Churn Reduction

Radhika Haresh Luvani

29/11/2018

Contents

1. Introduction ……………………………………………………………………………………………3
1.1. Problem Statement…………………………………………………………………………….3
1.2. Data……………………………………………………………………………………………3

2. Methodology……………………………………………………………………………………………5
2.1. Pre-processing……………………………………………………………………………………5
2.1.1. Missing Value Analysis……………………………………………………………………7
2.1.2. Outlier Analysis……………………………………………………………………………..7
2.1.3. Feature Selection…………………………………………………………………………10
2.2. Modeling…………………………………………………………………………………………12
2.2.1. Model Selection…………………………………………………………………………...12
2.2.2. Logistic Regression………………………………………………………………………13
2.2.3. Random Forest……………………………………………………………………………...15
3. **Conclusion…………………………………………………………………………………………….16**
3.1. Model Evaluation………………………………………………………………………………16
3.1.1. Accuracy and FNR…………………………………………………………………………..17
3.2. Model selection…………………………………………………………………………….…17

Appendix A - R code……………………………………………………………………………………18

References……………………………………………………………………………………………….20

Chapter 1

Introduction

1.1 Problem Statement

Churn (loss of customers to competition) is a problem for companies because it is more expensive to acquire a new customer than to keep your existing one from leaving. This problem statement is targeted at enabling churn reduction using analytics concepts.

1.2 Data

Our task is to build churn models which will predict the loss of customer depending on multiple predictors. We have the same data set as train and test data so we can combine the data. Given below is a sample of the data set that we are using to predict the churn score:

Table 1.2.1: Sample Data (Columns: 1-9)

account length	**international plan**	**voice mail plan**	**number vmail messages**	**total day minutes**	**total day calls**	**total day charge**	**total eve minutes**	**total eve calls**
128	no	yes	25	265.1	110	45.07	197.4	99
107	no	yes	26	161.6	123	27.47	195.5	103
137	no	no	0	243.4	114	41.38	121.2	110
84	yes	no	0	299.4	71	50.9	61.9	88
75	yes	no	0	166.7	113	28.34	148.3	122

Table 1.2.2: Sample Data (Columns: 10-18)

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**
16.78	244.7	91	11.1	10	3	2.7	1	False.
16.62	254.4	103	11.45	13.7	3	3.7	1	False.
10.3	162.6	104	8.86	12.2	5	3.29	0	False.
5.26	196.9	89	8.41	6.6	7	1.78	2	False.
12.61	186.9	121	9.18	10.1	3	2.73	3	False.

As you can see in the table below we have the following 16 variables, using which we have to correctly predict the customer behavior:

Table 1.2.3: Predictor Variables

S.No.	**Predictor**
1	Account Length
2	International Plan
3	Voicemail Plan
4	Number Of Voicemail Messages
5	Total Day Minutes Used
6	Total Day Calls Made
7	Total Day Charge
8	Total Evening Minutes
9	Total Evening Calls
10	Total Evening Charge
11	Total Night Minutes
12	Total Night Calls
13	Total Night Charge
14	Total International Minutes Used
15	Total International Calls Made
16	Total International Charge

Chapter 2

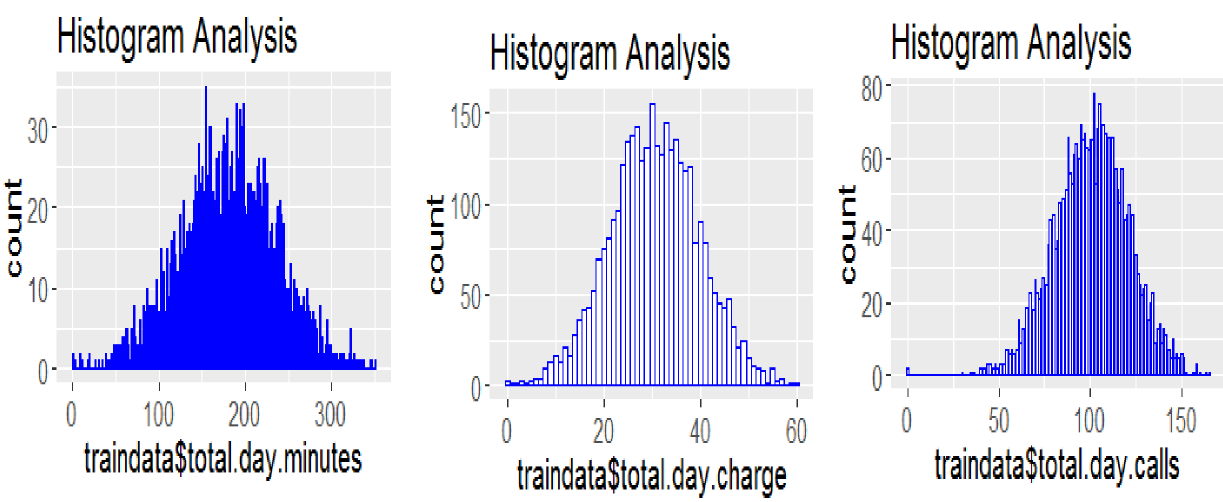
Methodology

2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms looking *at* data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. To start this process we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability count functions of the variable.

In the fig (2.1) it's showing an analysis of the individual predictors and its count with the help of Histogram Analysis.

