Predicting Churn out Model

*Pranjul Gupta*

*25 January 2018*

**Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [**1**](#page3) | [**Introduction**](#page3) | | | **3** |
|  | [1.1](#page3) | [Problem Statement](#page3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 3 |
|  | [1.2](#page3) | [Data](#page3). | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| [**2**](#page5) | [**Methodology**](#page5) | | | **6** |
|  | [2.1](#page5) | [Pre Processing](#page5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . | | 6 |
|  |  | [2.1.1](#page5) | [Outlier Analysis](#page5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
|  |  | [2.1.2](#page11) | [Feature Selection](#page11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 15 |
|  | [2.2](#page13) | [Modeling](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 12 |
|  |  | [2.2.1](#page13) | [Model Selection](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
|  |  | [2.2.2](#page13) | [Random Forest](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
|  |  | [2.2.3](#page15) | [Regression Trees](#page15) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
|  |  | [2.2.4](#page15) | [Classification](#page15) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| [**3**](#page16) | [**Conclusion**](#page16) | |  | **19** |
|  | [3.1](#page16) | [Model Evaluation](#page16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 19 |
|  |  | [3.1.1](#page16) | [Accuracy](#page16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  |  | [3.1.2](#page16) | [True positive rate](#page16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [3.2](#page17) | [Model Selection](#page17) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 21 |
| [**Appendix A - Extra Figures**](#page18) | | | | **22** |
| [**Appendix B - R Code**](#page24) **and Python** | | | | **26** |
|  |  | [Predictor variable Histograms (Fig:](#page24) [2.1, 2.2](#page6) ). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 26 |
|  |  | [Predictor variable Histograms with density (Fig:](#page24) [2.3)](#page7) . . . . . . . . . . . . . . . . . . . . . . . . . . | | 26 |
|  |  | [Predictors boxplot with churn (Fig:](#page24) [2.4, 2.5, 2.6)](#page19) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
|  |  | [Effect of outliers (Fig:](#page24) [4.1, 4.2, 4.3)](#page8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
|  |  | [Predictor boxplots (Fig:](#page25) [4.4)](#page9) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
|  |  | Complete R File. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . .. . . . . . . . . .. . . . | | 27 |
|  |  | Complete Python File. . . . . . . . . .. . . . . . . . . .. . . . . . . . . . . . . .. . . . .. . . . .. . . . . . | | 29 |
| [**References**](#page30) | | |  | **32** |

**Chapter 1**

**Introduction**

* 1. **Problem Statement**

Predicting the churning out rate of customer for Telecom Company according to user’s behavior dependent on their usage of calls, minutes and other attributes. Churn out meaning is the no. of customer we are losing each year. The aim of the model is to predict customer churn out, and reducing the cost to get a new customer, in fact getting a new customer is more expensive comparing to keeping the old one. With analytics concept we can predict for customer, which have probability to leave the company. We can give those customer extra support and attention.

* 1. **Data**

Our task is to build classification models which will classify the churn out of customer depending on multiple call usage factors. Given below is a sample of the data set that we are using to predict the churning out of the customer:

**Table 1.1**: Churning Out Sample Data (Columns: 1-8)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **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 | |

**Table 1.2**: Churning Out Sample Data (Columns: 9-16)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **total day minutes** | **total day calls** | **total day charge** | **total eve minutes** | **total eve calls** | **total eve charge** | **total night minutes** | **total night calls** | | 265.1 | 110 | 45.07 | 197.4 | 99 | 16.78 | 244.7 | 91 | | 161.6 | 123 | 27.47 | 195.5 | 103 | 16.62 | 254.4 | 103 | | 243.4 | 114 | 41.38 | 121.2 | 110 | 10.3 | 162.6 | 104 | | 299.4 | 71 | 50.9 | 61.9 | 88 | 5.26 | 196.9 | 89 | | 166.7 | 113 | 28.34 | 148.3 | 122 | 12.61 | 186.9 | 121 | | 223.4 | 98 | 37.98 | 220.6 | 101 | 18.75 | 203.9 | 118 | | 218.2 | 88 | 37.09 | 348.5 | 108 | 29.62 | 212.6 | 118 | | 157 | 79 | 26.69 | 103.1 | 94 | 8.76 | 211.8 | 96 | | 184.5 | 97 | 31.37 | 351.6 | 80 | 29.89 | 215.8 | 90 | | |

**Table 1.3**: Churning Out Sample Data (Columns: 16-21)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **total night charge** | **total intl minutes** | **total intl calls** | **total intl charge** | **number customer service calls** | **Churn** | | 11.01 | 10 | 3 | 2.7 | 1 | False. | | 11.45 | 13.7 | 3 | 3.7 | 1 | False. | | 7.32 | 12.2 | 5 | 3.29 | 0 | False. | | 8.86 | 6.6 | 7 | 1.78 | 2 | False. | | 8.41 | 10.1 | 3 | 2.73 | 3 | False. | | 9.18 | 6.3 | 6 | 1.7 | 0 | False. | | 9.57 | 7.5 | 7 | 2.03 | 3 | False. | | 9.53 | 7.1 | 6 | 1.92 | 0 | False. | | 9.71 | 8.7 | 4 | 2.35 | 1 | False. | |

**Table 1.4**: Churning Out predictor Variables

|  |  |
| --- | --- |
| **Serial no.** | **Predictor Variable** |
| 1 | **state** |
| 2 | **account length** |
| 3 | **area code** |
| 4 | **phone number** |
| 5 | |  | | --- | | **Total night charge** | |
| 6 | |  | | --- | | **total intl** | |
| 7 | |  | | --- | | **international plan** | |
| 8 | |  | | --- | | **total intl call** | |
| 9 | |  | | --- | | **total intl charge** | |
| 10 | |  | | --- | | **number customer service calls** | |
| 11 | |  | | --- | | **voice mail plan** | |
| 12 | **number vmail messages** |
| 14 | |  | | --- | | **total day minutes** | |
| 15 | |  | | --- | | **total day calls** | |
| 16 | |  | | --- | | **total day charge** | |
| 17 | |  | | --- | | **total eve minutes** | |
| 18 | |  | | --- | | **total eve calls** | |
| 19 | |  | | --- | | **total eve charge** | |
| 20 | |  | | --- | | **total night minutes** | |
| 21 | **total night calls** |

**Chapter 2**

**Methodology**

**2.1** **Pre Processing**

Preprocessing is very important part of any predictive analysis, we need to select right data for the model. If you choose data which is biased and supporting some special cases, so it will generate biased and inaccurate results. We can’t choose right data just looking on it, so we used analysis methods and graphs to see the right data for model. We use box plot, histogram, scatter plot and various graph for this, we are using the term **Exploratory Data Analysis** for all these process.

In figure 2.1, we have plotted PDF (probability density function) for each numeric variable which are actually affecting the result of churning out customers, we can see some variable are skewed, some are left and right skewed. In the graphs, we have histograms, and the blue line showing density graph overall the data. Which are normally distributed, we will see that data have low stats at starting and high in mid, again low at edges. We need to closely watch data for model

**2.1.1 Outlier Analysis**

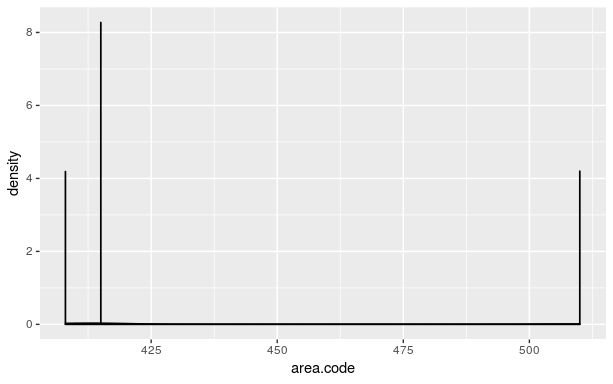
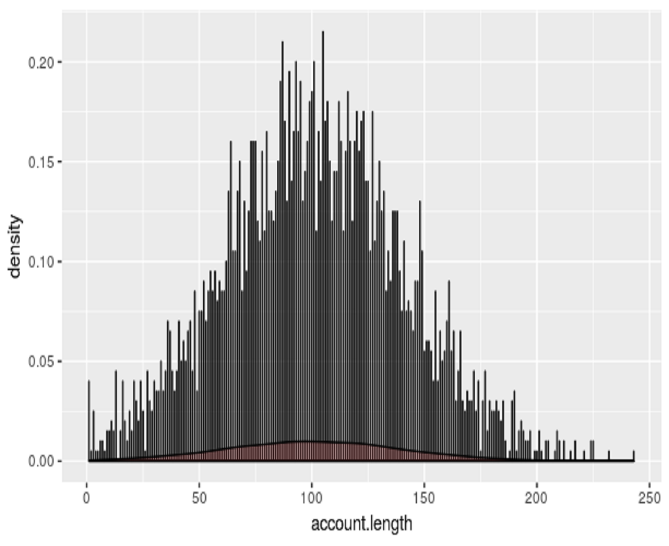
We can clearly observe the predictive variables are skewed, for example “total day calls”, “total eve calls”, “total night calls”, “total night calls”. We can see the probability distribution and mean of these is changed because of these outliers. And we can see the in plots. So these are affecting the result of actual model in last, so we need to handle these things without affecting the actual intent of result. The blue line for PDF, and red for mean distribution.

