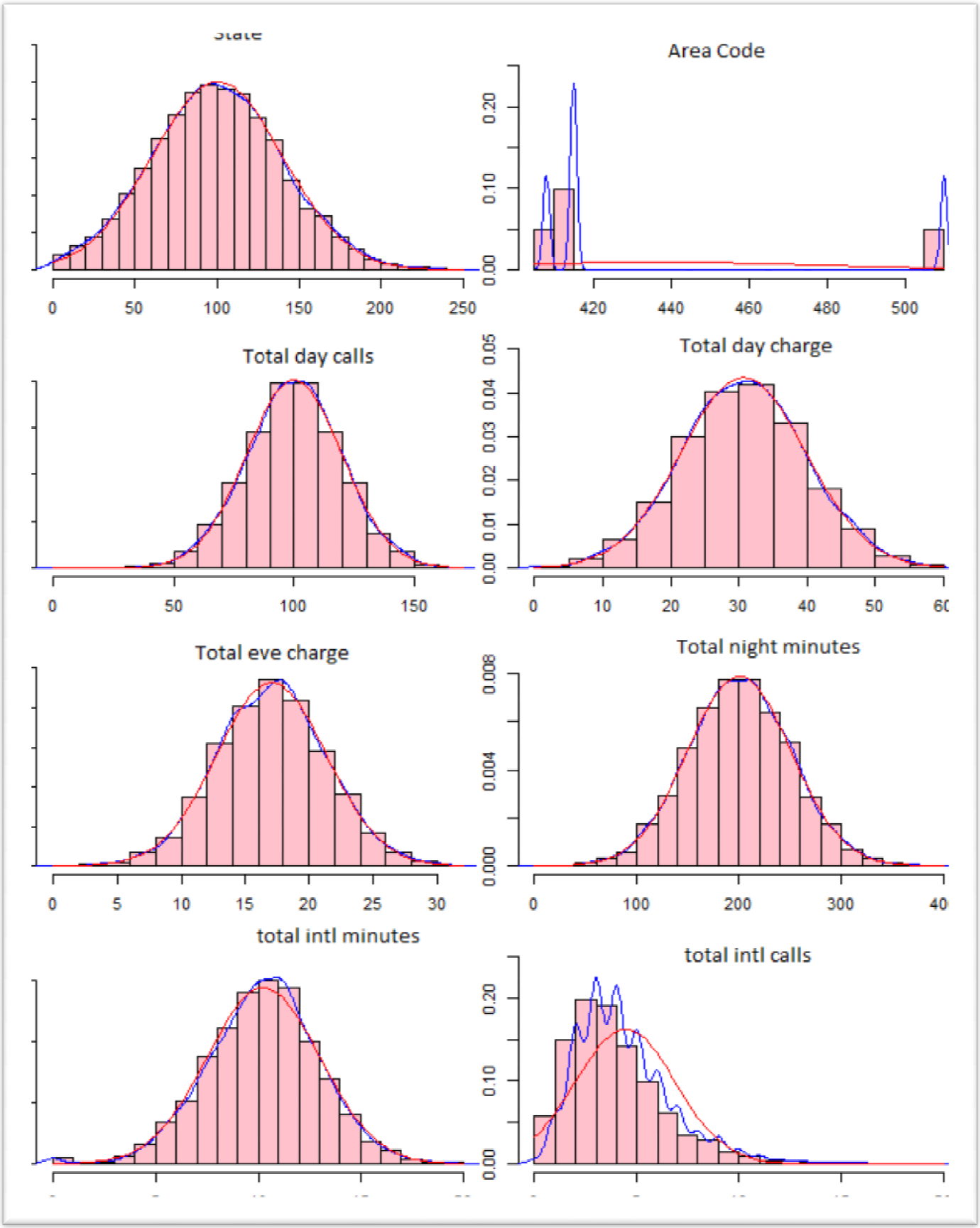


Figure 2.1 Probability Distribution Plot (a)

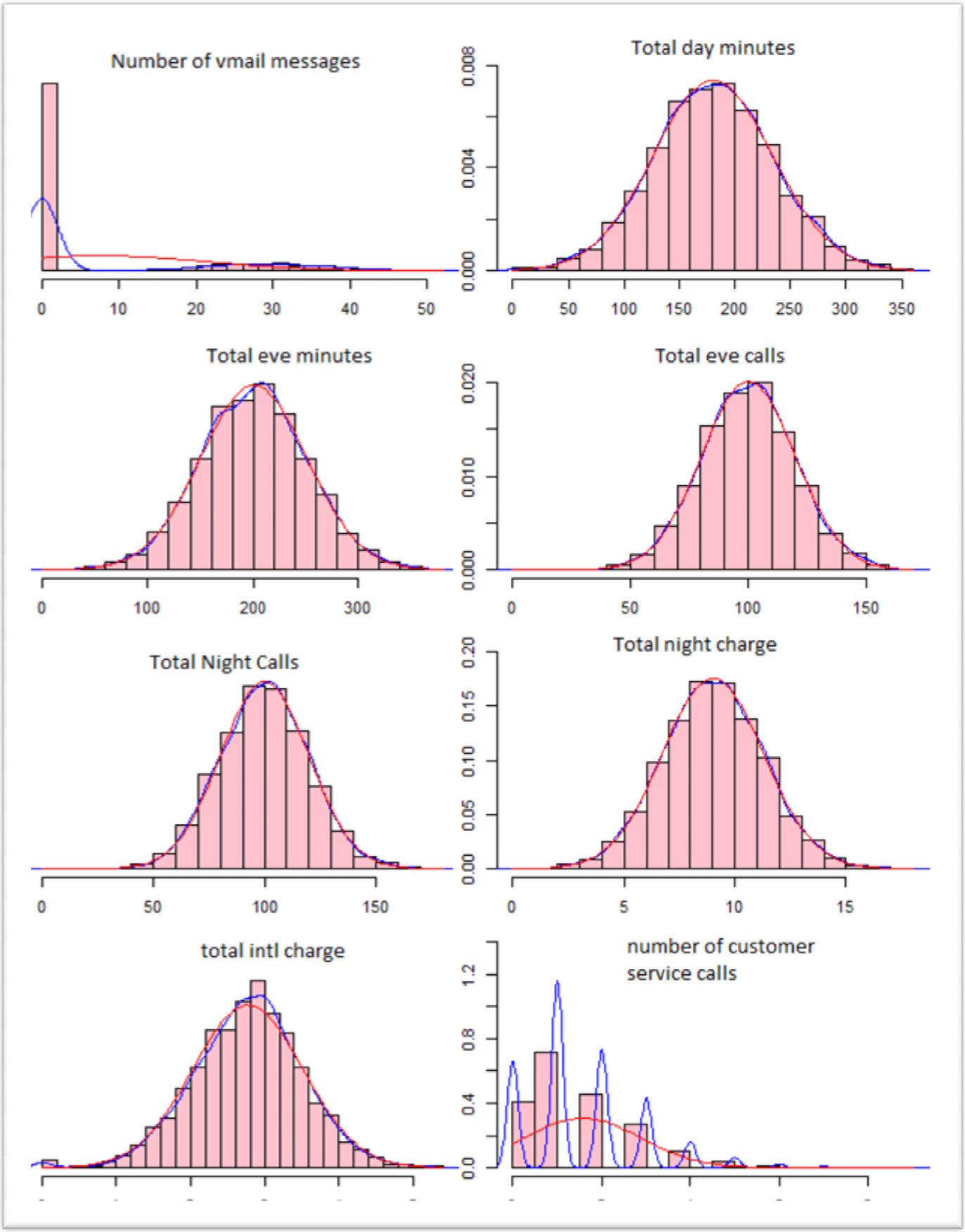


Figure 2.2 Probability Distribution Plot (b)

2.2.Exploratory Data Analysis

From our histogram, we have seen that the no. of customers who chose not to churn is more than the Customers who churned, we have plotted the box plot for all the variables we have seen that most people who churned out had an international plan, this could mean that international plan was an important variable in people churning out, also most of them did not have voice mail service activated. these provide useful insights to the data and they can be noted and passed on to help prevent churn of customers.

