***PROJECT REPORT ON***

# *CHURN PREDICTION*

(***Based on R and Python)***

**Presented by**

**ANKITA**

**(ankii.sharan1292@gmail.com)**

*02 JUNE 2018*

**CONTENTS**

**PAGE NO.**

1. **INTRODUCTION…………………………………………………….....................**3
   1. Problem Statement………………………………………………………………………………………………………….3
   2. Data…………………………………………………………………………………………………………………………………3
2. **METHODOLOGY…………………………………………………………………………**5
   1. Pre Processing…………………………………………………………………………………………………………………5
      1. Missing values………………………………………………………………………………………………….5
      2. Outlier analysis………………………………………………………………………………………………...5
      3. Feature Selection……………………………………………………………………………………………..9
3. **MODELING………………………………………………………………………………..**11
   1. M L Algorithm…………………………………………………………………………………………………………………11
      1. Model Selection……………………………………………………………………………………………….12
      2. Logistic Regression…………………………………………………………………………………………..12
      3. Random forest…………………………………………………………………………………………………13
      4. KNN Imputation……………………………………………………………………………………………….14
      5. Naïve Bayes……………………………………………………………………………………………………..15
4. **MODEL EVALUATION and MODEL EVALUATION………………………………………………………………………………16**
5. **CONCLUSION………………………………………………………………………………17**

**APPENDIX-A LIST OF FIGURES………………………………………………………18**

**APPENDIX-B OUTPUT TERMS……………………………………………………….24**

**REFERENCE……………………………………………………...............................25**

**CHAPTER – 1**

**INTRODUCTION**

Churn refers to when customer ceases their relationship with the company. These days this is common problem for the companies because it is more expensive to acquire a new customer than to prevent the existing customer from leaving. Being able to predict and prevent the existing customer from leaving can offer huge savings to business. One of the best ways to make the predictions is with the help of machine learning techniques.

* 1. **PROBLEM STATEMENT**

Churn loss leads to huge business loss. The aim of this project is to predict the churn score (who are likely to move) based on the previous patterns and behavior of the customers. To make these predictions we are using R and Python codes and algorithms.

* 1. **DATA**

The data we are using for prediction comes under Classification category and depending on the problem category we will develop and select the algorithms to predict the churn score.

We are using 18 predictors to predict the churn score which are listed in table 1.1. These predictors are very useful to make the Churn prediction.

**Table 1.1:** Variables Name

|  |  |
| --- | --- |
| No. | Variable |
| 1 | State |
| 2 | Account Length |
| 3 | Area code |
| 4 | International Plan |
| 5 | Voicemail Plan |
| 6 | No. of Voicemail Message |
| 7 | Total Day Minutes |
| 8 | Total Day Calls |
| 9 | Total Day Charges |
| 10 | Total Evening Minutes |
| 11 | Total Evening Calls |
| 12 | Total Evening Charges |
| 13 | Total Night Minutes |
| 14 | Total Night Calls |
| 15 | Total Night Charges |
| 16 | Total International Calls |
| 17 | Total International Minutes |
| 18 | Total International Charges |
| 19 | No of Customer Services Call made |

Here is the sample of data on which we are going to apply the various techniques and algorithms and predict the target class.

**Table: 1.2** Churn Dataset (Columns: 1 to 10)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State | Account  Length | Inter  national  Plan | Voicemail  Plan | No. of  Voicemail  Messages | Total  Day  Minutes | Total Day Calls | Total  Day  Charge | Total  Eve.  minutes | Total Eve  Calls |
| KS | 128 | No | Yes | 25 | 265.1 | 110 | 45.07 | 197.4 | 99 |
| OH | 107 | No | Yes | 26 | 161.6 | 123 | 27.47 | 195.5 | 103 |
| NJ | 137 | No | No | 0 | 243.4 | 114 | 41.38 | 121.2 | 110 |
| OH | 84 | Yes | No | 0 | 299.4 | 71 | 50.9 | 61.9 | 88 |
| OK | 75 | Yes | No | 0 | 166.7 | 113 | 28.34 | 148.3 | 122 |
| AL | 118 | Yes | No | 0 | 223.4 | 98 | 37.98 | 220.6 | 101 |
| MA | 121 | No | Yes | 24 | 218.2 | 88 | 37.09 | 348.5 | 108 |

