greatlearning
Learning for Life

GREAT LAKES
EXECUTIVE LEARNING

Predictive Modelling Project – Logistic Regression, kNN ,Naïve Bayes

By Ramprasad Mohan

Telecom Customer Churn Prediction

Table of Contents

1 Project Objective

The project objective is to solve the business problem of a Telecom firm facing customer churn. Customer Churn is a burning problem for many Telecom companies. In this project, we simulate one such case of customer churn where we would work on a data of postpaid customers with a contract. The data has information about the customer usage behavior, contract details and the payment details. The data also indicates which customers cancelled their service were. Based on this past data, we need to build a predictive model which can predict whether a customer will cancel their service in the future or not.
Managerial Report is to be prepared.

2 Data Dictionary

Below are the eleven variables or columns in the data set and its explanation:

- Churn : if the customer cancelled service (1) or not (0) [predictor variable]
- AccountWeeks : Number of weeks the customer had active account
- ContractRenewal : If customer recently renewed contract (1) or not (0)
- DataPlan : If the customer has a data plan (1) or not (0)
- DataUsage: Monthly data usage in gigabytes (GB)
- CustServCalls: Number of calls into customer service
- DayMins : Average daytime minutes per month
- DayCalls : Average number of daytime calls
- MonthlyCharge : Average monthly bill
- OverageFee : largest overage fee in last 12 months
- RoamMins : Average number of roaming minutes

3 Step by Step approach

We shall follow step by step approach to arrive to the conclusion as follows:

- Exploratory Data Analysis, descriptive statistics
- Outlier treatment and Multicollinearity check
- Splitting the data to training and testing dataset
- Build Logistic Regression model using training set
- Predict using testing set and evaluate the model with confusion matrix and ROC
- Normalize or scale the data and then Build kNN model using scaled training set
- Predict using testing set and evaluate the model metrics
- Build Naïve Bayes model using training set
- Predict using testing set and evaluate the model metrics
- Compare the three models based on the accuracy to find which model is best suited.
- Inferences and insights

4 Solution

1 EDA - Basic data summary, Univariate, Bivariate analysis, graphs

After importing the data, we could see from the below result that the Telecom data has 3333 observations and 11 variables (The variable names are available in Data dictionary section). All of them are numeric variables.

```
'data.frame':   3333 obs. of  11 variables:
 $ Churn           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AccountWeeks    : num  128 107 137 84 75 118 121 147 117 141 ...
 $ ContractRenewal: num  1 1 1 0 0 0 1 0 1 0 ...
 $ DataPlan        : num  1 1 0 0 0 0 1 0 0 1 ...
 $ DataUsage       : num  2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
 $ CustServCalls   : num  1 1 0 2 3 0 3 0 1 0 ...
 $ DayMins         : num  265 162 243 299 167 ...
 $ DayCalls        : num  110 123 114 71 113 98 88 79 97 84 ...
 $ MonthlyCharge   : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
 $ OverageFee      : num  9.87 9.78 6.06 3.1 7.42 ...
 $ RoamMins        : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
```

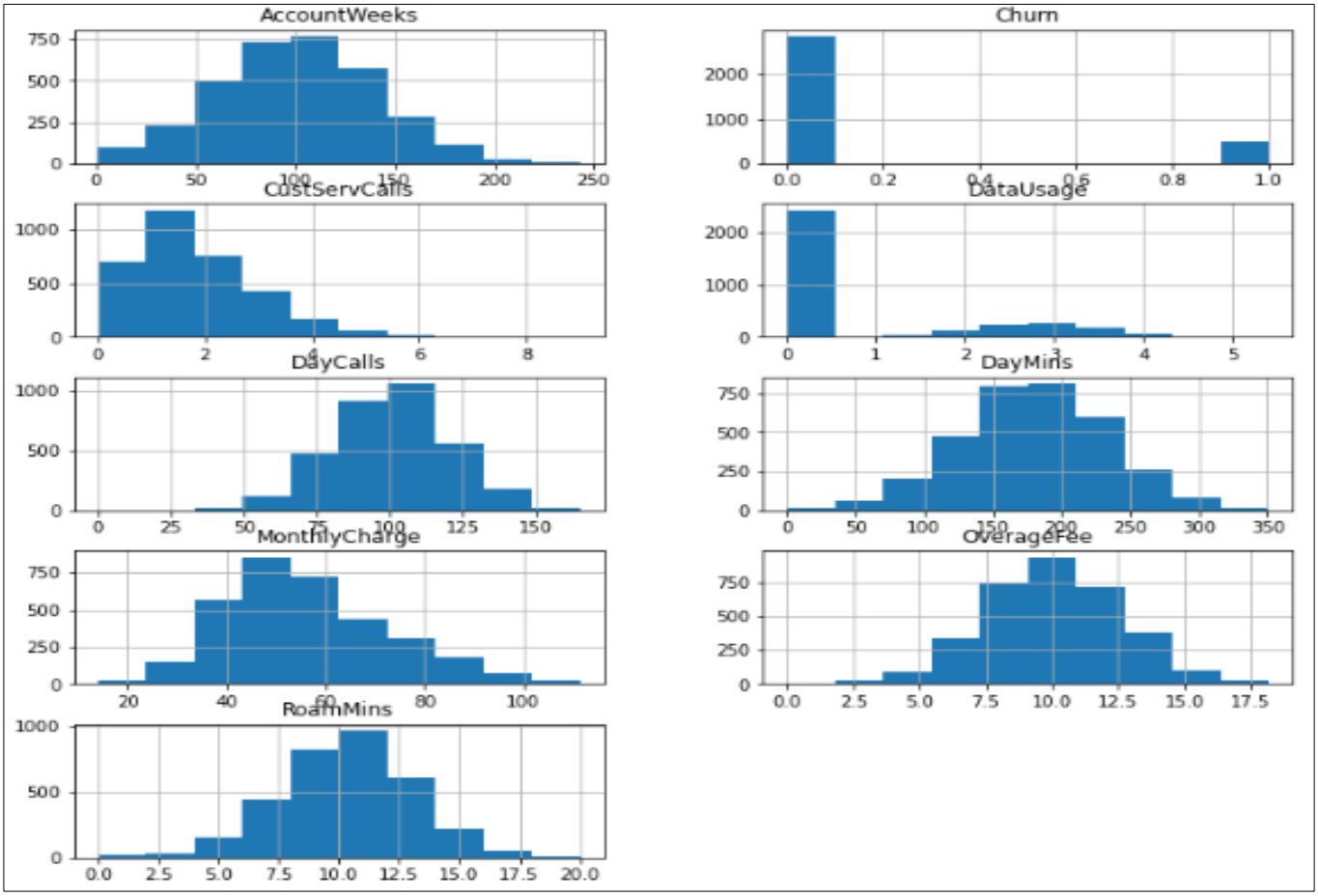
Next we check the summary of the dataset. From the below table, by looking at the median and the mean numbers, it gives us an idea that Data Usage is highly skewed and Customer service calls are less skewed, rest of them are all normally distributed. We will plot the data to see further.

Churn	AccountWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls
Min. :0.0000	Min. : 1.0	Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.000	Min. : 0.0	Min. : 0.0
1st Qu.:0.0000	1st Qu.: 74.0	1st Qu.:1.0000	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:1.000	1st Qu.:143.7	1st Qu.: 87.0
Median :0.0000	Median :101.0	Median :1.0000	Median :0.0000	Median :0.0000	Median :1.000	Median :179.4	Median :101.0
Mean :0.1449	Mean :101.1	Mean :0.9031	Mean :0.2766	Mean :0.8165	Mean :1.563	Mean :179.8	Mean :100.4
3rd Qu.:0.0000	3rd Qu.:127.0	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.:1.7800	3rd Qu.:2.000	3rd Qu.:216.4	3rd Qu.:114.0
Max. :1.0000	Max. :243.0	Max. :1.0000	Max. :1.0000	Max. :5.4000	Max. :9.000	Max. :350.8	Max. :165.0

MonthlyCharge	OverageFee	RoamMins
Min. : 14.00	Min. : 0.00	Min. : 0.00
1st Qu.: 45.00	1st Qu.: 8.33	1st Qu.: 8.50
Median : 53.50	Median :10.07	Median :10.30
Mean : 56.31	Mean :10.05	Mean :10.24
3rd Qu.: 66.20	3rd Qu.:11.77	3rd Qu.:12.10
Max. :111.30	Max. :18.19	Max. :20.00

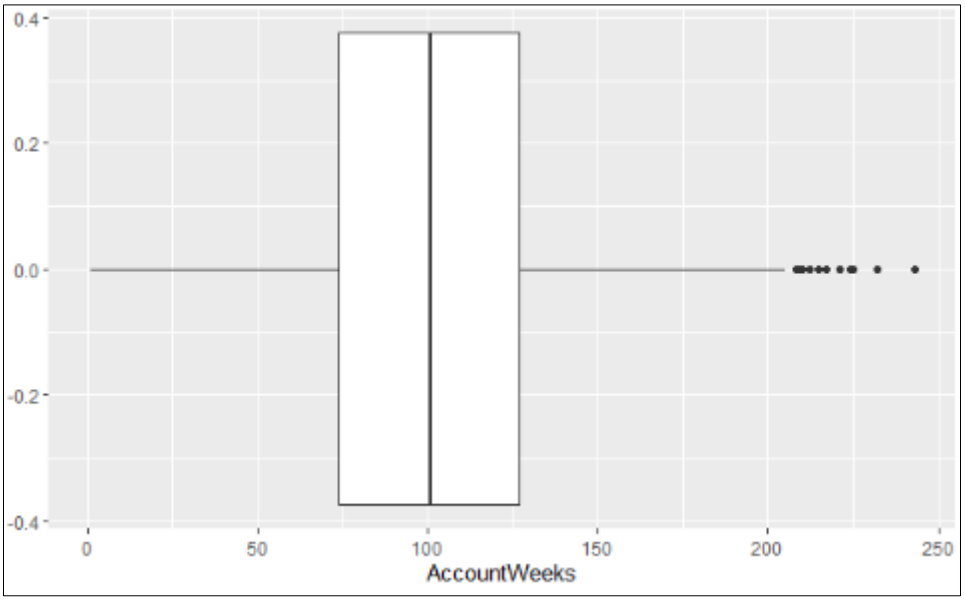
- In contract renewal , the number of records having value ‘1’ is very high compared to ‘0’
- DataUsage is highly skewed with minimum usage 0 GB being very high number of records.
- DayMins , Day Calls, Overage Fee, Roam mins are perfectly normal distributions
- In DataPlan, the customers without Data plan is more than those with data plan
- Customer service call one time by the customers is maximum
- The data is slightly imbalanced with less churn count and most customers have not cancelled service and has high counts.

Histograms of the variables



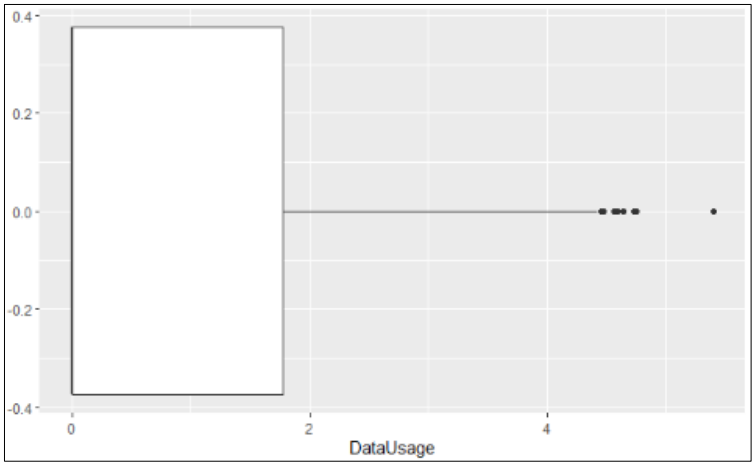
Univariate Analysis (Boxplot) for continuous variable:

Boxplot of Account Weeks:



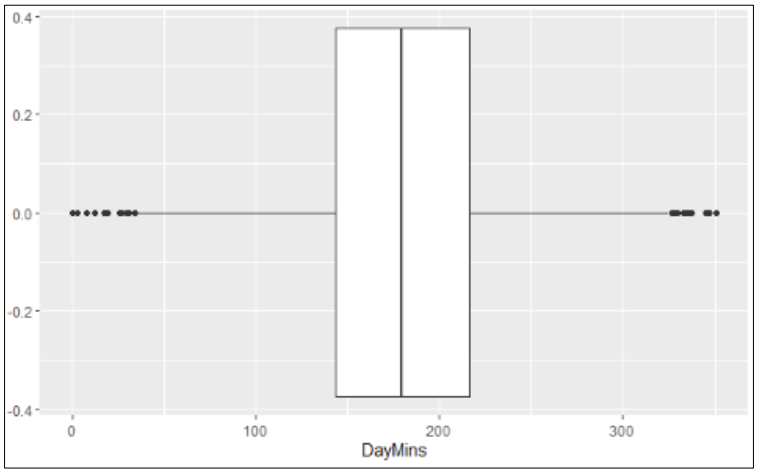
It is slightly left – skewed with outliers in one side.

Boxplot of Data Usage:



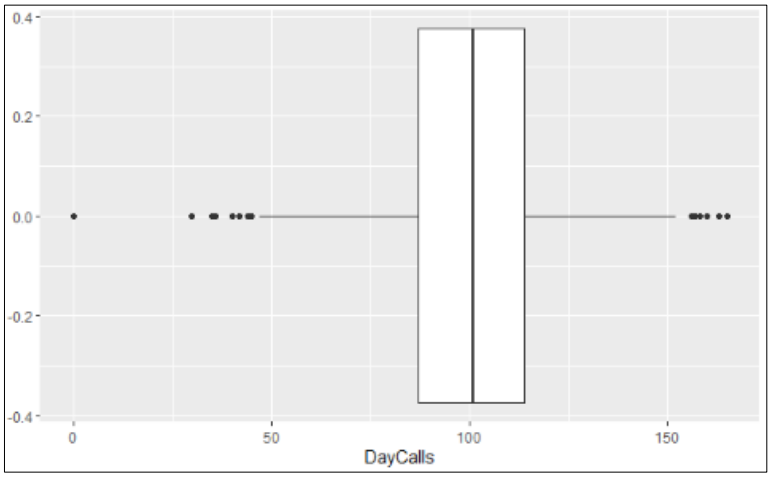
It is highly left – skewed with median and mean being '0' as the most occurring value.

Boxplot of Day Mins:



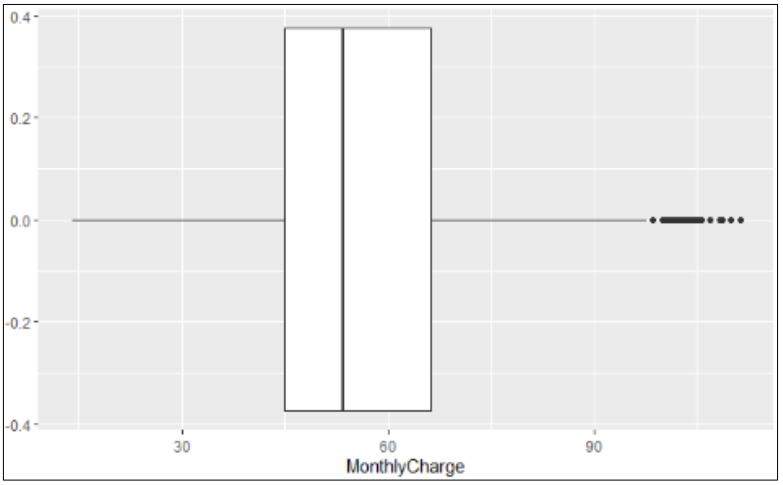
It is normally distributed with outliers on both the sides.

Boxplot of Day Calls:



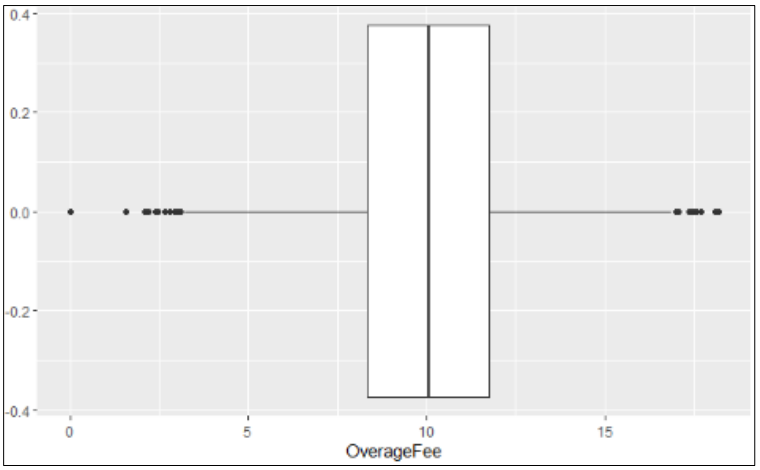
It is slightly rights skewed, with extreme outliers on one side.

Boxplot of Monthly Bill Charge:



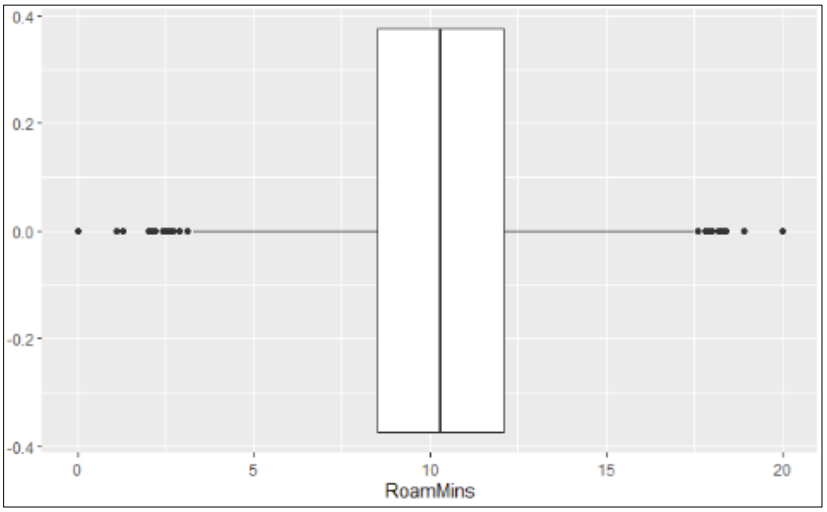
It is slightly left skewed with all outliers on right side only

Boxplot of Overage Fee:



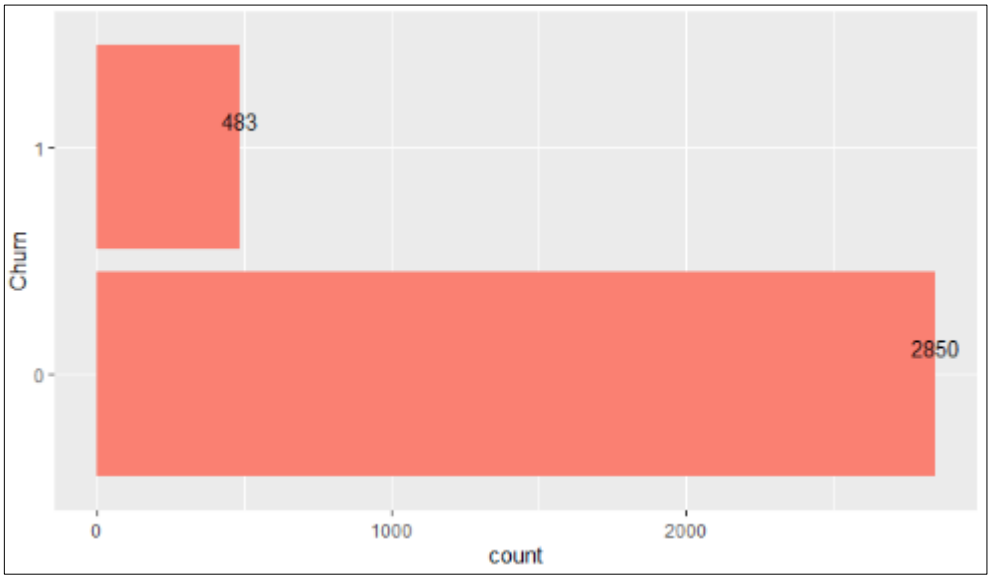
It is slightly right – skewed with extreme outliers on one side.

Boxplot of Roaming Minutes:



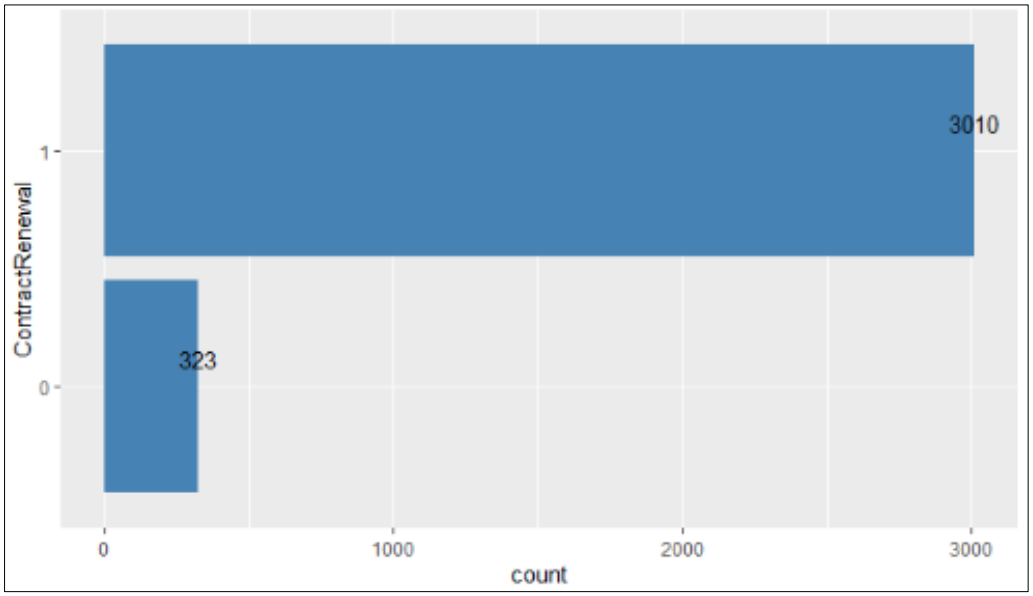
Univariate Analysis (Barplot) for factor variables:

Barchart of Churn Rate:



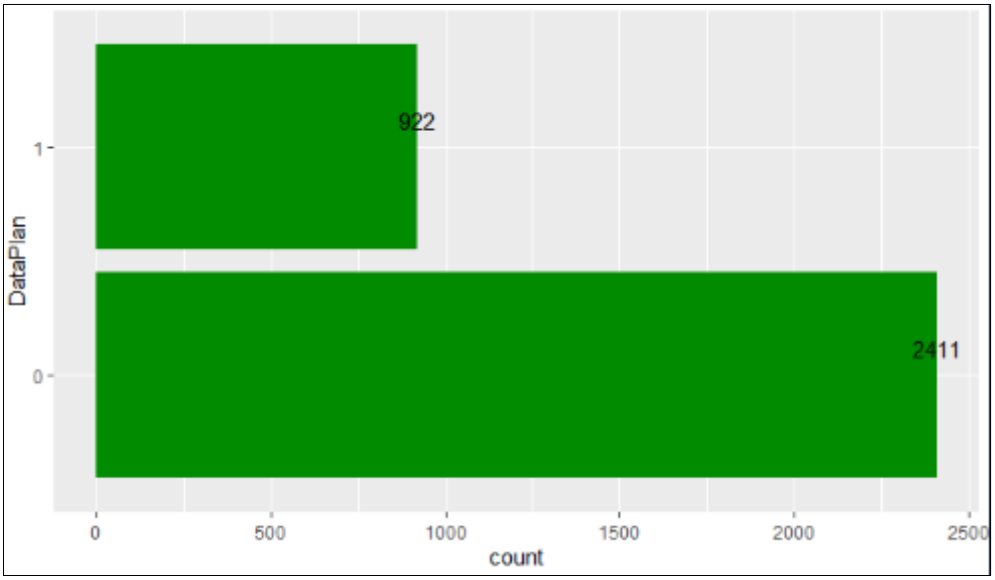
There are 483 customers who cancelled the service out of 3333 customers, which comes out to be 14.5% of the Churn ratio.

Barchart of Contract Renewal:



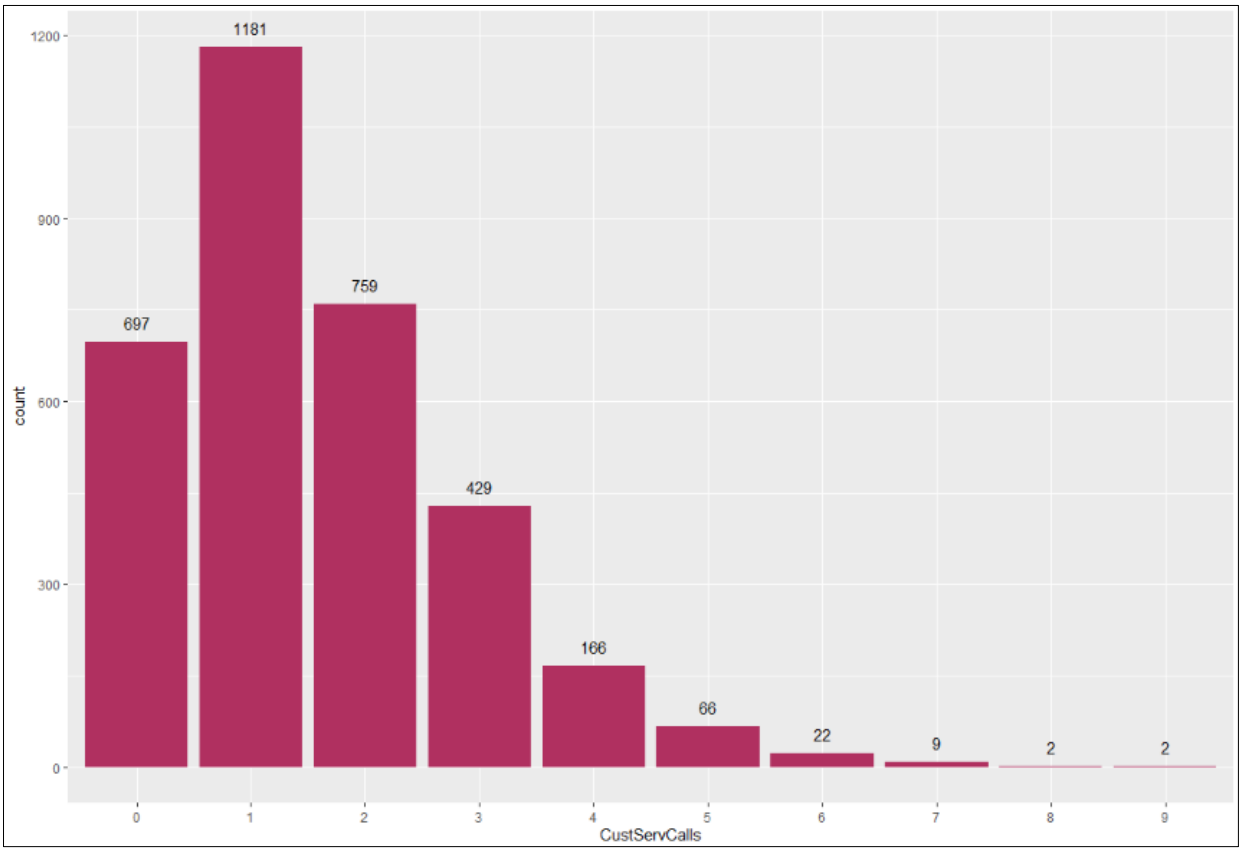
There are 3010 Contract renewal out of 3333 which comes out that only 9.6 % did not renew the contract.

Barchart of Contract Renewal:



2411 customers did not have Data plan for the postpaid service hence majority of the customers fall without data plan.