Various methods for outliers, one of the other steps of **pre-processing** apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers.We visualize the outliers using *boxplots*.

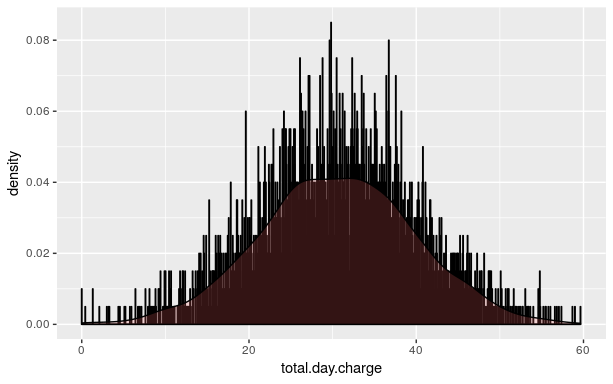
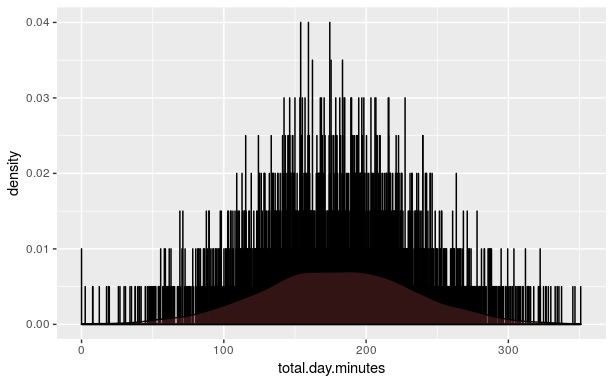
In preprocessing, we filter the data on the basis missing values, outliers [Turkey’s method1] and skewness2 tendency of data. Here we are the whole population, data set instances are comparatively less, and it would not generate any bias results.

1In Turkey’s Method, outliers have been defined as the data points which are ±1*.*5*SD*, and should be removed from the data. It was given by J. W. Tukey in his famous 1977 book *Exploratory Data Analysis.*

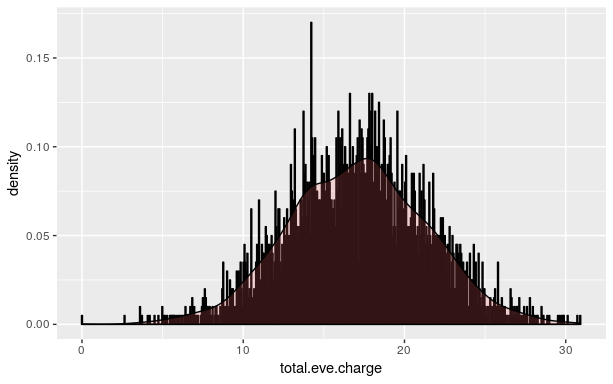
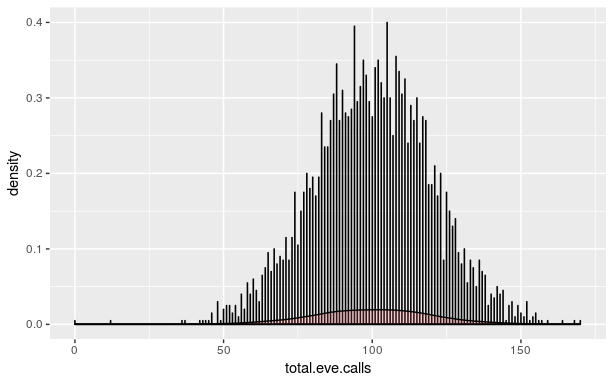
**2**Skewness is asymmetry in a statistical distribution, in which the curve appears distorted or skewed either to the left or to the right.



account.length area.code

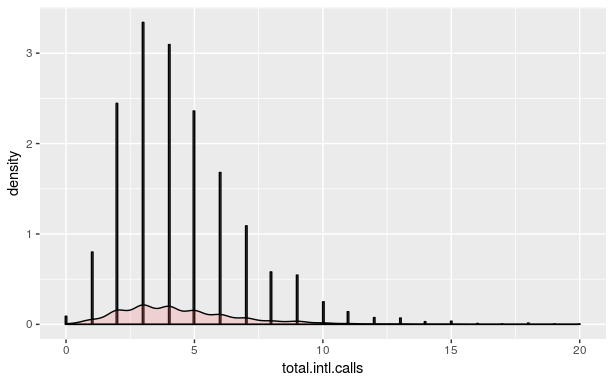
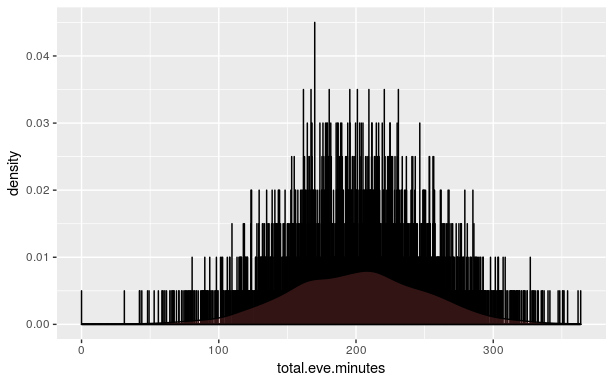


Total.day.minutes Total.day.charge

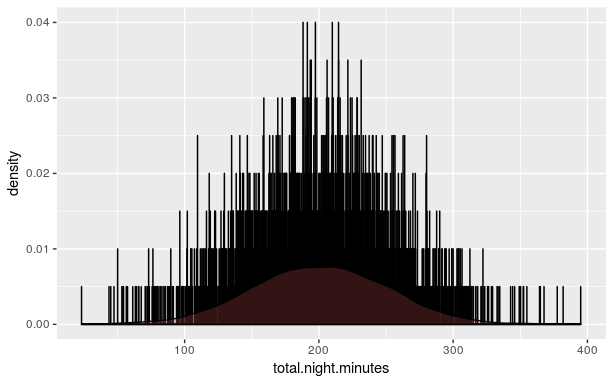
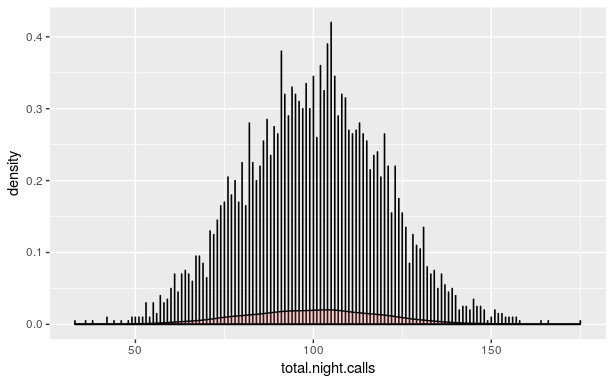