Fir 2.1 Predictors histogram analysis



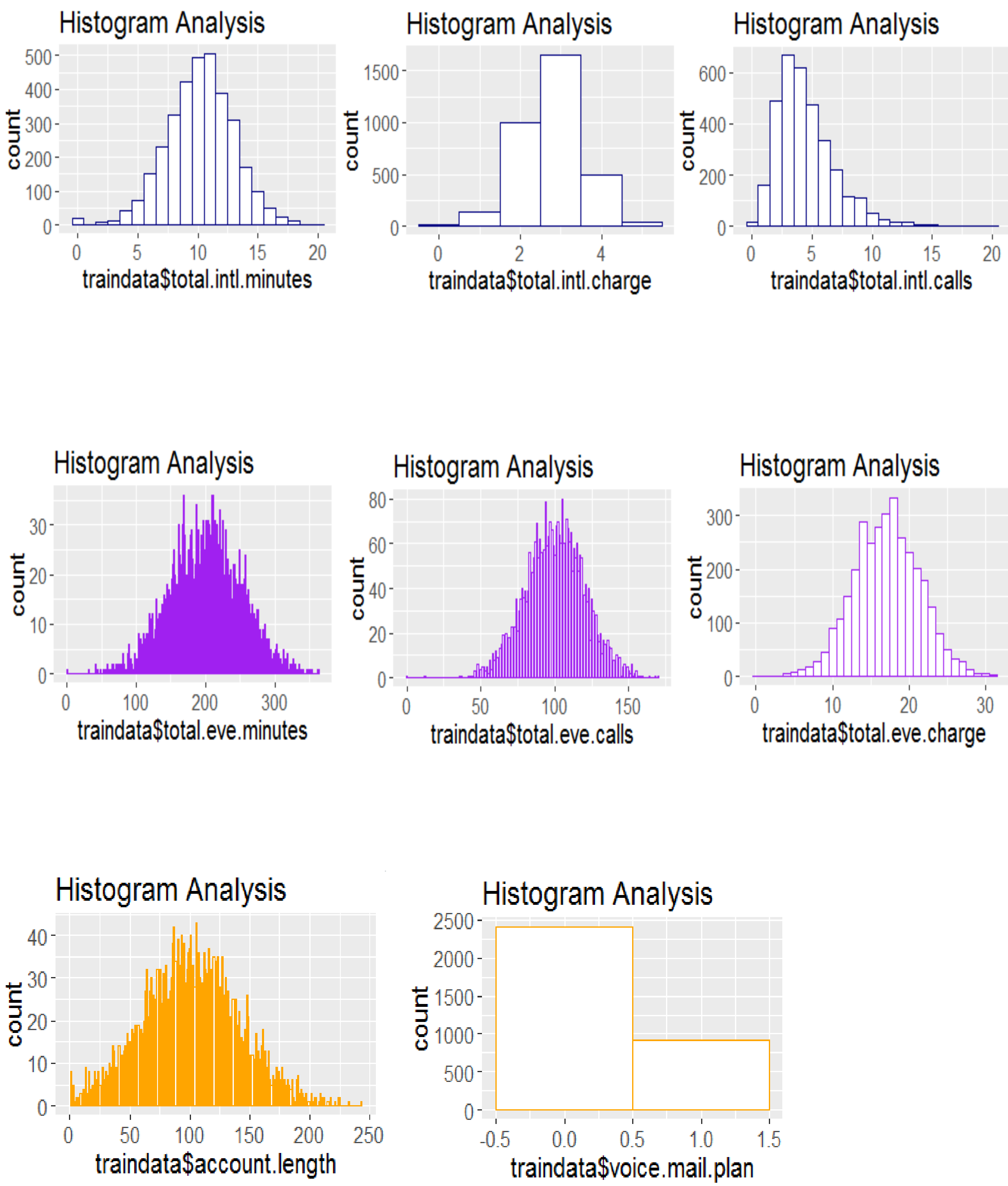

Histogram Analysis
count
traindata$total.intl.minutes
Histogram Analysis
count
traindata$total.intl.charge
Histogram Analysis
count
traindata$total.intl.calls
Histogram Analysis
count
traindata$total.eve.minutes
Histogram Analysis
count
traindata$total.eve.calls
Histogram Analysis
count
traindata$total.eve.charge
Histogram Analysis
count
traindata$account.length
Histogram Analysis
count
traindata$voice.mail.plan