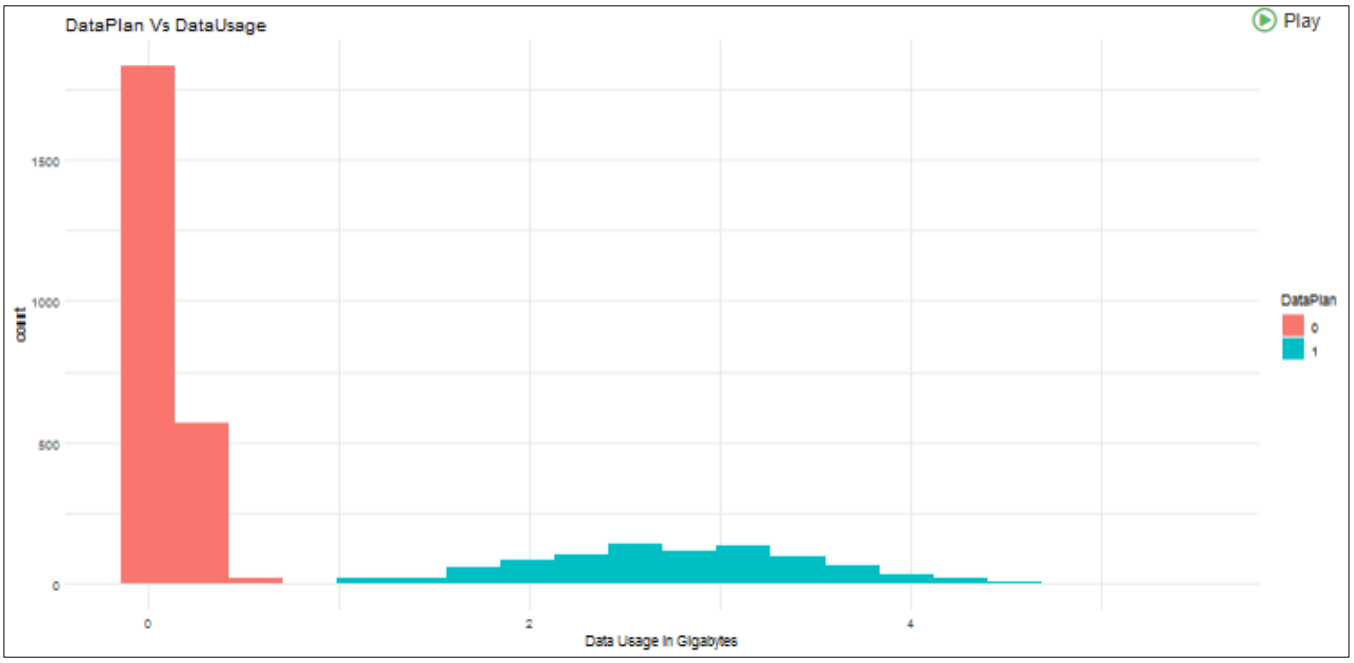
Barchart of Customer Service Calls:



We could see that the number of customers who reached customer service once is the highest (1181) followed by call of two times and zero times

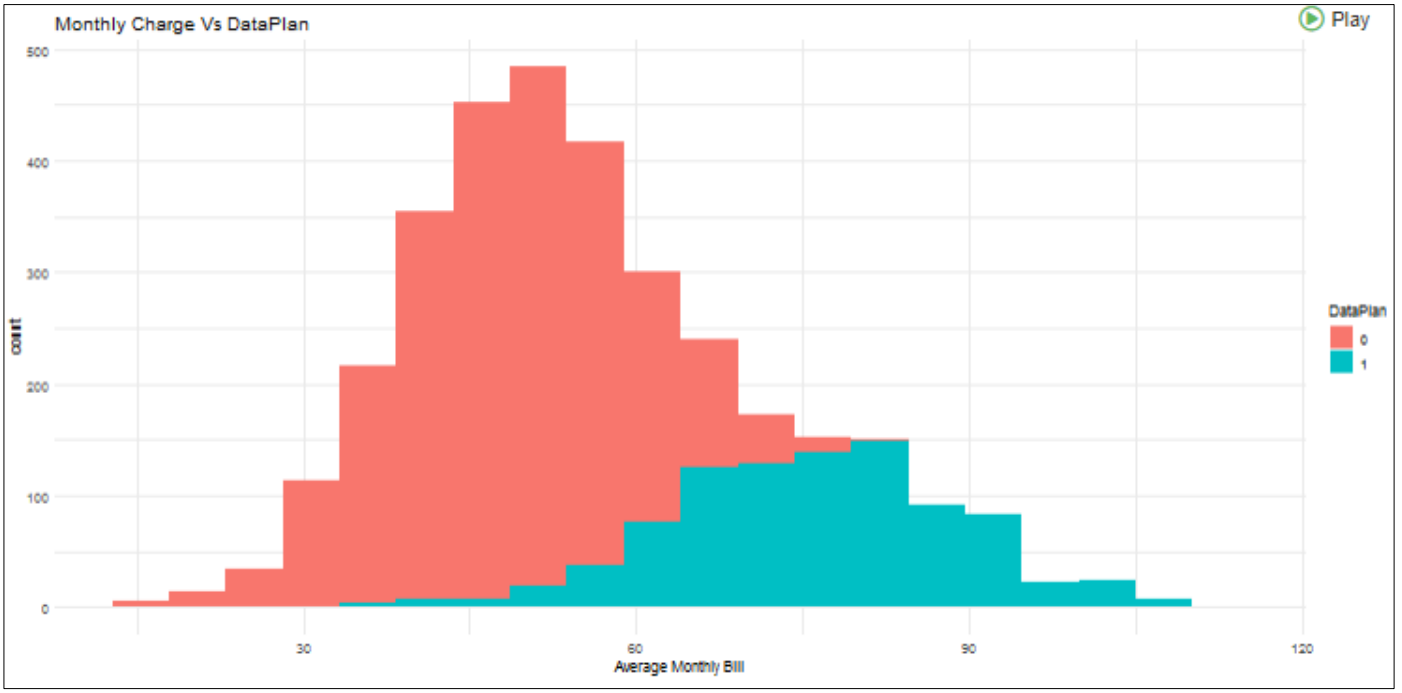
Bivariate Analysis:

Data Plan Vs Data Usage:



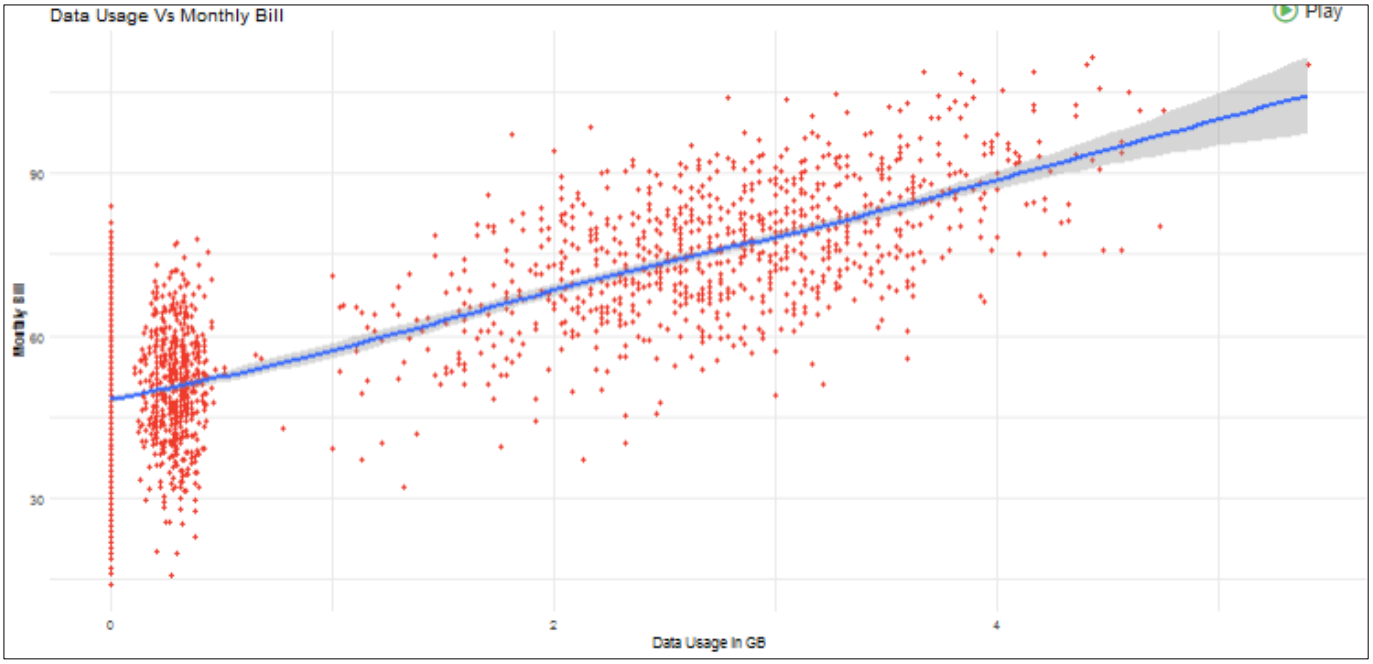
Since majority of the customers are without data plan (2100 customers), we could see that their data usage is below 1 GB usage. While those customers with data plan have normal distribution between 1 GB data usage to 5 GB data and maximum count of 300 customers.

Monthly Charge Vs Data Plan :



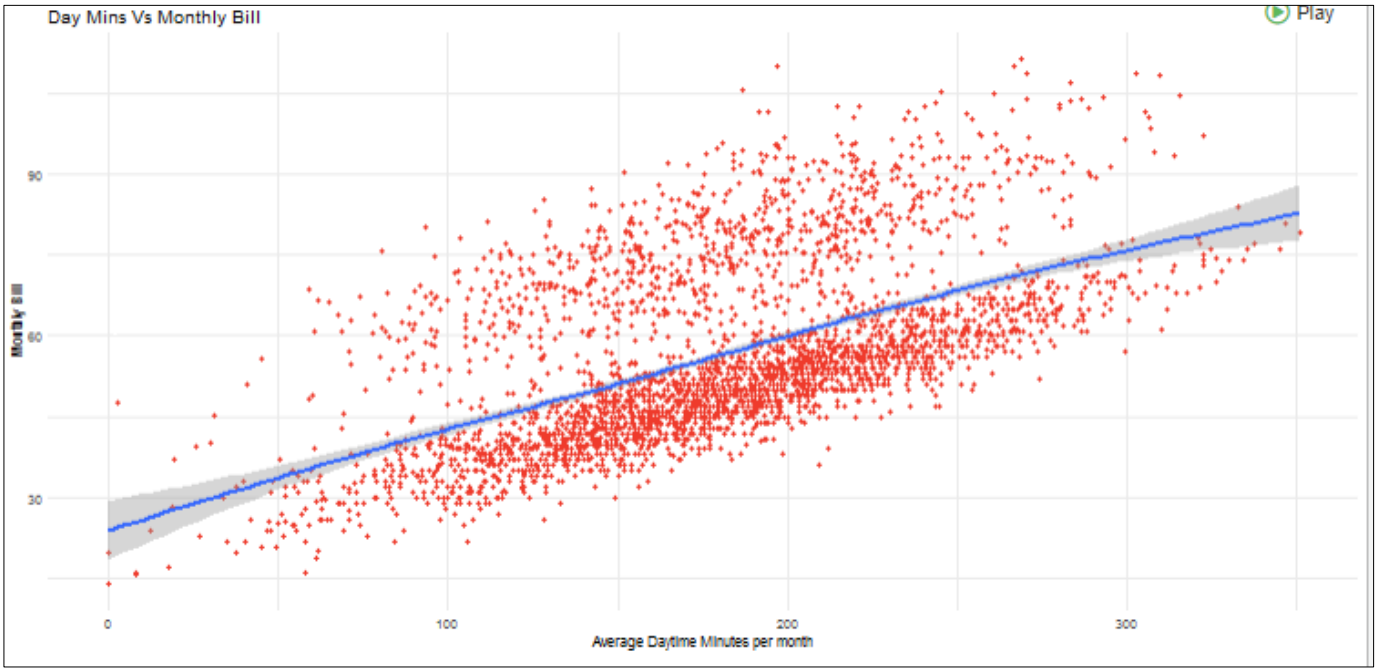
We could see that the monthly bill is high for customers with Dataplan compared to users with non data – plan. The average monthly bill is between 60 – 115 Rupees for data plan users

Data Usage Vs Monthly Bill :



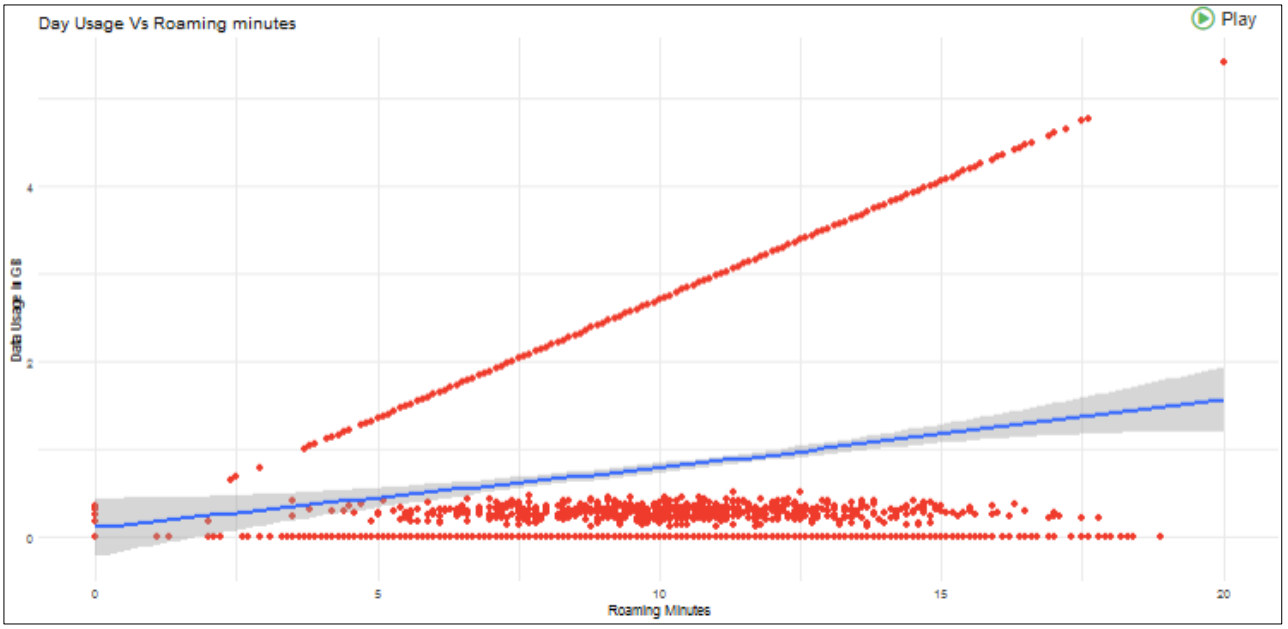
We can see that the as the data usage is increasing, the monthly bill is also increasing with strong linear relationship. From the scatter we can see that there are increasing monthly bill even for non – data plan users which are due to other charges due to roaming, call time etc.

Day Mins Vs Monthly Bill :



These above two variables has the highest correlation among all, we could see that as the Average daytime minutes of talk time increases the monthly bill also increases in linear fashion as depicted above.

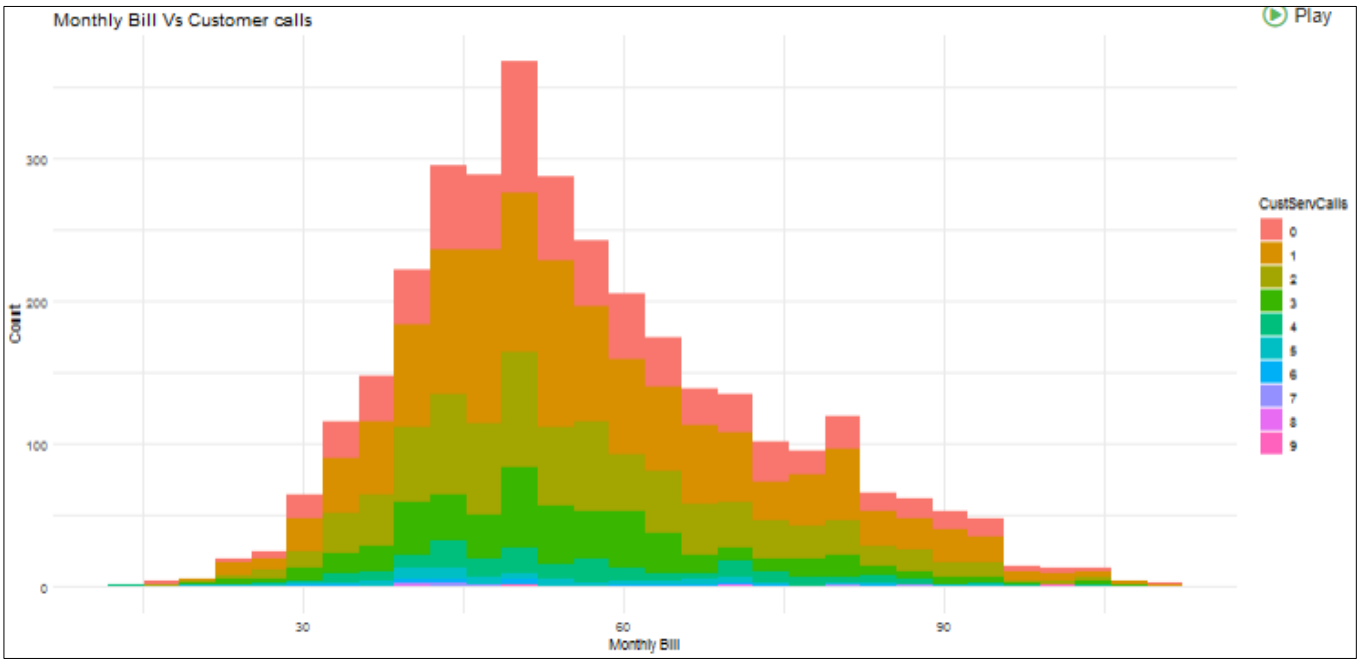
Data Usage Vs Roaming minutes:



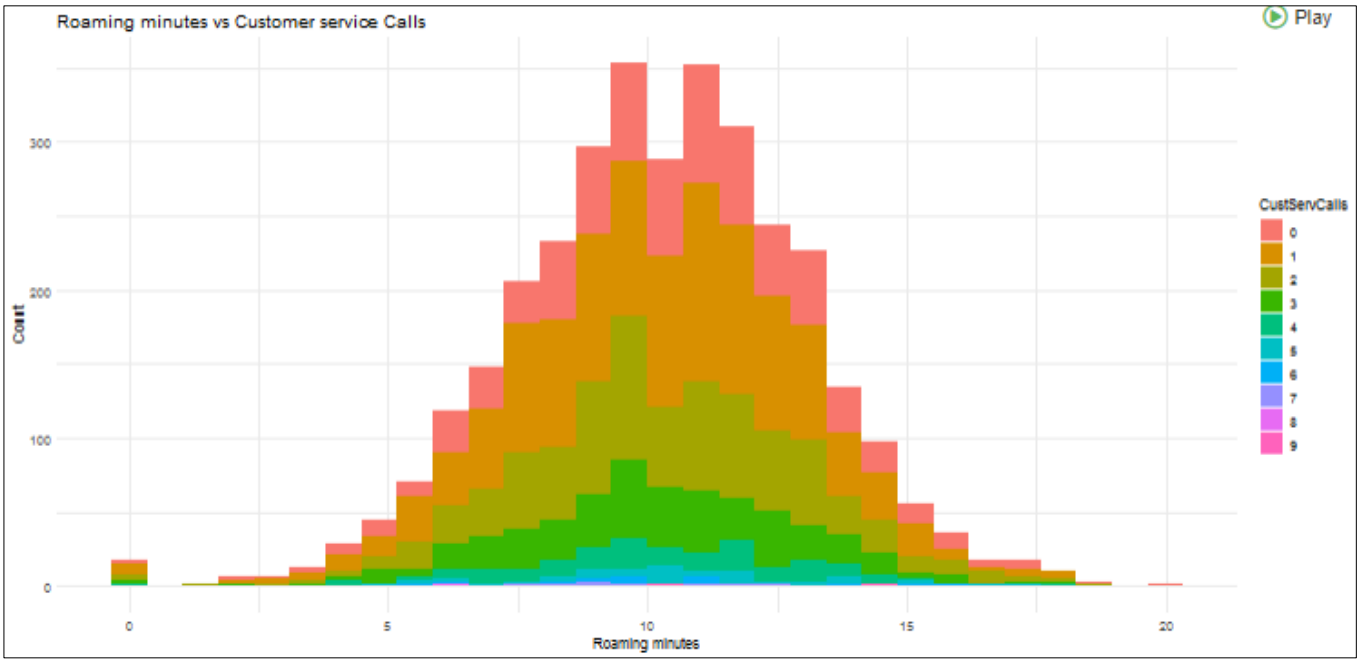
This shows that for data plan users as the roaming minutes increases the data usage for the users are also highly correlated and increases with high slope. This shows that these type of customers are frequent travelers who not only use more roaming talk time but also data usage equivalent to the roaming minutes.

Monthly Bill Vs Customer Service Calls:

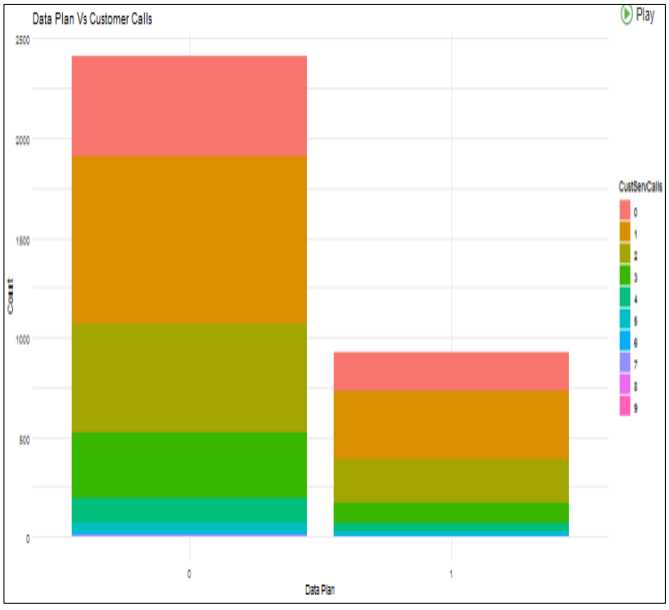
We could see that the customers with more than monthly bill of 50 have more than one customer calls.

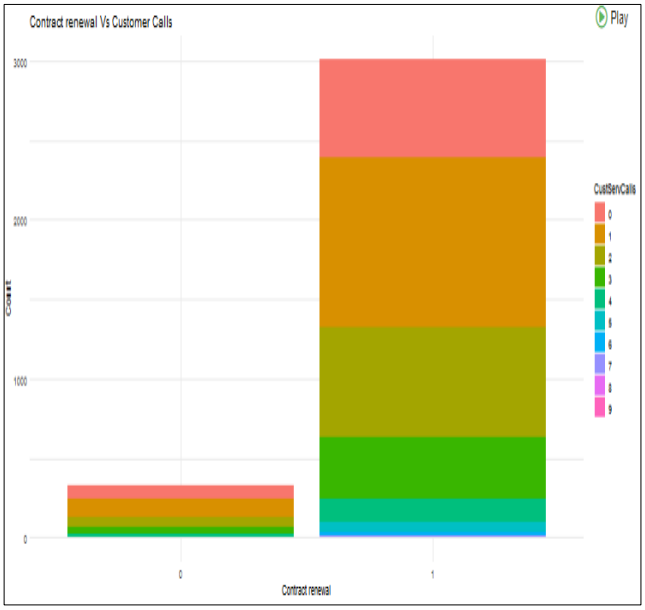


Roaming minutes Vs Customer Service Calls:



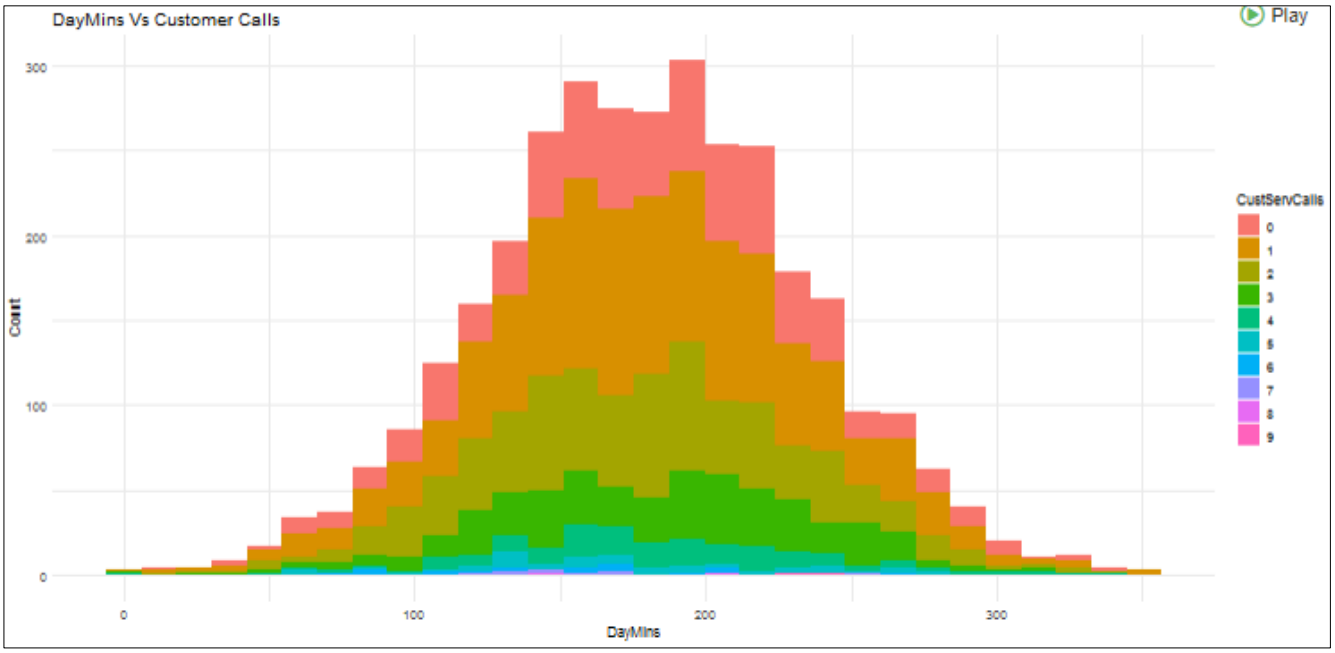
Data Plan & Contract renewal vs Customer Service Calls





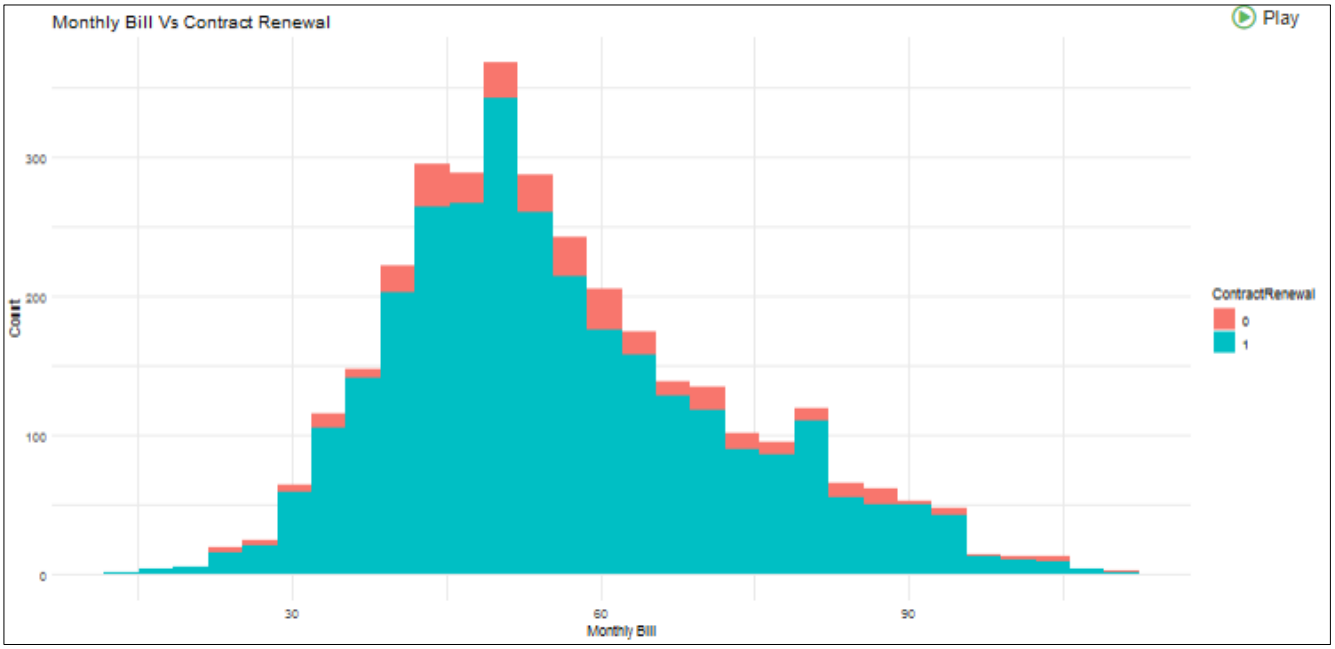
Customers without data plan has more customer calls to the call center compared to data plan users. Also those who renewed contract have more number of customer calls of more than one calls.