**Table: 1.3** Churn Dataset (Columns: 11 to 18)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Total Eve. Charge | Total Night Minutes | Total Night Calls | Total Night Charge | Total Intnl Minutes | Total Intnl Calls | Total Intnl Charge | No. of Cust. Service Calls | Churn |
| 16.78 | 244.7 | 91 | 11.01 | 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 | 7.32 | 12.2 | 5 | 3.29 | 0 | False |
| 5.26 | 196.9 | 89 | 8.86 | 6.6 | 7 | 1.78 | 2 | False |
| 12.61 | 186.9 | 121 | 8.41 | 10.1 | 3 | 2.73 | 3 | False |
| 18.75 | 203.9 | 118 | 9.18 | 6.3 | 6 | 1.7 | 0 | False |
| 29.62 | 212.6 | 118 | 9.57 | 7.5 | 7 | 2.03 | 3 | False |

**CHAPTER – 2**

**METHODLOLGY**

Data is the backbone on data science. When we work on any project 80% of the time goes on data understanding, data cleaning and data preparation as driving the data according to problem statement is very necessary. If the quality of data is good the model will predict better and results in high accuracy. For cleaning the data we go for various techniques given below and then we apply algorithms on cleaned data.

**2.1 Pre Processing**

Data which we receive from clients are messy data and are in different format. We cannot feed these data directly in the model as model understands only structured format. In data pre processing we clean and prepare the data and make them in structured format according to model category. We should observe the data and understand them. Looking at data and understanding them with the help of different tools and graph and visualizations is called Exploratory Data Analysis. It is one of the very important steps as driving the data according to problem statement is very necessary and to drive the data we need to understand our data first.

As our dataset consists of both numerical and categorical data first we checked the structure of data and converted them into required format.

**2.1.1 Missing Value Analysis**

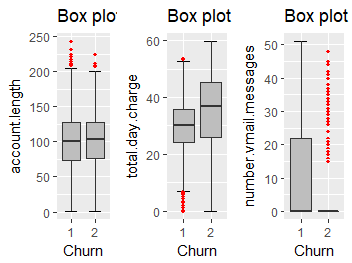
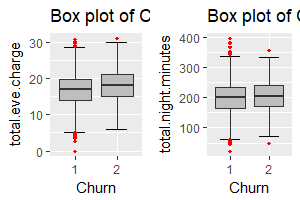
After converting our data into structured format we did missing value analysis. Missing Values are the values or data points which are not present in our dataset. There are multiple methods to deal with missing values such as mean method, KNN imputation as presence of missing values in our data set will results in poor accuracy.

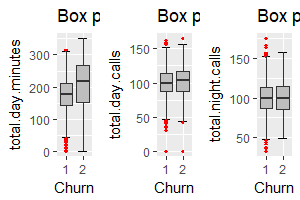
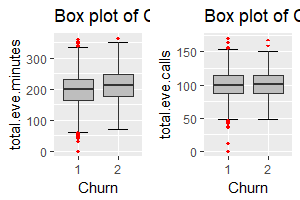
In our dataset we checked for missing values and there are no missing values present in our dataset.

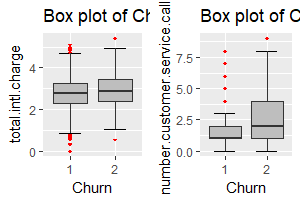
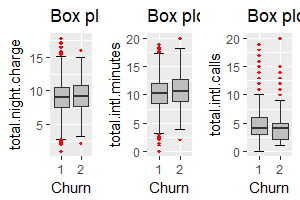
**2.2.2 Outlier Analysis**

Outlier analysis is a pre processing technique used to check for abnormal values in the data set clean them and transform the data into a proper shape. There are many different techniques like Graphical Tools (Box Plot Method), Statistical technique (Grubbs test), R Package outlier for outlier analysis. Presence of outlier in our data leads to poor data quality and contamination, low quality measurement and manual errors. The best way to look at outlier is to understand business process i.e. how data is generated and how is the business flow.