2.3.Outlier analysis:

Outlier is an observation which is inconsistent with the rest of the dataset there are many techniques to remove outliers from the dataset, graphical technique is the boxplot method, statistical technique is the Grubbs test, the outlier package in r and using experiment like replacing the outliers with NA, the disadvantage with Grubbs test is that it works only on normally distributed data and very few datasets are normally distributed, r package uses the mean concept, it will calculate the mean of whole variables and compare all the values with mean, and if the values is falling away from the mean it will count as an outlier these method are less effective than the Boxplot as boxplot gives the accurate model for regardless the data being normalized or not.From the histograms and Probability Distributions we see that a few of the variables are skewed which are most likely due to the presence of outliers, In fig 2.3, we have plotted the boxplot of all the 19 independent variables. we have plotted the box plot for all the variables we have seen that most people who churned out had an international plan, this could mean that international plan was an important variable in people churning out, also most of them did not have voice mail service activated,

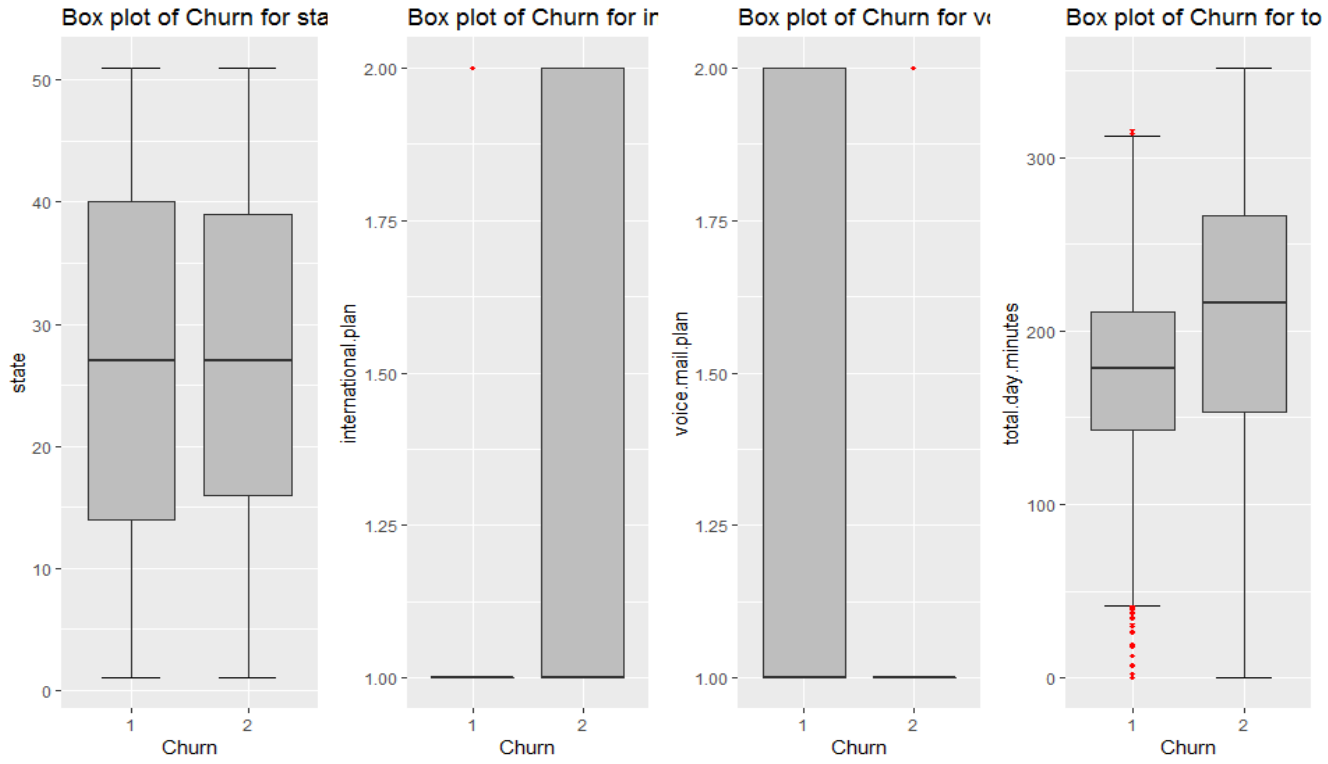


Fig 2.3 Box Plot Analysis (a)

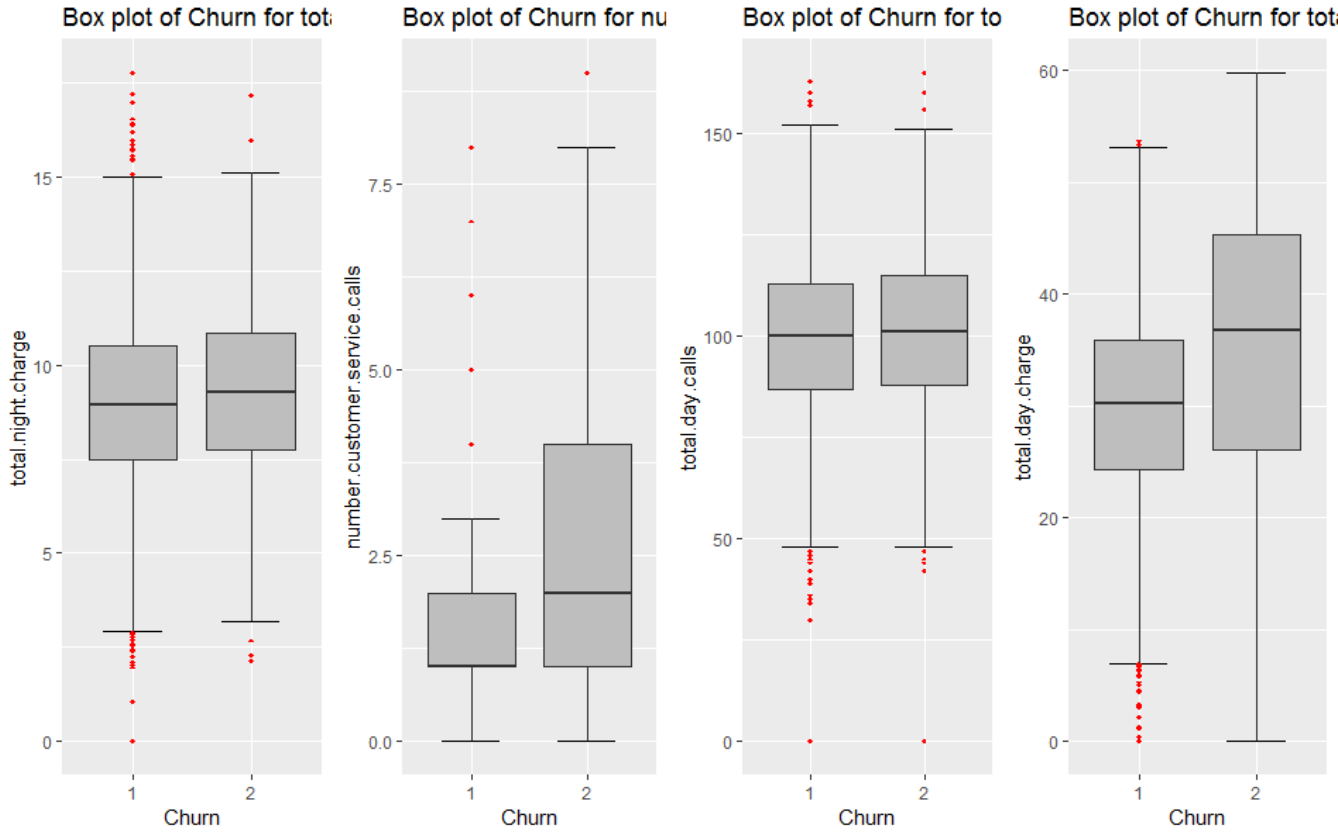


Fig 2.4 Box Plot Analysis (b)