Day Mins Vs Customer Service Calls

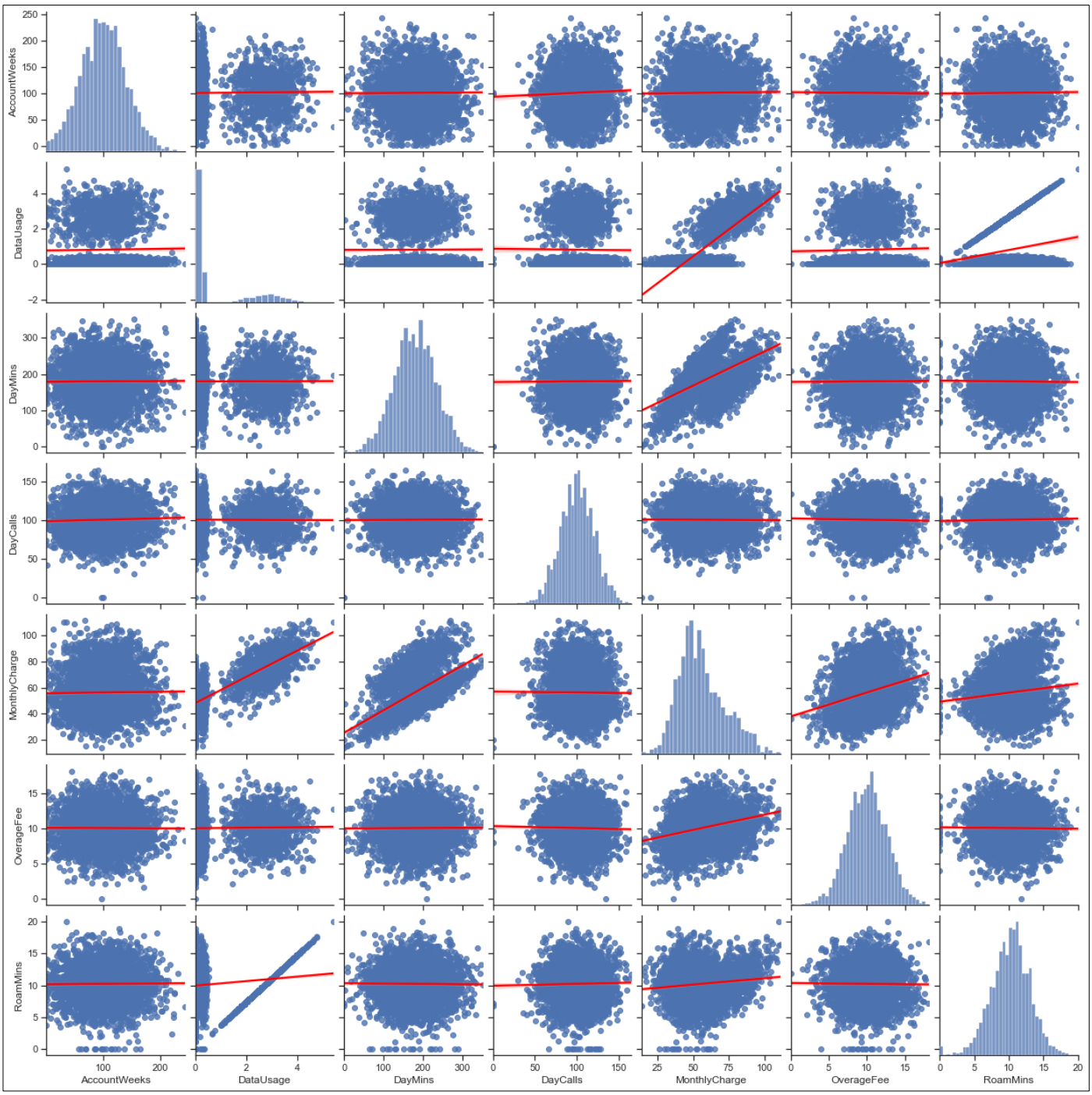


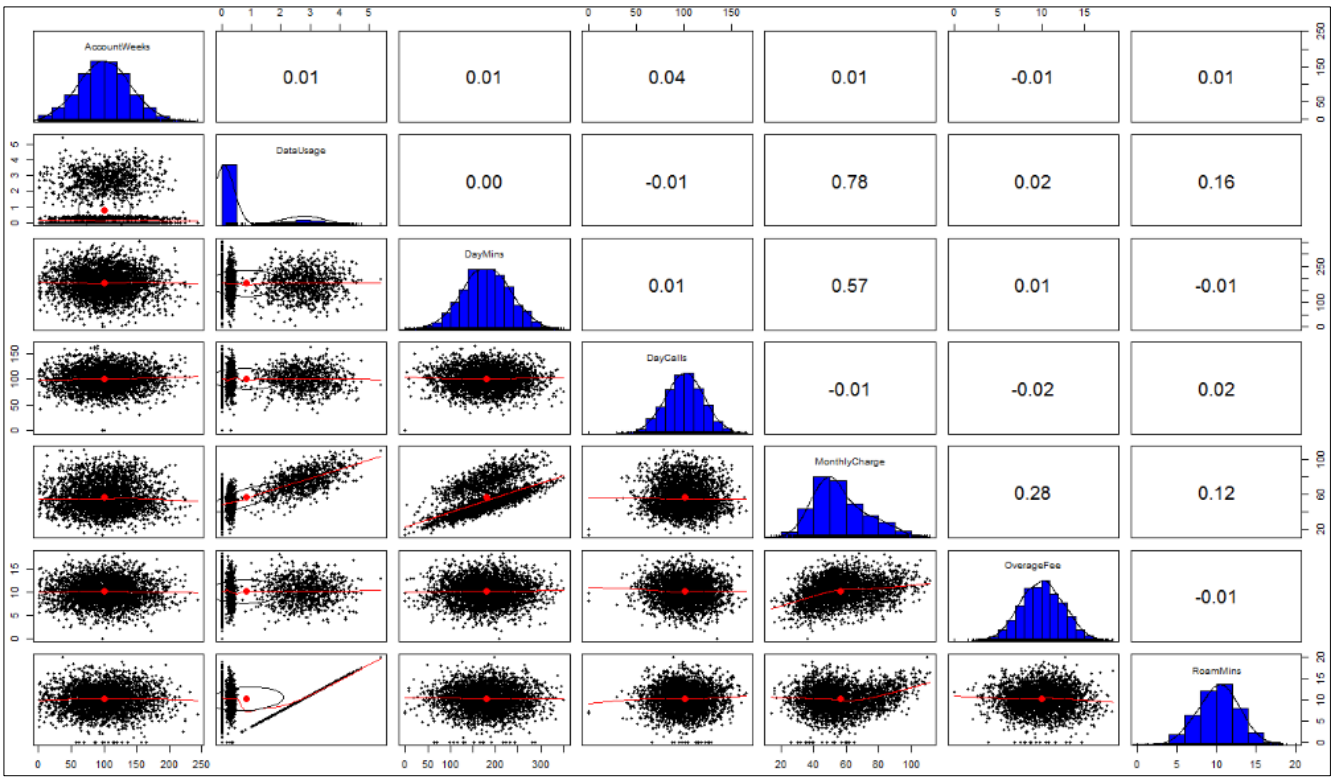
Maximum customer service calls is between the ranges of 150 to 250 Day call minutes.

Monthly Bill Vs Contract Renewal



Multivariate Analysis for continuous variable:

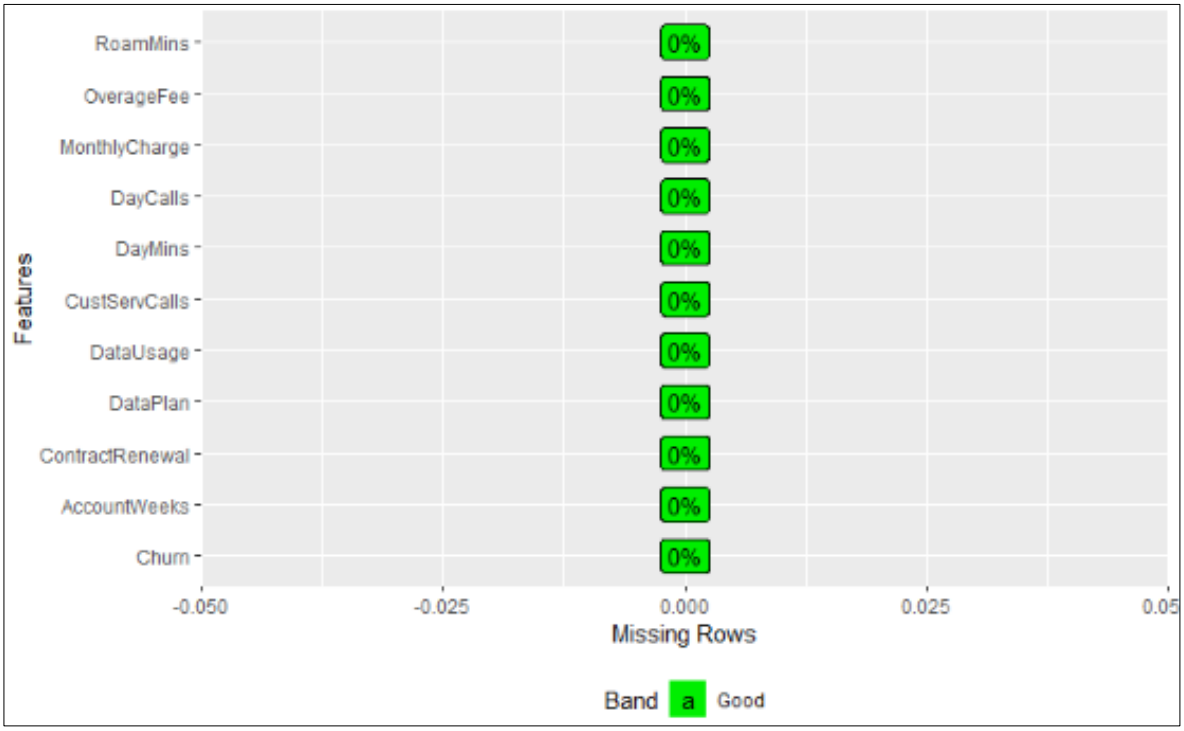




2 EDA - Check for Outliers and missing values and check the summary of the dataset

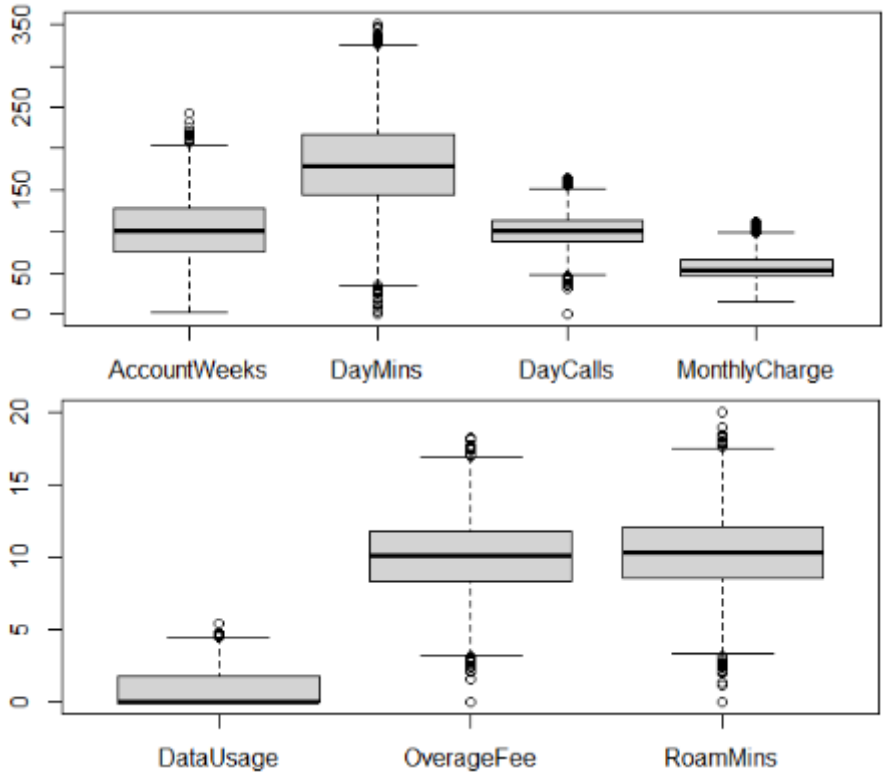
Missing Values:

Next we are checking for any missing or null values in the dataset using the Data Explorer package, we could see that there are no null or missing values in the data and the dataset is fine for model building.



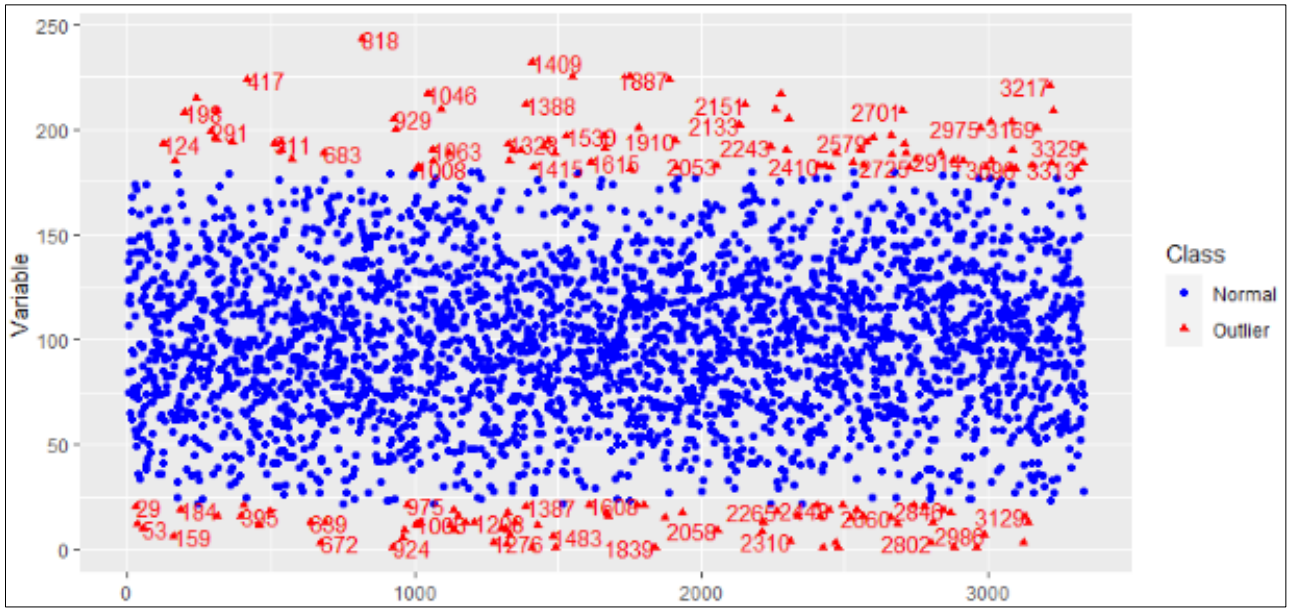
Outlier Check and Corrections:

Next we are checking for outliers using the boxplot and OutlierDetection package for each of the continuous variable. We could see that the account weeks , Daymins, DayCalls, Datausage, Monthly charge, Overage fee and RoamMins have some outliers hence using outlier detection we are plotting the outliers and instead of removing the outlier's we are limiting the extreme value or which is above 99% quantile to be within the range.

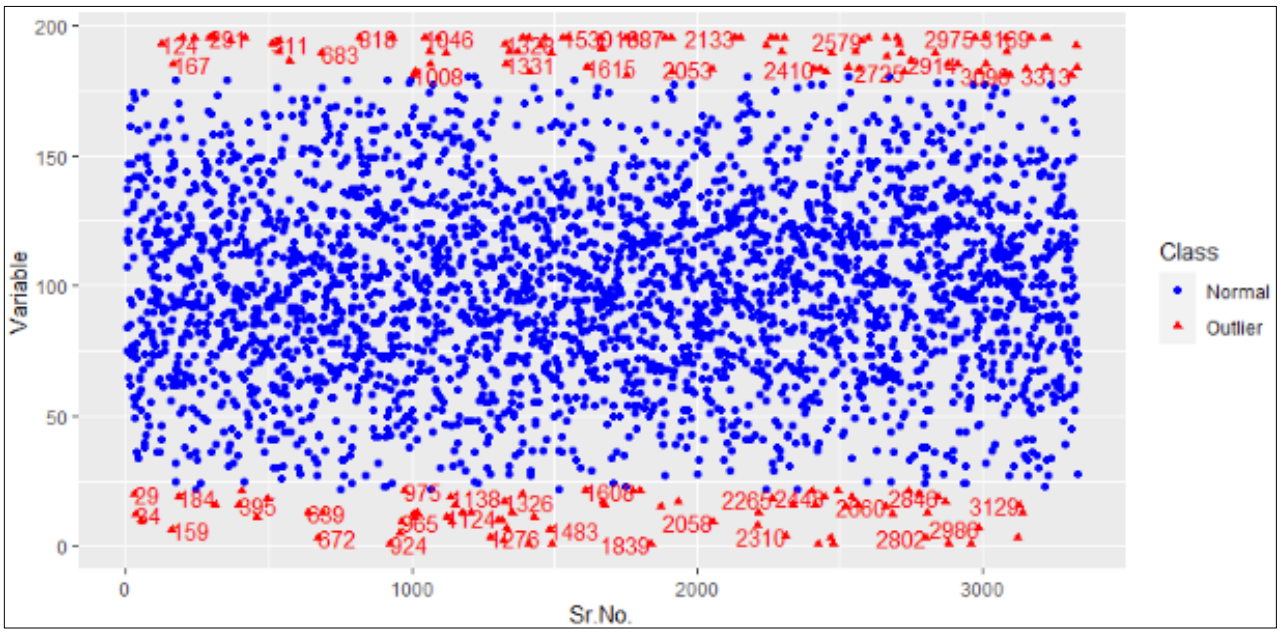


Outlier correction for AccountWeeks :

Before



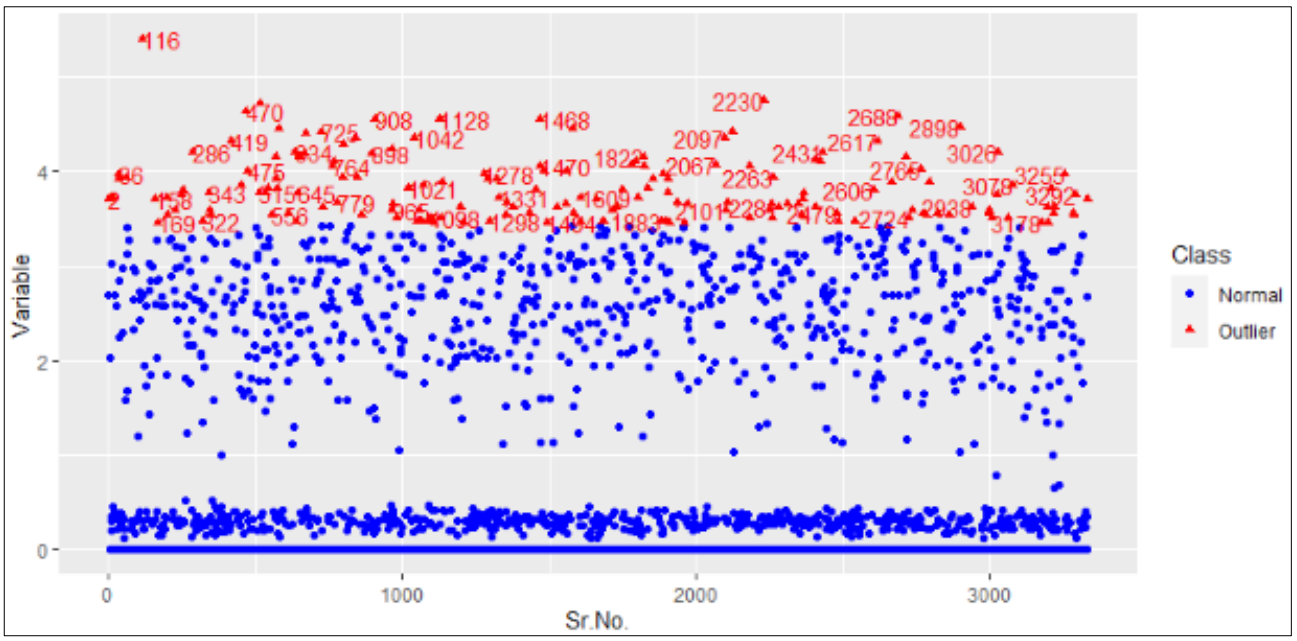

```
Telecom$AccountWeeks[which(Telecom$AccountWeeks > 195)] <- 195
```



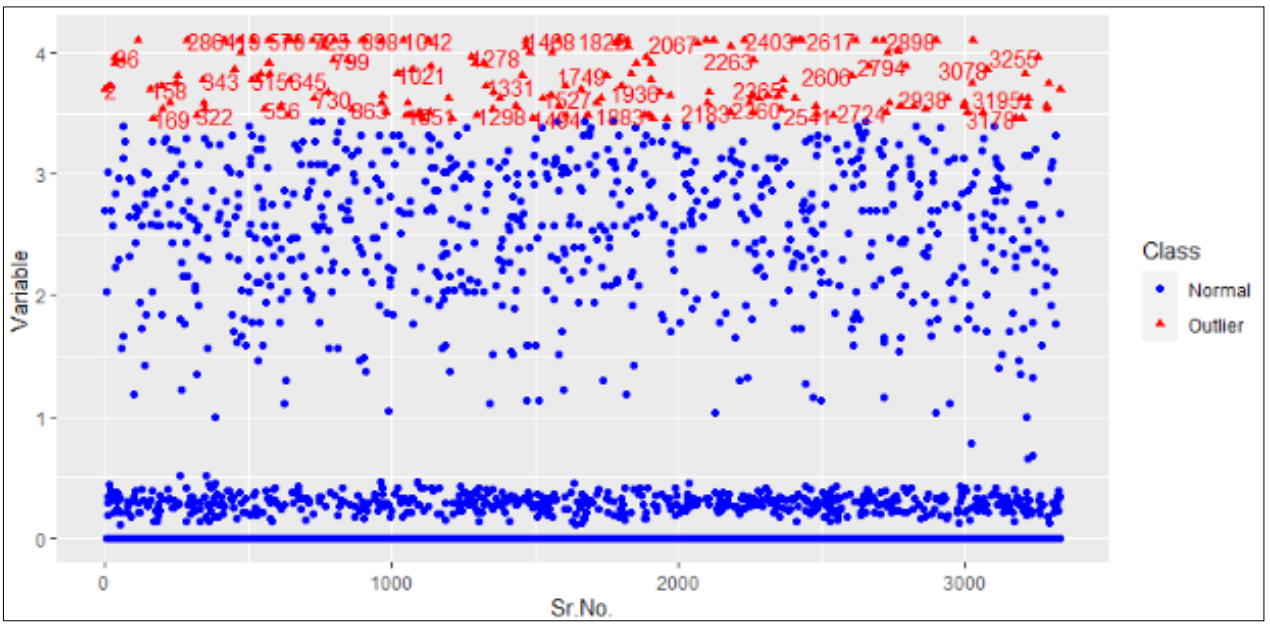
After

Outlier correction for Data Usage:

Before



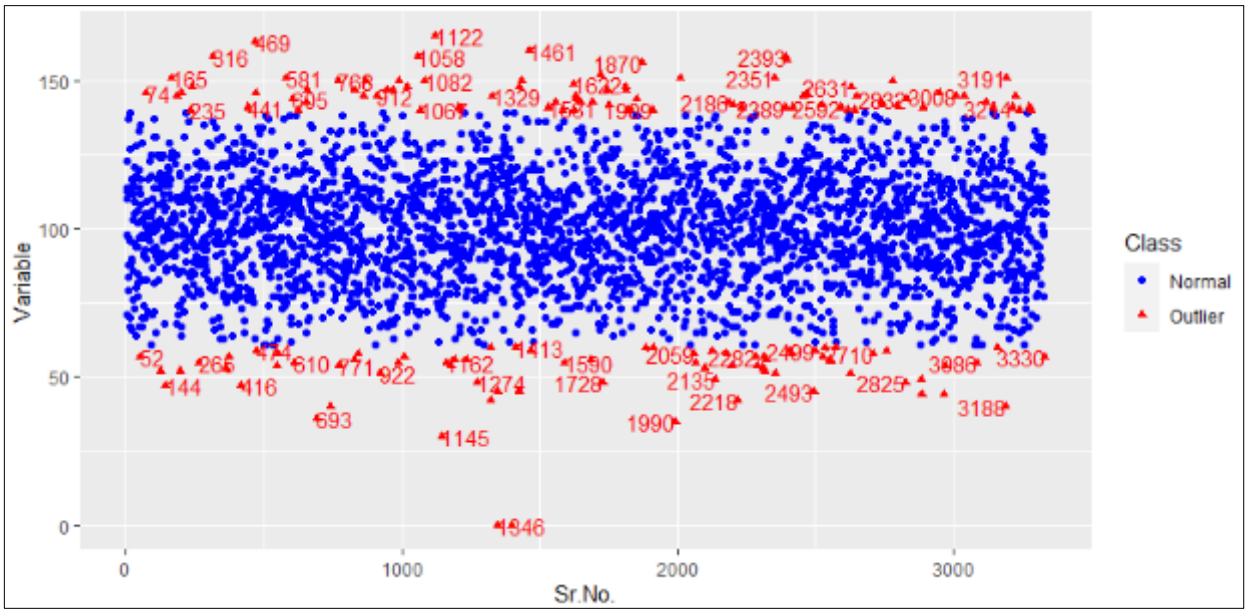

```
Telecom$DataUsage[which(Telecom$DataUsage > 4.10)] <- 4.10
```



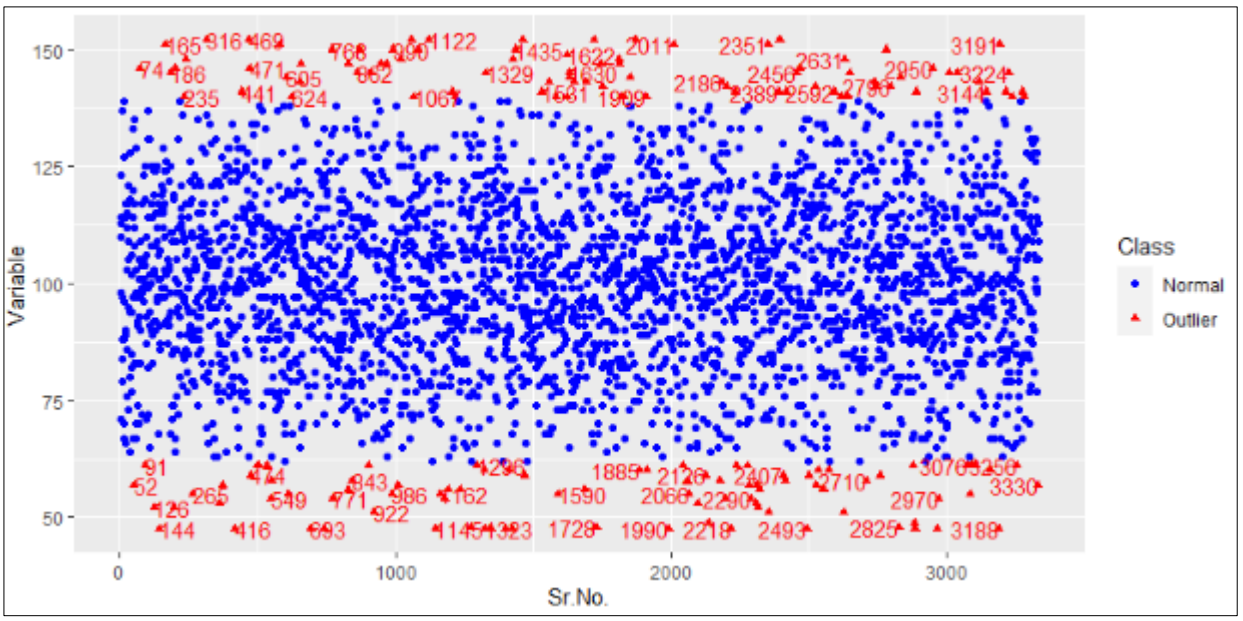
After

Outlier correction for Day Calls:

Before



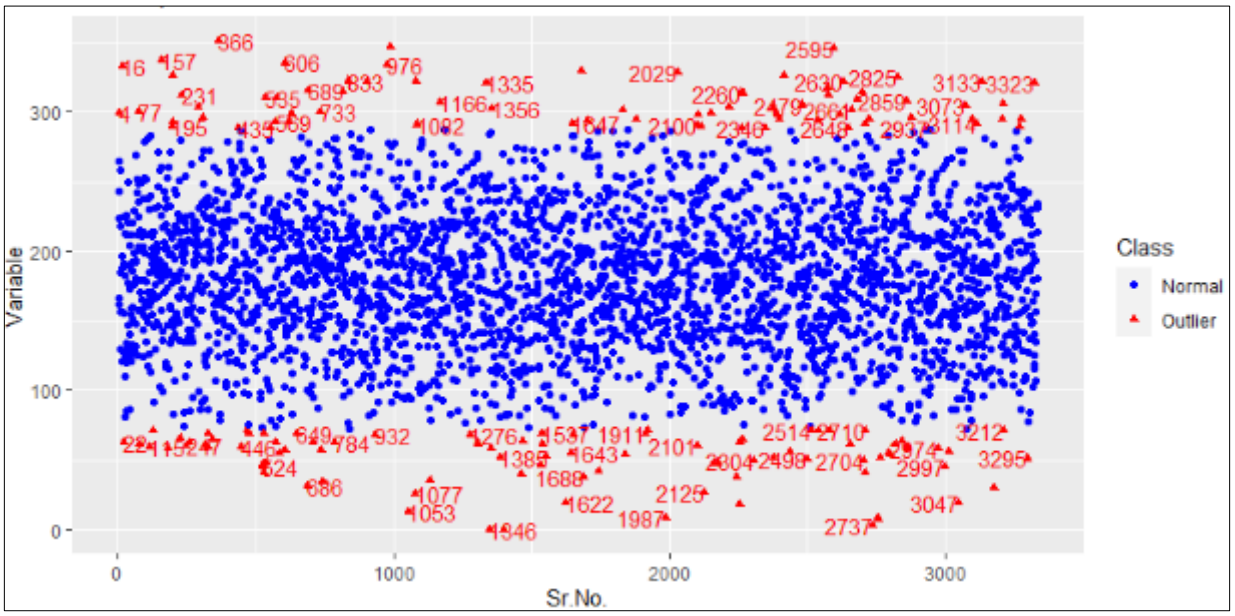

```
Telecom$DayCalls[which(Telecom$DayCalls > 152)] <- 152
Telecom$DayCalls[which(Telecom$DayCalls < 47.66)] <- 47.66
```