Total.eve.class Total.eve.charge

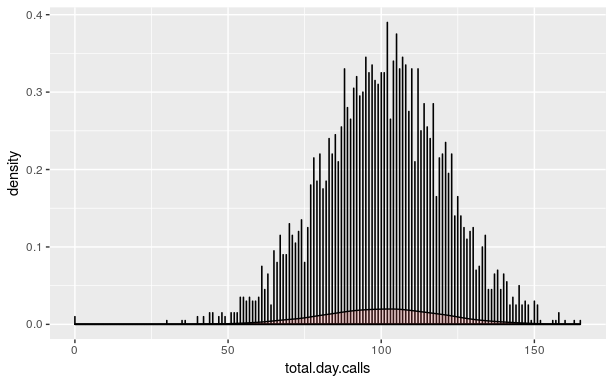
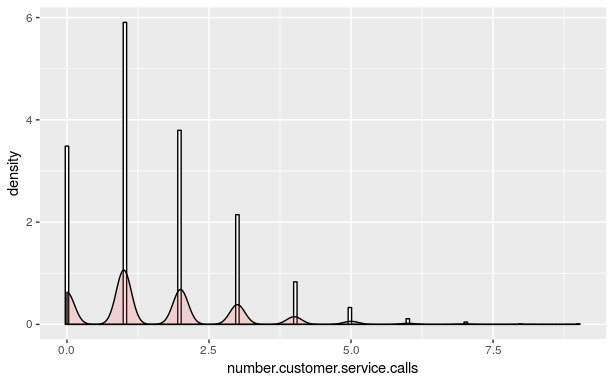
**Figure 2.1: Kernel Density graphs of predictor variables (**[**R code in Appendix**](#code)**)**



Total.eve.minutes Total.intl.calls

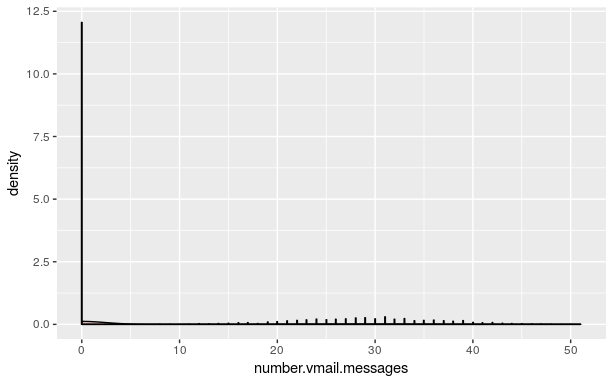
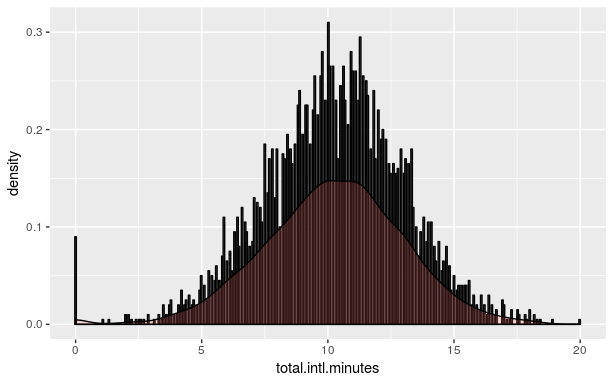


Total.night.calls total.night.minutes

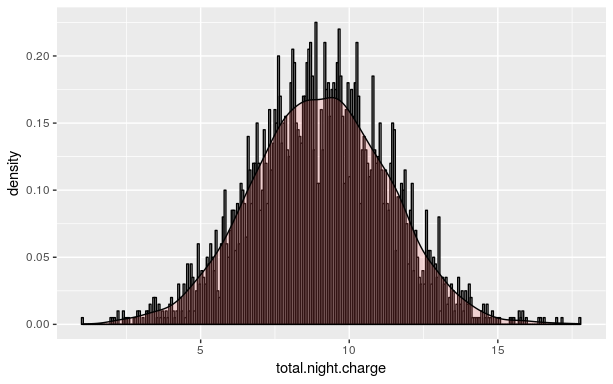
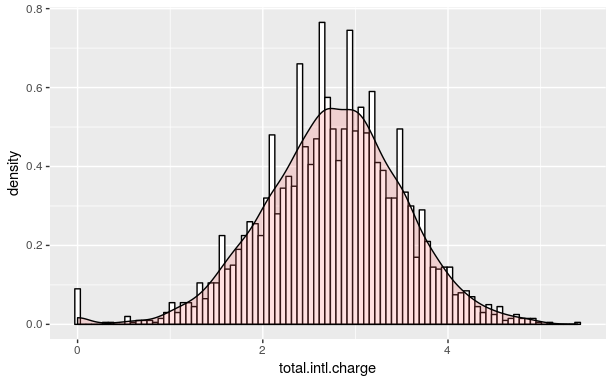


Number.customer.service.calls Total.day.calls

**Figure 2.2: kernel Density graphs of predictor variables (**[**R code in Appendix**](#code)**)**



Total.intl.minutes number.vmail.messages



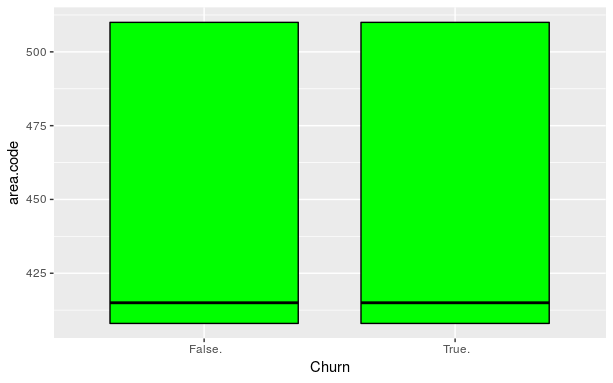
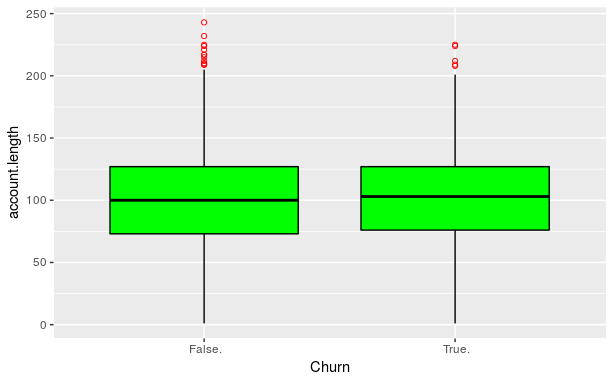
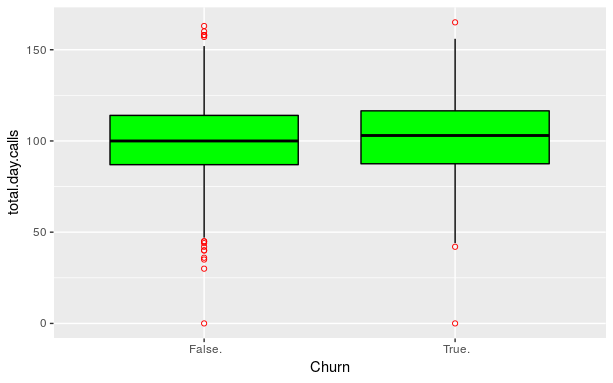
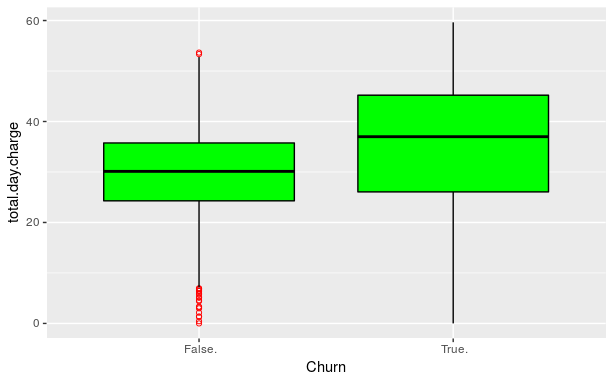
Total intl charge Total night charge

**Figure 2.3: Probability Density Functions of predictor variables (R code in Appendix)**

Other useful inferences can also be drawn from these plots. For example, if we see from the diagrams, the box plots are for true and false value are differently plotted, in false category the no. of outliers are more than true category, also mean and coordinates are different. Some of them are very close, like charges and minutes have parallel proportionate, if minutes increases the charge will also increase. Plotting the phone-number will not affect the target.