Fig 2.1

2.1.1 Missing Value Analysis

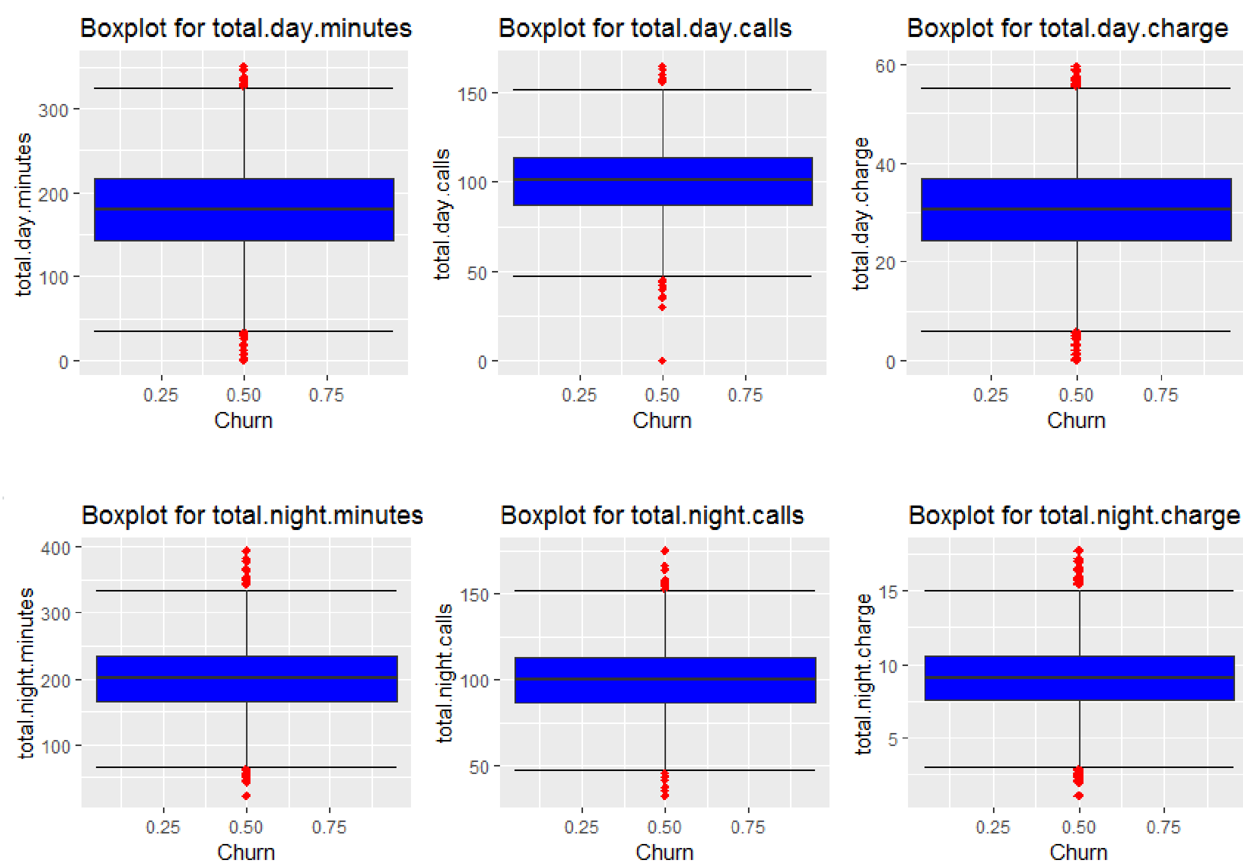
A missing value can signify a number of different things in your data. Data mining methods vary in the way they treat missing values. Typically, they ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values.