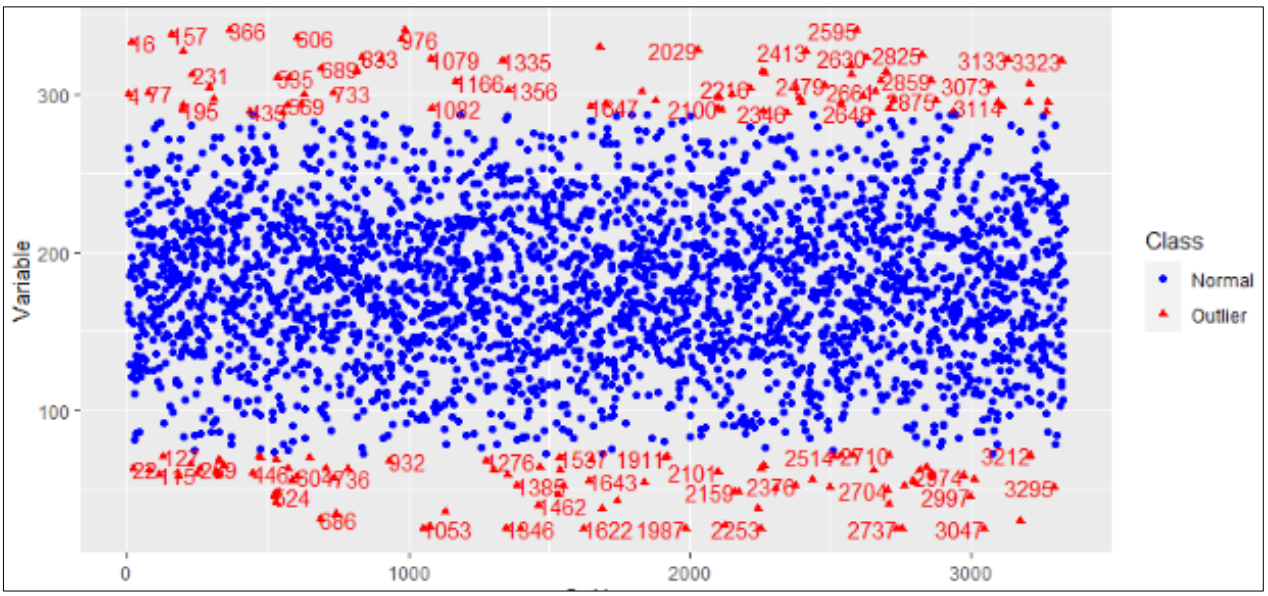
After

Outlier correction for Day Mins:

Before



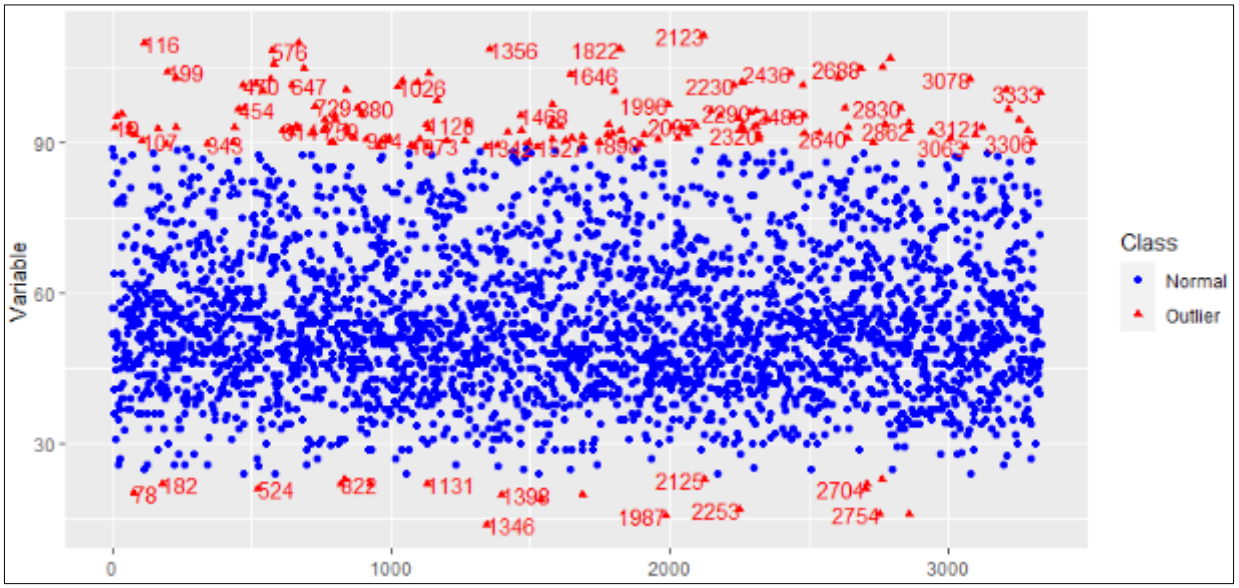

```
Telecom$DayMins[which(Telecom$DayMins > 340)] <- 340
Telecom$DayMins[which(Telecom$DayMins < 25 )] <- 25
```



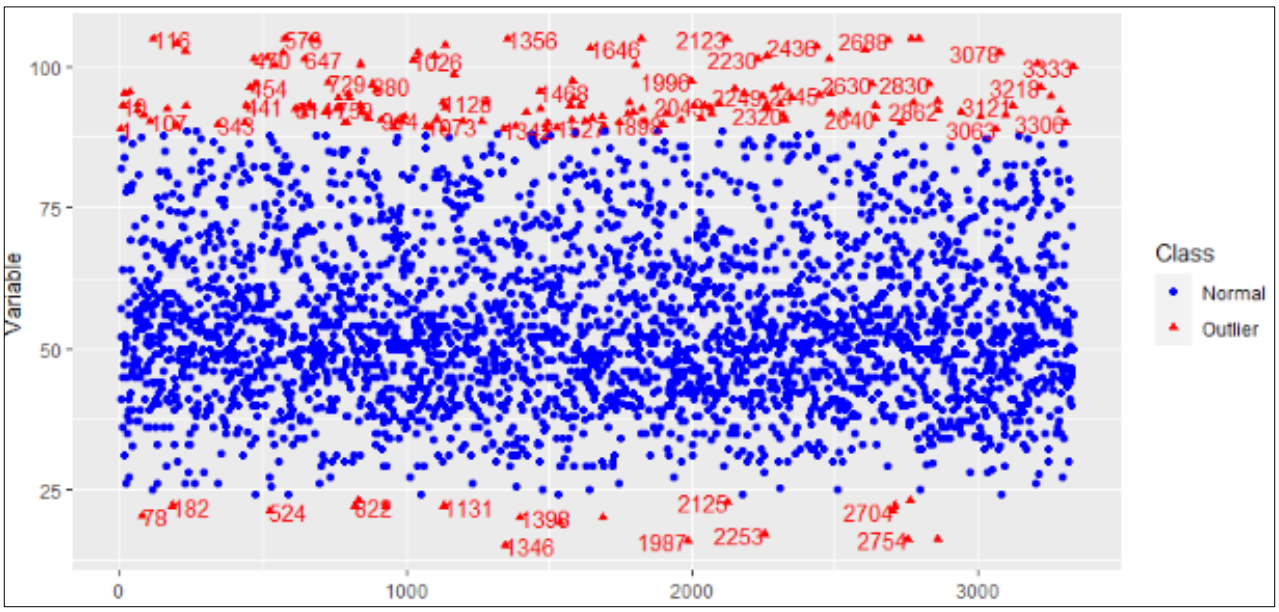
After

Outlier correction for Monthly Charge:

Before



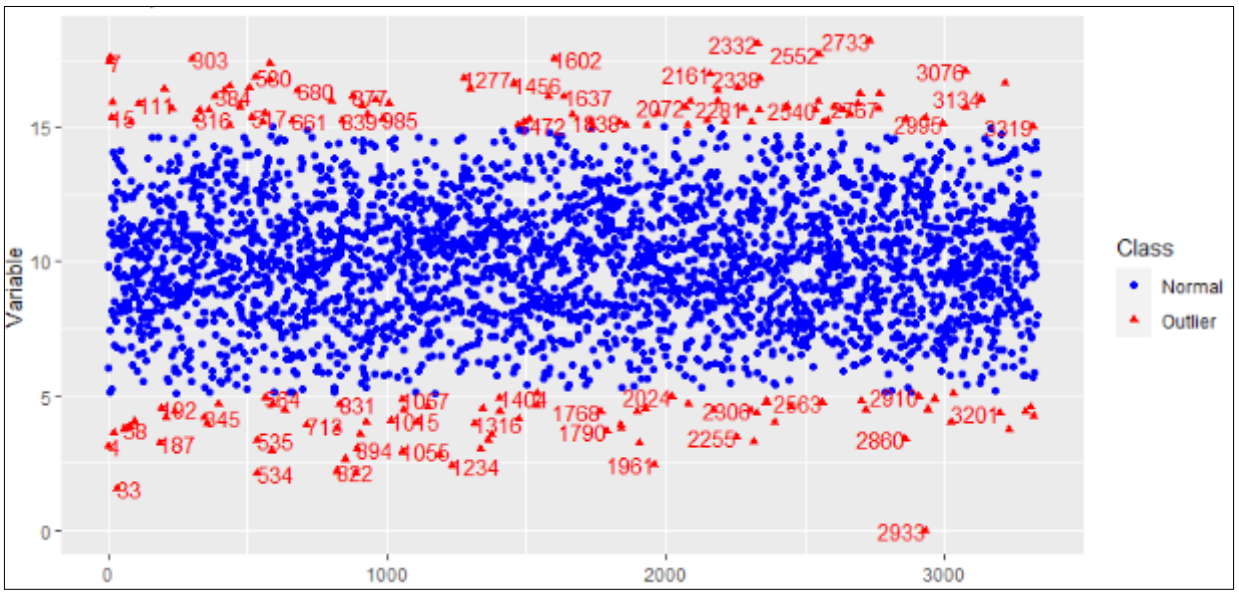

```
Telecom$MonthlyCharge[which(Telecom$MonthlyCharge > 105)] <- 105
Telecom$MonthlyCharge[which(Telecom$MonthlyCharge < 15 )] <- 15
```



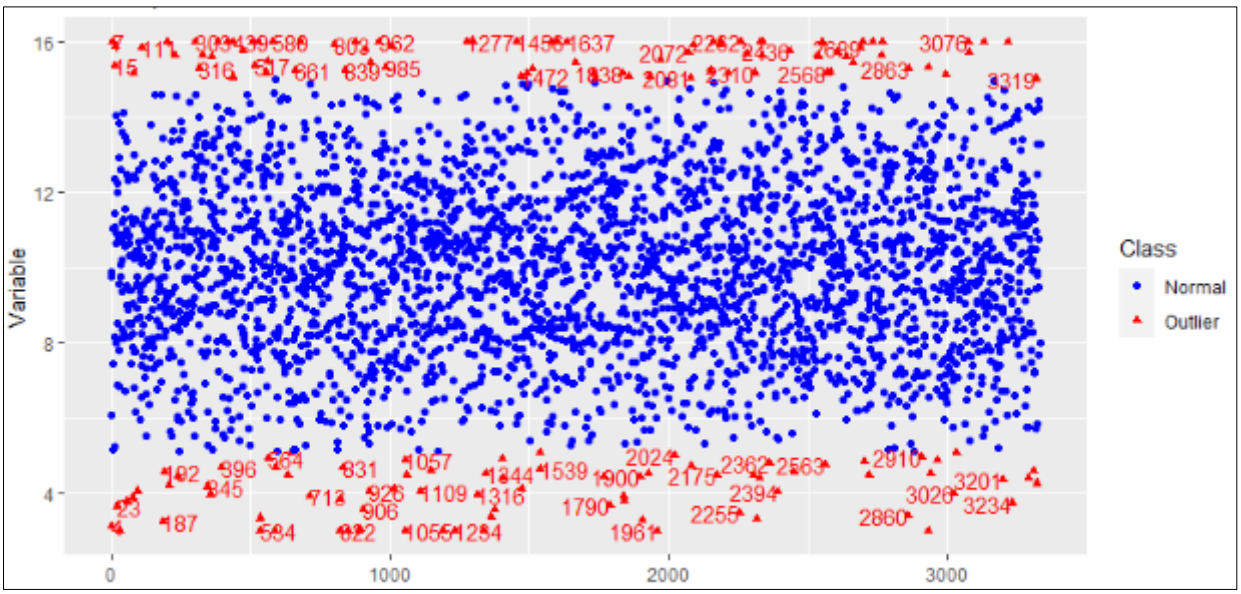
After

Outlier correction for Overage Fee:

Before

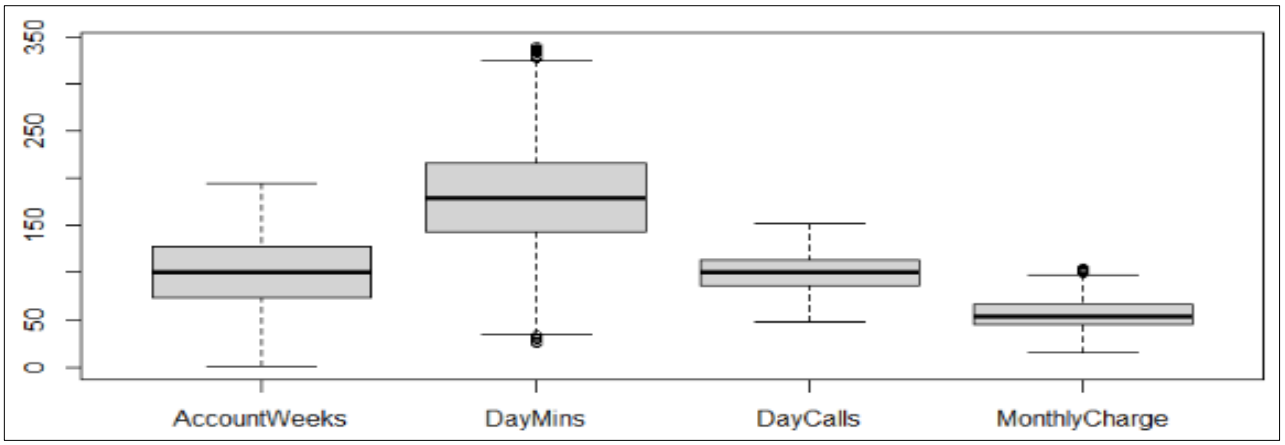


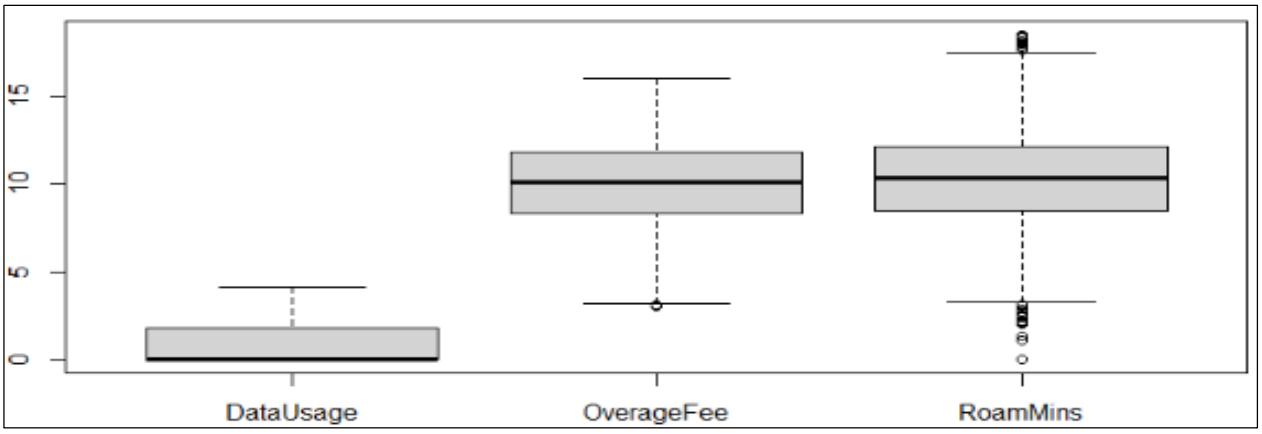

```
Telecom$OverageFee[which(Telecom$OverageFee > 16)] <- 16
Telecom$OverageFee[which(Telecom$OverageFee < 3 )] <- 3
```



After

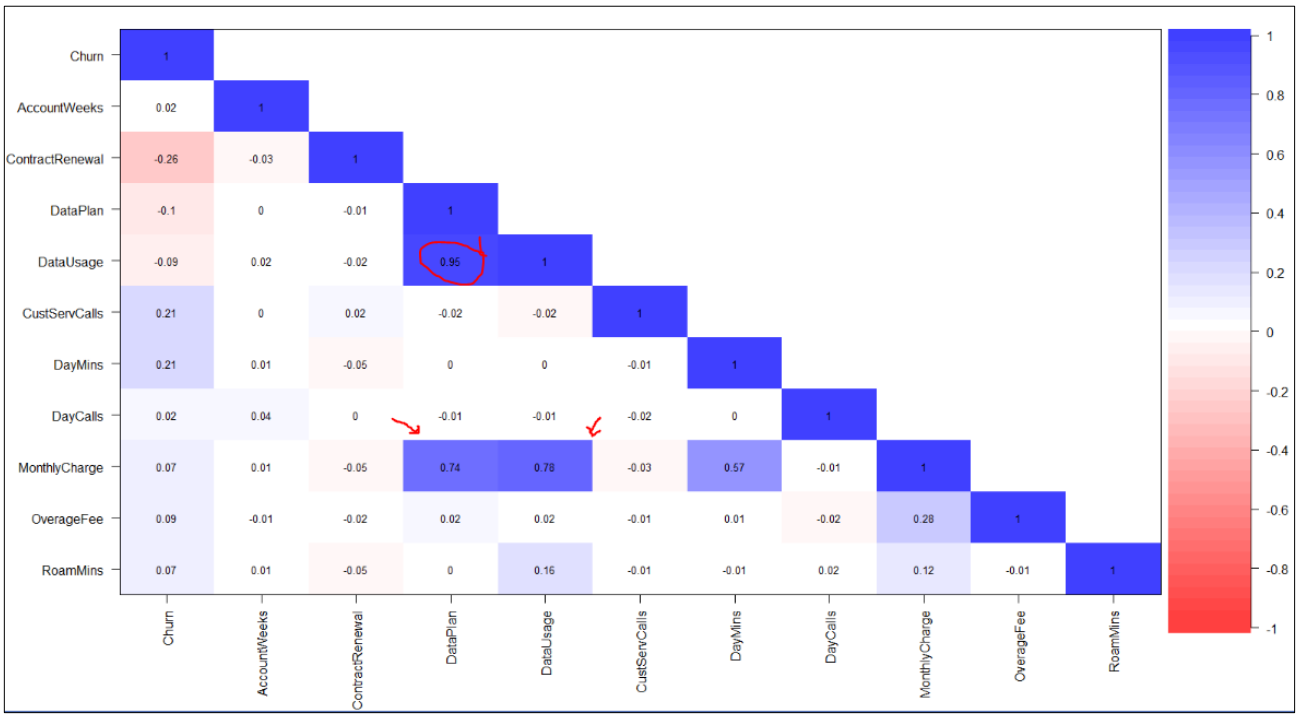
Once the outlier has been limited and imputed with the 99% quantile values below is the boxplot of the variables with the outliers replaced or limited. The Roaming minutes have very few outliers hence we are making minor outlier correction.

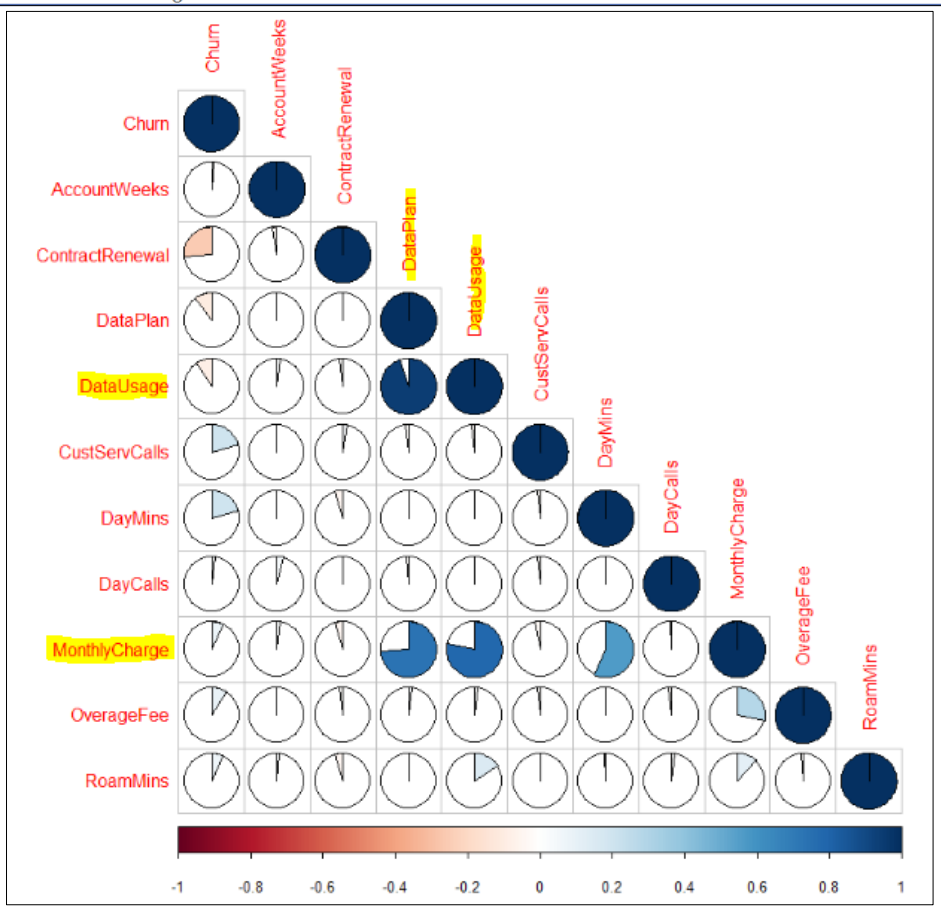




3 Check for Multicollinearity - Plot the graph based on Multicollinearity & treat it

First we are creating a correlation plot with all the variables for multicollinearity. We could see from the below plot that DataPlan and DataUsage is highly correlated. Next we could see that the MonthlyCharge is correlated with DataPlan and Data Usage. Hence we need to further analyze the three variables i.e MonthlyCharge , DataPlan and DataUsage.





Variance Inflation factor calculation :

For a given predictor (p), multicollinearity can be assessed by computing a score called the **variance inflation factor** (or **VIF**), which measures how much the variance of a regression coefficient is inflated due to multicollinearity in the model.
The smallest possible value of VIF is one (absence of multicollinearity). As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity.

Below are the VIF score calculated for the predictor variables by running a test regression model with all the variables and using car package to calculate the vif score.

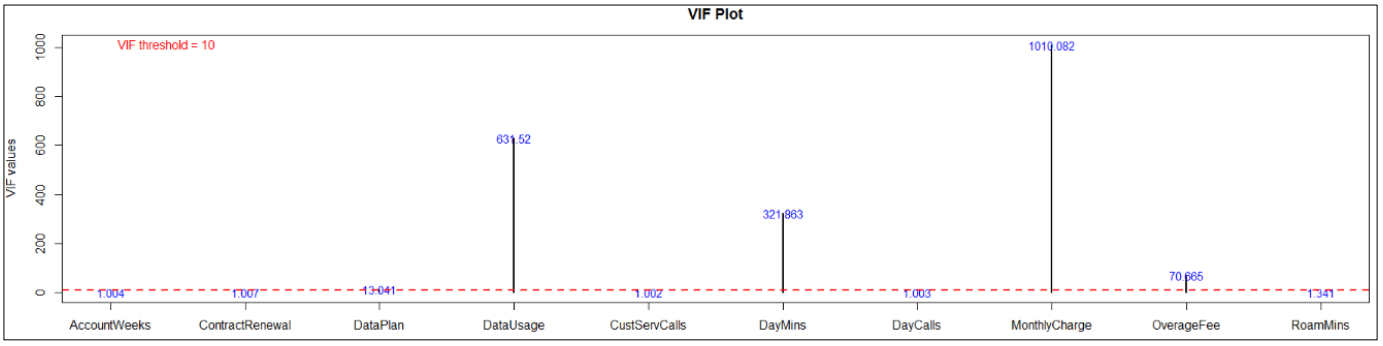
```
logittest <- glm(Telecom$Churn ~ . ,family = 'binomial',data = Telecom)
```

```
AccountWeeks ContractRenewal       DataPlan      DataUsage   CustServCalls        DayMins       DayCalls
    1.003608       1.057710      14.896708     468.055348       1.082030     265.279470       1.005540
 MonthlyCharge   OverageFee      RoamMins
   781.525766     60.286692       1.192551
```

Apart from AccountWeeks, ContractRenewal , CustomerServcalls, DayCalls, RoamMins all other has high vif score. When faced with multicollinearity, the concerned variables should be removed or perform PCA, since the presence of multicollinearity implies that the information that this variable provides about the response is redundant in the presence of the other variables. We are further checking multicollinearity with other tests using mctest package.

MCTEST

MCTEST package provides the overall multicollinearity diagnostic measures which includes determinant of correlation matrix, R-squared from regression of all x's on y, Farrar and Glauber chi-square test for detecting the strength of collinearity over the complete set of regressors, Condition Index, Sum of reciprocal of Eigenvalues, Theil's and Red indicator. The individual multicollinearity diagnostic measures are Klein's rule, variance inflation factor (VIF), Tolerance (TOL), Corrected VIF (CVIF), Leamer's method, F & R^2 relation, Farrar & Glauber F-test, and IND1 & IND2 indicators. The package also indicates which regressors may be the reason of collinearity among regressors.