We will again plot of all the independent variable after removal of outlier. We plotted the box according to churn classification.

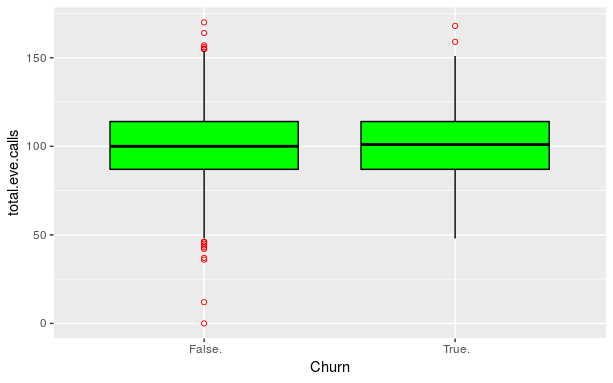
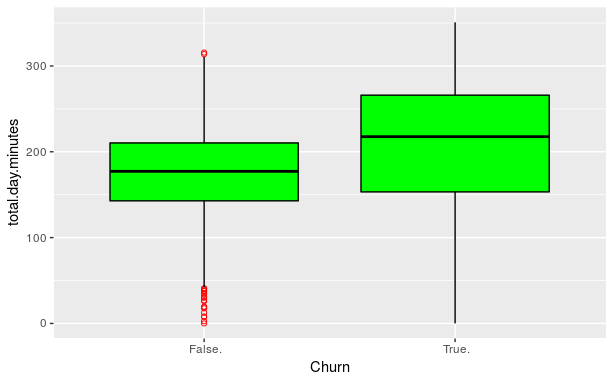
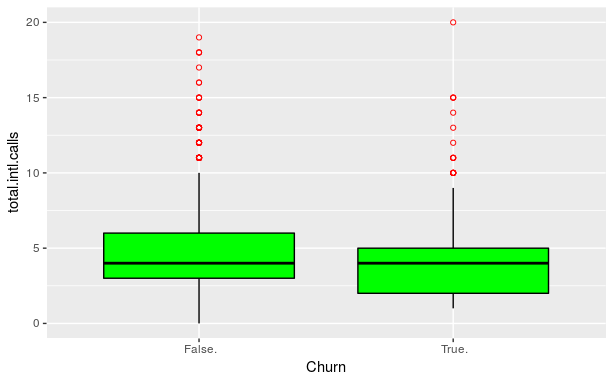
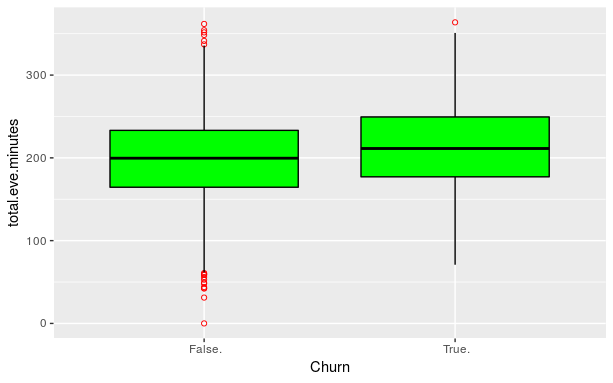
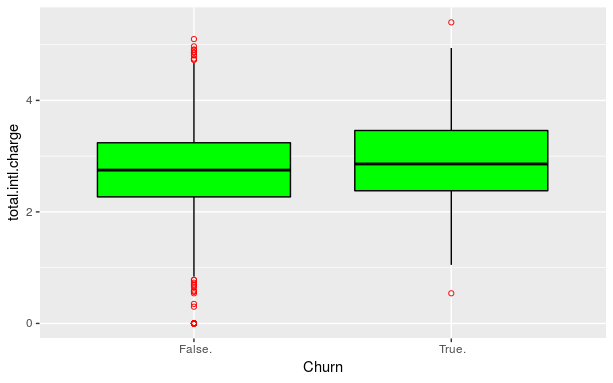
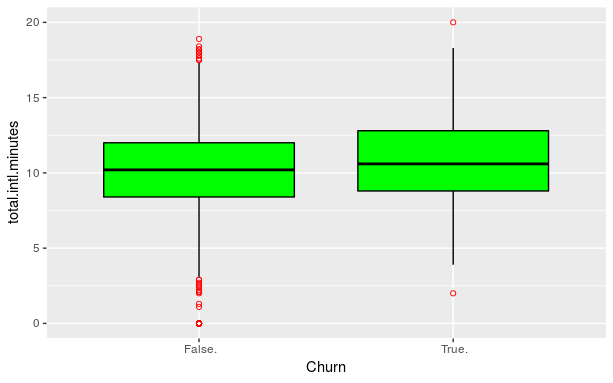
**Box plot model for each predictor with churn**

Account.length Area.code

Total.day.charge Total.day.calls

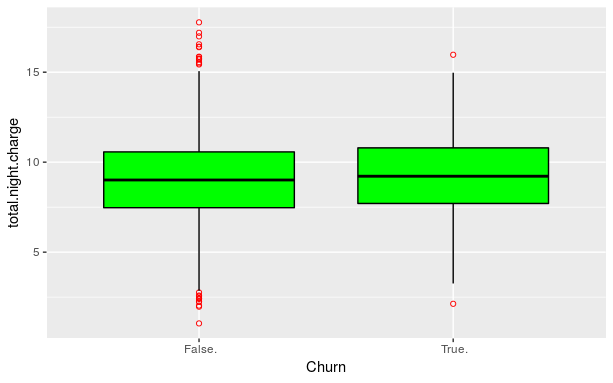
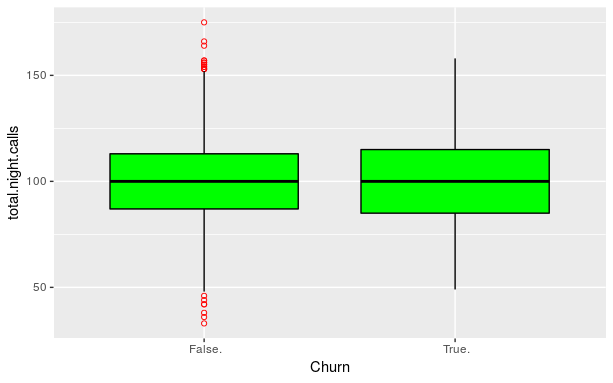
**Figure 2.4: Box Plots of predictor variables with Churn (**[**R code in Appendix**](#code)**)**

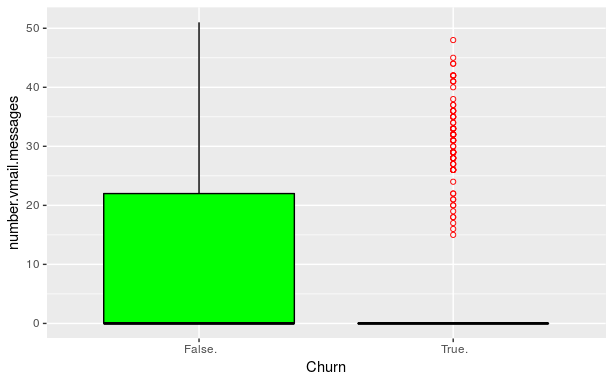
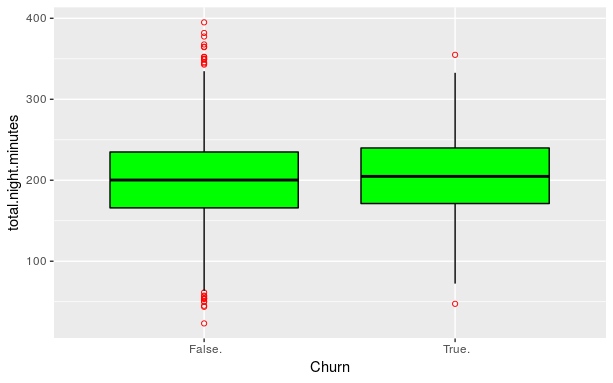
As we can observe from box plot Figure 2.4, that area code don’t have outliers, because area code have unique values, range of area code we can’t define, it’s like factor in numeric data, area code is id particular place. Also outliers of minutes and charge matches with each other.