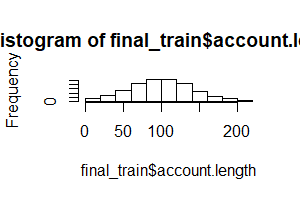
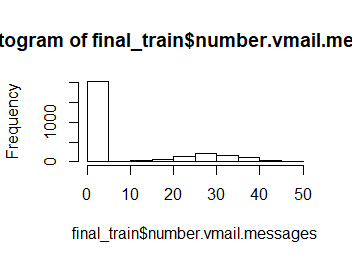
In our project we detect the outlier with the help of Box plot Method. We have separated numerical and categorical variables as outlier analysis is applicable only on numerical variables. To check where the outliers are present in our dataset we plotted predictors versus Box plot graphs (Figure2.1). We have used Box plot method to remove the outliers. After removing the outliers we checked our data with the help of histogram plots and box plots. The graph without outliers is shown in figure 2.2. In figure 2.1 we have plotted box plot for 15 predictor variables with respect to Churn (predicted). The red dots are the outliers and we can see there is extreme overlap between conditions. Similar and somewhat cases can be seen for test data (appendix 1). In figure 2.2 after removal of outliers data points are no more overlapping.

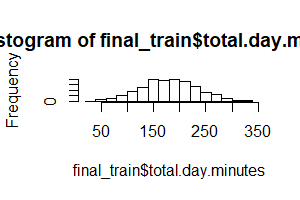
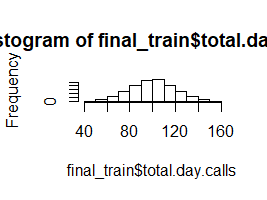
 

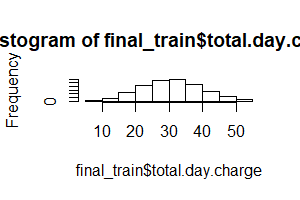
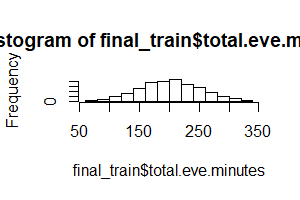
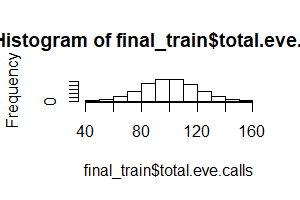
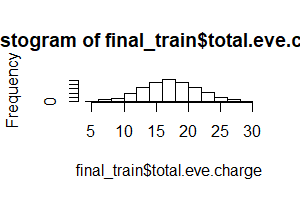


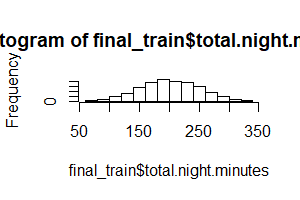
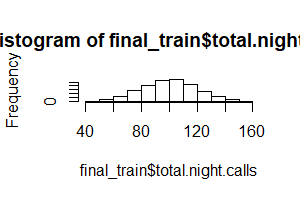


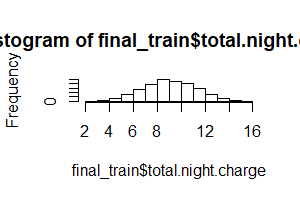
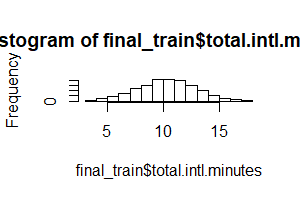
**Fig 2.1:** Box plot of predictors versus churn of train data (with outliers)

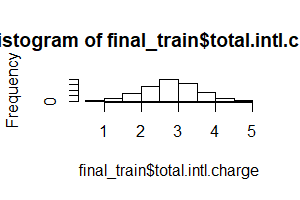
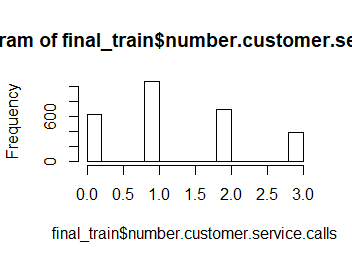
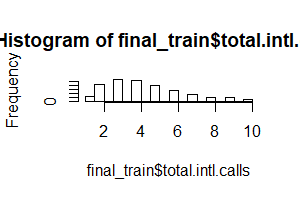
 