Multicollinearity Diagnostic:

```
All Individual Multicollinearity Diagnostics Result

                    VIF     TOL           Wi           Fi Leamer       CVIF Klein   IND1   IND2
AccountWeeks     1.0037 0.9964       1.3493       1.5184 0.9982     1.0220     0 0.0027 0.0070
ContractRenewal  1.0067 0.9934       2.4681       2.7775 0.9967     1.0251     0 0.0027 0.0128
DataPlan        13.0407 0.0767    4445.6973    5002.9146 0.2769    13.2789     1 0.0002 1.7848
DataUsage      631.5198 0.0016  232801.9276  261980.9835 0.0398   643.0556     1 0.0000 1.9299
CustServCalls    1.0022 0.9978       0.8289       0.9328 0.9989     1.0206     0 0.0027 0.0043
DayMins        321.8626 0.0031  118469.6161  133318.4260 0.0557   327.7420     1 0.0000 1.9270
DayCalls         1.0029 0.9972       1.0542       1.1863 0.9986     1.0212     0 0.0027 0.0055
MonthlyCharge 1010.0817 0.0010  372575.4029  419273.4635 0.0315  1028.5325     1 0.0000 1.9311
OverageFee       70.6651 0.0142   25721.9160   28945.8637 0.1190    71.9559     1 0.0000 1.9056
RoamMins          1.3414 0.7455     126.0344     141.8313 0.8634     1.3659     1 0.0020 0.4919

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test

AccountWeeks , DataPlan , DayMins , DayCalls , MonthlyCharge , OverageFee , coefficient(s) are
non-significant may be due to multicollinearity

R-square of y on all x: 0.1756
```

```
Overall Multicollinearity Diagnostics

                        MC Results detection
Determinant |X'X|:          0.0001         1
Farrar Chi-Square:      31668.4806         1
Red Indicator:              0.2362         0
Sum of Lambda Inverse:   2052.5268         1
Theil's Method:             3.5926         1
Condition Number:         408.3265         1

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test
```

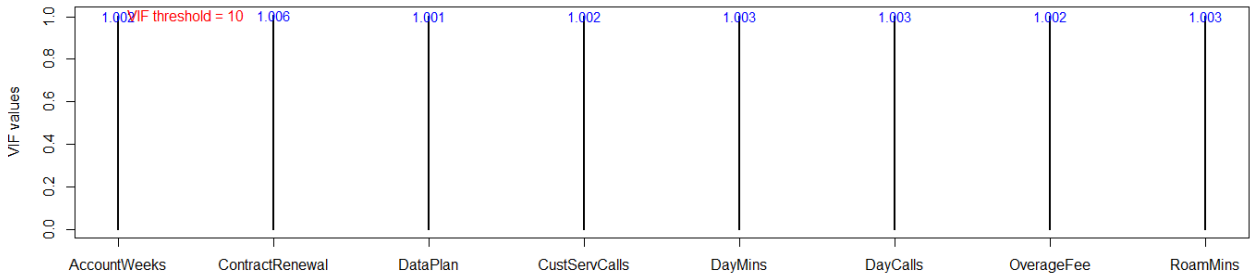
Stepwise iteration to remove multicollinearity :

The multicolliniearity has caused the inflated VIF values for correlated variables, making the model unreliable. We will thus use a stepwise variable reduction custom function using VIF values. The function works like this in the following way: The function uses three arguments. The first is a matrix or data frame of the explanatory variables, the second is the threshold value to use for retaining variables, and the third is a logical argument indicating if text output is returned as the stepwise selection progresses. The output indicates the VIF values for each variable after each stepwise comparison. The function calculates the VIF values for all explanatory variables, removes the variable with the highest value, and repeats until all VIF values are below the threshold. The final output is a list of variable names with VIF values that fall below the threshold. Using the custom function **'vif_fun.R'** available at https://beckmw.wordpress.com/2013/02/05/collinearity-and-stepwise-vif-selection/ we have invoked and ran the function with our dataset. Below is the output generated.

```
var              vif
 AccountWeeks     1.00365448681719
 ContractRenewal  1.00668465685899
 DataPlan         13.0407089443646
 DataUsage        631.519815819631
 CustServCalls    1.00224511450893
 DayMins          321.862637668325
 DayCalls         1.00285523455923
 MonthlyCharge    1010.08174125547
 OverageFee       70.665135244838
 RoamMins         1.34135097211736

removed:  MonthlyCharge 1010.082

 var              vif
 AccountWeeks     1.00363683026502
 ContractRenewal  1.00635211123941
 DataPlan         12.9659668649587
 DataUsage        13.3021947329112
 CustServCalls    1.00171121796624
 DayMins          1.00331091345461
 DayCalls         1.00271943780344
 OverageFee       1.00159923007229
 RoamMins         1.33995392418384

removed:  DataUsage 13.30219

"AccountWeeks"    "ContractRenewal" "DataPlan"
"CustServCalls"   "DayMins"         "DayCalls"
"OverageFee"      "RoamMins"
```

We could see that the iteration has removed the variables 'MonthlyCharge' and 'DataUsage' which led to multicollinearity among the dataset hence we have to perform regression analysis with these two variables removed. We are checking the multicollinearity again by performing test logistic regression without 'MonthlyCharge' and 'DataUsage' with rest others included. The test model is again run with mctest package

Running the MCtest again after removing MonthlyCharge and DataUsage:



All Individual Multicollinearity Diagnostics Result

	VIF	TOL	Wi	Fi	Leamer	CVIF	Klein	IND1	IND2
AccountWeeks	1.0023	0.9977	1.0835	1.2645	0.9989	1.0058	0	0.0021	0.8702
ContractRenewal	1.0061	0.9940	2.8784	3.3592	0.9970	1.0096	0	0.0021	2.3030
DataPlan	1.0010	0.9991	0.4514	0.5268	0.9995	1.0045	0	0.0021	0.3630
CustServCalls	1.0017	0.9983	0.7893	0.9211	0.9992	1.0052	0	0.0021	0.6343
DayMins	1.0028	0.9972	1.3492	1.5745	0.9986	1.0064	0	0.0021	1.0829
DayCalls	1.0027	0.9973	1.2915	1.5072	0.9986	1.0062	0	0.0021	1.0367
OverageFee	1.0016	0.9984	0.7550	0.8811	0.9992	1.0051	0	0.0021	0.6068
RoamMins	1.0029	0.9971	1.3744	1.6039	0.9986	1.0064	0	0.0021	1.1031

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test

AccountWeeks , DayCalls , coefficient(s) are non-significant may be due to multicollinearity

R-square of y on all x: 0.1745

Overall Multicollinearity Diagnostics

	MC Results	detection
Determinant |X'X|:	0.9895	0
Farrar Chi-Square:	35.0504	0
Red Indicator:	0.0195	0
Sum of Lambda Inverse:	8.0210	0
Theil's Method:	-1.2003	0
Condition Number:	28.1326	0

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test

Upon removing the 'MonthlyCharge' and 'DataUsage' variables the multicollinearity no longer exist in the model and the data is now best suitable to perform logistic regression and other analysis.

4 Apply & Interpret Logistic Regression

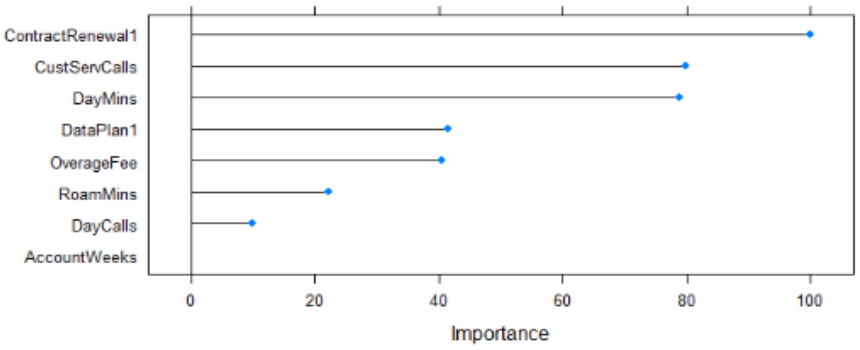
After identifying the variable causing the multicollinearity , we are removing those two variables in earlier section and performing logistic regression with the below formula on the train dataset.

```
Churn ~ AccountWeeks + ContractRenewal + DataPlan + CustServCalls + DayMins +
DayCalls + OverageFee + RoamMins
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9824  -0.5105  -0.3582  -0.2215   2.9770

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -6.020843   0.650627  -9.254  < 2e-16 ***
AccountWeeks    0.001122   0.001692   0.663  0.50712
ContractRenewal1 -2.040869   0.169207 -12.061  < 2e-16 ***
DataPlan1      -0.918583   0.170182  -5.398 6.75e-08 ***
CustServCalls   0.456878   0.046739   9.775  < 2e-16 ***
DayMins         0.012229   0.001266   9.658  < 2e-16 ***
DayCalls        0.005800   0.003258   1.780  0.07502 .
OverageFee      0.145610   0.027551   5.285 1.26e-07 ***
RoamMins        0.076535   0.023908   3.201  0.00137 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1934.3  on 2333  degrees of freedom
Residual deviance: 1547.6  on 2325  degrees of freedom
AIC: 1565.6

Number of Fisher Scoring iterations: 5
```

The output of the logistic regression is shown above, the variable importance plot for the above built logistic regression model is shown below.

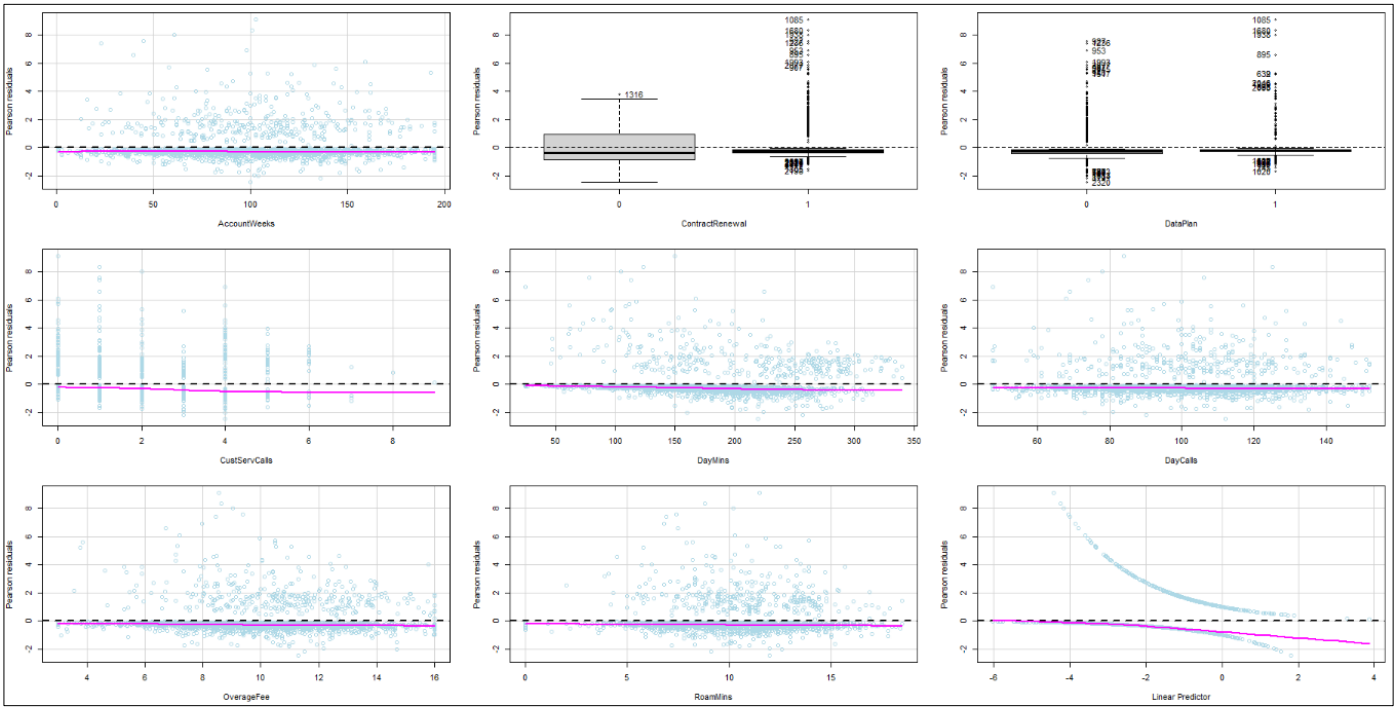
```
> caret::varImp(lgisticfinal)
                 Overall
ContractRenewal1 12.061339
CustServCalls     9.775182
DayMins           9.658149
DataPlan1         5.397660
OverageFee        5.285164
RoamMins          3.201207
DayCalls          1.780357
AccountWeeks      0.663322
```



We could see that the logistic regression model built shows 3 variables are highly significant in predicting the Churn, ContractRenewal of value 1 (If customer recently renewed), CustServCalls & DayMins , followed by these three variables OverageFee , dataplan and Roaming minutes are significant. The least significant are the Account weeks and day calls. Also, note the AIC score is 1565.6. The model having least AIC Score would be the most preferred and optimized one.

AIC: The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models.

Next we are checking the residual plots for the predictor variables using car package and below are the plots for residuals.



Next we will check the significance of the model built by using Log likelihood ratio using lrtest function.

```
Likelihood ratio test

Model 1: Churn ~ AccountWeeks + ContractRenewal + DataPlan + CustServCalls +
    DayMins + DayCalls + OverageFee + RoamMins
Model 2: Churn ~ 1
  #Df  LogLik Df Chisq Pr(>Chisq)
1   9 -773.79
2   1 -967.14 -8 386.7  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation of Log Likelihood ratio test: This test is based out of two hypothesis as shown below

H0: All betas are zero
H1: At least 1 beta is nonzero

From the log likelihood output, we can see that the intercept of model 2 is -967.14 (i.e model with itself)
When we take the full model, -773.79 variance was unknown to us. So we can say that, 1 - (-773.79 /-967.14)= 19.99% of the uncertainty inherent in the intercept only model is calibrated by the full model. Chisq likelihood ratio is significant. Also the p value suggests that we can accept the Alternate Hypothesis that at least one of the beta is not zero. So Model is significant.

Mcfadden's R squared Test : Next we check whether our model robustness or by using Mcfadden's pseudo R squared Test and below is the output.

```
fitting null model for pseudo-r2
          llh       llhNull            G2      McFadden          r2ML          r2CU
-773.7890355 -967.1400908  386.7021105     0.1999204     0.1526845     0.2710049
```

The McFadden's pseudo-R Squared test suggests that atleast 19.99% (Approx 20%) variance of the data is captured by our Model, which suggests it's a robust model.

Explanatory Power of Odds Ratio and Probability :

Next we are finding out the power of Odds and Probability of the variables impacting on Customer Churn.

```
(Intercept)  AccountWeeks ContractRenewal1        DataPlan1    CustServCalls         DayMins         DayCalls
0.002427623   1.001123000      0.129915768      0.399084196      1.579136512     1.012303727     1.005817110
 OverageFee        RoamMins
1.156744488     1.079539704

(Intercept)  AccountWeeks ContractRenewal1        DataPlan1    CustServCalls         DayMins         DayCalls
0.002421744   0.500280592      0.114978277      0.285246733      0.612273334     0.503057125     0.501450060
 OverageFee        RoamMins
0.536338215     0.519124353
```

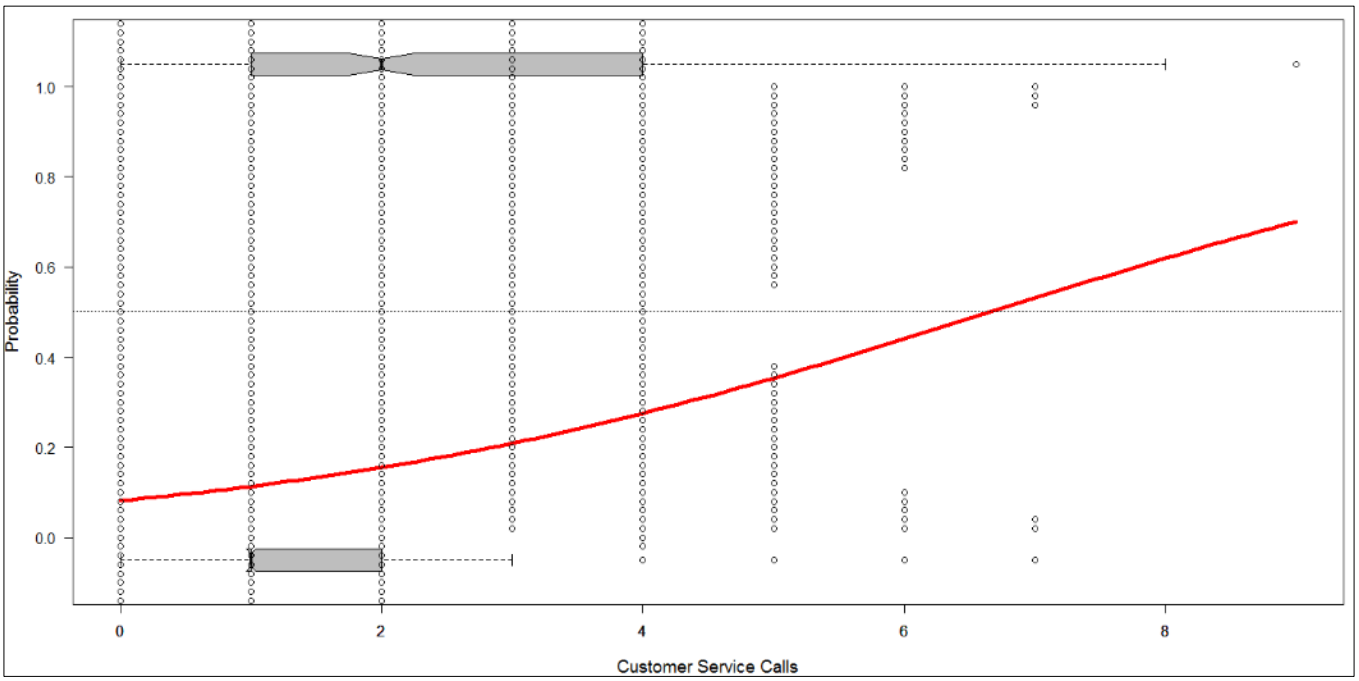
If a particular Variable as shown in following table is increased by 'One Unit', the odds of customer churn (Vs. not churning) and the probability of Customer Churn is shown in the following table

Variables	Odds Ratio	Probability
AccountWeeks	1.0011	50.03%
ContractRenewal1	0.1299	11.50%
DataPlan1	0.3991	28.53%
CustServCalls	1.5791	61.23%
DayMins	1.0123	50.30%
DayCalls	1.0058	50.14%
OverageFee	1.1567	53.63%
RoamMins	1.0795	52%

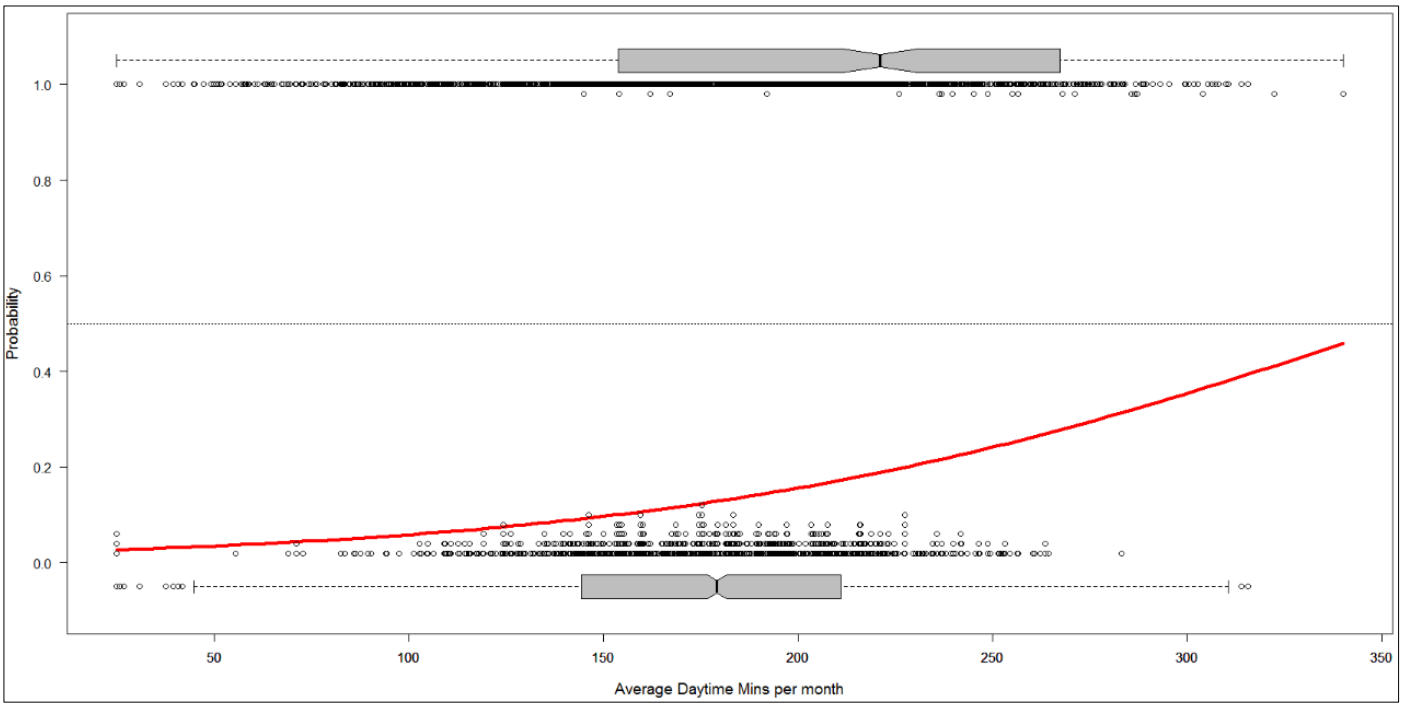
When the Customer renews Contract the odds customer will churn is 0.1299 compared to when the customer does renew. Similarly, when the customer opt for Data Plan the odds the customer will churn is 0.399 compared to when the customer doesn't opt for DataPlan. The customer churn is explained maximum by the Odds Ration of contract renewal which shows that 88.5 % probability of churn if the customer does renew the contract.

Next we are plotting the sigmoid curves of various variables with churn with the help of popbio()

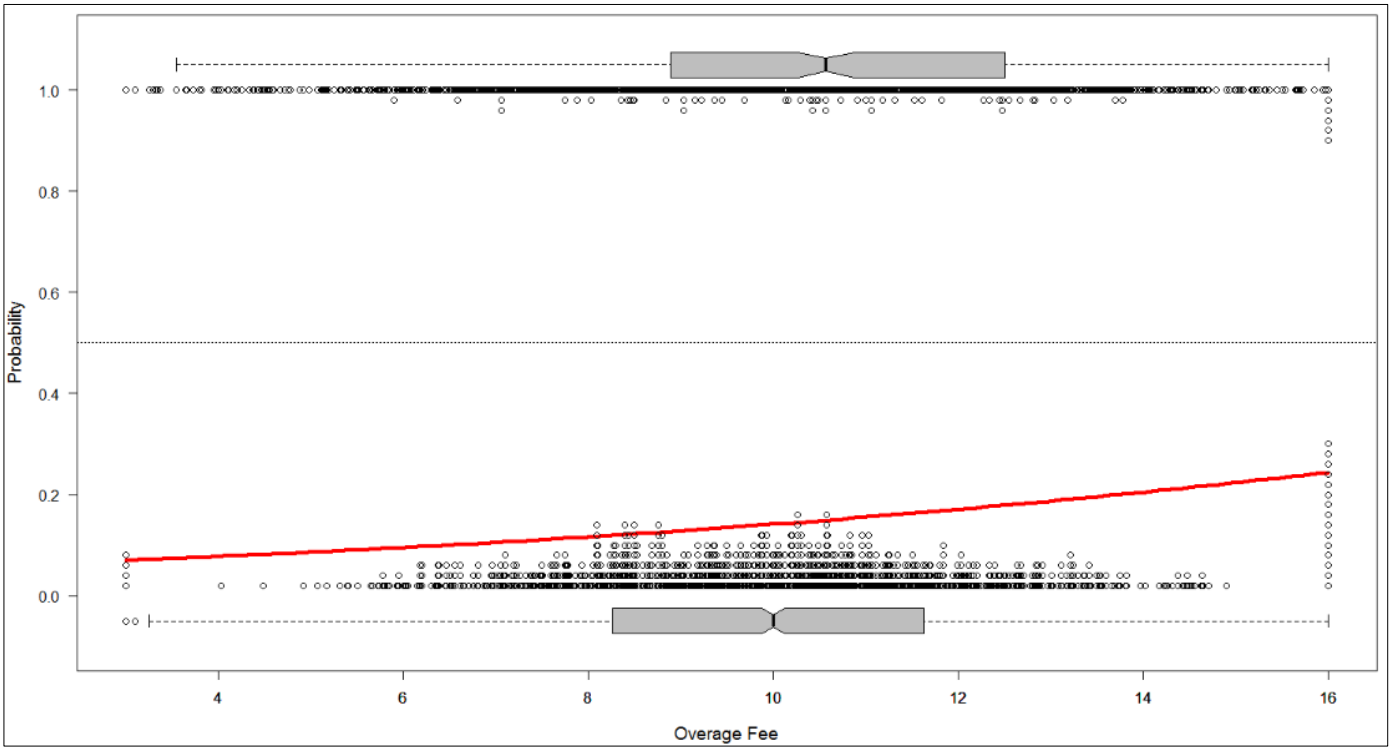
Customer Service Calls vs Churn



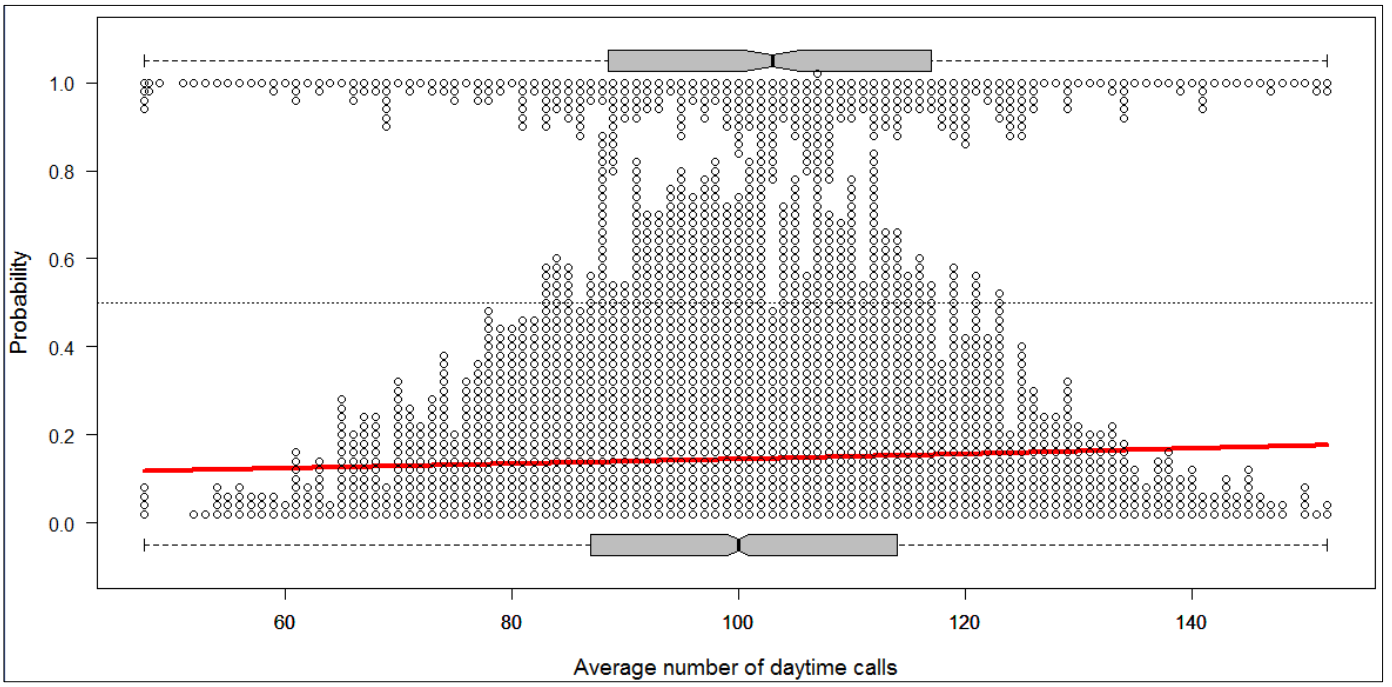
DayMins vs Churn



Overage Fee vs Churn

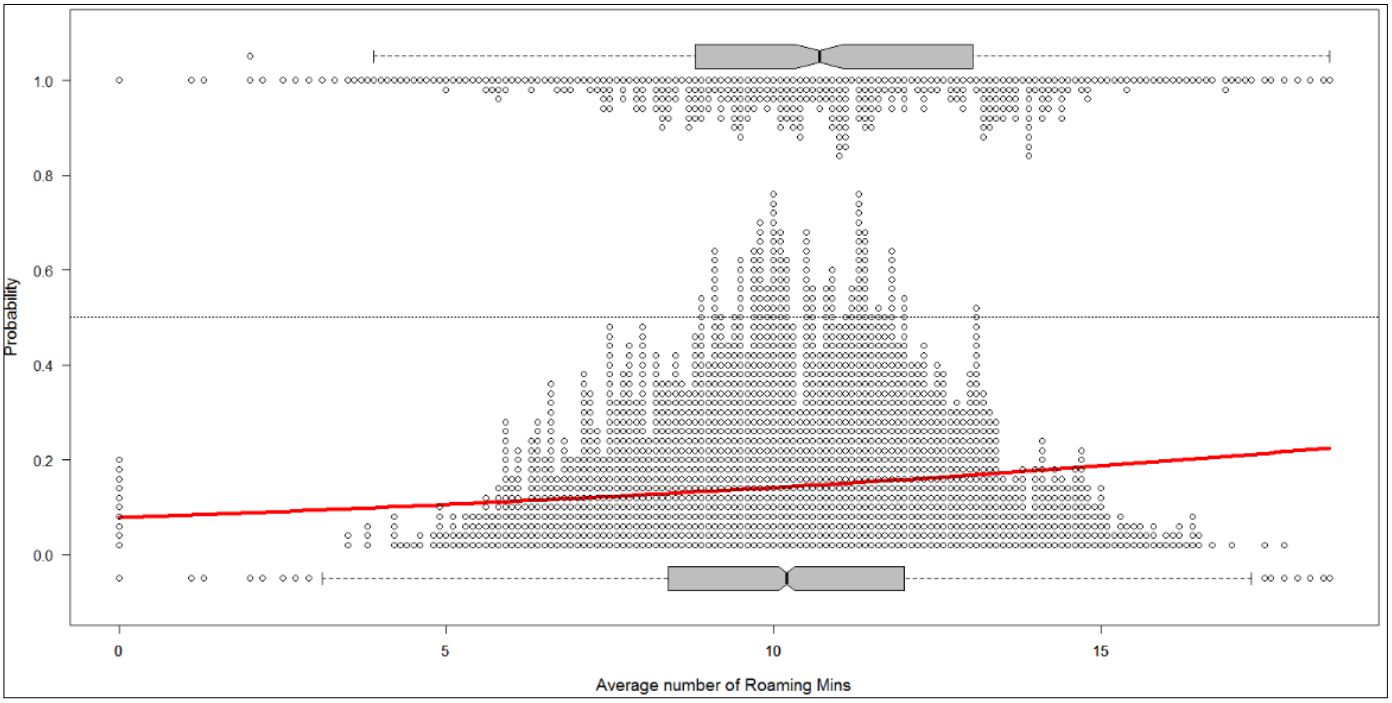


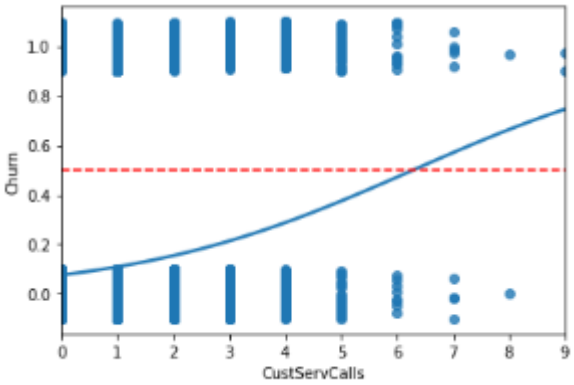
Day Calls vs Churn

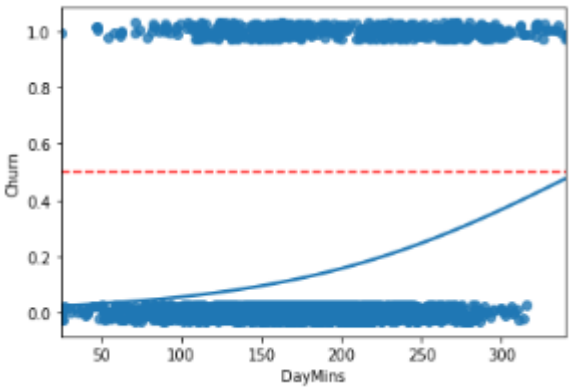


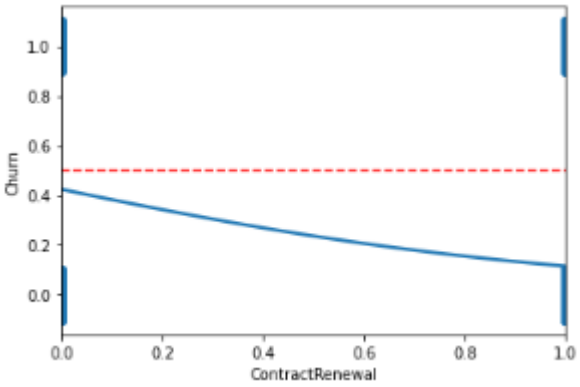
It shows that the Day Calls is least significant as the sigmoid curve is nearly a straight line and less than 0.5 cut off value.

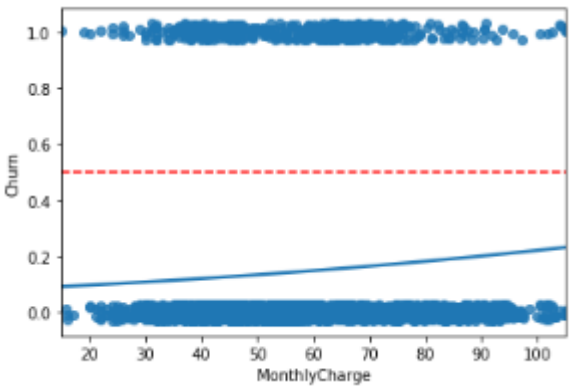
Roaming Minutes vs Churn











From all the above plots we could see that the Customer service calls , DayMins and Contract Renewal are highly significant and nearly trace a sigmoid curve with all the others having a straight line.

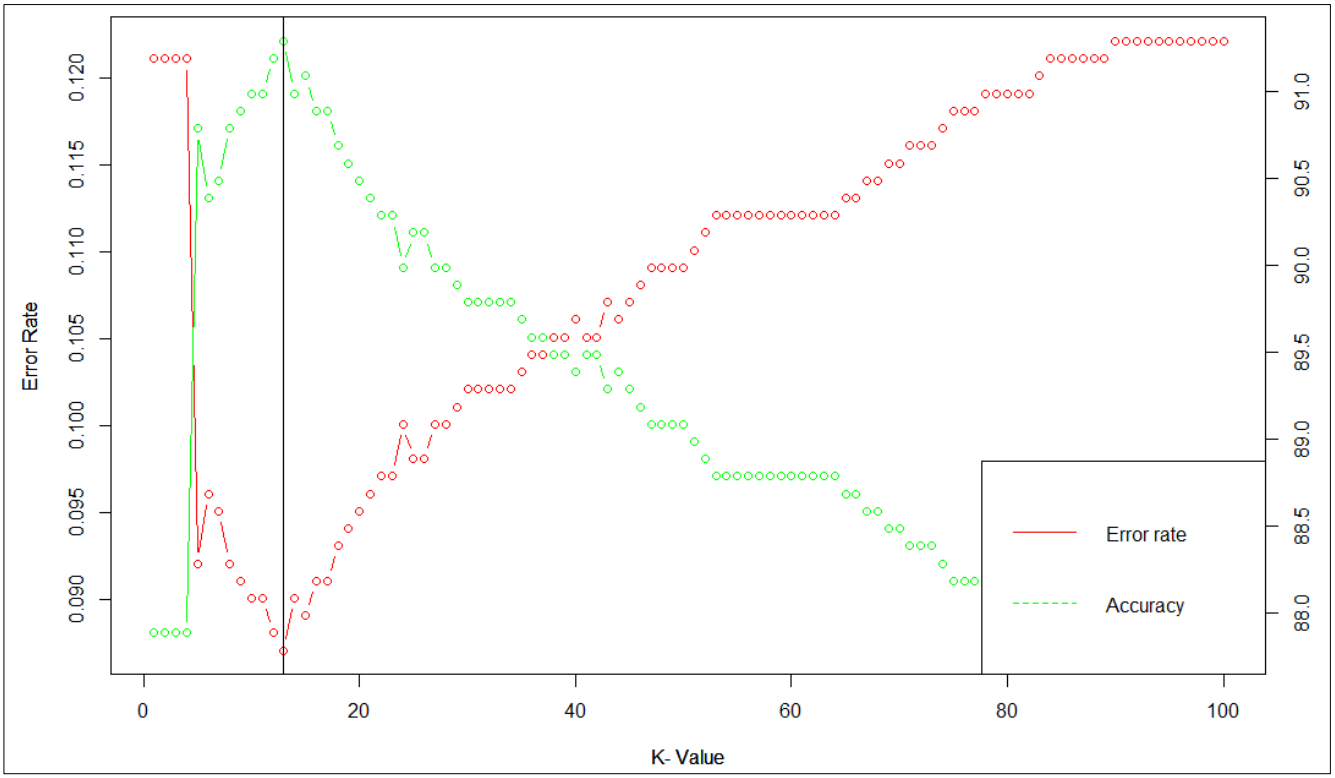
5 Apply & Interpret KNN Model

Since KNN method uses distance calculations, scaling or normalizing of data is a prerequisite hence we are performing minmax normalization on the original data set and then splitting it into training and testing set. Below is the head of the training set after min – max normalization.

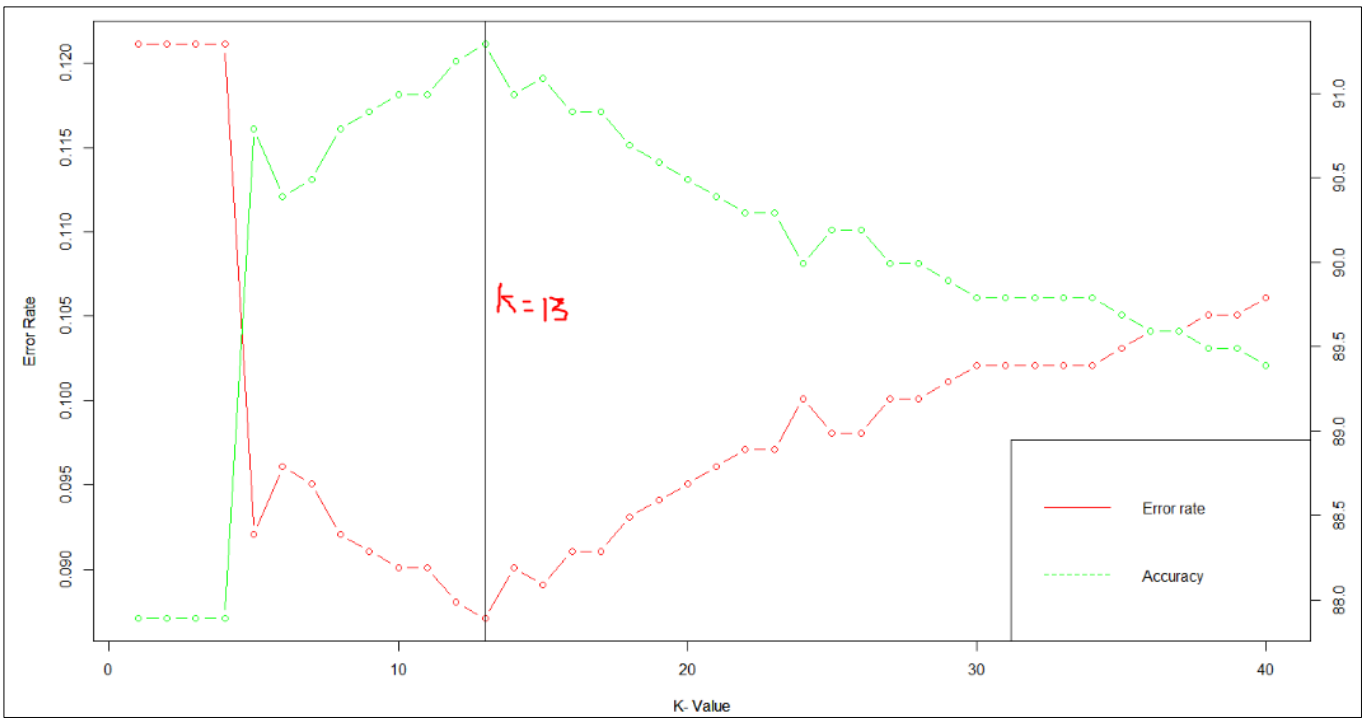
Churn	AccountWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins
0	0.6546392	1	1	0.6585366	0.1111111	0.7622222	0.5974698	0.8222222	0.528461538	0.5405405
0	0.7010309	1	0	0.0000000	0.0000000	0.6933333	0.6358060	0.4111111	0.235384615	0.6594595
0	0.4278351	0	0	0.0000000	0.2222222	0.8711111	0.2236918	0.4666667	0.007692308	0.3567568
0	0.3814433	0	0	0.0000000	0.3333333	0.4498413	0.6262220	0.2888889	0.340000000	0.5459459
0	0.6030928	0	0	0.0000000	0.0000000	0.6298413	0.4824612	0.4666667	0.617692308	0.3405405
0	0.6185567	1	1	0.4951220	0.3333333	0.6133333	0.3866207	0.8033333	1.000000000	0.4054054

Optimal K value :

We would be using kknn package to perform the K- nearest neighbor analysis. Since the function requires K – value or K – neighbors to be provided before hand, we have to determine the optimal number of neighbors. Hence we are running two custom loops, one for determining the accuracy and other for error rate for a range of K – values from one neighbors to 100 neighbors . Below is the output of the plots (Refer the code for the loops)



We could see that beyond K – value of 40 , the accuracy reduces drastically and error rate also increases, hence we are running the loops again with K – value range between 1 and 40 and below is the output generated.

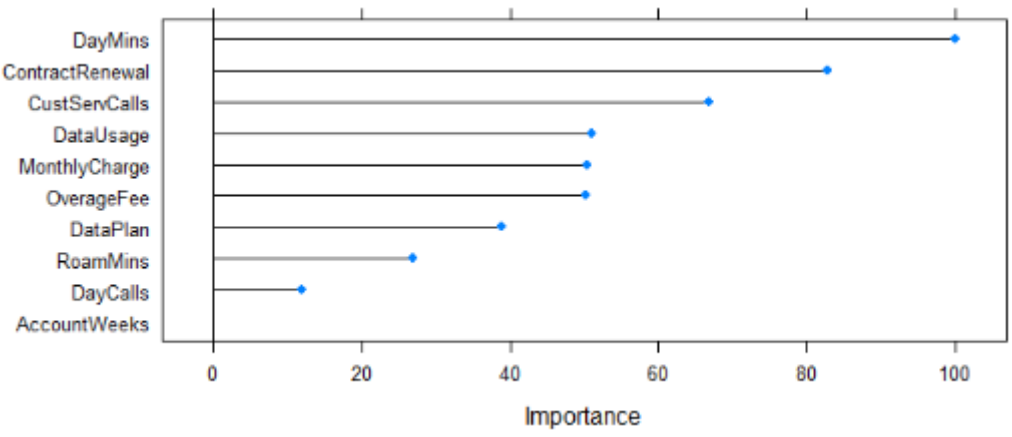


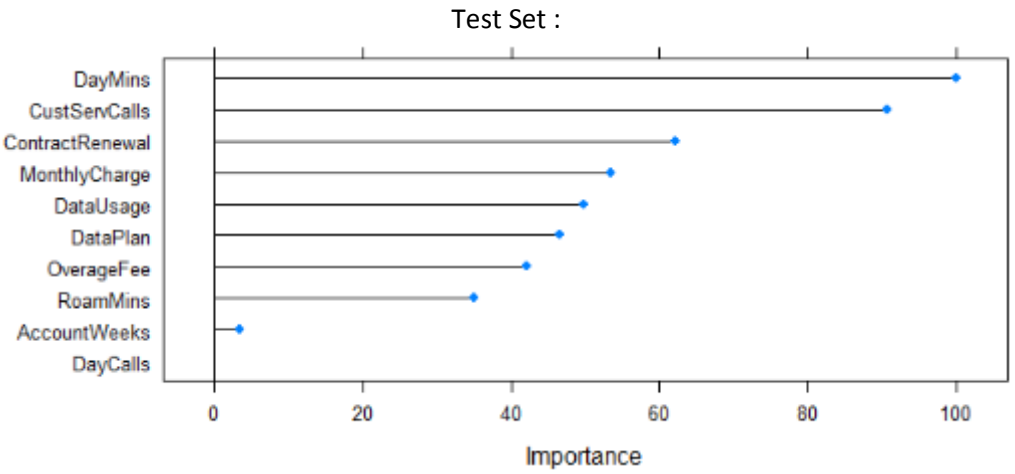
We could see from above that at K – value of 13 , the accuracy of the model is highest and error rate is the least. Hence we would choose K = 13 as the optimal value inorder to proceed with building the K - NN

After we run the k-NN model with 13 neighbors and predict with the test dataset with our model build, we next check the confusion matrix and other metrics which will be discussed in the next sections.

The variable importance for the k-NN model for the trained model with train dataset is shown below. Note that k-NN is a non – parametric model which does not have any assumptions or restrictions on multicollinearity. Hence we have ran the model with all the variables. K-NN model shows DayMins , Contract renewal , Customer Service calls as the most importance feature variables for customer churn.

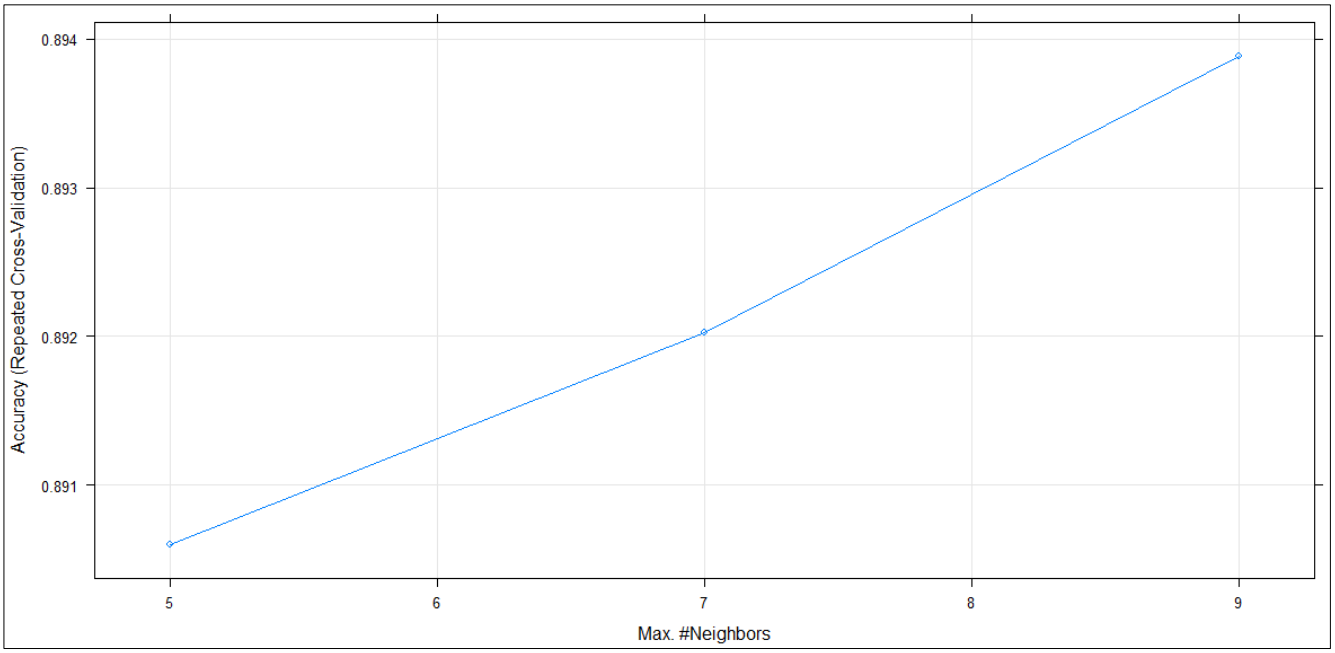
Train Set :





With Cross validation:

We are running the K – NN model with 10 fold cross validation inorder to improve the accuracy and goodness of fit of the model. We would be using the caret package to perform the same. Incase of cross validation. The optimal value of K is chosen to be 9, this is due to automated selection process of the caret package to choose optimal neighbors when performing iterations with the cross validation folds.



6 Apply & Interpret Naïve Bayes Model

Since Naïve Bayes has naïve assumption that all the variables must be independent of each other, the model must be independent of multicollinearity, hence we would have to perform the model with the two variables Monthly Charge and Data Usage removed. Apart from multicollinearity we are also performing normality test on the continuous variables and check for normality.

Normality test :