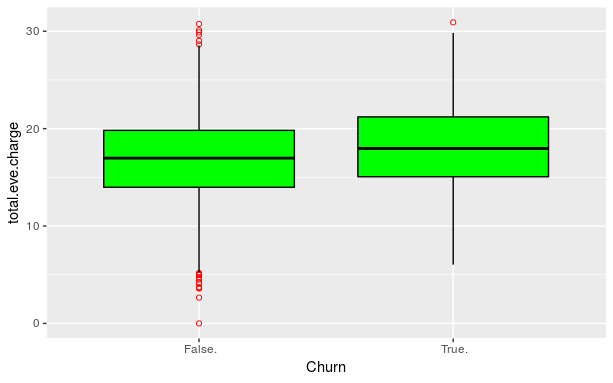
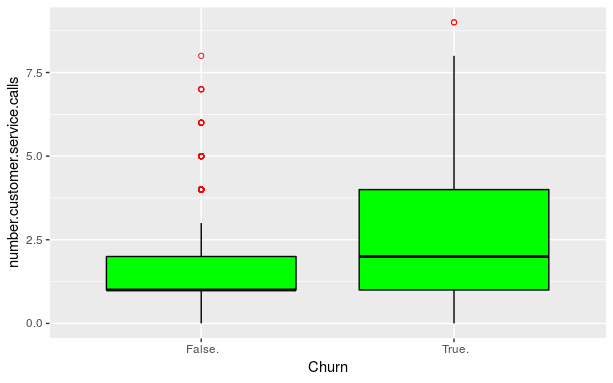
Total.day.minutes Total.eve.calls Total.eve.minutes Total.intl.calls

Total.intl.minutes Total.intl.charges

**Figure 2.5: Box Plots of predictor variables with churn (**[**R code in Appendix**](#code)**)**

Total.night.calls Total.night.charge

Total night minutes Vmail.messages

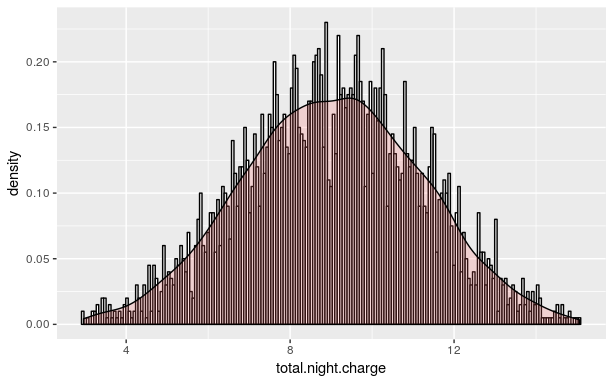
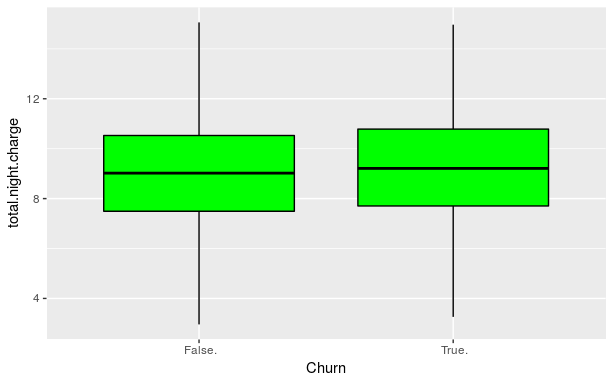
Number.customer.service.calls Total.eve.charge

**Figure 2.6- Predictor box plot with churn out classification**

As we observed from above box plots, mean is changed in the influence of outliers, and these one will affect the final calculation also, we are removing these outliers with relevant data. We used various methods to fulfill that data. In fact this data set have less no. of instances. We applied this one all the data and replaces with KNN.

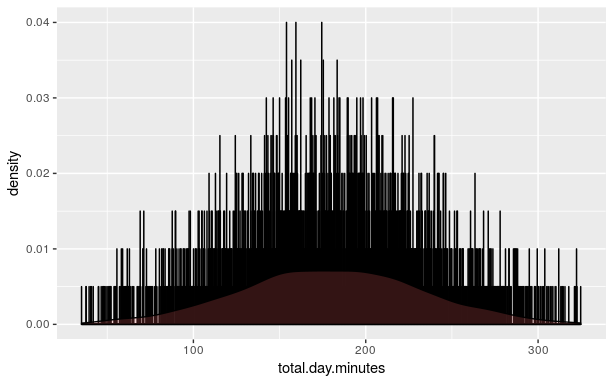
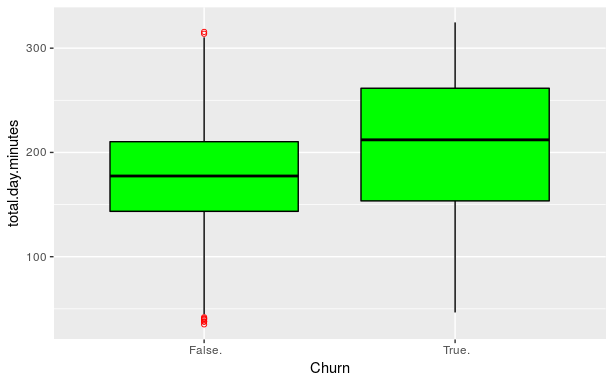
So here we are showing the affect outlier analysis after the outlier removal. We are taking some of them as example.

**Here we plotted *night charge box plot* and density is normal after the outlier removal**



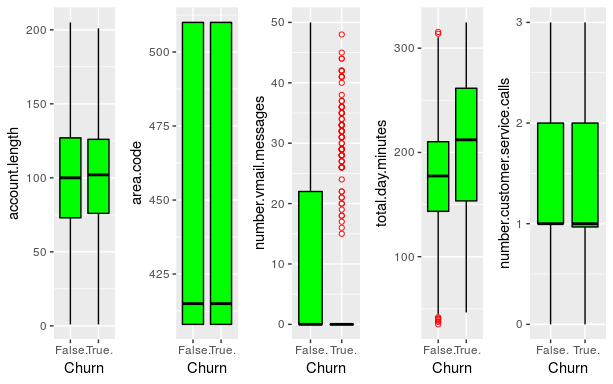
Total.night.charge Total.night.charge

**Here we plotted *night charge box plot* and density, after the outlier removal**

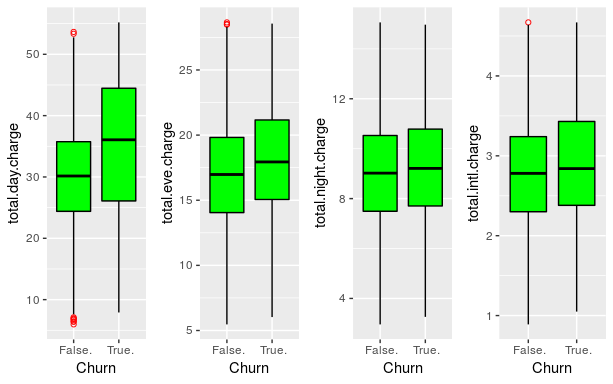


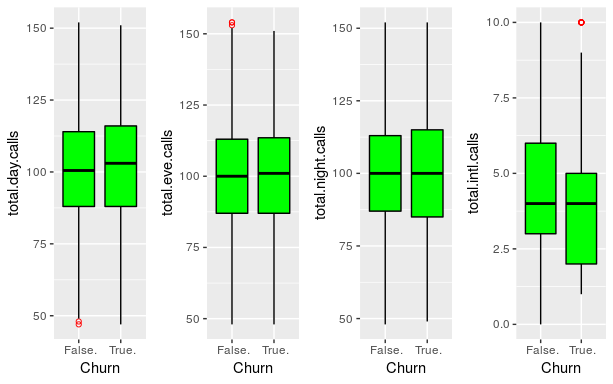
**Figure 2.7, you can see the clear difference in the diagram.**