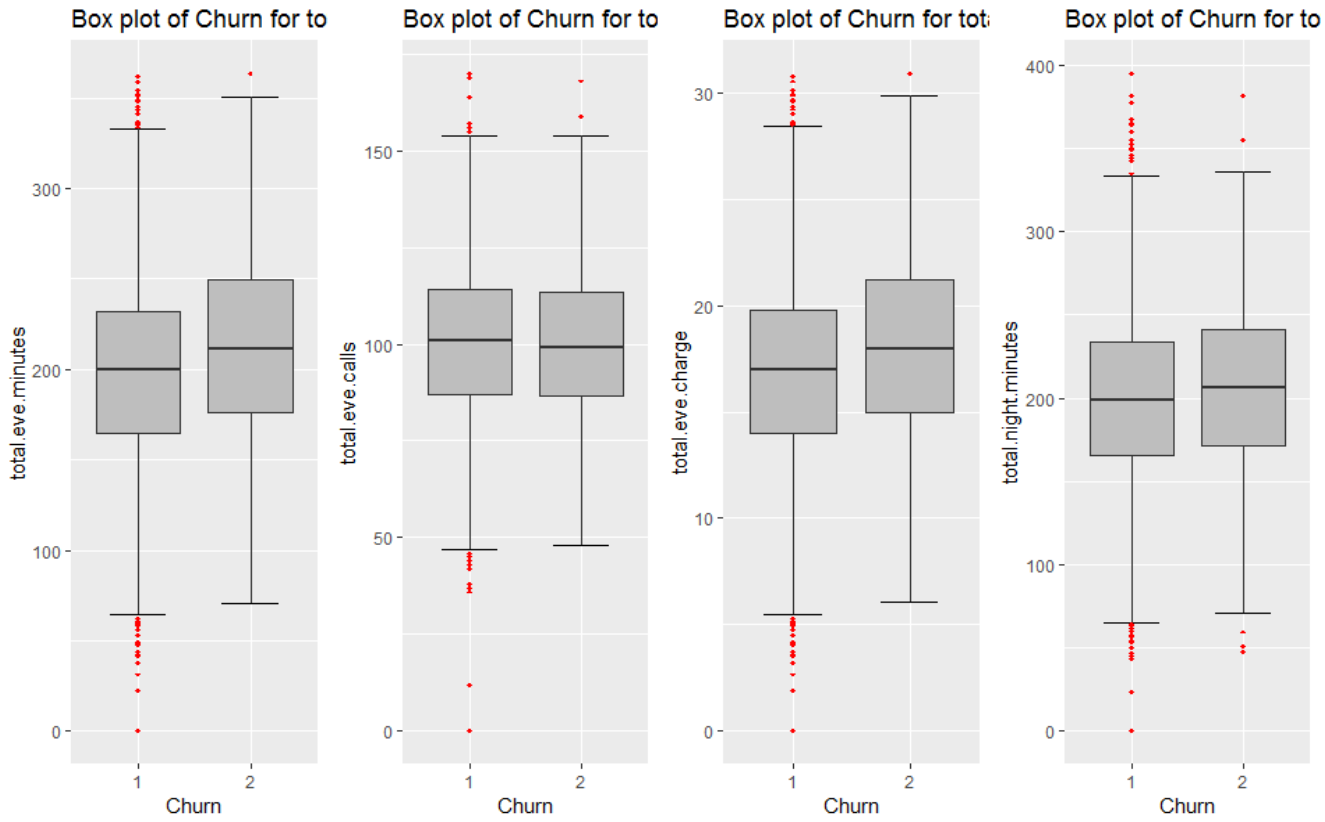


Fig 2.4 Box Plot Analysis (c)

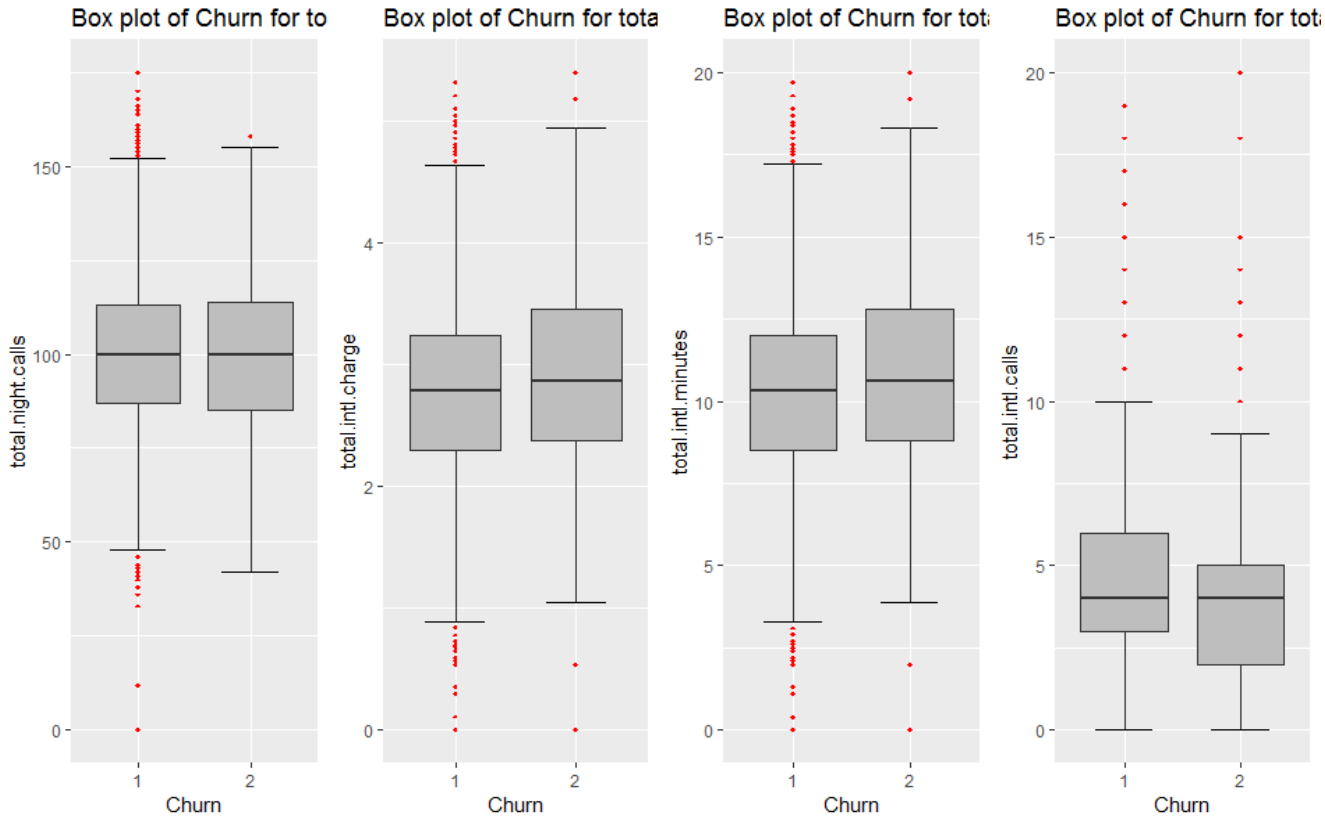


Fig 2.4 Box Plot Analysis (d)