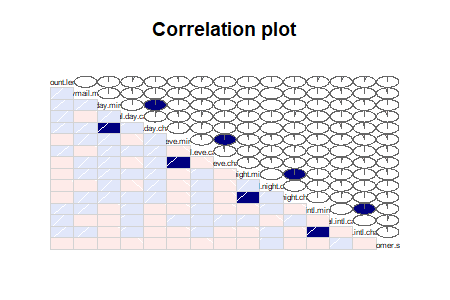
**Fig 2.2:** Histogram of train data (without outliers)

**2.2.3 Feature Selection**

Selecting subset of relevant features for model construction is known as Feature Selection. When we get raw data we have multiple variables and with the help of variable selection we have to extract relevant data. It is also called attribute selection or Dimensionality Reduction. We cannot use all the features because some features may be carrying the same information or irrelevant information which does not impact the business solution. To reduce complexity we adopt feature selection technique to extract meaningful features out of it. This in turn helps us to avoid the problem of multi colinearity. Correlation Analysis, Chi Square Test and other ML algorithm like Random Forest are some of the methods of dimensionality reduction.

The very important point about dimensionality reduction is there should be low dependency between two independent variables but there should be very high dependency between an independent and a dependent variable.

In this project we have used correlation analysis and chi square test for dimensionality reduction. Correlation Analysis is a statistical technique to measure the dependencies between continuous variables. Chi square test is used when we have categorical variables. It works on contingency table and hypothesis testing to measure the dependencies. Figure 2.3 represents correlation plot for train data and shows how and whether the variables are dependent or not. Also chi square values for train data are listed in table 2.1 below. We will avoid those variables whose p value is greater than 0.05 for categorical data. For continuous variables we will see the correlation plot and measure the dependencies between the variables whether they are highly positively correlated or highly negatively correlated. The deep blue color shows that the variables are highly positively correlated where as light pink color shows they are strongly negatively correlated.



**Fig 2.3:** Correlation plot for train data

Chi square value of train data

|  |  |
| --- | --- |
| Variable Name | P value |
| State | 0.01892 |
| Area code | 0.7649 |
| International Plan | < 2.2e-16 |
| Voice mail plan | 1.932e-07 |

**Table 2.1:** Chi Square Value of Train data

After applying correlation analysis and chi square test out of 20 predictors we are left with 18 predictor variables and the target variable i.e. Churn. The variables we are considered after applying correlation analysis and chi square test are:

1. State
2. Account length
3. International plan
4. Voicemail plan
5. No. of voicemail message
6. Total day minutes
7. Total day calls
8. Total day charge
9. Total evening minutes
10. Total evening calls
11. Total evening charge
12. Total night call
13. Total night minutes
14. Total night charge
15. Total international minutes
16. Total international call
17. Total international charge
18. Number of customer services call made
19. Churn(target variable)

**CHAPTER – 3**

**MODELING**

**3.1 MACHINE LEARNING ALGORITHM**

Machine Learning means programming computers to optimize a performance criterion using example data or past experience. With help of historical data we try to extract patterns and save it in ML itself. Once a new test case comes in we use that historical pattern to apply on new data to predict its class level.

**3.1.1 Model Selection**

In our early stages of analysis during pre-processing we have come to understand the customer behavior pattern on test data.

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

1. Nominal

2. Ordinal

3. Interval

4. Ratio

If the dependent variable, in our case **Churn** is categorical the only 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. The dependent variable we are dealing with is Categorical, for which classification is preferred according to problem statement. We should always start our model building from the simplest to more complex. Here we have used different ML Algorithm on trained data and then applied it on test to predict the future values in both R and Python. Different algorithm gave different results with different accuracy and False Negative Rate.