**Box plot for each predictor after outlier analysis**

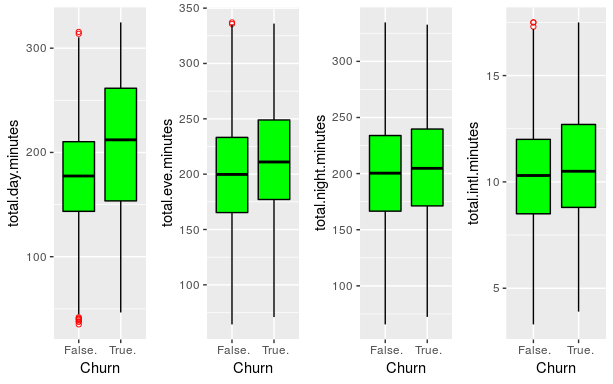
****

**Figure 2.8**

****

**Figure 2.9**

**Figure 2.10**

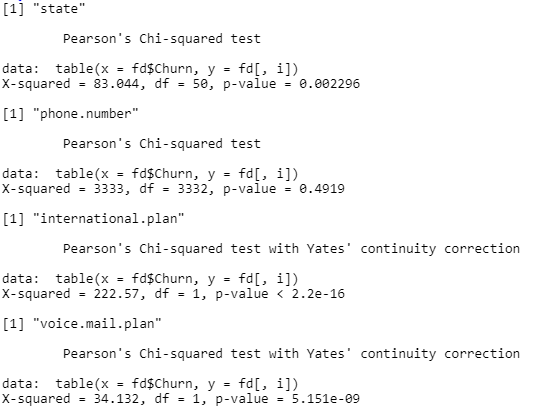
****

**Figure 2.11**

**2.1.2 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 *Correlation plot to select the features for numeric variable and chi2square method for categorical values.* Below all the category variables, for feature selection we are using chi2\_squared test.

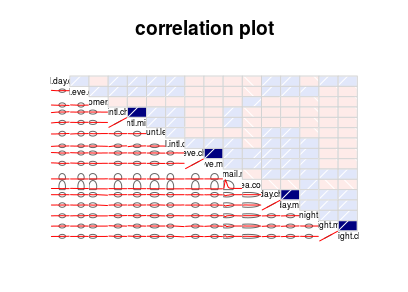
**We have 4 category variables here, state, phone.number, international.plan, voice.mail.plan.**



**Figure 2.12** Above all the observation of Chi2square test

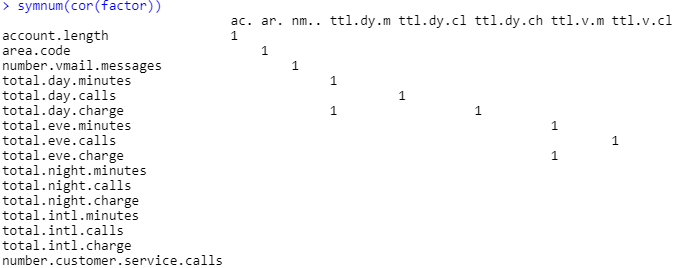
One thing that becomes clear from above predictor variable importance of numeric variable is seen by correlation plot, the darkest blue section depicting that they are highly correlated, and with chi2square test we see that some of the variable have no significance in prediction. Example phone-number adding no information in prediction.

We have 16 different variables in numeric, below is the correlation plot of these variables.

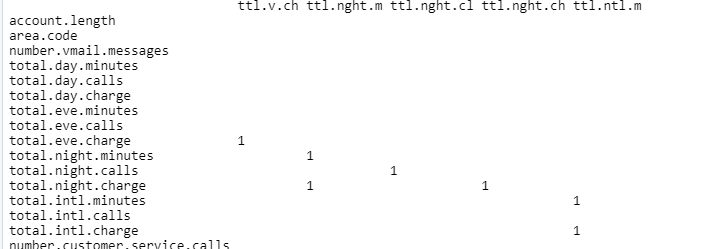


**Figure 2.13**

From the graph we are taking the observation graphical manner and easy to filter the data. A very simple way of looking at correlations in the data is shown above. From the plot we are observing that intl.minute and charge correlated, eveminutes and charge, dayminutes and charge, night.minutes and charge are in dark blue and they are highly correlated. So we can ignore these variables for modelling.



**Figure 2.14**



**Figure 2.15**

**2.2 Modeling**

**2.2.1 Model Selection**

In our early stages of analysis during pre-processing we have come to understand that true and false values are there, but true values are very less than false observations. As we researched online 5-10% churn is normal in telecom industry. So it is obvious to have less true values.

The dependent variable can fall in either of the four categories:

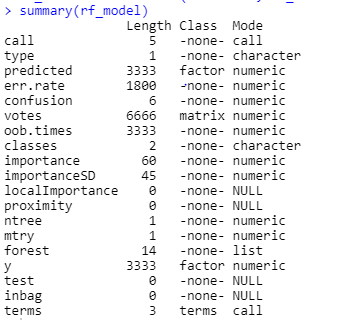
1. Nominal
2. Ordinal
3. Interval
4. Ratio

If the dependent variable, in our case *Churn,* is Ordinal value (yes/no), so in predictive analysis that we can perform is **Classification**, and if the dependent variable is Interval or Ratio the normal method is to do a **Regression** analysis, or classification after binning. But this data set having both kind of variable like categorical as well as numerical values. So here we applied some method to take best fit, Random forest, naïve bay’s, logistic regression.

There for we started to build a model with random forest, then applied other methods also.

**2.2.2 Random Forest**

In above figure, the summary stating all the variables of model. In random forest we used all the filtered variables for prediction. From fig this, we noted p value of variables with respect to churn output. As we applied random forest in this, started with 300 trees and then applies 500, 600 as further. We got results in return and saw good improvement in accuracy of model.



**Figure 2.16**

**2.2.3 Regression rules**

Here, we stating some of the rules of regression trees.

[1] "account.length<=200.5 & international.plan %in% c(' no') & number.vmail.messages<=4 & total.day.minutes<=261 & number.customer.service.calls<=1.99675139258772 & number.customer.service.calls<=1.00098642708666"

[2] "account.length<=200.5 & international.plan %in% c(' no') & number.vmail.messages<=4 & total.day.minutes<=261 & number.customer.service.calls<=1.99675139258772 & number.customer.service.calls>1.00098642708666"

**2.2.4 Classification**

In this case classification become important, specially the case of true value, if we classify the value false if it is true, we could not able to maintain that customer, in this data set have mixed values so we used those methods which take numeric and categorical values. There are 21 independent variables and after feature selection, there are 16 variables rest.

**Chapter 3**

**Conclusion**

**3.1 Model Evaluation**

Now that we have number of 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. Accuracy
2. True positive rate
3. False negative rate

In our case of churn out model, the false negative rate is more important than accuracy, we observed that id customer fall in true but predicted as false, then we can’t able to give extra assistance to that particular customer.

,

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating error metrics for this and finding TP and TF rate.

**3.1.1 Accuracy**

Accuracy is defined as how many values fall in right category in predicted cross validated with real values of train data, in below figure we will see the actual metrics and code. We are evaluating on the basis of that.

#applying Random forest. # directly use data

rf\_model=randomForest(Churn ~ ., df\_reducted,

importance=TRUE,ntree=600)

test=read.csv("Test\_data.csv", header=T)

rf\_predict=predict(rf\_model,test[,-21])

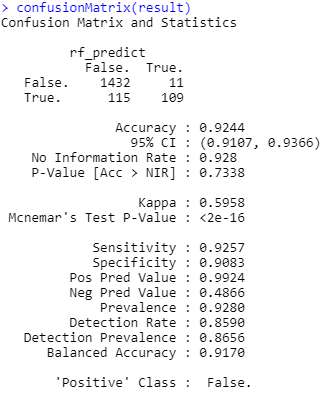
result=table(test$Churn,rf\_predict)

confusionMatrix(result)

**Figure 3.1**

**3.1.2 True Positive Rate**

True positive rate is defined as all true predicted correctly. The formula of this total predicted positive divided by all the real positive values.



**Figure 3.2 (**[**code in appendix**](#code2)**)**

We calculated all the stats, but as we observed all the details, accuracy is approx. 92% but the false negative rate is too high, so this is a problem for this. In the data set, we observed true values are less than false values in churn, 70% less than false values, so this can be *data imbalance* problem, so we replicated true value in data set and got better results

Here is result of metrics of various methods result in table below.