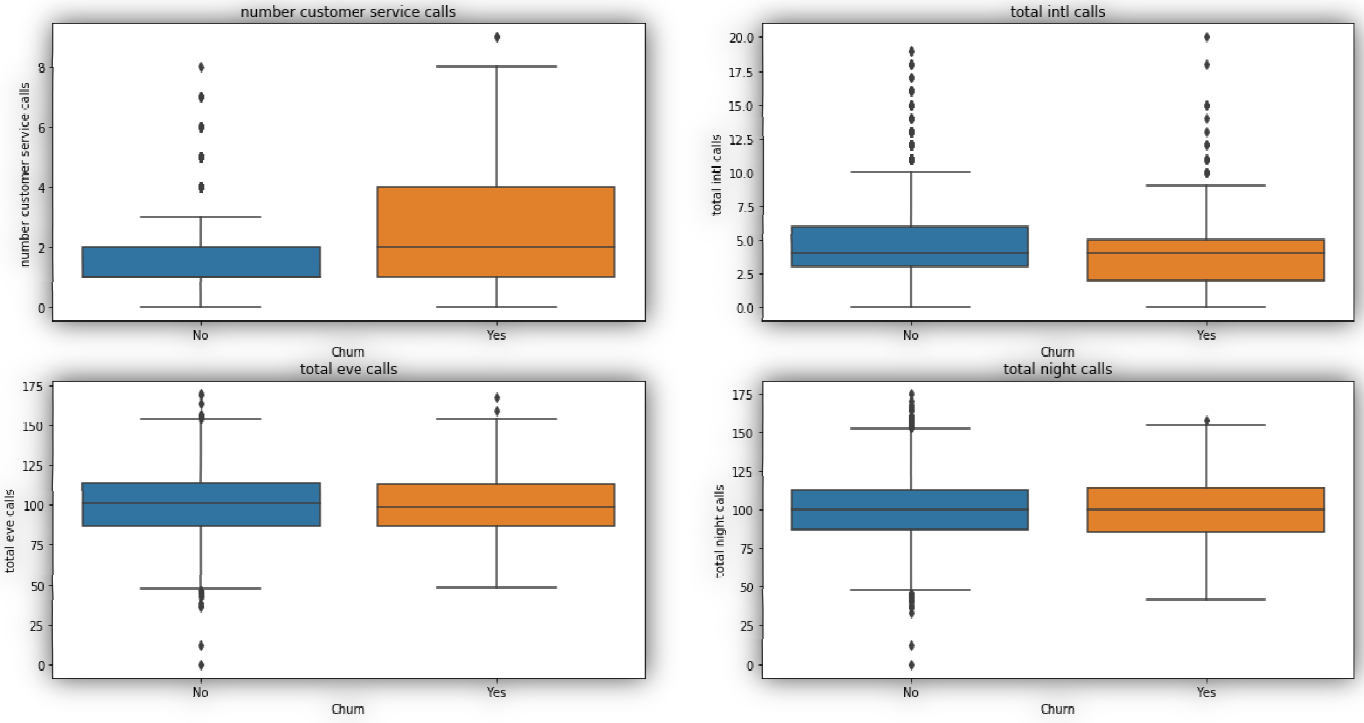


Fig 2.4 Churn Vs Variables Boxplot

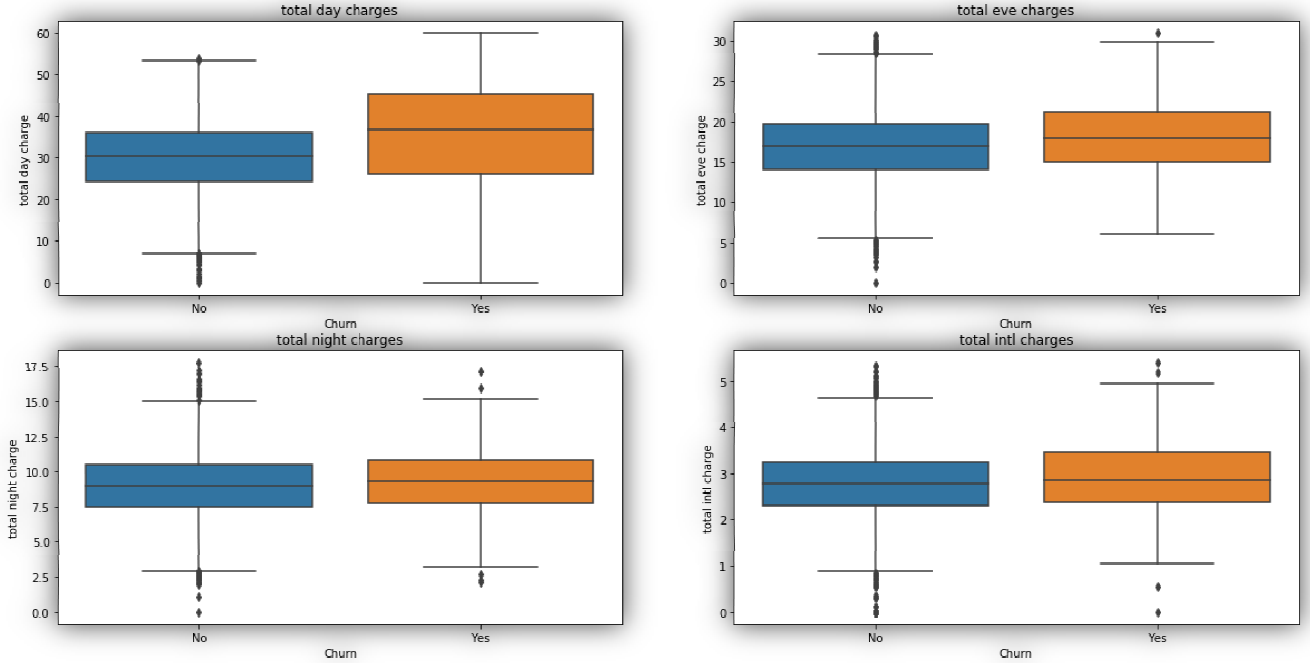


Fig 2.5 Churn Vs Variables Boxplot

2.4.Feature Selection

Feature Selection is an important step in data processing and modelling as it can help us avoid unnecessary variables which might bloat our model and can even have negative effects on our model. Feature selection is selecting a subset of feature from the dataset which reduces the complexity of the data, time consumption and memory consumption; we can find out the variables which are not contributing to our dataset, we have used random forest to perform feature selection

```
> impvar=randomForest(Churn~.,data=df,ntree=100,keep.forest=F,importance=T)
> importance(impvar,type=1)
                               MeanDecreaseAccuracy
 state                                    -0.4403871
account.length                           -2.3493366
area.code                                 0.2956045
international.plan                       34.0151916
voice.mail.plan                           9.8253945
number.vmail.messages                    11.7763111
total.day.minutes                        21.4937099
total.day.calls                           0.6994652
total.day.charge                         21.5521870
total.eve.minutes                        13.1449206
total.eve.calls                          -1.1155171
total.eve.charge                         13.6038847
total.night.minutes                       9.9388797
total.night.calls                        -1.6335349
total.night.charge                        9.5586598
total.intl.minutes                       14.5368054
total.intl.calls                         20.5239717
total.intl.charge                        12.2832951
number.customer.service.calls            49.2247016
```

We see that account length and state are not important to the predictor variable in our analysis and hence we can ignore them,

2.5.Correlation analysis:

One method to check which variable doesn't give much information to the model is via correlation analysis, Correlation coefficient can be calculated by the formula (1-r2)=1/VIF, from the original formula VIF = 1/(1 - r²), Below we have used correlation analysis on our data and found out that total day charge is highly correlated with total day minutes similarly evening and night call are correlated with evening and night charge.