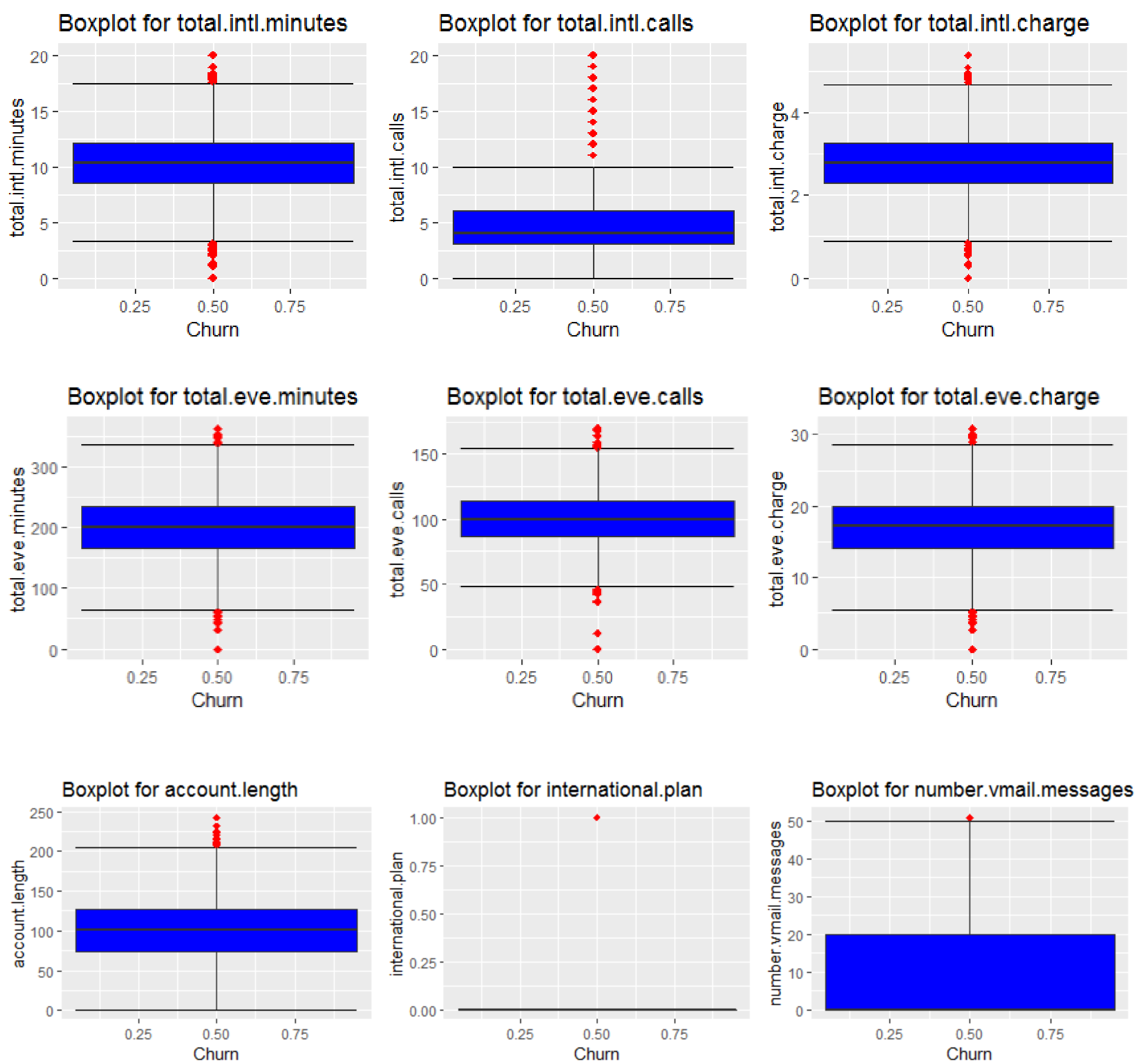
2.1.2 Outlier Analysis

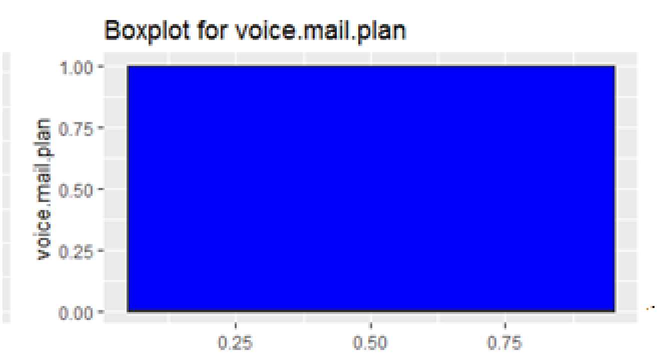
An outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.

Observations inconsistent with rest of the dataset Global Outlier. Fig (2.1.2.1) will show the effect of outliers in each predictor, with the help of boxplot. In figure 2.1.2.1 we have plotted the boxplots of the 16 predictor variables with respect to churn . A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set.

Figure 2.1.2.1: Outliers in each predictor

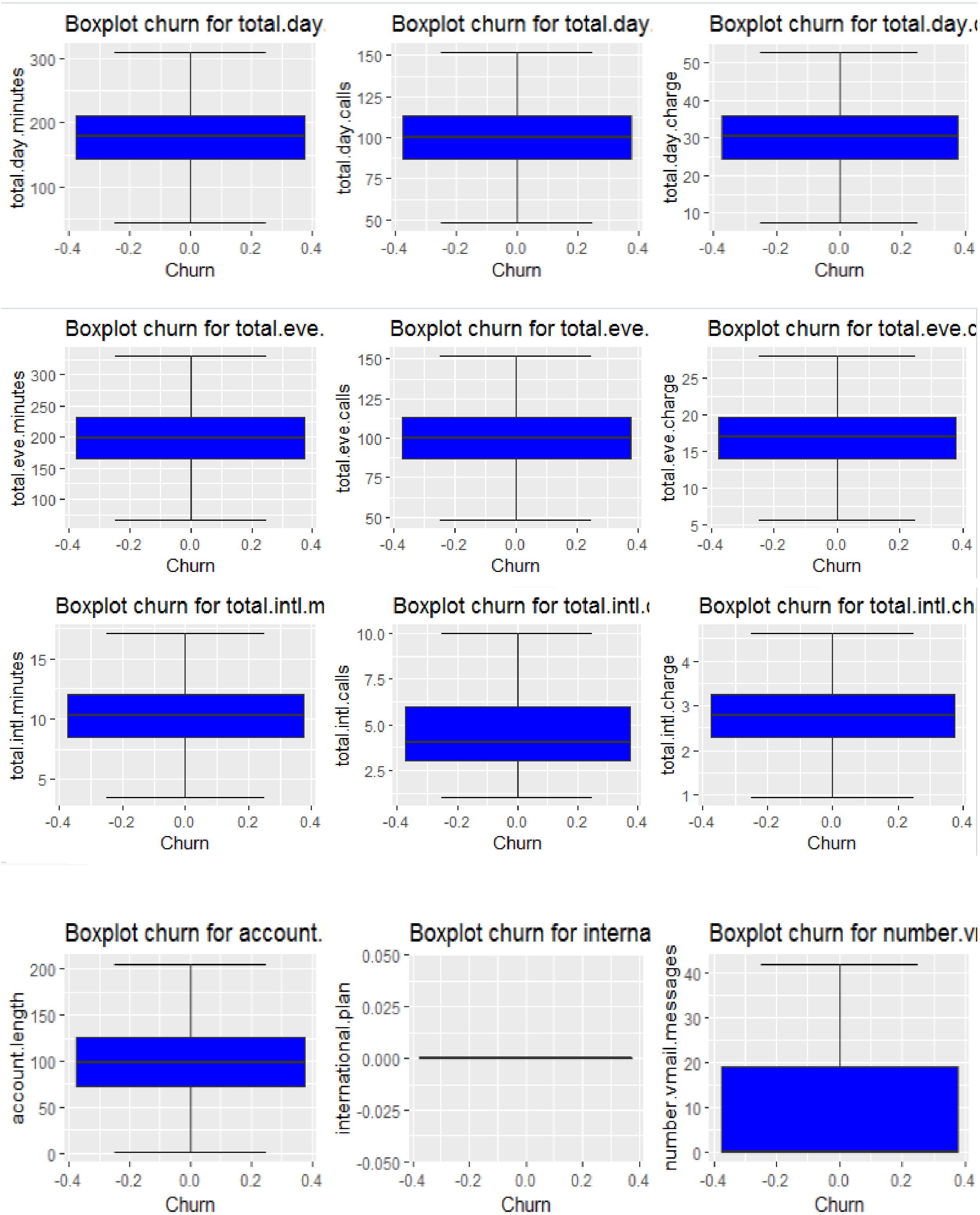



Boxplot for total.intl.minutes
total.intl.minutes
Churn
Boxplot for total.intl.calls
total.intl.calls
Churn
Boxplot for total.intl.charge
total.intl.charge
Churn
Boxplot for total.eve.minutes
total.eve.minutes
Churn
Boxplot for total.eve.calls
total.eve.calls
Churn
Boxplot for total.eve.charge
total.eve.charge
Churn
Boxplot for account.length
account.length
Churn
Boxplot for international.plan
international.plan
Churn
Boxplot for number.vmail.messages
number.vmail.messages
Churn


Boxplot for voice.mail.plan
voice.mail.plan

Effect of after removing the outliers form the given data in each predictor.