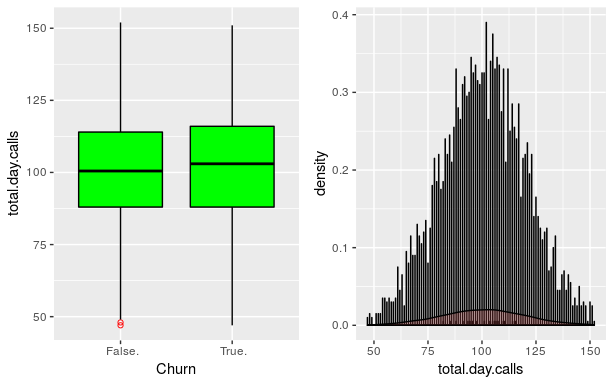
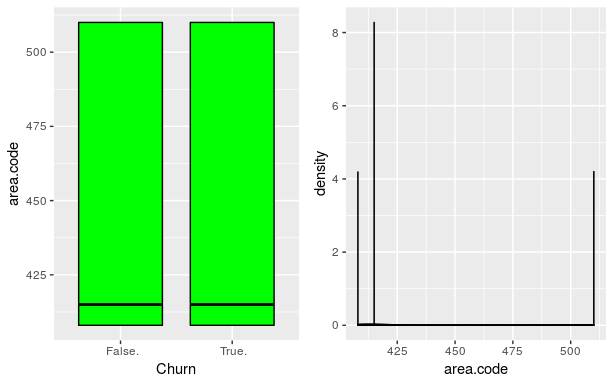
**Figure 3.3**

**3.2 Model Selection**

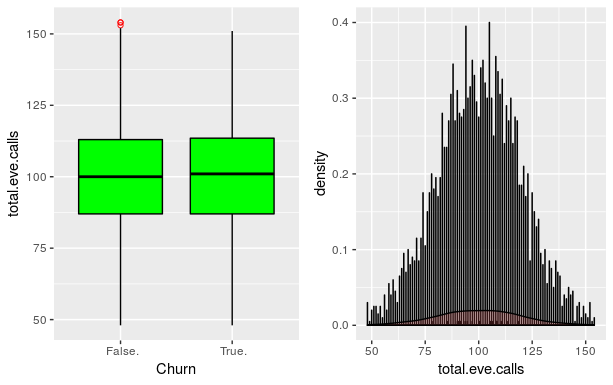
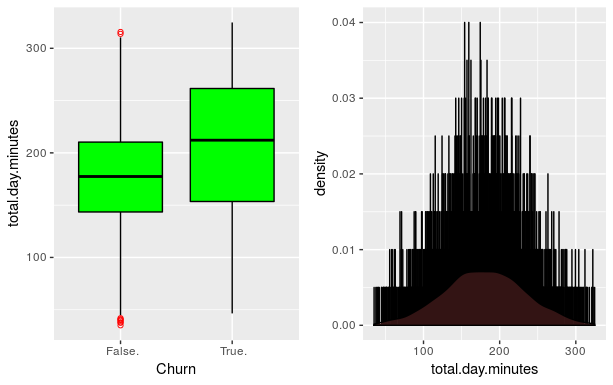
We can see all three models perform comparatively on average and therefore we can select either of the models without any loss of information.

**Appendix A - Extra Figures**

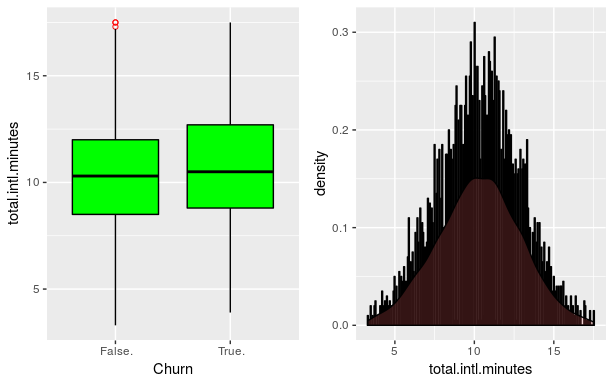
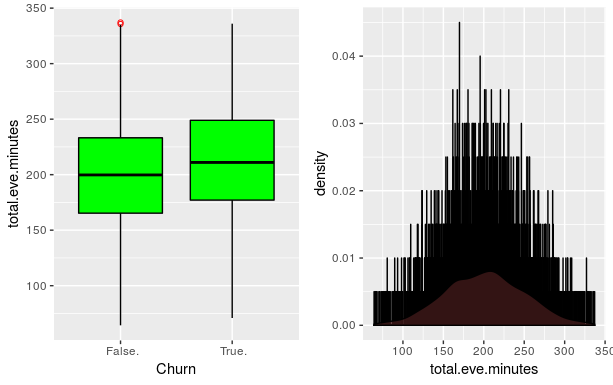
We plotted box and histogram here in figure to observe the effect after outlier removal.



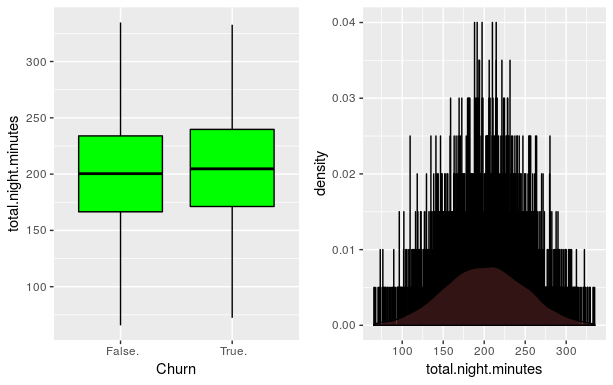
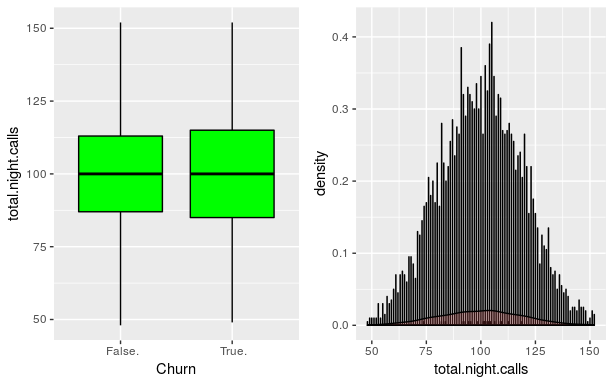
**Figure 4.1**



**Figure 4.2**



**Figure 4.3**



**Figure 4.4** Effect of outliers on predictive variable.

**Appendix B - R Code**

#installing required packages

rm(list=ls())

install.packages(c('corrgram','gridExtra','dplyr','ggplot2','DMwR','RandomForest','inTrees','Rcpp','caret','e107'))

library(ggplot2)

library(dplyr)

library(gridExtra)

library(grid)

library(DMwR)

library(corrgram)

library(randomForest)

library(inTrees)

library(Rcpp)

library(caret)

library(e1071)

#setting working directory

setwd("/cloud/project/Project-ed")

#reading csv file in data frame

df= read.csv("Train\_data.csv",header=T)

#calculating missing values

missing=data.frame(apply(df,2,function(x){sum(is.na(x))}))

missing$coloumns=row.names(missing)

colnames(missing)[1]="values"

row.names(missing)=NULL

missing=missing[,c(2,1)]

#histogram plot with density

for (i in 1:length(cname))

{

nam=paste0("hist",i)

assign(nam, ggplot(df, aes\_string(x=cname[i])) +

geom\_histogram(aes(y=..density..), colour="black", fill="white",binwidth=0.06))

}

#histogram plot with density

for (i in 1:length(cname))

{

nam=paste0("hist",i)

assign(nam, ggplot(df, aes\_string(x=cname[i])) +

geom\_histogram(aes(y=..density..), colour="black", fill="white",binwidth=0.06)+

geom\_density(alpha=.2, fill="#FF6666") )

}

#Outlier analysis, through Box plot

#structure of dataframe

str(df)

#selecting numeric variable

d=sapply(df,is.numeric)

fd=df[,d]

cname=colnames(fd)

for (i in 1:length(cname))

{

#plotting box plot here to analyse outlier

nam=paste0("r",i)

assign(nam,ggplot(data=df, aes\_string(x="Churn",y=cname[i]))+

geom\_boxplot(fill="green", color="black",outlier.color = "red",outlier.shape = 1))