The Machine Learning Model we applied in R and Python are as follows:

**Logistic Regression**

Logistic Regression is statistical model for classification. Its possible outcome could be classes or probabilities. The outcome of logistic regression is given below.

***FOR “R”***

Null deviance: 1917.0 on 2788 degrees of freedom

Residual deviance: 1253.2 on 2721 degrees of freedom

AIC: 1389.2

logistic\_pred (where 1 = no, 2 = yes)

1(no) 2(yes)

1(no) 909 22

2(yes) 84 40

Accuracy: 0.8995

95% CI: (0.8798, 0.917)

No Information Rate: 0.9412

P-Value [Acc > NIR]: 1

Kappa: 0.3817

Mcnemar's Test P-Value: 3.126e-09

Sensitivity: 0.9154

Specificity: 0.6452

Pos Pred Value: 0.9764

Neg Pred Value: 0.3226

Prevalence: 0.9412

Detection Rate: 0.8616

Detection Prevalence: 0.8825

Balanced Accuracy: 0.7803

'Positive' Class: 1

#accuracy = 89.95

# FNR = 35.00

Here our accuracy is 89.95 where as our False negative rate is 35.00. In any model we should have high accuracy and low false negative rate. Our accuracy as well as false negative rate is quite satisfying. Here total no of observations (n = 1055) out of which false negative rate is 85 i.e. 35% which a client is often interested to know.

Difference between null deviance and residual deviance should be high. If the difference is low

Which means our model does not contain much variance to explain our target variable.

**Random Forest**

Random Forest is an ensemble technique that consists of many decision trees. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It is called random forest as we are building n no. of trees randomly. In other words, to build the decision trees it selects randomly n no of variables and n no of observations to build each decision tree. It means to build each decision tree on random forest we are not going to use the same data. The method combines Breimans “bagging” idea and the random selection of features.

***(For “R”)***

rf\_predict

1 (no) 2(yes)

1(no) 923 8

2(yes) 41 83

Accuracy: 0.9536

95% CI: (0.9391, 0.9654)

No Information Rate: 0.9137

P-Value [Acc > NIR]: 4.091e-07

Kappa: 0.7469

Mcnemar's Test P-Value: 4.844e-06

Sensitivity: 0.9575

Specificity: 0.9121

Pos Pred Value: 0.9914

Neg Pred Value: 0.6694

Prevalence: 0.9137

Detection Rate: 0.8749

Detection Prevalence: 0.8825

Balanced Accuracy: 0.9348

'Positive' Class: 1

Accuracy is 95.36% and FNR is 77.28

***“For Python”***

Accuracy is 95.63% and FNR is 81.45

When we applied Random Forest Algorithm in R and Python and checked for accuracy and False Negative Rate it was not reliable. The False negative ratio was not very high which is not considerable.

**KNN Imputation**

KNN stands for K Nearest Neighbor. KNN is simple that stores all available cases and classifies new cases based on similar measure. It is a supervised learning technique and can be applied on both classification and regression. It is a lazy learning method because it never stores the pattern from training data. In KNN Imputation whenever we get a set of new test data to predict the target label it will take all the values of test data and try to calculate the distance between that test data with all the training observations or historical observations present in the data. This algorithm is time consuming but bit accurate from all other algorithms as it consider all the observations irrespective of patterns and variance.

***(For “R”)***

knn\_predict

1(no) 2(yes)

1(no) 870 61

2(yes) 73 51

Accuracy: 0.873

95% CI: (0.8514, 0.8925)

No Information Rate: 0.8938

P-Value [Acc > NIR]: 0.9861

Kappa: 0.3609

Mcnemar's Test P-Value: 0.3420

Sensitivity: 0.9226

Specificity: 0.4554

Pos Pred Value: 0.9345

Neg Pred Value: 0.4113

Prevalence: 0.8938

Detection Rate: 0.8246

Detection Prevalence: 0.8825

Balanced Accuracy: 0.6890

'Positive' Class: 1

Accuracy is 87.3 and FNR is 54.46

***(“For Python***”***)***

Accuracy is 87.29 and FNR is 54.46

In KNN Imputation, for R accuracy is 87.3 which are acceptable but False Negative Rate is 54.46 which are high and not considerable. In Python also the accuracy is high which is acceptable but false negative rate is high which is not acceptable.