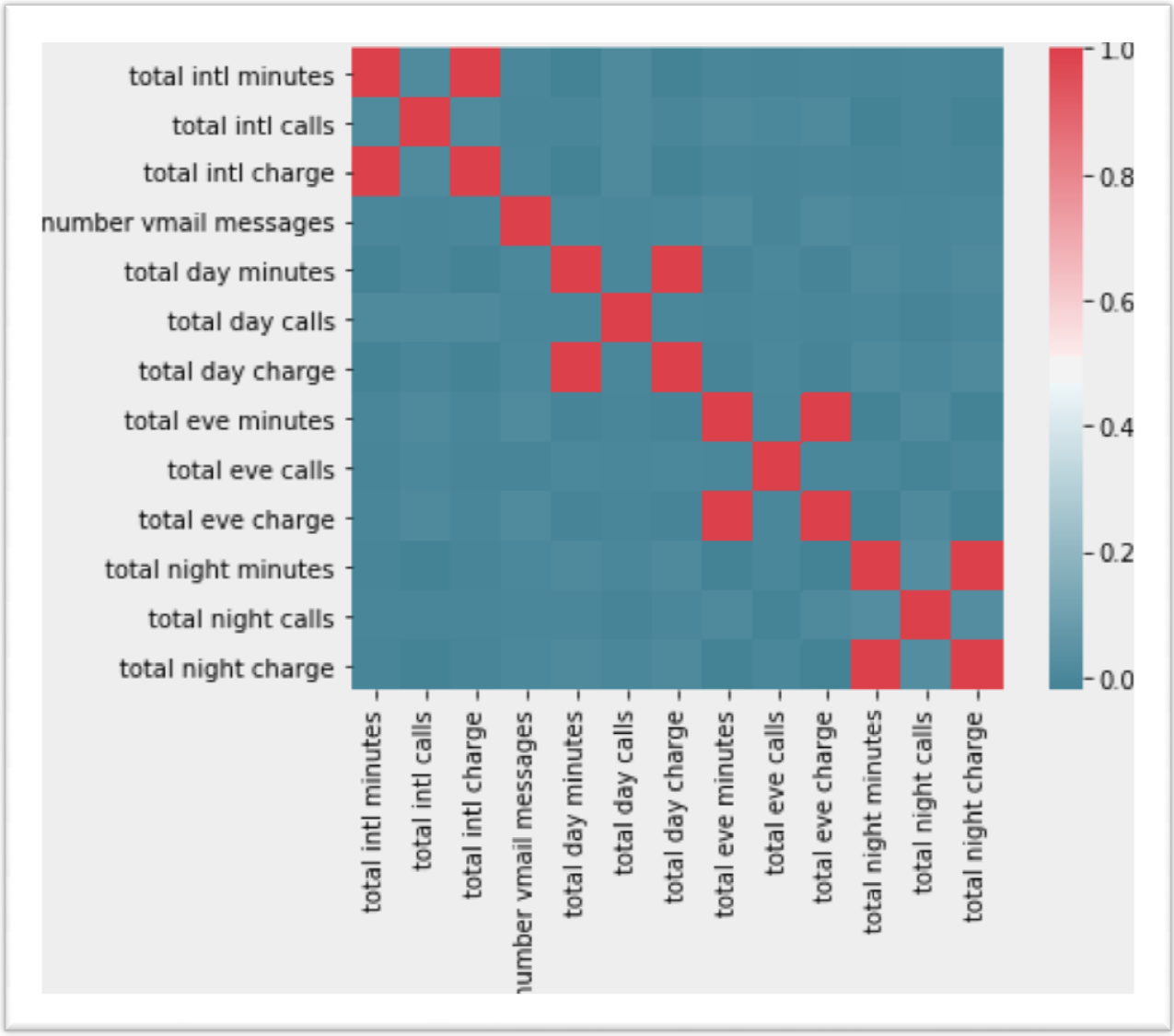


Fig 2-3 Correlation Plot

2.6.Modelling:

To Model our data, we are going to use classification methods, a single model is to be selected which gives the best result. We know from preprocessing that the method to use would be classification method, there are 3 popular methods for Classification which we are going to use today, and they are

1. Random Forest
2. NaiveBayes
3. Decision Tree

2.6.1. Random Forest

The random forest model is very good at handling tabular data with numerical features, fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target.

2.6.2. NaiveBayes:

NaiveBayes is used for data mining but we can also used for classification analysis, it is used only for classification models, it is a probabilistic classification and gives yes or no depending on the probability

2.6.3. Decision Tree:

Decison Tree is a predictive model based on a branching series of Boolean tests, Decision tree is a rule, each branch connects nodes with “and” and multiple branches are connected by “or”, we use C5.0 which is a very famous and useful decision tree algorithm, its main functions are multi split, Information gain, Rule based Pruning

Chapter 3

3. Conclusion

3.1.Model Evaluation

After modelling a few models for predicting the count, we have to choose the right one based on

1. Prediction Performance
2. Complexity
3. Efficiency and the ability to mould according to data

   Our main aim is the Prediction Performance and hence we choose the model which gives high prediction performance. Prediction Performance can be measured by comparing Predictions of the models with real values, the avg error is used to determine the best model

3.2.Model Selection

We check the performance metrics of the given models and choose the one which is performing the best.

3.2.1. Confusion Matrix:

The Confusion matrix is one of the most intuitive and easiest metrics used for finding the accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes.

We see that random forest gives the best performance metric with the Accuracy of 96% and FNR of 26%

```
RF_Predictions
       1    2
  1 1440    3
  2   59  165
```

Where as, decision tree method gave

```
   C50_Predictions
       1    2
  1 1435    8
  2   57  167
```

#Accuracy:92.68

#FNR: 30.80%

NaiveBayes method gave the result as

```
predicted
observed    1    2
       1 1394   49
       2  129   95
```

#Accuracy: 89.32%

#FNR: 57.58

Comparing the models, we choose the Random Forest method to predict the model

Appendix A-Extra Figures

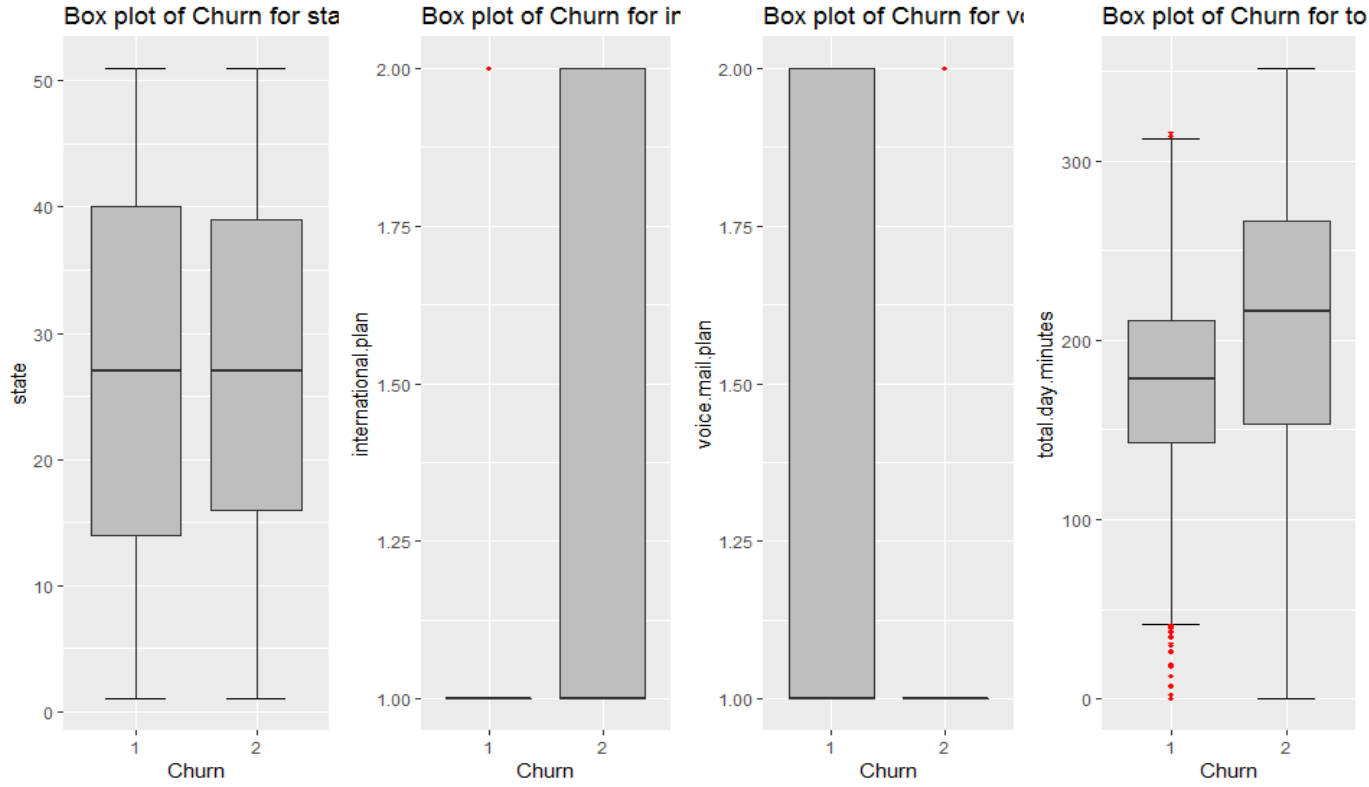


Fig 2.4 Box Plot Analysis (a)