}

grid.arrange(r5,r8,r11,r14,ncol=4)

grid.arrange(r4,r7,r10,r13,ncol=4)

grid.arrange(r6,r9,r12,r15,ncol=4)

grid.arrange(r1,r2,r3,r4,r16,ncol=5)

#plotting predictor variable boxplot with churn classification

for (i in 1:length(cname))

{

nam=paste0("after",i)

assign(nam,ggplot(data=df, aes\_string(x="Churn",y=cname[i]))+

geom\_boxplot(fill="green", color="black",outlier.color = "red",outlier.shape = 1))

}

#Effect of outliers in boxplot

grid.arrange(r5,r8,ncol=2)

grid.arrange(r4,r7,r10,r13,ncol=4)

grid.arrange(r6,r9,r12,r15,ncol=4)

grid.arrange(r1,r2,r3,r4,r16,ncol=5)

#predictor box plots

for(i in cname)

{

value= df[,i][df[,i] %in% boxplot.stats(df[,i])$out]

print(length(value))

df[,i][df[,i] %in% value]=NA

}

#imputing values with KNN method

df=knnImputation(df,k=3)

for (i in 1:length(cname))

{

nam=paste0("r",i)

assign(nam,ggplot(data=df, aes\_string(x="Churn",y=cname[i]))+

geom\_boxplot(fill="green", color="black",outlier.color = "red",outlier.shape = 1))

}

#now feature selection plot numeric with corrgram package

corrgram(df[,d],order=TRUE,lower.panel= panel.ellipse,

upper.panel = panel.shade, text.panel = panel.txt, main="correlation plot" )

#plotting with chi square for categorical variable

factor\_index=sapply(df,is.factor)

fd=df[,factor\_index]

for(i in 1:4 )

{

print(names(fd)[i])

print( chisq.test(table(x=fd$Churn,y=fd[,i])) )

}

#subset the original data frame after removal of correlated variable

df\_reducted= subset(df,select= -c(phone.number,total.intl.charge,total.eve.charge,total.day.charge,total.night.charge))

test= subset(test,select= -c(phone.number,total.intl.charge,total.eve.charge,total.day.charge,total.night.charge))

# plotting histogram plot of cont. variable after removal of outlier

cname=colnames(df\_reducted)

for(i in 1:length(cname))

{

we=df\_reducted[,i]

if(is.numeric(we))

{

assign(paste0("we",i),

ggplot(data=df\_reducted, aes\_string(x=cname[i]))+ geom\_histogram(position="identity",fill = "red", alpha = 0.4))

}

}

#applying Random forest. # directly use data

rf\_model=randomForest(Churn ~ ., df\_reducted, importance=TRUE,ntree=600)

#getting RF object to trees

tree\_list=RF2List(rf\_model)

#extracting rules from tree

ex\_rules=extractRules(tree\_list,df\_reducted[,-16])

#showing some of rules

ex\_rules[1:2,]

#showing various variable of rules in presentable variable

read\_rules=presentRules(ex\_rules,colnames(df\_reducted))

read\_rules[1:2,]

#showing rule metrics of rules

rule\_metric=getRuleMetric(ex\_rules,df\_reducted[,-16],df\_reducted$Churn)

# we get to error

rule\_metric[23,]

# predicting the result on test data

test=read.csv("Test\_data.csv", header=T)

rf\_predict=predict(rf\_model,test[,-21])

result=table(test$Churn,rf\_predict)

confusionMatrix(result) #generating confusion metrics

#now applyin naïve bayes to build model

nb\_model=naiveBayes(Churn ~ ., data=df\_reducted )

nb\_prediction=predict(nb\_model, test[,1:15], type='class')

conf\_mat=table(observed=test[,16],predicted=nb\_prediction)

confusionMatrix(conf\_mat)

#implemented logistic for model

lit\_model=glm(Churn ~ ., data=df\_reducted, family='binomial')

summary(lit\_model)

log\_prediction=predict(lit\_model, newdata=test, type="response")

log\_prediction=ifelse(log\_prediction<0.5,0,1)

log\_table=table(test$Churn,log\_prediction)

confusionMatrix(log\_table)

**Python Code**

#loading all the packages

import numpy as np

import pandas as pd

from scipy.stats import chi2\_contingency

import matplotlib.pyplot as plt

import os

import seaborn as sns #for plotting

from sklearn.ensemble import RandomForestClassifier

#from sklearn.neighbours import

from fancyimpute import KNN

#setting directory for project

os.chdir("E:\Data Science\Edwisor\Project")

os.getcwd()

#reading files in train and test

train=pd.read\_csv("Train\_data.csv", encoding="ISO-8859-1")

test=pd.read\_csv("Test\_data.csv",encoding="ISO-8859-1")

#dimensions of data

train.shape

#sample of data

train.head()

#missing value in data

missing=pd.DataFrame(train.isnull().sum())

missing=missing.reset\_index()

missing=missing.rename(columns= {'index':"missing",0:'value'})

#missing percentage

missing['percentage']=(missing['value']/len(missing))\*100

#sorting values and saving the file

missing=missing.sort\_values("value",ascending=False).reset\_index(drop=True)

missing.to\_csv("missing.csv",index=False)

#outlier analysis

num\_col=train.select\_dtypes(include=[np.number])

col=num\_col.columns.tolist() #this one has all data cols

#box plot

%matplotlib inline

for i in col :

r1=plt.boxplot(train[i])

#getting cols for outlier and replacing with na

for i in col :

# for practice a=train.loc[:,i] #assigning variable

q75=np.percentile(train.loc[:,i],75)

q25=np.percentile(train.loc[:,i],25)

iqr=q75 - q25

min=q25 - (1.5\*iqr)

max=q75 + (1.5\*iqr)

train.loc[train[i]>max,i]=np.nan

train.loc[train[i]>max,i]=np.nan

#firain-lleng data with mean value

train=train.fillna(train.mean())

#getting only numeric variable

num\_col=train.select\_dtypes(include=[np.number])

#setting the width and height of the plot

f, ax= plt.subplots(figsize=(7,5))

#replacing with data with mean value

train=train.fillna(train.mean())

#getting only numeric variable

num\_col=train.select\_dtypes(include=[np.number])

#setting the width and height of the plot

f, ax= plt.subplots(figsize=(7,5))

#creating correlation object fro this

corr = num\_col.corr()

#plot using seaborn library

sns.distplot(train['account length'], kde=True, rug=False)

sns.heatmap(corr,mask= np.zeros\_like(corr,dtype=np.bool), cmap=sns.diverging\_palette(220,10,as\_cmap=True), square=True, ax=ax)

plt.show()

#chi square test

fact\_col=train.select\_dtypes(exclude=[np.number])

col=fact\_col.columns.tolist()

for i in col:

print(i)

chi,p, dof,ex=chi2\_contingency(pd.crosstab(train['Churn'],fact\_col[i]))

print(p)

#selecting only selected features

train\_reducted=train.drop(["Churn","phone number","total day charge","total eve charge","total night charge","total intl charge"],axis=1)

y\_test=test.values[:,20]

test=test.drop(["Churn","phone number","total day charge","total eve charge","total night charge","total intl charge"],axis=1)

#now data is reducted, and with outlier values, applying random forest

rf\_model=RandomForestClassifier(n\_estimators=100)

x=pd.get\_dummies(train\_reducted) #getting string into dumming they are accepting these ones only

y=train.values[:,20] # it will take value only

rf\_model=rf\_model.fit(x,y)

#predicting here

x=pd.get\_dummies(test)

rf\_prediction=rf\_model.predict(x)

#making cross table for this.

cm= pd.crosstab(rf\_prediction,y\_test)

tp=cm.iloc[0,0]

fn=cm.iloc[0,1]

fp=cm.iloc[1,0]

tn=cm.iloc[1,1]

((tp+tn)\*100)/(tp+fn+fp+tn) #92 % accuracy

#false positive rate important in this scenario

**References**

**Bibliography:**

1- Hands on Machine Learning With Python Paperback – John Anderson (Author)

# 2- Fundamentals of Mathematical Statistics Paperback – SC Gupta (author)

**Websites:**

**1-** <https://www.r-statistics.com>

**2-** <https://www.kaggle.com/rtatman/welcome-to-data-science-in-r>