Fig 2.1.2.2 removed outliers from predictor



2.1.3 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Below we have used *Random Forests* to perform features selection.

Data Preparation and Feature Selection In this study used publically available dataset. On other hand, the feature selection is an important step in knowledge discovery process, to identify those relevant variables or attributes from the large number of attributes in a dataset which are too relevant and reduce the computational cost [2]. To make sure that only relevant features are included into decision table that also reduces the computational Cost and address to P1 (Which features are more indicative for churn prediction in telecom sector?),
The selection of most appropriate attributes from the dataset.

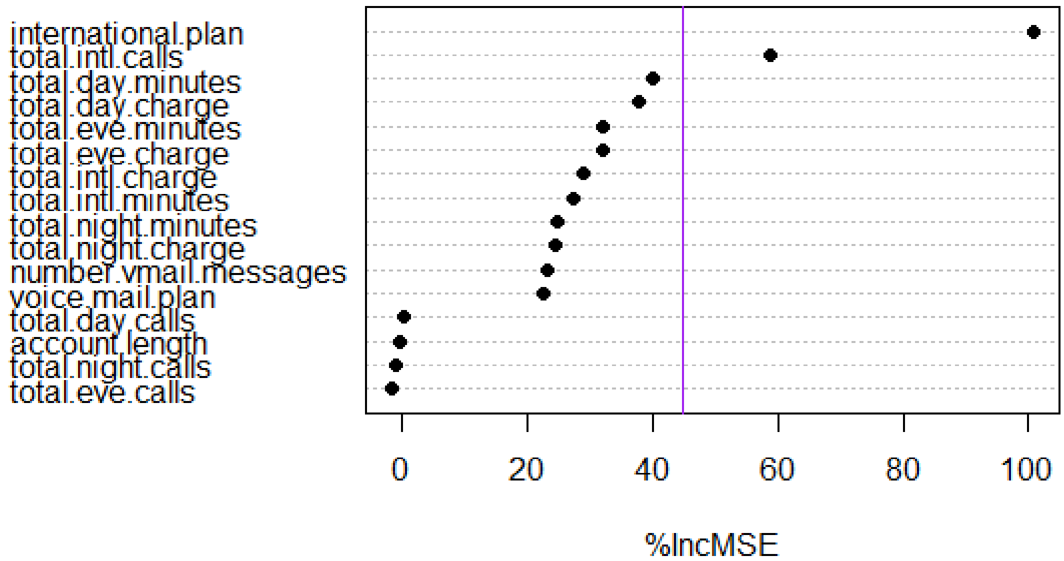
```
RF_model = randomForest(Churn ~ . ,data= train, importance = TRUE, ntree = 500 , ntry = 500)
print(RF_model)
importance(RF_model)
```

	%IncMSE	IncNodePurity
account.length	-1.4196768	16.80921
international.plan	97.0872527	31.44387
voice.mail.plan	22.2509904	9.53794
number.vmail.messages	22.0646337	14.07548
total.day.minutes	36.8822105	48.87765
total.day.calls	-1.1170274	15.55453
total.day.charge	37.1850610	51.46790
total.eve.minutes	32.6036260	30.96953
total.eve.calls	-1.8075459	14.88887
total.eve.charge	32.1063611	31.03664
total.night.minutes	26.1524133	22.17340
total.night.calls	-0.4019971	16.24340
total.night.charge	25.1948378	21.03689
total.intl.minutes	29.6568238	21.41301
total.intl.calls	57.2737471	28.64616
total.intl.charge	27.7016175	20.74130

Fig 2.1.3.1 Feature Selection

1 Churn Prediction in Telecommunication Industry Using Rough Set Approach
Adnan Amin1, Changez Khan1, Saeed Shehzad2, Imtiaz Ali1, Sajid Anwar1,* 2016.

Graphical representation of Feature selection and how this will help us to know the predictor of churning the customer with the help of random forest model.