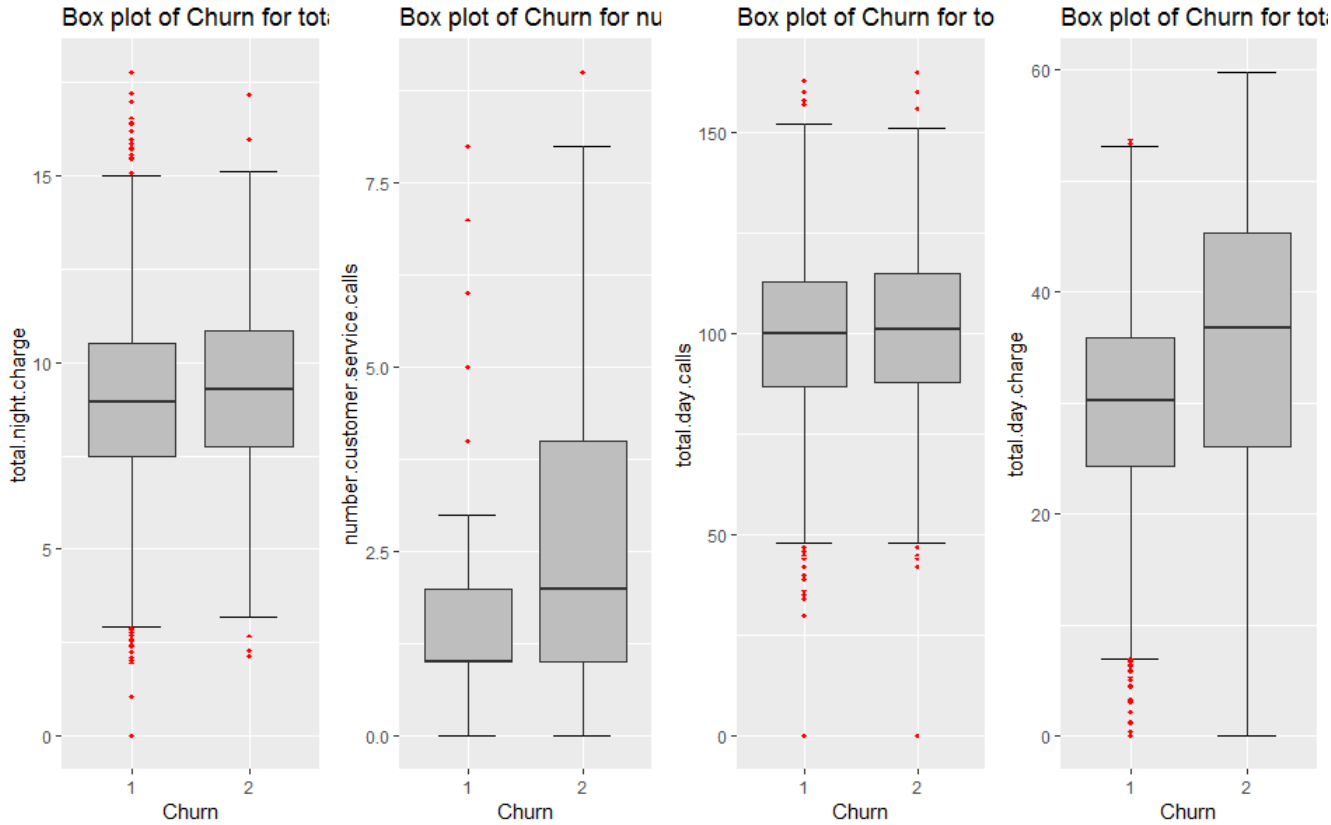


Fig 2.4 Box Plot Analysis (b)

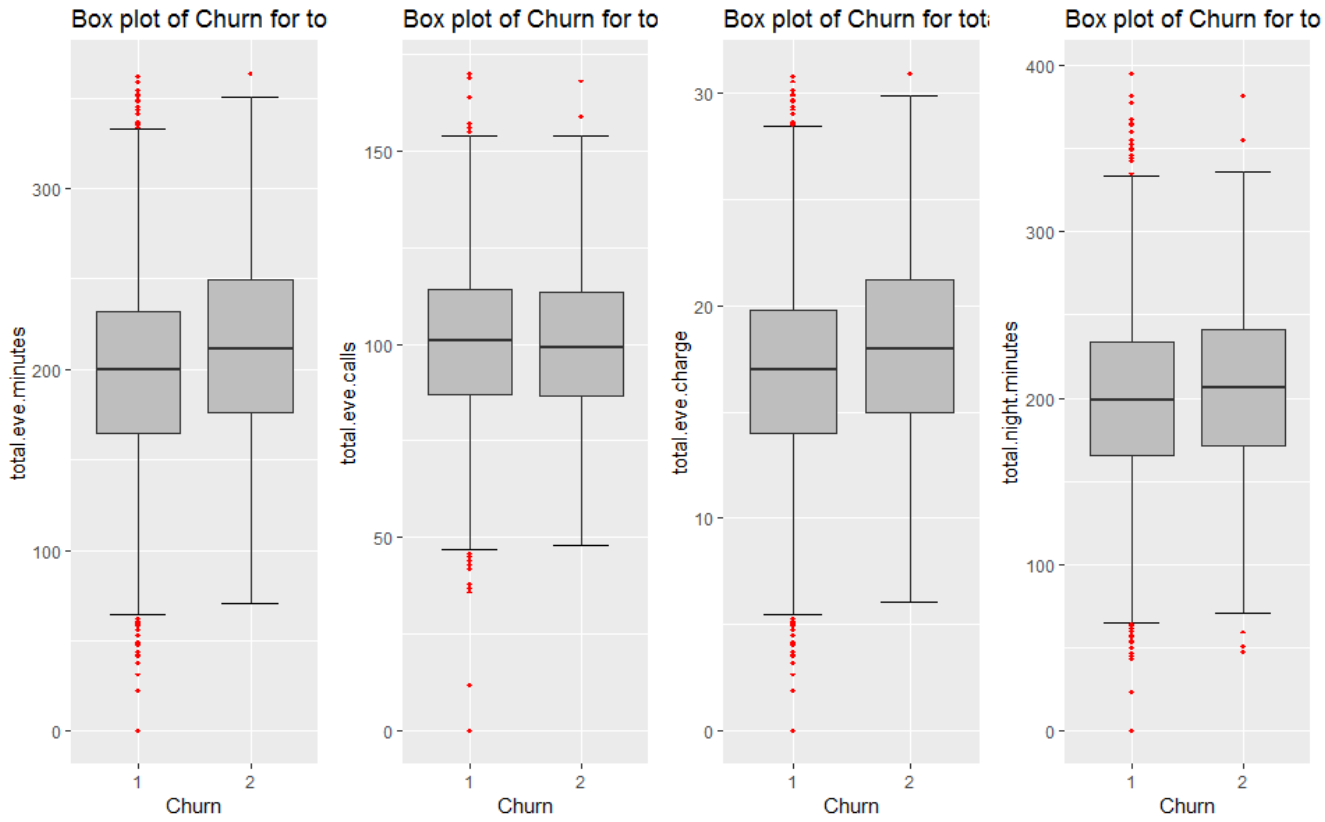


Fig 2.4 Box Plot Analysis (c)