```
$multivariateNormality
           Test         Statistic p value Result
1 Mardia Skewness 9485.04361783102       0     NO
2 Mardia Kurtosis 162.212187021344       0     NO
3             MVN              <NA>    <NA>     NO

$univariateNormality
          Test      Variable Statistic   p value Normality
1 Shapiro-Wilk AccountWeeks     0.9965  <0.001       NO
2 Shapiro-Wilk   DataUsage      0.6699  <0.001       NO
3 Shapiro-Wilk     DayMins      0.9991  0.1002       YES
4 Shapiro-Wilk   DayCalls       0.9979   2e-04       NO
5 Shapiro-Wilk MonthlyCharge    0.9711  <0.001       NO
6 Shapiro-Wilk  OverageFee      0.9973  <0.001       NO
7 Shapiro-Wilk   RoamMins       0.9935  <0.001       NO

$Descriptives
              n        Mean   Std.Dev Median   Min   Max   25th   75th         Skew     Kurtosis
AccountWeeks  3333 100.9171917 39.428020 101.00  1.00 195.0  74.00 127.00  0.02929880 -0.28953769
DataUsage     3333   0.8136364  1.264748   0.00  0.00   4.1   0.00   1.78  1.24872989 -0.07258366
DayMins       3333 179.8096910 54.321365 179.40 25.00 340.0 143.70 216.40 -0.01370244 -0.09018037
DayCalls      3333 100.4732133 19.832671 101.00 47.66 152.0  87.00 114.00 -0.04190104 -0.18193678
MonthlyCharge 3333  56.2965797 16.397707  53.50 15.00 105.0  45.00  66.20  0.58329581 -0.06473645
OverageFee    3333  10.0467897  2.508362  10.07  3.00  16.0   8.33  11.77 -0.04833532 -0.19134899
RoamMins      3333  10.2367237  2.790022  10.30  0.00  18.5   8.50  12.10 -0.25103041  0.58680068
```

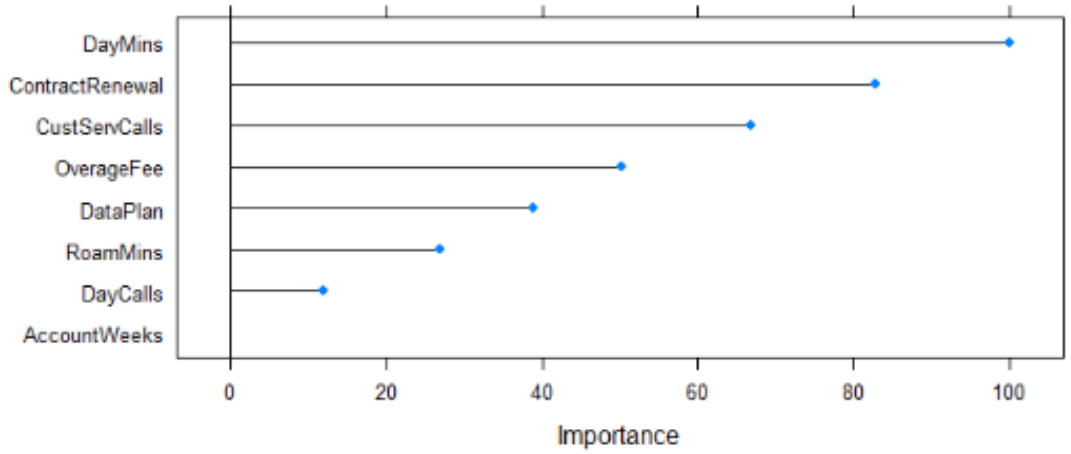
Though only Daymins are shown to be normal distributed , the p value for other variables are all less than 0.05 which shows that all the other variables are nearly normally distributed hence it is suitable for Naïve Bayes model. The alternate hypothesis is that at least one of the variables are normally distributed.

After the tests , we are running the naïve bayes model on the train set and below are the prior probabilities of various variables with churn as dependent variable.

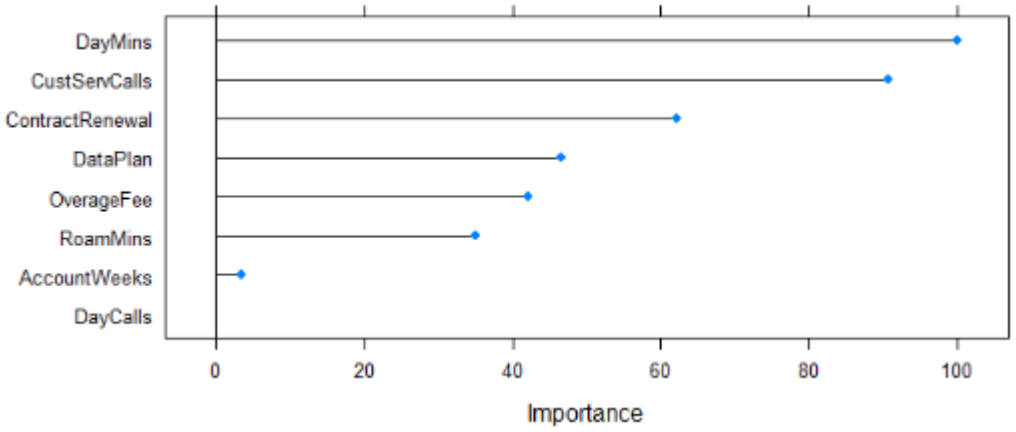
```
$apriori
grouping
        0         1
0.8547558 0.1452442

$tables
$tables$AccountWeeks
      [,1]     [,2]
0 100.8937 38.93021
1 103.2124 38.74193

$tables$ContractRenewal
        var
grouping          0          1
       0 0.06165414 0.93834586
       1 0.29498525 0.70501475

$tables$DataPlan
        var
```

```
grouping         0         1
       0 0.7022556 0.2977444
       1 0.8289086 0.1710914

$tables$CustServCalls
      [,1]     [,2]
0 1.464160 1.169700
1 2.162242 1.810617

$tables$DayMins
      [,1]     [,2]
0 176.7696 49.68654
1 207.4336 70.62451

$tables$DayCalls
      [,1]     [,2]
0 100.3801 19.73616
1 102.1907 21.39271

$tables$OverageFee
       [,1]     [,2]
0  9.934792 2.475232
1 10.620177 2.522482

$tables$RoamMins
      [,1]     [,2]
0 10.20140 2.780224
1 10.71209 2.825801


$levels
[1] "0" "1"

$call
NaiveBayes.default(x = X, grouping = Y)
```

Train :



Test :



Above are the variable importance plot produced by the naïve bayes algorithms for both training and test sets.

7 *Confusion matrix interpretation for all models*

The following are the performance metrics that are derived for the three Machine Learning models and compared.

1) Confusion Matrix :

A confusion matrix is a technique for summarizing the performance of a classification algorithm. It consist of the below set of values.

- True Positives: The predicted positive (yes) and the actual positive is same
- True Negatives: The predicted negative (No) and the actual negative is same
- False Positives (Type 1 Error): The value is predicted as positive (yes) but the actual value is Negative in the data.
- False Negatives (Type 2 Error): The value is predicted as negative (no) but the actual value is positive in the data.

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

2) Accuracy :

It is a measure of how accurately our model classify the data points in terms of percentage. In general less the value of false prediction rate, the more is the accuracy.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN)$$

3) Sensitivity or Recall :

Sensitivity/Recall is the ratio between how much were correctly identified as positive compared to how much were actually positive. Lesser the false negative rates, higher is the sensitivity of the model.

$$\text{Sensitivity} = TP / FN+TP$$

4) Specificity :

Specificity of a classifier is the ratio between how much were correctly classified as negative to how much was actually negative.

$$\text{Specificity} = TN/FP+TN$$

5) Precision :

It is a measure of how much were correctly classified as positive out of all positives. Among the points identified as Positive by the model, how many are really Positive ones.

$$\text{Precision} = TP / TP + FP$$

6) F1 Score

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall when there is an uneven class distribution (large number of Actual Negatives). The harmonic mean of precision and recall gives f1 score.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

7) Kappa Coefficient :

It basically tells us how much better the classification model is performing over the performance of a ML model that simply guesses at random according to the frequency of each class. It is a coefficient value that vary between 0 to 1 with ranges indicating 0–0.20 as slight, 0.21–0.40 as fair, 0.41–0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1 as almost perfect agreement.

Logistic Regression (Without cross validation)

Train Set

Below is the confusion matrix for the train data set for logistic , Accuracy of the model is 86.42%

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1949  271
         1   46   68

               Accuracy : 0.8642
                 95% CI : (0.8496, 0.8778)
    No Information Rate : 0.8548
    P-Value [Acc > NIR] : 0.1025

                  Kappa : 0.245

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.20059
            Specificity : 0.97694
         Pos Pred Value : 0.59649
         Neg Pred Value : 0.87793
              Precision : 0.59649
                 Recall : 0.20059
                     F1 : 0.30022
             Prevalence : 0.14524
         Detection Rate : 0.02913
   Detection Prevalence : 0.04884
      Balanced Accuracy : 0.58877

       'Positive' Class : 1
```

Test Set :

Below is the confusion matrix for the test data set , accuracy of the model is 86.19%, accuracy for the test dataset will be always low compared to train set. In this case its slightly low hence it performed well in both the data sets.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 837 120
         1  18  24

               Accuracy : 0.8619
                 95% CI : (0.8389, 0.8827)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.313

                  Kappa : 0.2064

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.16667
            Specificity : 0.97895
         Pos Pred Value : 0.57143
```

```
          Neg Pred Value : 0.87461
               Precision : 0.57143
                  Recall : 0.16667
                      F1 : 0.25806
              Prevalence : 0.14414
          Detection Rate : 0.02402
    Detection Prevalence : 0.04204
       Balanced Accuracy : 0.57281

        'Positive' Class : 1
```

Logistic Regression with cross validation

Train Set

Below is the confusion matrix for the train data set after 10 fold cross validation, we could see that the accuracy of the model remained the same with 86.42 % upon cross validation which shows that the model has been very robust with cross validation as well.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1949  271
         1   46   68

               Accuracy : 0.8642
                 95% CI : (0.8496, 0.8778)
    No Information Rate : 0.8548
    P-Value [Acc > NIR] : 0.1025

                  Kappa : 0.245

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.20059
            Specificity : 0.97694
         Pos Pred Value : 0.59649
         Neg Pred Value : 0.87793
              Precision : 0.59649
                 Recall : 0.20059
                     F1 : 0.30022
             Prevalence : 0.14524
         Detection Rate : 0.02913
   Detection Prevalence : 0.04884
      Balanced Accuracy : 0.58877

       'Positive' Class : 1
```

Test Set

Below is the confusion matrix for the test data set , accuracy of the model is 85.89% which is very slightly less than the accuracy for the train dataset. Hence the model we build is really robust.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 829 115
         1  26  29

               Accuracy : 0.8589
                 95% CI : (0.8357, 0.8799)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.4149

                  Kappa : 0.2301

 Mcnemar's Test P-Value : 1.254e-13

            Sensitivity : 0.20139
            Specificity : 0.96959
         Pos Pred Value : 0.52727
         Neg Pred Value : 0.87818
              Precision : 0.52727
                 Recall : 0.20139
                     F1 : 0.29146
             Prevalence : 0.14414
         Detection Rate : 0.02903
   Detection Prevalence : 0.05506
      Balanced Accuracy : 0.58549

       'Positive' Class : 1
```

KNN model (Without cross validation)

Train Set

Below is the confusion matrix for the train data set for the K-NN model, Accuracy of the model is 93.53% which is highest among other two.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1989  145
         1    6  194

               Accuracy : 0.9353
                 95% CI : (0.9246, 0.9449)
    No Information Rate : 0.8548
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.686

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.57227
            Specificity : 0.99699
         Pos Pred Value : 0.97000
         Neg Pred Value : 0.93205
              Precision : 0.97000
                 Recall : 0.57227
                     F1 : 0.71985
             Prevalence : 0.14524
```

```
         Detection Rate : 0.08312
   Detection Prevalence : 0.08569
      Balanced Accuracy : 0.78463

       'Positive' Class : 1
```

Test Set

Below is the confusion matrix for the test data set, Accuracy of the model is 91.29 %

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 848  80
         1   7  64

               Accuracy : 0.9129
                 95% CI : (0.8937, 0.9297)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 3.030e-08

                  Kappa : 0.5528

 Mcnemar's Test P-Value : 1.171e-14

            Sensitivity : 0.44444
            Specificity : 0.99181
         Pos Pred Value : 0.90141
         Neg Pred Value : 0.91379
              Precision : 0.90141
                 Recall : 0.44444
                     F1 : 0.59535
             Prevalence : 0.14414
         Detection Rate : 0.06406
   Detection Prevalence : 0.07107
      Balanced Accuracy : 0.71813

       'Positive' Class : 1
```

KNN model with cross validation

Train Set

Below is the confusion matrix for the train data set for the K-NN model after subjecting to 10 fold cross validation. Accuracy of the model is 93.53% which has been increased, however it is slightly overfit model compared to previous since k-nn is highly lazy algorithm and if same train dataset is used for testing, it is likely to overfit the model.