**Naïve Bayes**

Naïve Bayes is one of the most practical learning methods works on Bayes theorem of Probability to predict the classes of unknown dataset. Naïve Bayes is one of the supervised learning algorithm which works on probabilities. This algorithm allows us to predict a class for a set of features predictors using the probability. So this is also known as Probabilistic Algorithm for classifier. It is “assumptions of independence”. It means Naïve Bayes classifier assumes that the presence of a particular feature in the dataset is unrelated to the presence of any other features in the data. We do not need to apply correlation analysis for this algorithm because even though two independent variables are highly correlated this will holdno value in Naïve Bayes model as this model assumes that each independent variable is contributing separately.

***(“For R”)***

naive\_predict

1(no) 2(yes)

1(no) 913 18

2(yes) 47 77

Accuracy: 0.9384

95% CI: (0.9221, 0.9521)

No Information Rate: 0.91

P-Value [Acc > NIR]: 0.0004357

Kappa: 0.6695

Mcnemar's Test P-Value: 0.0005147

Sensitivity: 0.9510

Specificity: 0.8105

Pos Pred Value: 0.9807

Neg Pred Value: 0.6210

Prevalence: 0.9100

Detection Rate: 0.8654

Detection Prevalence: 0.8825

Balanced Accuracy: 0.8808

'Positive' Class: 1

Accuracy is 93.84 and FNR is 37.90

***(“For Python”)***,

Accuracy is 91.09 and FNR is 39.72

In Naïve Bayes for R and Python both accuracy and false negative rate is acceptable. So we can consider this algorithm for our project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Accuracy R | FNR R | Accuracy Python | FNR Python |
| Random Forest | 95.36 | 77.28 | 95.63 | 81.45 |
| Logistic Regression | 89.95 | 35.00 | - | - |
| KNN | 87.3 | 54.46 | 87.29 | 54.46 |
| Naïve Bayes | 93.84 | 37.90 | 91.09 | 39.72 |

**Table 3.1:** Summary of accuracy and FNR

**CHAPTER – 4**

**MODEL EVALUATION & MODEL SELECTION**

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. In confusion matrix we have two classes “yes” and “no”. Some other important metric of confusion matrix are:

* **True positives (TP):** These are cases in which we predicted yes (they have churned), and they do moved out or churned.
* **True negatives (TN):** We predicted no, and they do not moved out or churned.
* **False positives (FP):** We predicted yes, but they don't actually move out or churned. (Also known as a "Type I error.")
* **False negatives (FN):** We predicted no, but they actually have churned or moved out. (Also known as a "Type II error.")

For classification we go for confusion matrix, False negative rate and Recall. In the project we have calculated accuracy with the help of confusion matrix and FNR for both R and python. Besides accuracy, False Negative Rate is important error metric which a client generally demands. Here to evaluate our model besides accuracy we have calculated False Negative Rate (in which the prediction is no but the customer have actually churned out or move).

Based on these evaluations LOGISTIC REGRESSION is best suited for R codes which here has highest accuracy and lowest False Negative Rate and NAÏVE BAYES is best model for Python for the given dataset.

**CHAPTER-5**

**CONCLUSION**

In this project **“Churn Analysis”** we have applied different models to predict the final result. For each algorithm we got different accuracy and false negative rate which are acceptable in some case and not considerable in some cases. We applied the algorithm on both R and Python for the same dataset as mentioned in the project.

In **R** we have selected LOGISTIC REGRESSION for predictions as it gave the highest accuracy and lowest false negative rate which is 89.95% accurate and have 35.00 false negative rate.

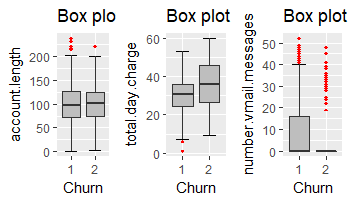
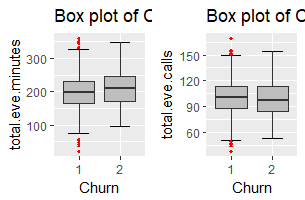
According to Logistic Regression, False Negative Rate is 35.00 and 84 customer are going to churn.