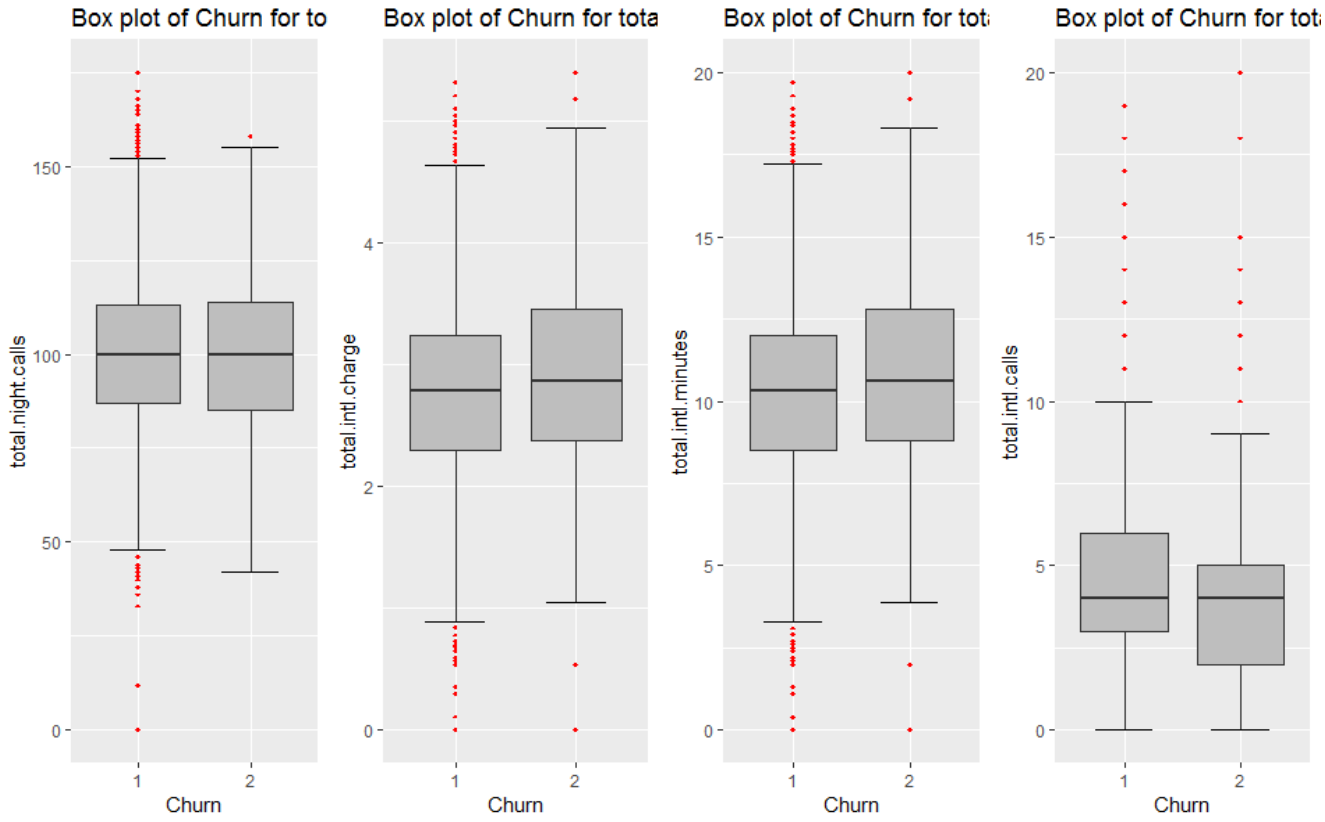
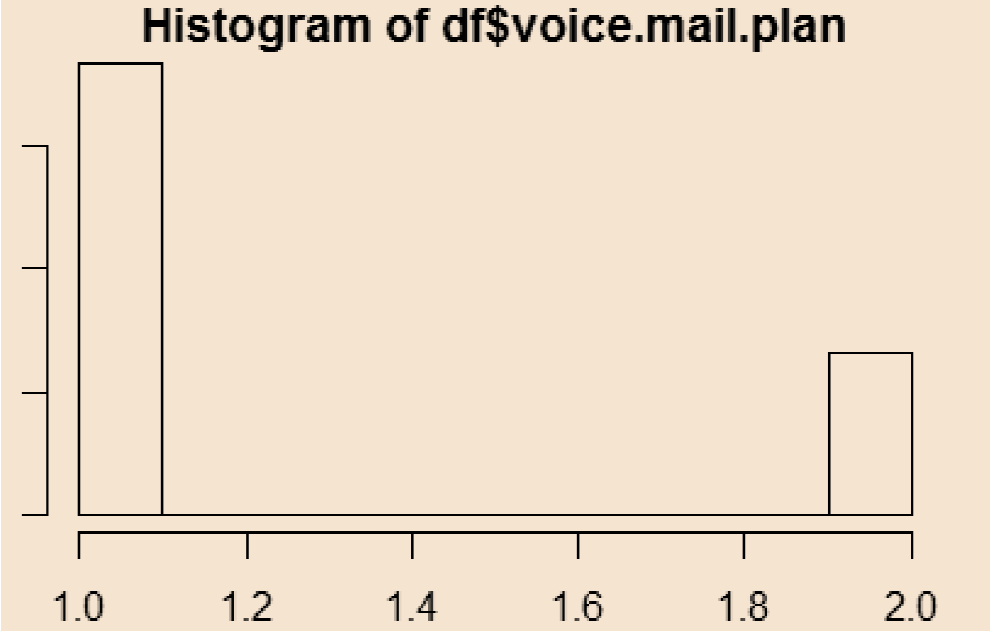
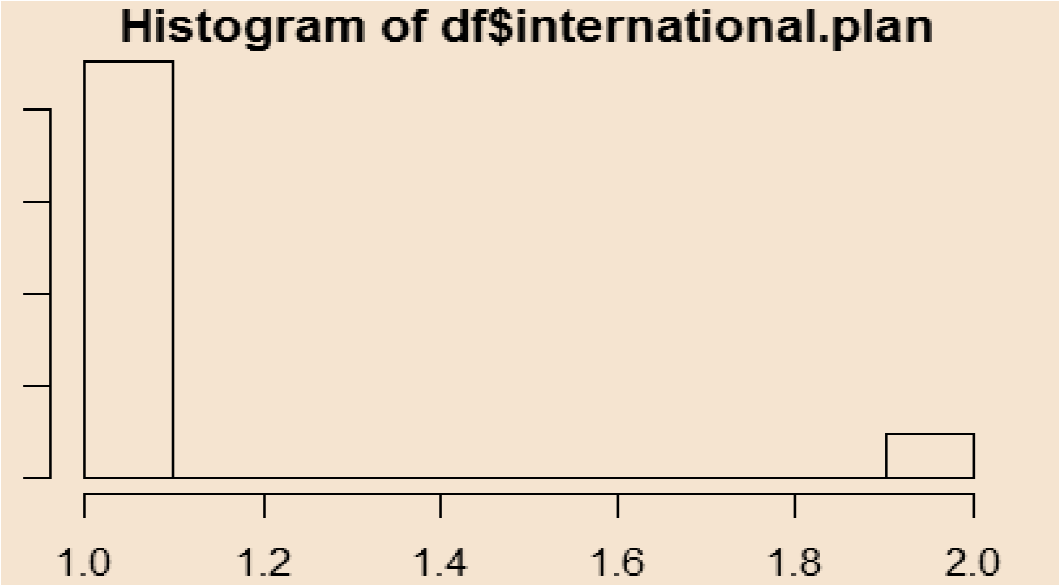


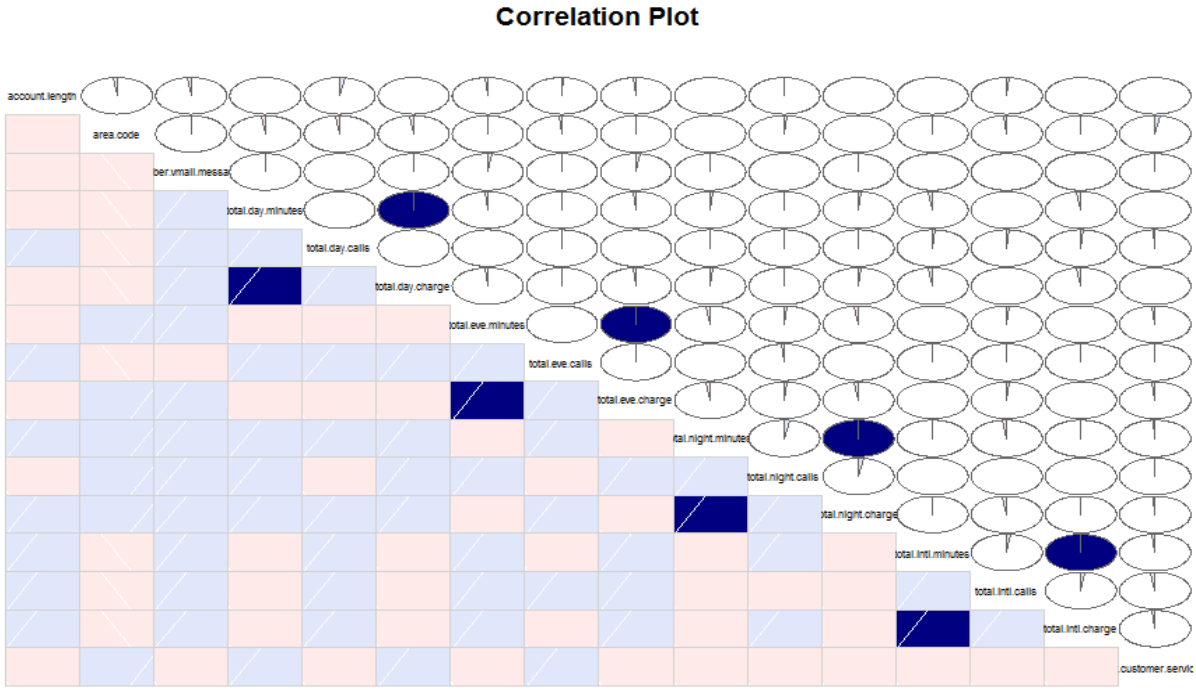
Fig 2.4 Box Plot Analysis (d)



Histogram for Voice mail plan



Histogram for International plan



Correlation Plot

Appendix B – R Code

Histogram:

```
multi.hist(numeric_data,main=NA,dcol=c("blue","red"),dlty=c("solid","solid"),bcol="pink")
h1=hist(df$Churn)
hist(df$voice mail plan)
hist(df$international plan)
```