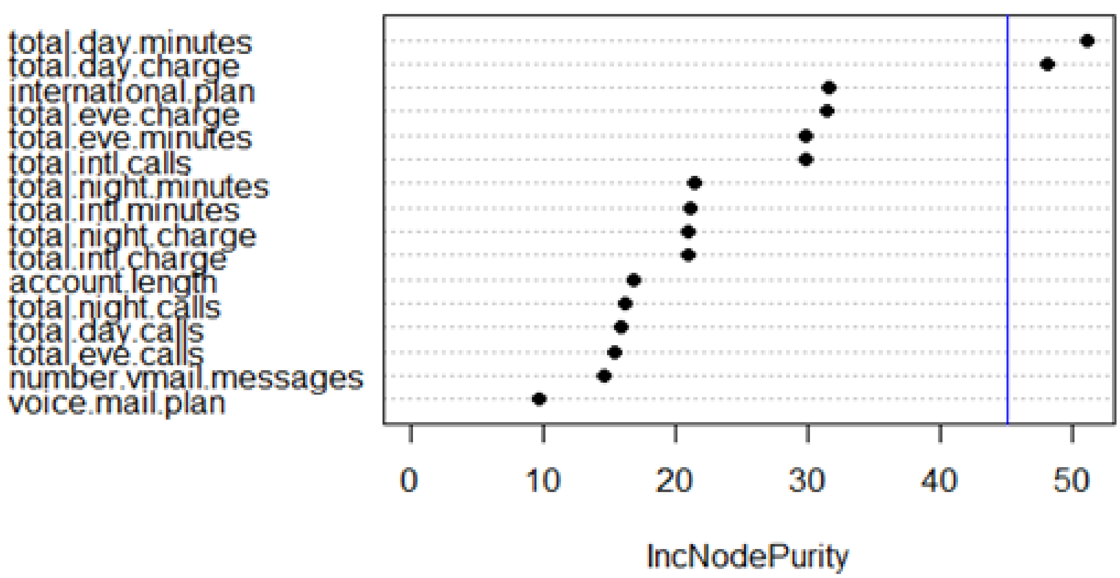


Fig 2.1.2.2 graphical representation of Feature scal

2.2 Modeling

2.2.1 Model Selection

In our early stages of analysis during pre-processing we have come to understand that data behaves the same way and we can combine the data and use because it behaves the same way. Generate the models for the given data.
The dependent variable can fall in either of the four categories:

1. Nominal
2. Binomial
3. Ordinal
4. Multinomal

Here, we have independent variable as factor which means regression techniques will be suitable for the given data. Churn has and two categories which means we going to use binomial functions to analyses. Logistic regression and Random forest will be an appropriate algorithm to test the data.

1. Exhaustive Algorithm
2. Genetic Algorithm
3. Covering Algorithm
4. RSES LEM2 Algorithm

- Exhaustive Algorithm: It takes subsets of features incrementally and then returns the deducts of required One. It needs more concentration because it may lead to extensive computations in case of complex and large decision table. It is based on Boolean reasoning approach.

- Genetic Algorithm: It is based on order-based GA coupled with heuristic and this evolutionary method is presented by it is used to reduce the computational cost in large and complex decision table.[2]

- Covering Algorithm: it is customized implementation of the LEM2 idea and implemented in RSES Covering method. It was introduced by Jerzy Grzymala .

- RSES LEM2 Algorithm: it is a separate-&-conquer technique paired with lower and upper approximation Of rough set theory and it is based on local covering determination of each object from the decision class, It is implementation of LEM2.[1]

---

1 Churn Prediction in Telecommunication Industry Using Rough Set Approach
Adnan Amin1, Changez Khan1, Saeed Shehzad2, Imtiaz Ali1, Sajid Anwar1,* 2016.
2 Bazan J., Nguyen H.S., Nguyen S.H., Synak P., Wróblewski J. Rough Set Algorithms in Classification Problem. Physica-Verlag, Heidelberg, New York, (2000)

2.2.2 Logistic Regression

R-code

```
#Logistic Regression
logit_model = glm(Churn ~ ., data = train, family = "binomial")

#summary of the model
summary(logit_model)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5596  -0.5466  -0.4016  -0.2565   2.9717

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -6.591e+00  6.560e-01 -10.047  < 2e-16 ***
account.length         2.908e-03  1.314e-03   2.213 0.026888 *
international.plan     1.800e+00  1.367e-01  13.165  < 2e-16 ***
voice.mail.plan       -2.138e+00  5.631e-01  -3.798 0.000146 ***
number.vmail.messages  3.659e-02  1.751e-02   2.089 0.036688 *
total.day.minutes      4.541e+00  3.100e+00   1.465 0.142943
total.day.calls        6.647e-04  2.619e-03   0.254 0.799625
total.day.charge      -2.665e+01  1.824e+01  -1.461 0.143970
total.eve.minutes      6.074e-01  1.559e+00   0.390 0.696781
total.eve.calls       -1.234e-03  2.687e-03  -0.459 0.646032
total.eve.charge      -7.067e+00  1.834e+01  -0.385 0.699957
total.night.minutes   -4.868e-01  8.302e-01  -0.586 0.557611
total.night.calls     -3.480e-03  2.639e-03  -1.319 0.187272
total.night.charge     1.090e+01  1.845e+01   0.591 0.554577
total.intl.minutes     1.281e+00  4.963e+00   0.258 0.796286
total.intl.calls      -7.436e-02  2.315e-02  -3.213 0.001315 **
total.intl.charge     -4.462e+00  1.838e+01  -0.243 0.808251
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2856.8  on 3524  degrees of freedom
Residual deviance: 2424.9  on 3508  degrees of freedom
AIC: 2458.9

Number of Fisher Scoring iterations: 5
```

Therefore, this is the maximum accuracy that we can get from this model. International plan , voice mail plan and total international are the significant as per the model.

Python code-

```
#Built Logistic Regression
import statsmodels.api as sm

logit = sm.Logit(train['churn'], train[train_cols]).fit()

logit.summary()
```