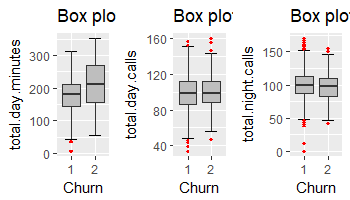
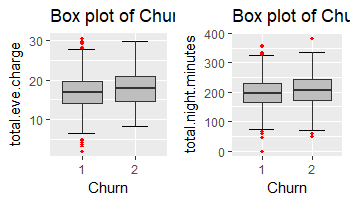
In **Python,** the best model is Naïve Bayes giving the highest accuracy and lowest False Negative Rate which is 91.09 and 39.72 respectively.

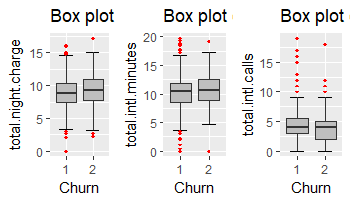
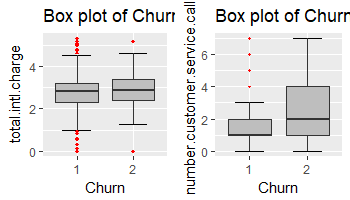
***APPENDIX***

**APPENDIX – A (Extra Figures)**

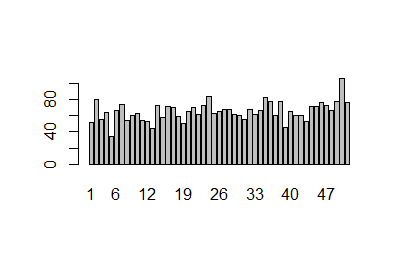
**Box Plot for Test Data**

** **

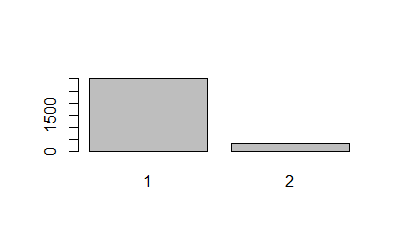
****

** **

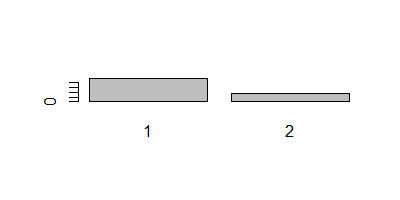
**Bar Plot for Categorical data (Train data)**

****

**(Bar Plot of State)**

****

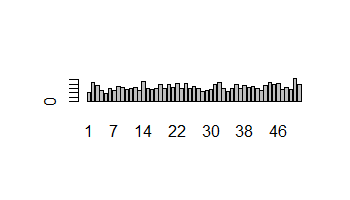
**(Bar Plot of International Plan)**

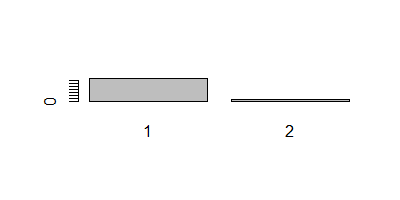
****

**(Bar Plot of Voice Mail Plan)**

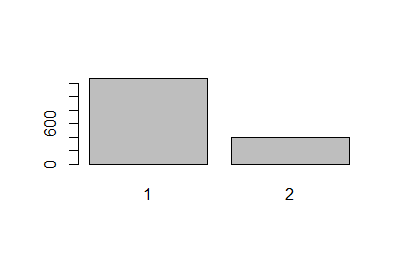
**Bar Plot for Categorical Data(Test Data)**

**(Bar Plot for State)**

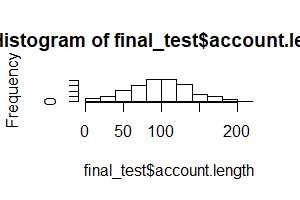
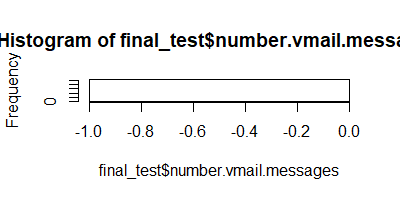