```
          Reference
Prediction    0    1
         0 1988  116
         1    7  223

               Accuracy : 0.9473
                 95% CI : (0.9374, 0.956)
    No Information Rate : 0.8548
```

```
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7551

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.65782
            Specificity : 0.99649
         Pos Pred Value : 0.96957
         Neg Pred Value : 0.94487
              Precision : 0.96957
                 Recall : 0.65782
                     F1 : 0.78383
             Prevalence : 0.14524
         Detection Rate : 0.09554
   Detection Prevalence : 0.09854
      Balanced Accuracy : 0.82715

       'Positive' Class : 1
```

Test Set

Below is the confusion matrix for the test data set, Accuracy of the model is 95 % which is highly overfitted model.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 851  46
         1   4  98

               Accuracy : 0.9499
                 95% CI : (0.9345, 0.9626)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7692

 Mcnemar's Test P-Value : 6.7e-09

            Sensitivity : 0.6806
            Specificity : 0.9953
         Pos Pred Value : 0.9608
         Neg Pred Value : 0.9487
              Precision : 0.9608
                 Recall : 0.6806
                     F1 : 0.7967
             Prevalence : 0.1441
         Detection Rate : 0.0981
   Detection Prevalence : 0.1021
      Balanced Accuracy : 0.8379

       'Positive' Class : 1
```

Naïve Bayes (Without cross validation)

Train Set

Below is the confusion matrix for the train data set for naïve bayes model, Accuracy of the model is 87 %

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1936  243
         1   59   96

               Accuracy : 0.8706
                 95% CI : (0.8563, 0.884)
    No Information Rate : 0.8548
    P-Value [Acc > NIR] : 0.01498

                  Kappa : 0.3274

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.28319
            Specificity : 0.97043
         Pos Pred Value : 0.61935
         Neg Pred Value : 0.88848
              Precision : 0.61935
                 Recall : 0.28319
                     F1 : 0.38866
             Prevalence : 0.14524
         Detection Rate : 0.04113
   Detection Prevalence : 0.06641
      Balanced Accuracy : 0.62681

       'Positive' Class : 1
```

Test Set

Below is the confusion matrix for the test data set , accuracy of the model is 87.59%, accuracy for the test dataset is nearly same compared to train set. Hence the naïve bayes model built is highly robust.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 834 103
         1  21  41

               Accuracy : 0.8759
                 95% CI : (0.8538, 0.8957)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.03749

                  Kappa : 0.3409

 Mcnemar's Test P-Value : 3.49e-13

            Sensitivity : 0.28472
            Specificity : 0.97544
         Pos Pred Value : 0.66129
```

```
          Neg Pred Value : 0.89007
               Precision : 0.66129
                  Recall : 0.28472
                      F1 : 0.39806
              Prevalence : 0.14414
          Detection Rate : 0.04104
    Detection Prevalence : 0.06206
       Balanced Accuracy : 0.63008

        'Positive' Class : 1
```

Naïve Bayes with cross validation

Train Set

Below is the confusion matrix for naïve bayes train set after performing 10 fold cross validation. The accuracy has slightly reduced by 2% and has been made underfit, hence the cross validation has been made suitable for this model to tune it to be good fit model.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1994  327
         1    1   12

               Accuracy : 0.8595
                 95% CI : (0.8447, 0.8733)
    No Information Rate : 0.8548
    P-Value [Acc > NIR] : 0.2701

                  Kappa : 0.0581

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.035398
            Specificity : 0.999499
         Pos Pred Value : 0.923077
         Neg Pred Value : 0.859112
              Precision : 0.923077
                 Recall : 0.035398
                     F1 : 0.068182
             Prevalence : 0.145244
         Detection Rate : 0.005141
   Detection Prevalence : 0.005570
      Balanced Accuracy : 0.517448

       'Positive' Class : 1
```

Test Set

Below is the confusion matrix for the test data set , accuracy of the model is 86.79%, which has been made slightly under fit after 10 fold cross validation.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 855 132
         1   0  12

               Accuracy : 0.8679
                 95% CI : (0.8453, 0.8883)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.1499

                  Kappa : 0.1347

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.08333
            Specificity : 1.00000
         Pos Pred Value : 1.00000
         Neg Pred Value : 0.86626
              Precision : 1.00000
                 Recall : 0.08333
                     F1 : 0.15385
             Prevalence : 0.14414
         Detection Rate : 0.01201
   Detection Prevalence : 0.01201
      Balanced Accuracy : 0.54167

       'Positive' Class : 1
```

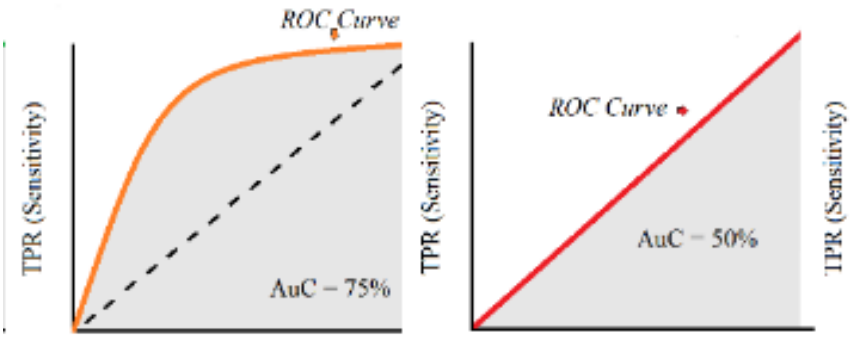
We could see that the K-NN model without cross validation performed well and has highest accuracy compared to other two. However k-nn is parametric and more like memory based algorithm which memorizes the train data and applies the same to testing data with insights gained from train. Compared to naïve bayes and logistic regression, both the models have very few differences in accuracy at the range of 1% to 2% . However Naïve bayes performed well with 1% high accuracy compared to logistic regression for both non cross validated and cross validated models. In model comparison we would check for other parameters such as specificity, sensitivity etc.

8 Interpretation of other Model Performance Measures for all the Models (KS, AUC, GINI)

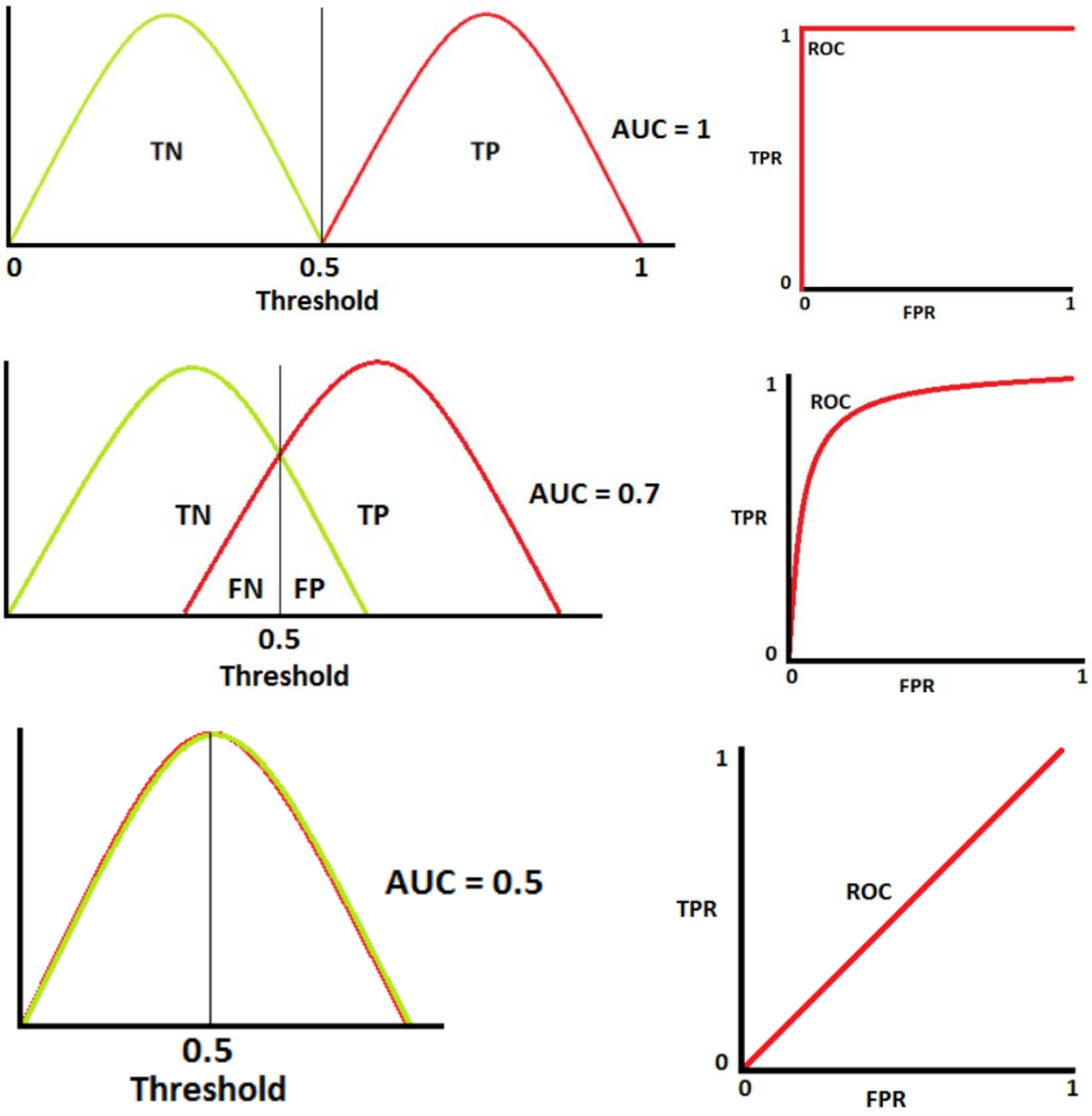
Receiver Operating Characteristics (ROC) Curve

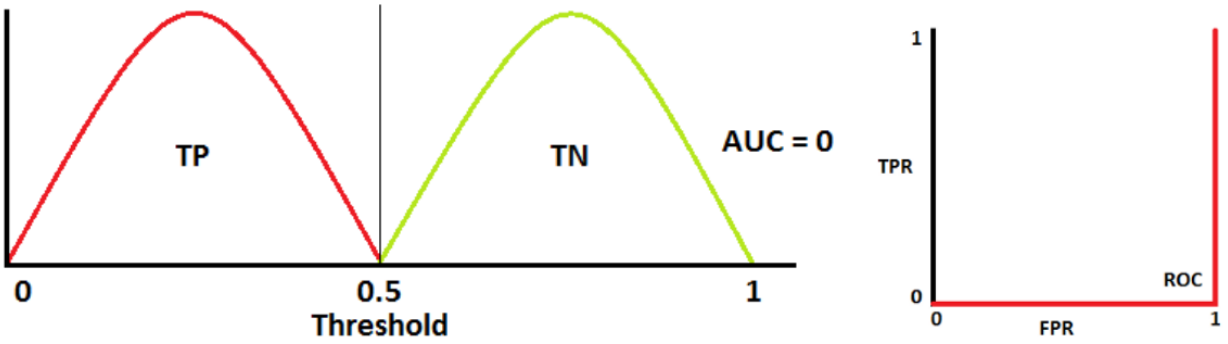
AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s

The ROC curve is plotted with True Positive Rates (TPR) against the False Positive Rates (FPR)where TPR is on y-axis and FPR is on the x-axis



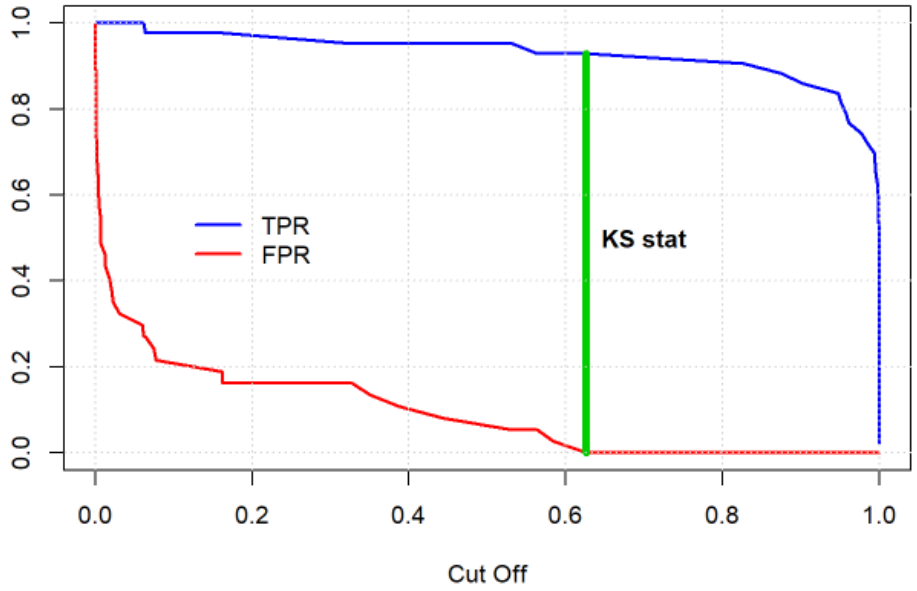
An excellent model has AUC near to the 1 which means it has good measure of separability. A poor model has AUC near to the 0 which means it has worst measure of separability, when AUC is 0.5, it means model has no class separation capacity. Below depicts the various scenarios.





Kolmogorov-Smirnov chart (KS)

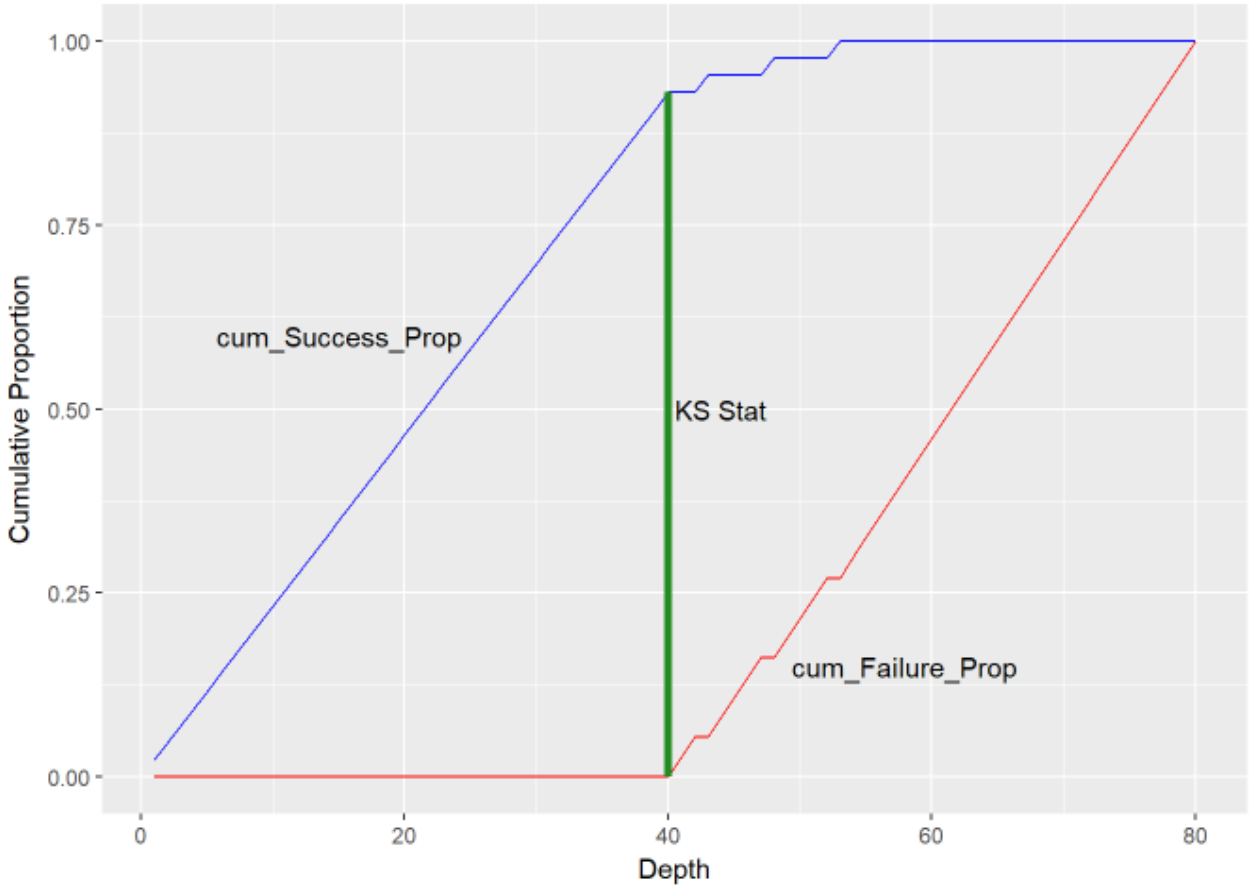
KS statistic is the maximum difference between TPR and FPR . In the figure below, a graphics of KS statistics is shown. Higher KS stat value is indicative of better model. If we have more than one models, then KS statistics can be used as a performance measure.



In the above figure, we took the cutoff along X-axis, and TPR and FPR along Y-axis. Instead we could take the population depth along X and proportion of 1s and 0s along Y. The max difference of two plots will be the KS stat. A more detailed explanation is given below:

1. From the model, we will have some probabilities of success. First, we need to sort the observations in descending order of these probabilities.
2. Now we will start with the first observation. This is almost guaranteed that the first observation will have an actual label 1. So for depth=1, the proporiton of success will be 1 divided by number of total 1s. And the proportion of failure would be 0 / number of total 1s.
3. Now for each possible depth, calculate the success and failure proportion. We have to keep in mind that this is cumulative proportion. So, at depth=n, success proportion would be number of 1s in first n rank ordered observation / number of total and failure proportion would be number of 0s in first n rank ordered observation divided by number of total 1's

4. Now we will plot the cumulative success and failure proportion against depth and calculate the KS stat as the maximum difference of the two.



Gini Coefficient

Gini coefficient is another measure of goodness of a binary classifier. The Gini coefficient is a ratio of two areas. The areas are:

1. The area between the ROC curve and the random model line (y=xy=x line, passing through (0,0)(0,0) and (1,1)(1,1))
2. The top left triangle above the random model line (y=xy=x line).

Now, the second area is just 0.5, since the domain and range of ROC are [0,1][0,1]. The first area is 0.5 less than the AUROC. So, Gini coefficient can be reduced to:

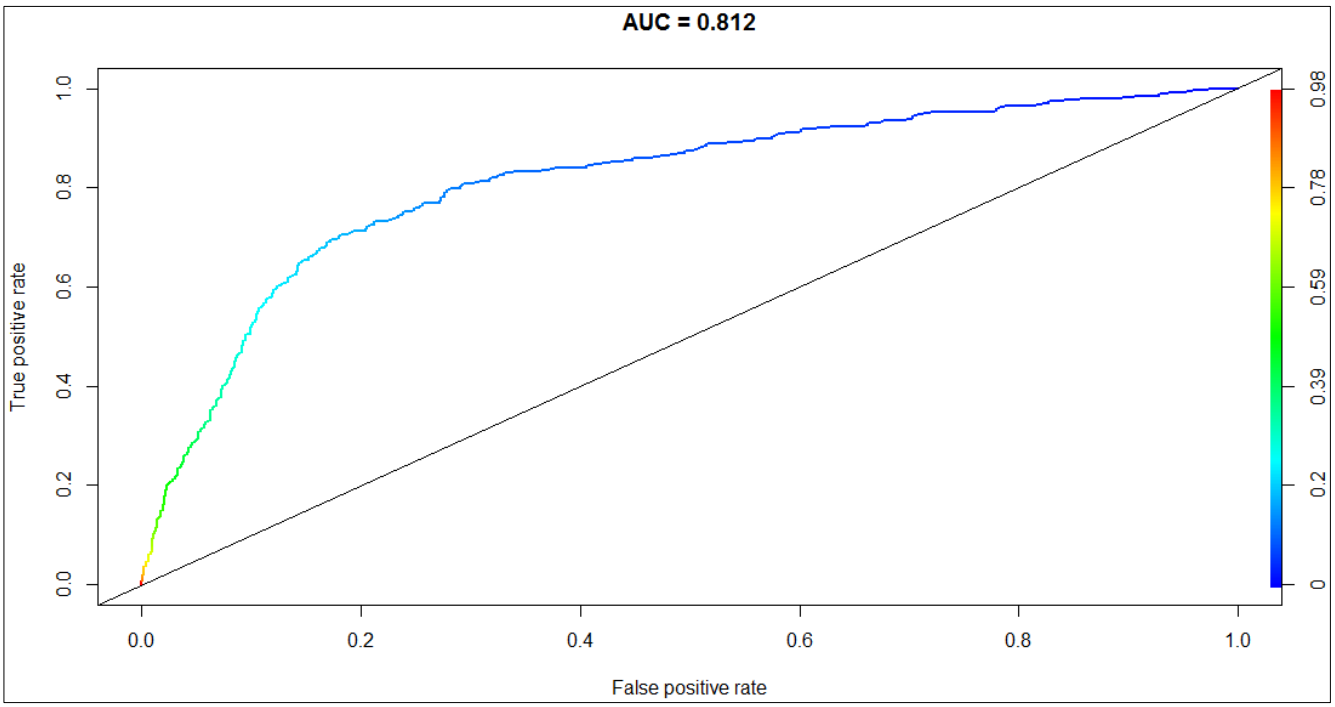
Gini Coefficient= AUROC−0.50.5 = AUROC0.5−1 = **2×AUROC−1**

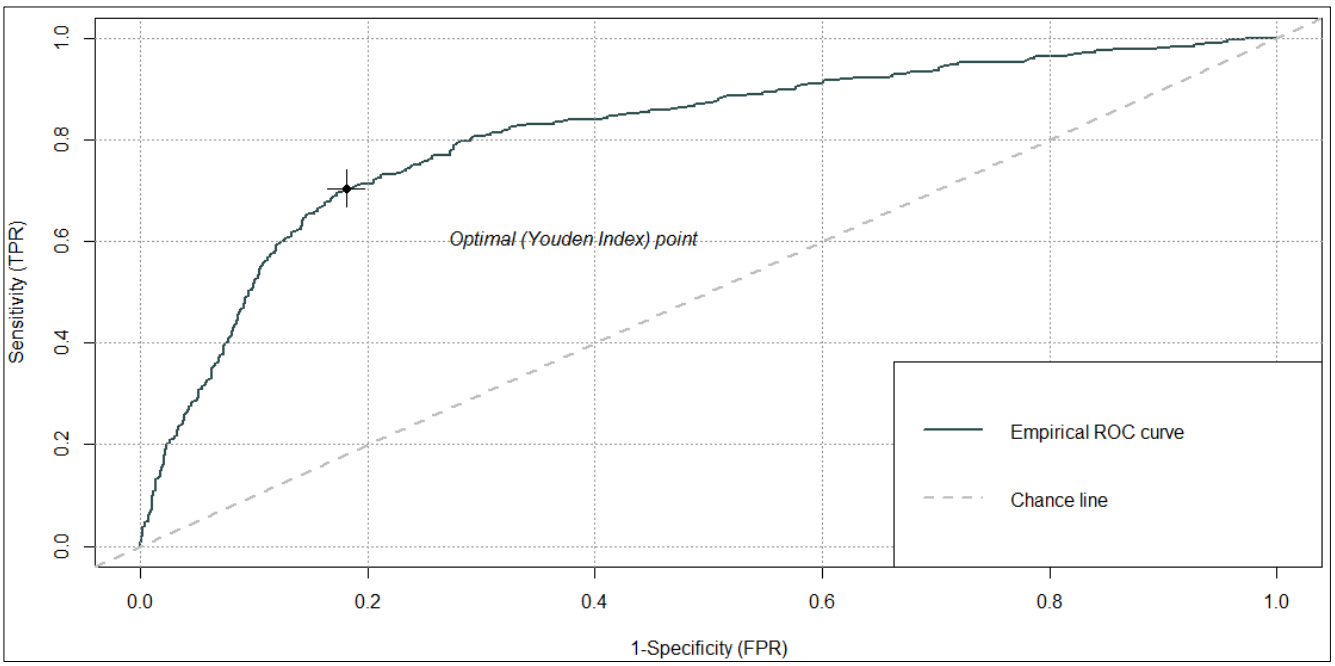
So, Gini coefficient is just a scaled version of AUROC , higher the Gini coefficient, better the model.

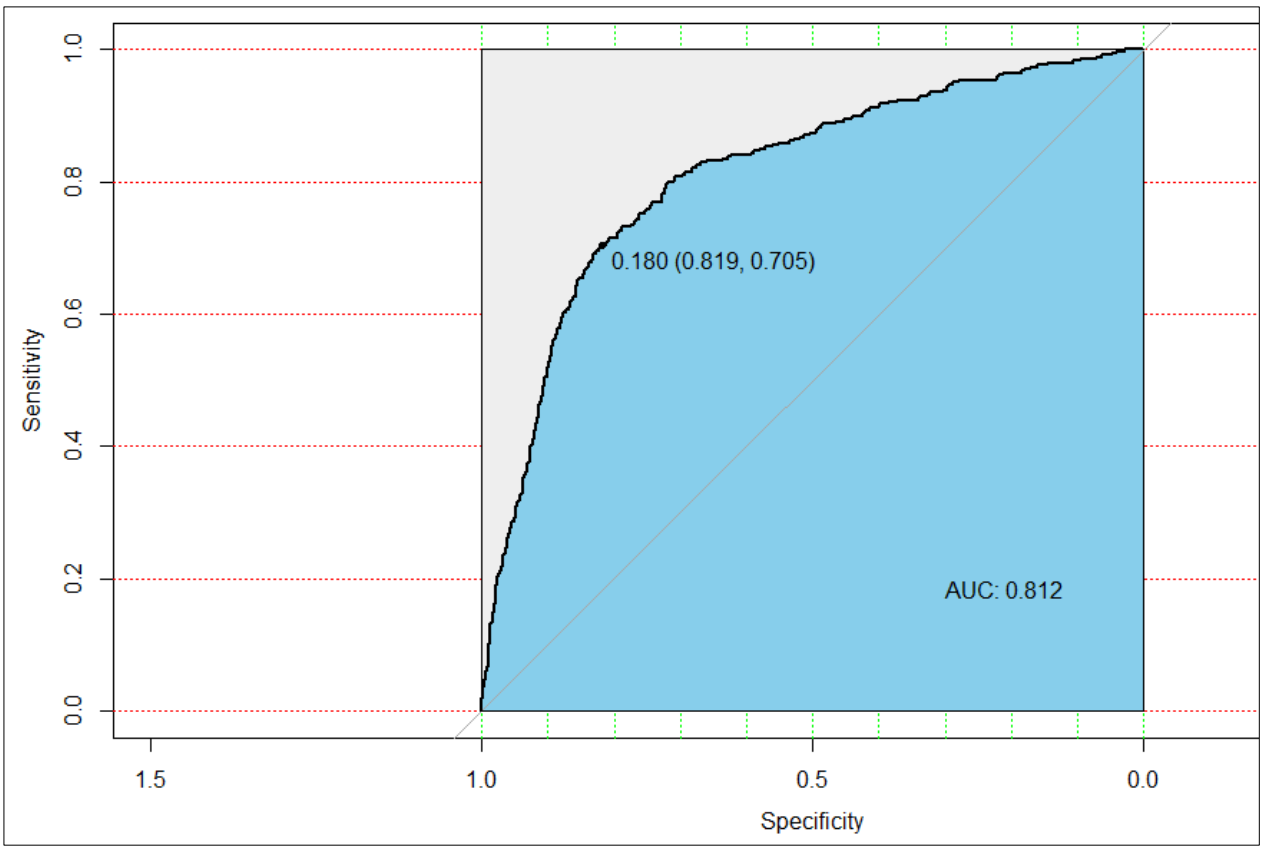
With above explanation on various model evaluation measures below are the calculated values for each model.