Logit Regression Results

Dep. Variable:	churn	**No. Observations:**	4020
Model:	Logit	**Df Residuals:**	4002
Method:	MLE	**Df Model:**	17
Date:	Sat, 01 Dec 2018	**Pseudo R-squ.:**	0.2261
Time:	19:22:16	**Log-Likelihood:**	-1270.0
converged:	True	**LL-Null:**	-1641.0
		LLR p-value:	1.042e-146

	coef	std err	z	P>|z|	[0.025	0.975]
account_length	-0.0009	0.001	-0.667	0.505	-0.003	0.002
number_vmail_messages	-0.0430	0.017	-2.499	0.012	-0.077	-0.009
total_day_minutes	-4.9572	3.022	-1.640	0.101	-10.881	0.967
total_day_calls	-0.0017	0.003	-0.681	0.496	-0.007	0.003
total_day_charge	29.0753	17.779	1.635	0.102	-5.771	63.922
total_eve_minutes	-0.2177	1.516	-0.144	0.886	-3.189	2.753
total_eve_calls	0.0026	0.003	0.984	0.325	-0.003	0.008
total_eve_charge	2.4860	17.834	0.139	0.889	-32.467	37.439
total_night_minutes	-0.1282	0.812	-0.158	0.874	-1.719	1.462
total_night_calls	0.0008	0.003	0.303	0.762	-0.004	0.006
total_night_charge	2.7602	18.033	0.153	0.878	-32.585	38.105
total_intl_minutes	3.5613	4.878	0.730	0.465	-6.000	13.123
total_intl_calls	0.0656	0.022	2.921	0.003	0.022	0.110
total_intl_charge	-13.4983	18.068	-0.747	0.455	-48.912	21.915
number_customer_service_calls	-0.5181	0.037	-14.050	0.000	-0.590	-0.446
international_plan_0	5.2664	nan	nan	nan	nan	nan
international_plan_1	3.1277	nan	nan	nan	nan	nan
voice_mail_plan_0	2.9670	nan	nan	nan	nan	nan
voice_mail_plan_1	5.4272	nan	nan	nan	nan	nan

2.2.2 Random Forest

```
                 Length Class  Mode
call               6   -none- call
type               1   -none- character
predicted       3525   -none- numeric
mse              500   -none- numeric
rsq              500   -none- numeric
oob.times       3525   -none- numeric
importance        32   -none- numeric
importanceSD      16   -none- numeric
localImportance    0   -none- NULL
proximity          0   -none- NULL
ntree              1   -none- numeric
mtry               1   -none- numeric
forest            11   -none- list
coefs              0   -none- NULL
y               3525   -none- numeric
test               0   -none- NULL
inbag              0   -none- NULL
terms              3   terms  call
```

Rules –

```
> exec[1:10,]
 [1] "X[,2]<=0.5 & X[,3]<=0.5 & X[,8]<=208.75 & X[,11]<=61.55"
 [2] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=45.245 & X[,8]<=208.75 & X[,11]>61.55 & X[,14]<=18.25"
 [3] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=45.245 & X[,8]<=208.75 & X[,11]>61.55 & X[,14]>18.25"
 [4] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]>45.245 & X[,8]<=208.75 & X[,9]<=108.5 & X[,11]>61.55"
 [5] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]>45.245 & X[,8]<=208.75 & X[,9]>108.5 & X[,11]>61.55"
 [6] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=41.965 & X[,8]>208.75 & X[,11]<=321.75 & X[,14]<=13.05"
 [7] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=41.965 & X[,8]>208.75 & X[,11]>321.75 & X[,14]<=13.05"
 [8] "X[,2]<=0.5 & X[,3]<=0.5 & X[,6]<=73.5 & X[,7]>41.965 & X[,8]>208.75 & X[,14]<=13.05"
 [9] "X[,2]<=0.5 & X[,3]<=0.5 & X[,6]>73.5 & X[,7]>41.965 & X[,8]>208.75 & X[,14]<=13.05"
[10] "X[,2]<=0.5 & X[,3]<=0.5 & X[,5]<=222.3 & X[,8]>208.75 & X[,8]<=209.25 & X[,14]>13.05"
```

Rule Metric

```
      len freq    err
 [1,] "4" "0.001" "0.25"
 [2,] "6" "0.365" "0.0637466773410989"
 [3,] "6" "0.001" "0.222222222222222"
 [4,] "6" "0.013" "0.230873698506111"
 [5,] "6" "0.008" "0.229591836734694"
 [6,] "6" "0.219" "0.095043915207733"
 [7,] "6" "0.001" "0.25"
 [8,] "6" "0.002" "0.234375"
 [9,] "6" "0.021" "0.108087821354851"
[10,] "6" "0.001" "0.25"
      condition
 [1,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,8]<=208.75 & X[,11]<=61.55"
 [2,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=45.245 & X[,8]<=208.75 & X[,11]>61.55 & X[,14]<=18.25"
 [3,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=45.245 & X[,8]<=208.75 & X[,11]>61.55 & X[,14]>18.25"
 [4,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]>45.245 & X[,8]<=208.75 & X[,9]<=108.5 & X[,11]>61.55"
 [5,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]>45.245 & X[,8]<=208.75 & X[,9]>108.5 & X[,11]>61.55"
 [6,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=41.965 & X[,8]>208.75 & X[,11]<=321.75 & X[,14]<=13.05"
 [7,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,7]<=41.965 & X[,8]>208.75 & X[,11]>321.75 & X[,14]<=13.05"
 [8,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,6]<=73.5 & X[,7]>41.965 & X[,8]>208.75 & X[,14]<=13.05"
 [9,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,6]>73.5 & X[,7]>41.965 & X[,8]>208.75 & X[,14]<=13.05"
[10,] "X[,2]<=0.5 & X[,3]<=0.5 & X[,5]<=222.3 & X[,8]>208.75 & X[,8]<=209.25 & X[,14]>13.05"
      pred
 [1,] "0.5"
 [2,] "0.0684292379471229"
 [3,] "0.333333333333333"
 [4,] "0.638297872340426"
 [5,] "0.357142857142857"
 [6,] "0.106355382619974"
 [7,] "0.5"
 [8,] "0.625"
 [9,] "0.876712328767123"
[10,] "0.5"
```

Chapter 3

Conclusion

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of Churn Data, the latter two, *Interpretability* and *Computation Efficiency*, do hold much significance. Therefore we will use *Predictive performance* as the criteria to compare and evaluate models. Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some Accuracy and False negative Rate (FNR).