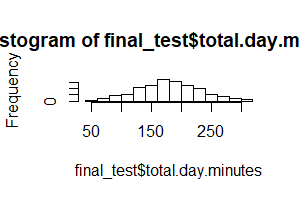
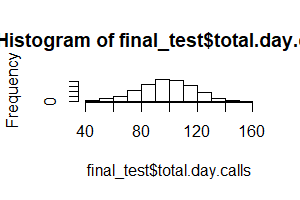
(**Bar Plot for international plan)**

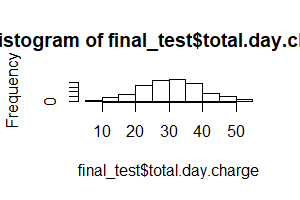
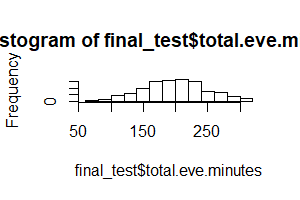
**(Bar Plot for voice mail plan)**

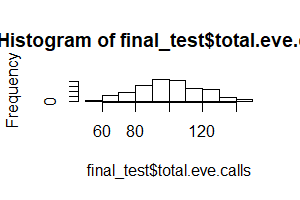
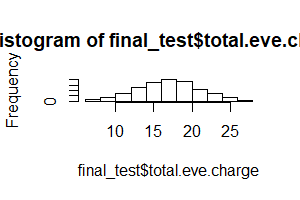
****

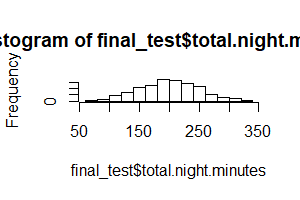
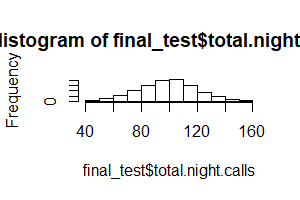
**Histogram for Test Data without Outliers**

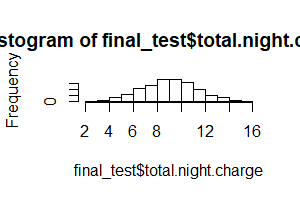
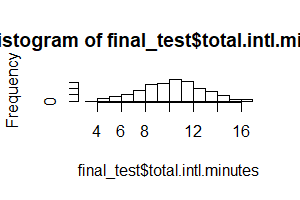
 

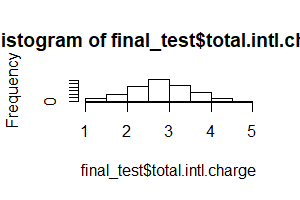
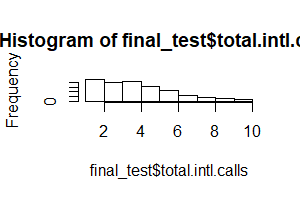
 

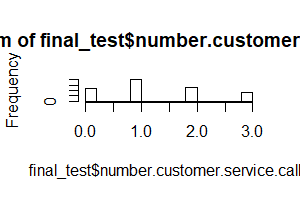
 



**APPENDIX – 2(Output Terms)**

**AIC (Alkaline Information Criteria)**

It adjusts the loc likelihood based on the number of observation and complexity of the model.

**BIC (Baisen Information Criteria)**

It is similar to AIC but has high penalty for models.

**Omnibus**

Provides combined statistical test for the presence of skewness and kurtosis. Basically it is breakdown of skewness and kurtosis.

**Skew and Kurtosis**

These tests are basically for time series dataset.

**Null Deviance**

It tells us how well the response variable is predicted by the model with intercept only.

**Residual Deviance**

It tells us how well the response variable is predicted by using null deviance and all other independent variables.

**REFERENCES**

1. **“EdWisor Videos”**
2. **“My study channel(youtube)”**
3. **“Introduction to deep computer vision” by** *John olafelwa and Moses olafelwa*