Feature Selection:

```
#Feature Selection
impvar=randomForest(Churn~.,data=df,ntree=100,keep.forest=F,importance=T)
importance(impvar,type=1)

## based on the importance we remove area code and state from test and train dataset
df$area.code=NULL
df$state=NULL
test$account.length=NULL
test$state=NULL
train$state=NULL
train$account.length=NULL
```

Correlation Analysis:

```
# Correaltion analysis
corrgram(df[,cnames], order = F,
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")
```

Boxplot Analysis:

```
#box plot analysis

for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i]), x = "Churn"), data = subset(df))+
           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])))
}
```

```
gridExtra::grid.arrange(gn1,gn4,gn5,gn7,ncol=4)
gridExtra::grid.arrange(gn15,gn19,gn8,gn9,ncol=4)
gridExtra::grid.arrange(gn10,gn11,gn12,gn13,ncol=4)
gridExtra::grid.arrange(gn14,gn18,gn16,gn17,ncol=4)
```

Full RCode:

```
#clean the workspace
rm(list=ls())

#set working directory
setwd("E:/study/data/Project#2")
getwd()
#load libraries
x=c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50",
"dummies", "e1071", "Information",
    "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine',
'inTrees','knitr','tidyr','xtable',"varhandle",'MASS','psych')
library(DMwR)
lapply(x, require, character.only = TRUE)
rm(x)

#load datasets
train =read.csv("Train.csv")
test=read.csv("Test.csv")
df=rbind(train,test)

#exploring the data: checking the data type of the variables and if there are any missing
values
str(df)
table(is.na(df))

##changing the factor to catagorical factor
#for df
for(i in 1:ncol(df)){

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

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

  }
}
#For train
for(i in 1:ncol(train)){

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

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

```
  }
}
#For test
for(i in 1:ncol(test)){

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

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

  }
}

##################exploratory analaysis##############
df$phone.number=NULL
train$phone.number=NULL
test$phone.number=NULL
df$state=unfactor(df$state)
df$international.plan=unfactor(df$international.plan)
df$voice.mail.plan=unfactor(df$voice.mail.plan)

numeric_index = sapply(df,is.numeric)
numeric_data = df[,numeric_index]
cnames = colnames(numeric_data)
#box plot analysis

for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i]), x = "Churn"), data = subset(df))+
           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])))
}
gridExtra::grid.arrange(gn1,gn4,gn5,gn7,ncol=4)
gridExtra::grid.arrange(gn15,gn19,gn8,gn9,ncol=4)
gridExtra::grid.arrange(gn10,gn11,gn12,gn13,ncol=4)
gridExtra::grid.arrange(gn14,gn18,gn16,gn17,ncol=4)




#Histograms

multi.hist(numeric_data,main=NA,dcol=c("blue","red"),dlty=c("solid","solid"),bcol="pink")

# Correaltion analysis
corrgram(df[,cnames], order = F,
```

```
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")


#Feature Selection
impvar=randomForest(Churn~.,data=df,ntree=100,keep.forest=F,importance=T)
importance(impvar,type=1)

## based on the importance we remove area code and state from test and train dataset
df$area.code=NULL
df$state=NULL
test$account.length=NULL
test$state=NULL
train$state=NULL
train$account.length=NULL

######

df$Churn=unfactor(df$Churn)
h1=hist(df$Churn)
#model development
#Decision tree for classification
#Develop Model on training data
C50_model = C5.0(Churn ~., train, trials = 100, rules = TRUE)

#Summary of DT model
summary(C50_model)

#write rules into disk
write(capture.output(summary(C50_model)), "c50Rules.txt")

#Lets predict for test cases

C50_Predictions = predict(C50_model, test, type = "class")

##Evaluate the performance of classification model
ConfMatrix_C50 = table(test$Churn, C50_Predictions)
confusionMatrix(ConfMatrix_C50)

#False Negative rate
FNR = FN/FN+TP
#accuracy
#FNR
## Random Forest method
RF_model = randomForest(Churn ~ ., train, importance = TRUE, ntree = 500)

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

```
exec = extractRules(treeList, train[-18])  # R-executable conditions
#
# #Visualize some rules
exec[1:2,]
#
# #Make rules more readable:
readableRules = presentRules(exec, colnames(train))
readableRules[1:2,]
#
# #Get rule metrics
ruleMetric = getRuleMetric(exec, train[,-18], train$Churn)  # get rule metrics
#
# #evaulate few rules
ruleMetric[1:2,]

#Presdict test data using random forest model
RF_Predictions = predict(RF_model, test[,-18],)

##Evaluate the performance of classification model
ConfMatrix_RF = table(test$Churn, RF_Predictions)
confusionMatrix(ConfMatrix_RF)
#Accuracy :96%
#FNR-26%
#False Negative rate
FNR = FN/FN+TP

#naive Bayes
library(e1071)

#Develop model
NB_model = naiveBayes(Churn ~ ., data = train)

#predict on test cases #raw
NB_Predictions = predict(NB_model, test[,1:17], type = 'class')

#Look at confusion matrix
Conf_matrix = table(observed = test[,18], predicted = NB_Predictions)
confusionMatrix(Conf_matrix)

#Accuracy: 89.32%
#FNR: 57.58


##****************************************************************************
**********************************
# #create NA on  outliners and compute them with mean, or median whichever gives better
outout

for(i in cnames){
  val = df[,i][df[,i] %in% boxplot.stats(df[,i])$out]
```

```
  print(length(val))
  df[,i][df[,i] %in% val] = NA
}

#
df$number.customer.service.calls [is.na(df$number.customer.service.calls )] =
mean(df$number.customer.service.calls, na.rm = T)
df$international.plan [is.na(df$international.plan )] = mean(df$international.plan, na.rm = T)
df$number.vmail.messages [is.na(df$number.vmail.messages )] =
mean(df$number.vmail.messages, na.rm = T)
df$total.day.minutes [is.na(df$total.day.minutes )] = mean(df$total.day.minutes, na.rm = T)
df$total.day.calls [is.na(df$total.day.calls )] = mean(df$total.day.calls, na.rm = T)
df$total.day.charge [is.na(df$total.day.charge )] = mean(df$total.day.charge, na.rm = T)
df$total.eve.minutes [is.na(df$total.eve.minutes )] = mean(df$total.eve.minutes, na.rm = T)
df$total.eve.calls [is.na(df$total.eve.calls )] = mean(df$total.eve.calls, na.rm = T)
df$total.eve.charge [is.na(df$total.eve.charge )] = mean(df$total.eve.charge, na.rm = T)
df$total.night.minutes [is.na(df$total.night.minutes )] = mean(df$total.night.minutes, na.rm =
T)
df$total.night.calls [is.na(df$total.night.calls )] = mean(df$total.night.calls, na.rm = T)
df$total.night.charge [is.na(df$total.night.charge )] = mean(df$total.night.charge, na.rm = T)
df$total.intl.minutes [is.na(df$total.intl.minutes )] = mean(df$total.intl.minutes, na.rm = T)
df$total.intl.calls [is.na(df$total.intl.calls )] = mean(df$total.intl.calls, na.rm = T)
df$total.intl.charge [is.na(df$total.intl.charge )] = mean(df$total.intl.charge, na.rm = T)
df$account.length [is.na(df$account.length  )] = mean(df$account.length , na.rm = T)
table(is.na(df))
## Random Forest method
train.index = createDataPartition(df$Churn, p = .80, list = FALSE)
train = df[ train.index,]
test  = df[-train.index,]

RF_model = randomForest(Churn ~ ., train, importance = TRUE, ntree = 500)

#Extract rules fromn random forest
#transform rf object to an inTrees' format
treeList = RF2List(RF_model)
#
# #Extract rules
exec = extractRules(treeList, train[-18])  # R-executable conditions
#
# #Visualize some rules
exec[1:2,]
#
# #Make rules more readable:
readableRules = presentRules(exec, colnames(train))
readableRules[1:2,]
#
# #Get rule metrics
ruleMetric = getRuleMetric(exec, train[,-18], train$Churn)  # get rule metrics
#
# #evaulate few rules
```

```
ruleMetric[1:2,]

#Presdict test data using random forest model
RF_Predictions = predict(RF_model, test[,-18],)

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