3.1.1 Accuracy and FNR

Accuracy and FNR is one of the Performance measure used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.
R code-

```
#Accuracy: 86.23
#FNR:  87.32

#Random forest
#Accuracy = 96.28
#FNR = 26.33
```

Python code

```
#check accuracy of model
 #accuracy_score(y_test, y_pred)*100
#((TP+TN)*100)/(TP+TN+FP+FN)

(FN*100)/(FN+TP)
#Accuracy: 85.85
#FNR: 3.419811
```

```
#False Negative rate
#(FN*100)/(FN+TP)

#Accuracy: 96.2
#FNR: 19.8529
```

3.2 Model Selection

We can see that both models have different performance comparatively and therefore we can select one of the two models with analysis of the information. On the basis of the curve we can see Random Forest model works better on the given data. Sometimes model works differently with different languages as per R code Random forest works the best but with Python Logistic works the best because we are interested in FNR and its comparatively less then Random forest.

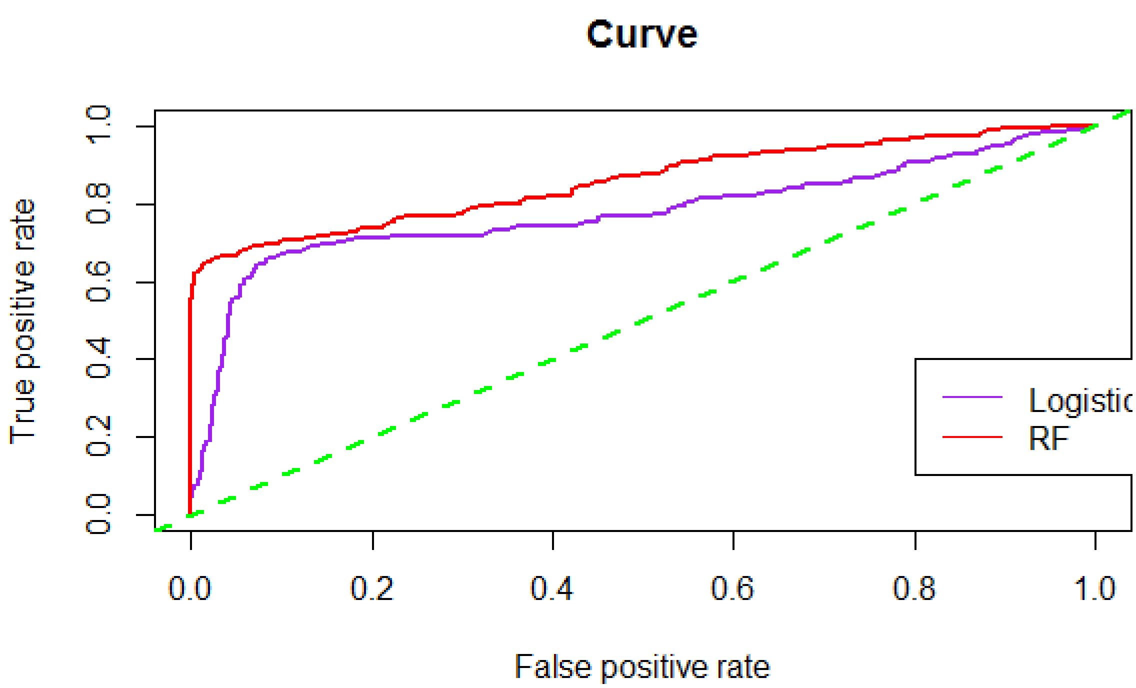


Fig – 3.2 Curve for model selection

Appendix A – R code

Histogram –

```
ggplot(data , aes(x = data$total.day.minutes ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "blue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.day.charge ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "blue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.day.calls ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "blue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.eve.minutes ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "purple")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.eve.calls ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "purple")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.eve.charge ))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "purple")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.intl.minutes))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "navyblue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.intl.charge))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "navyblue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$total.intl.calls))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "navyblue")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$account.length))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "orange")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))

ggplot(data , aes(x = data$voice.mail.plan))+
  geom_histogram(binwidth = 1 , fill = "white" ,  colour = "orange")+
  ggtitle("Histogram Analysis") +  theme(text=element_text(size=15))
```

Box plot for outlier check

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

numeric_data = data[,numeric_index]

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 = "blue" ,outlier.shape=18,
                         outlier.size=0.5, notch=FALSE) +
            theme(legend.position="bottom")+
            labs(y=cnames[i],x="Churn")+
            ggtitle(paste("Boxplot churn for",cnames[i])))
 }

# ## Plotting plots together
gridExtra::grid.arrange(gn1,gn2, gn4, ncol=3 )
gridExtra::grid.arrange(gn5, gn6,gn7,ncol=3)
gridExtra::grid.arrange(gn8,gn9,gn10 ,ncol=3)
gridExtra::grid.arrange(gn11,gn12, gn13 ,ncol=3)
gridExtra::grid.arrange(gn14,gn15,gn16 , ncol=3)
```

References

Churn Prediction in Telecommunication Industry Using Rough Set Approach
Adnan Amin1, Changez Khan1, Saeed Shehzad2, Imtiaz Ali1, Sajid Anwar1,* 2016.

Bazan J., Nguyen H.S., Nguyen S.H., Synak P., Wróblewski J. Rough Set Algorithms in Classification Problem. Physica-Verlag, Heidelberg, New York, (2000)