Logistic Regression

Train set :





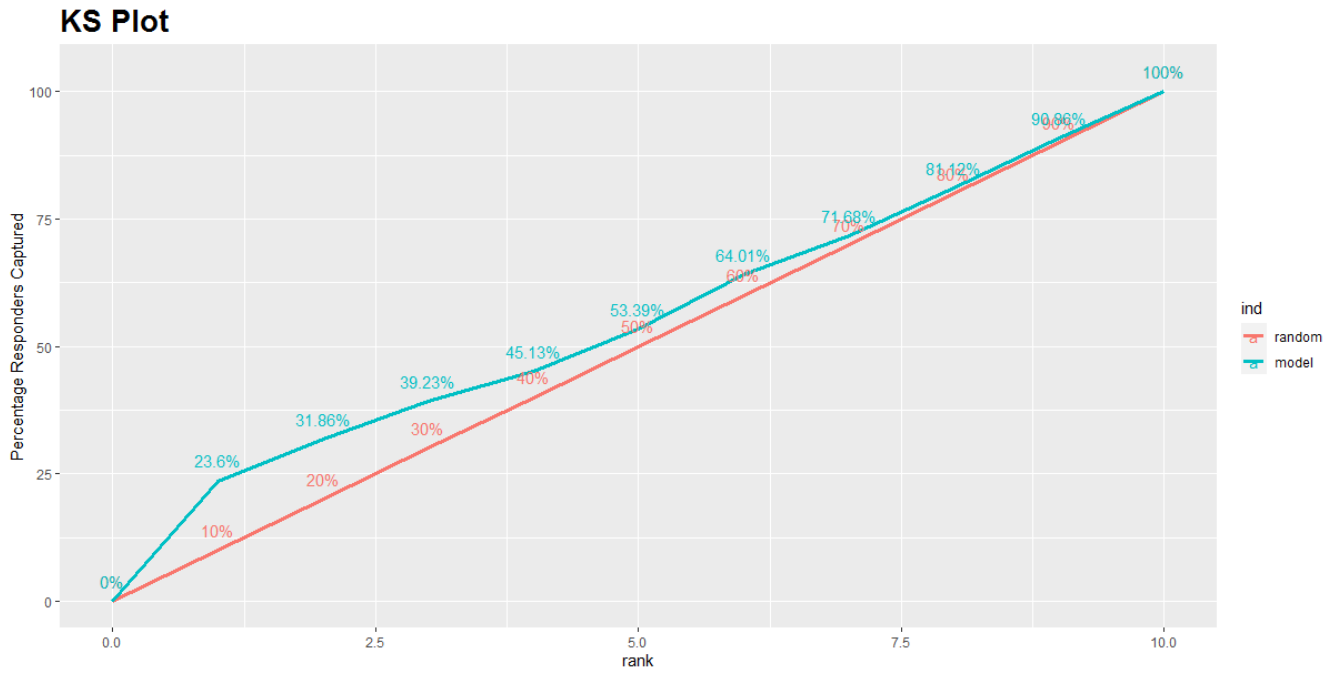


The AUC values and optimal youden index is shown above, the auc value is 81.2 % for the train dataset

KS Statistic and plot

rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders
1	233	153	80	33.84190	0.23598820	0.07669173
2	233	205	28	33.84190	0.08259587	0.10275689
3	233	208	25	33.84190	0.07374631	0.10426065
4	233	213	20	33.84190	0.05899705	0.10676692
5	233	205	28	33.84190	0.08259587	0.10275689
6	233	197	36	33.84190	0.10619469	0.09874687
7	233	207	26	33.84190	0.07669617	0.10375940
8	233	201	32	33.84190	0.09439528	0.10075188
9	233	200	33	33.84190	0.09734513	0.10025063
10	237	206	31	34.42288	0.09144543	0.10325815

cum_perc_responders	cum_perc_non_responders	difference
0.2359882	0.07669173	0.15929647
0.3185841	0.17944862	0.13913545
0.3923304	0.28370927	0.10862111
0.4513274	0.39047619	0.06085124
0.5339233	0.49323308	0.04069022
0.6401180	0.59197995	0.04813804
0.7168142	0.69573935	0.02107481
0.8112094	0.79649123	0.01471821
0.9085546	0.89674185	0.01181272
1.0000000	1.00000000	0.00000000

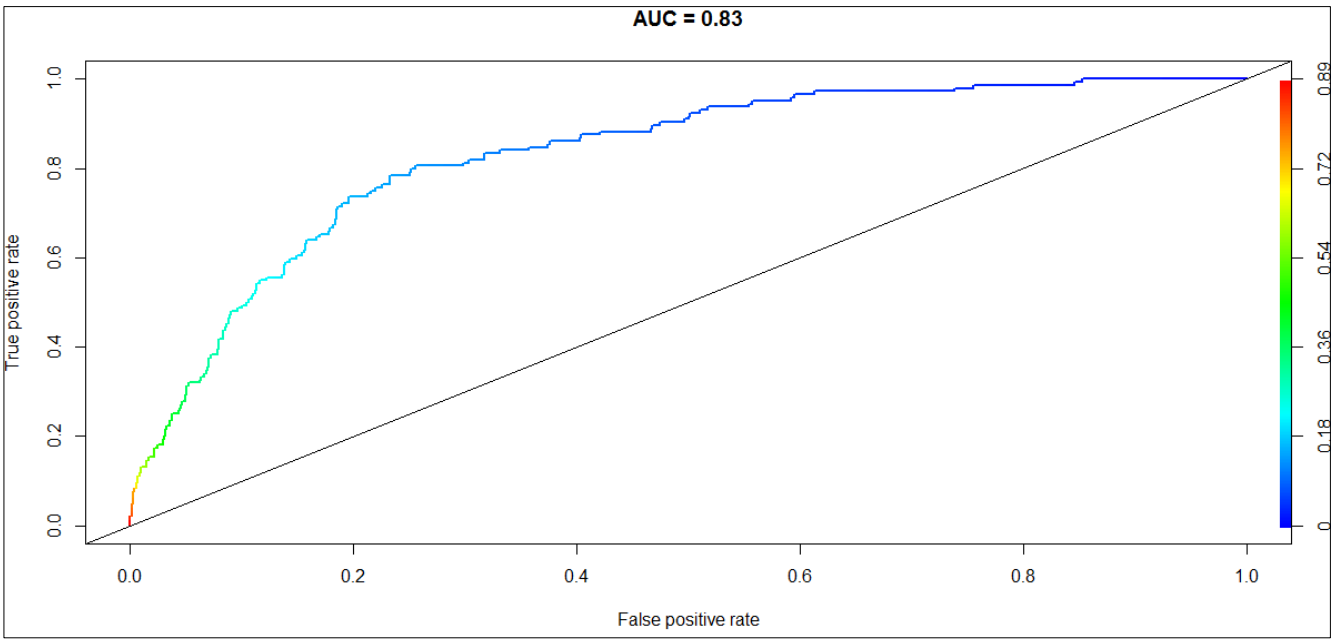


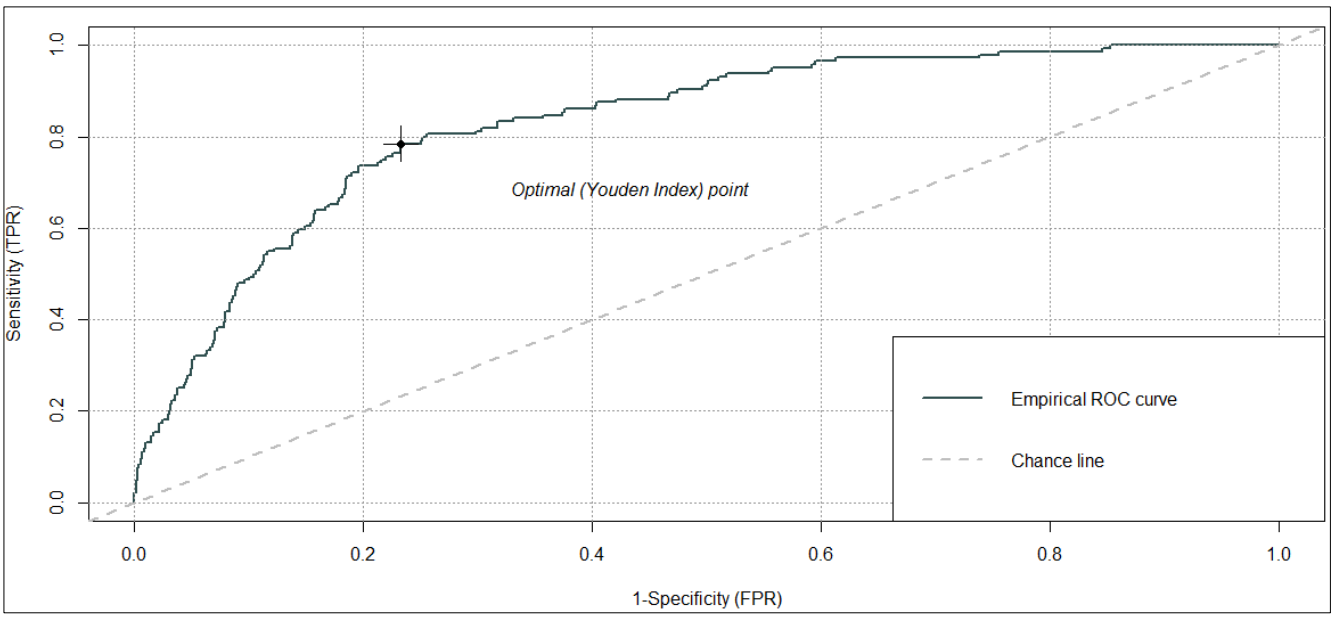
Gini coeff :

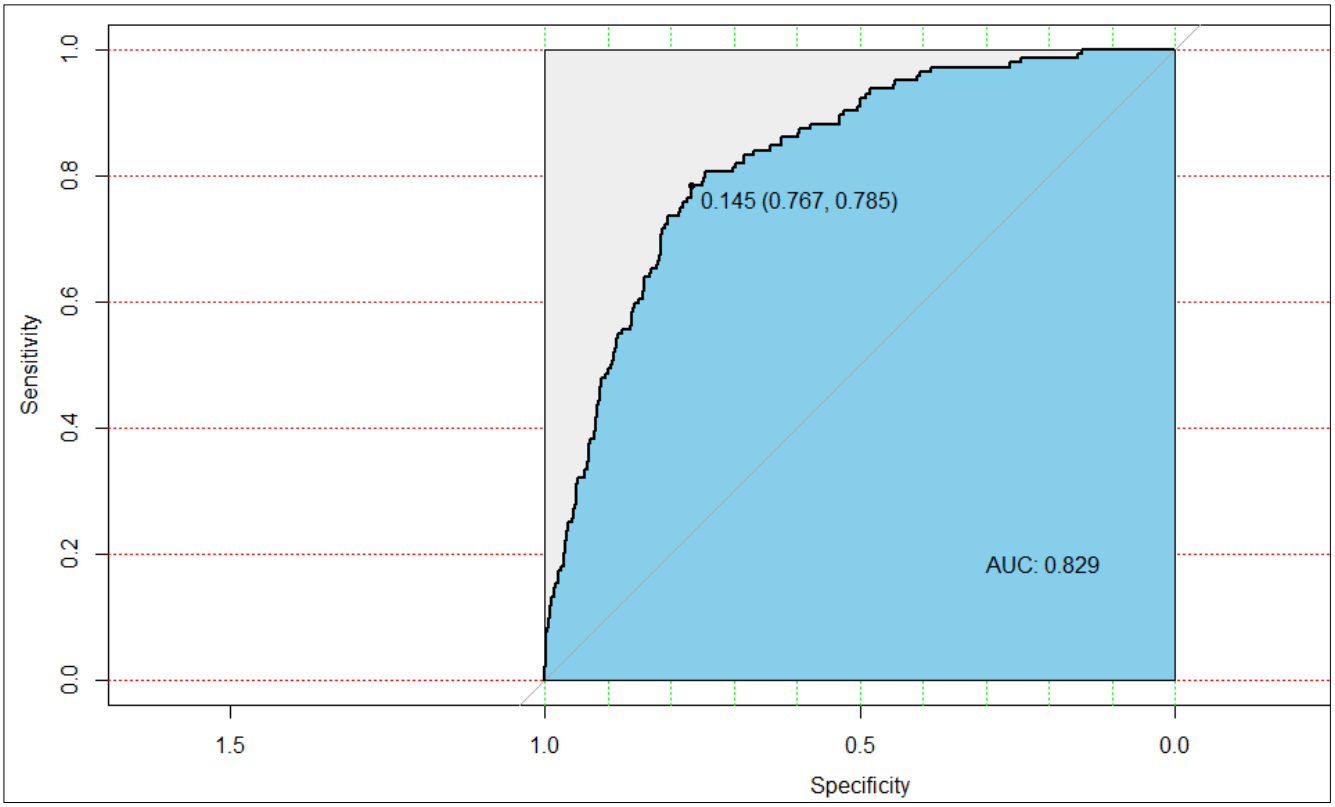
```
[1] 0.4519757
```

The Gini coeff and the KS ratio's are also good for this model

Test set :



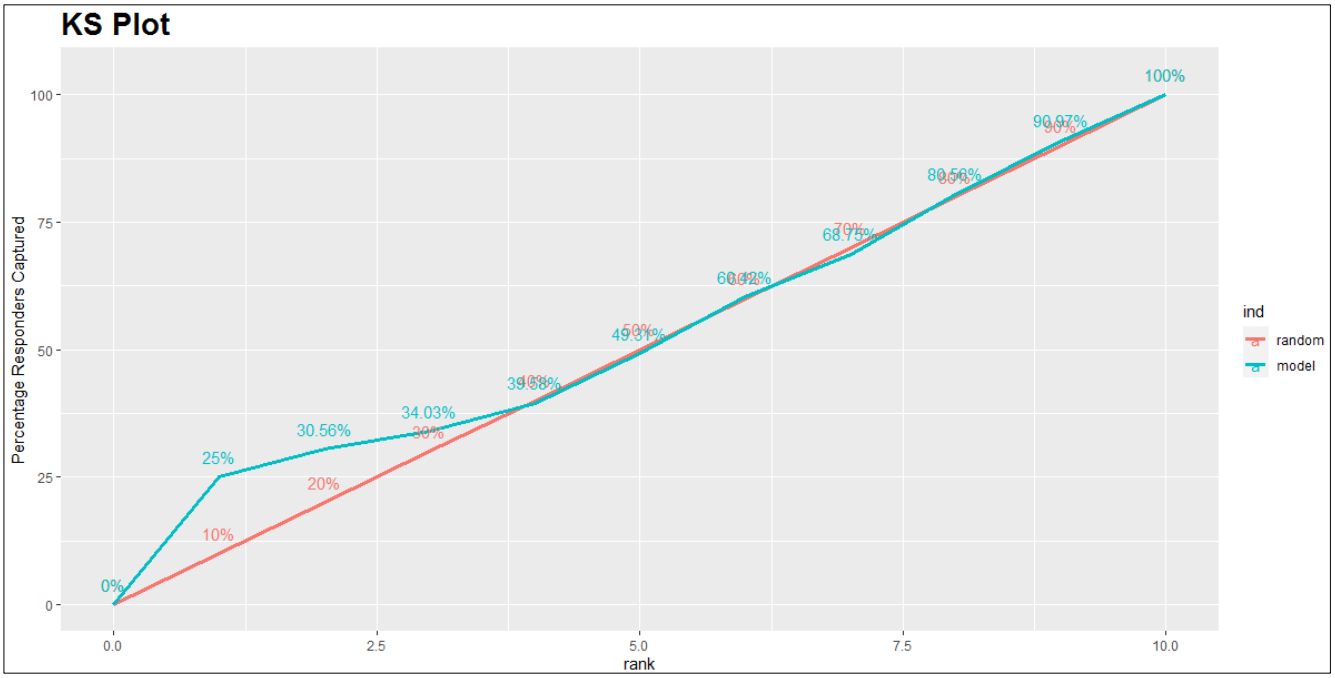




KS Statistics

rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders
1	100	64	36	14.41441	0.25000000	0.07485380
2	100	92	8	14.41441	0.05555556	0.10760234
3	100	95	5	14.41441	0.03472222	0.11111111
4	100	92	8	14.41441	0.05555556	0.10760234
5	100	86	14	14.41441	0.09722222	0.10058480
6	100	84	16	14.41441	0.11111111	0.09824561
7	100	88	12	14.41441	0.08333333	0.10292398
8	100	83	17	14.41441	0.11805556	0.09707602
9	100	85	15	14.41441	0.10416667	0.09941520
10	99	86	13	14.27027	0.09027778	0.10058480

cum_perc_responders	cum_perc_non_responders	difference
0.2500000	0.0748538	0.175146199
0.3055556	0.1824561	0.123099415
0.3402778	0.2935673	0.046710526
0.3958333	0.4011696	-0.005336257
0.4930556	0.5017544	-0.008698830
0.6041667	0.6000000	0.004166667
0.6875000	0.7029240	-0.015423977
0.8055556	0.8000000	0.005555556
0.9097222	0.8994152	0.010307018
1.0000000	1.0000000	0.000000000



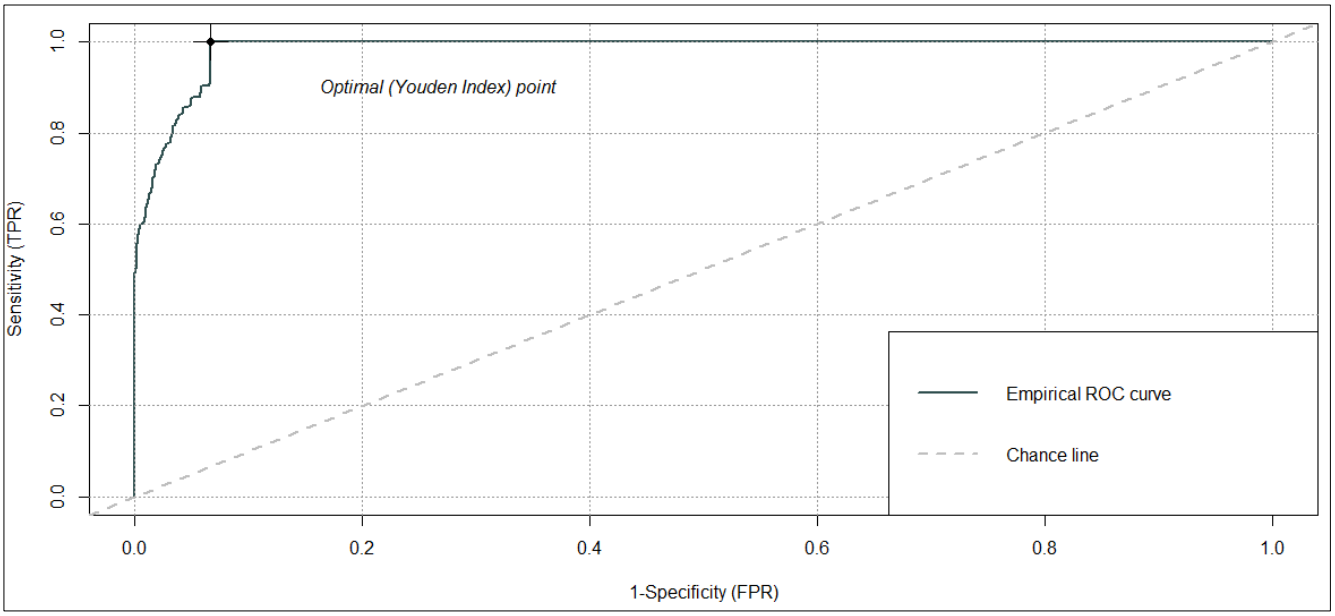
Gini coefficient:

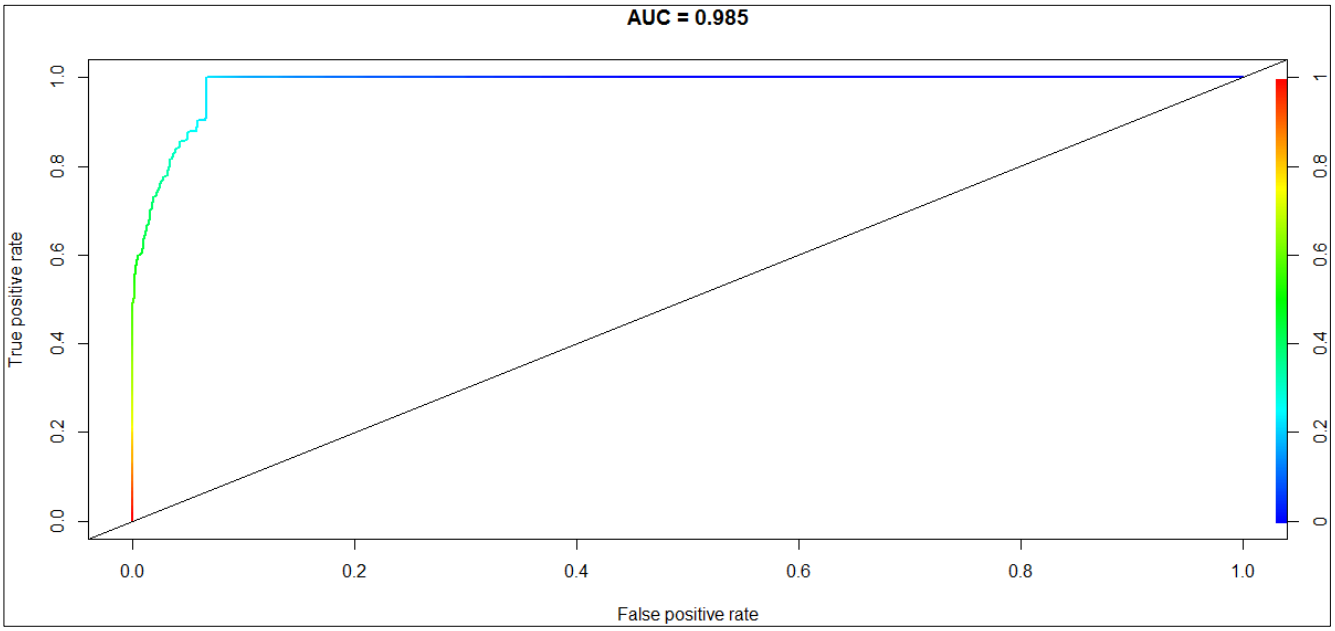
```
[1] 0.4728566
```

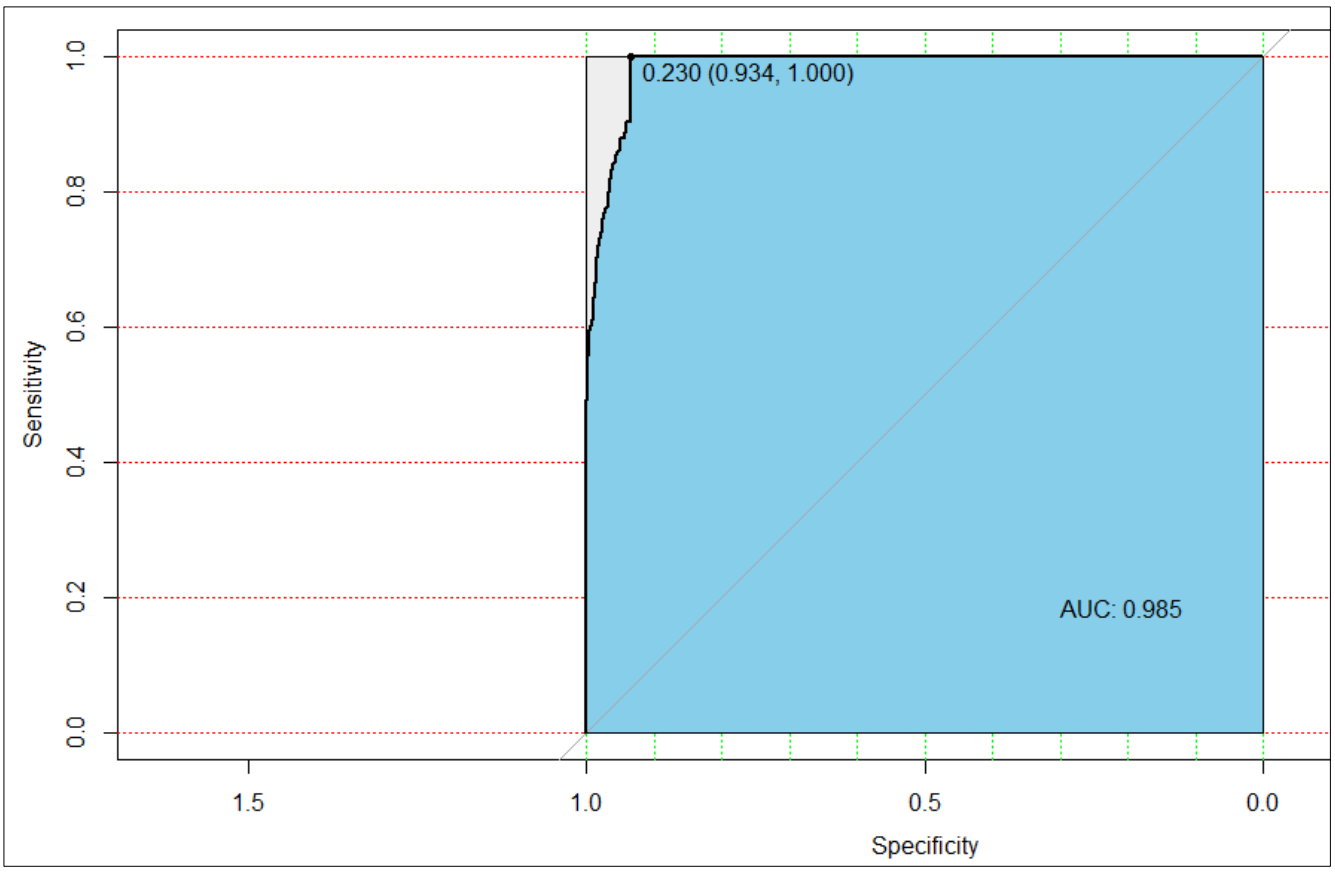
Both the gini coefficient and the K statistics show a good value

K - Nearest Neighbour

Train





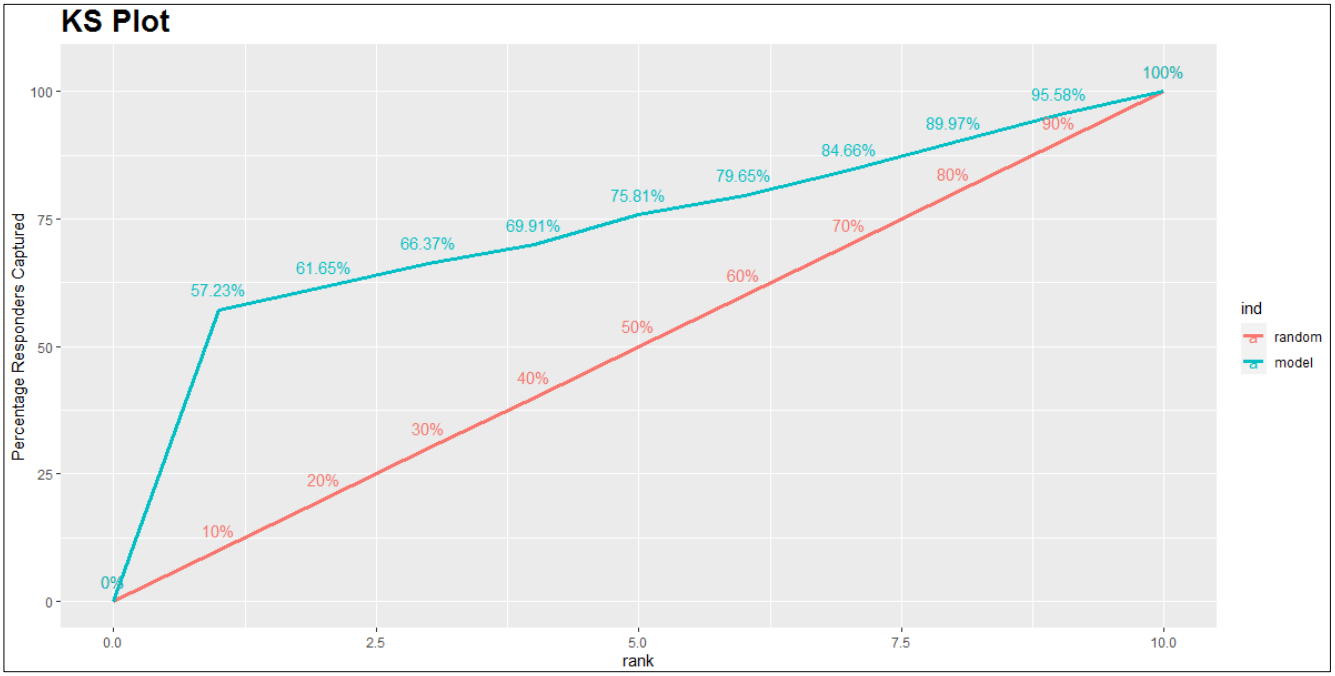


We could see that the AUC value is 98.5 % which is very high which shows that the k-nn is slightly overfit model in predicting the specificity and sensitivity.

KS Statistics :

rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders
1	233	39	194	33.84190	0.57227139	0.01954887
2	233	218	15	33.84190	0.04424779	0.10927318
3	233	217	16	33.84190	0.04719764	0.10877193
4	233	221	12	33.84190	0.03539823	0.11077694
5	233	213	20	33.84190	0.05899705	0.10676692
6	233	220	13	33.84190	0.03834808	0.11027569
7	233	216	17	33.84190	0.05014749	0.10827068
8	233	215	18	33.84190	0.05309735	0.10776942
9	233	214	19	33.84190	0.05604720	0.10726817
10	237	222	15	34.42288	0.04424779	0.11127820

cum_perc_responders	cum_perc_non_responders	difference
0.5722714	0.01954887	0.55272251
0.6165192	0.12882206	0.48769712
0.6637168	0.23759398	0.42612283
0.6991150	0.34837093	0.35074412
0.7581121	0.45513784	0.30297425
0.7964602	0.56541353	0.23104664
0.8466077	0.67368421	0.17292346
0.8997050	0.78145363	0.11825138
0.9557522	0.88872180	0.06703041
1.0000000	1.00000000	0.00000000



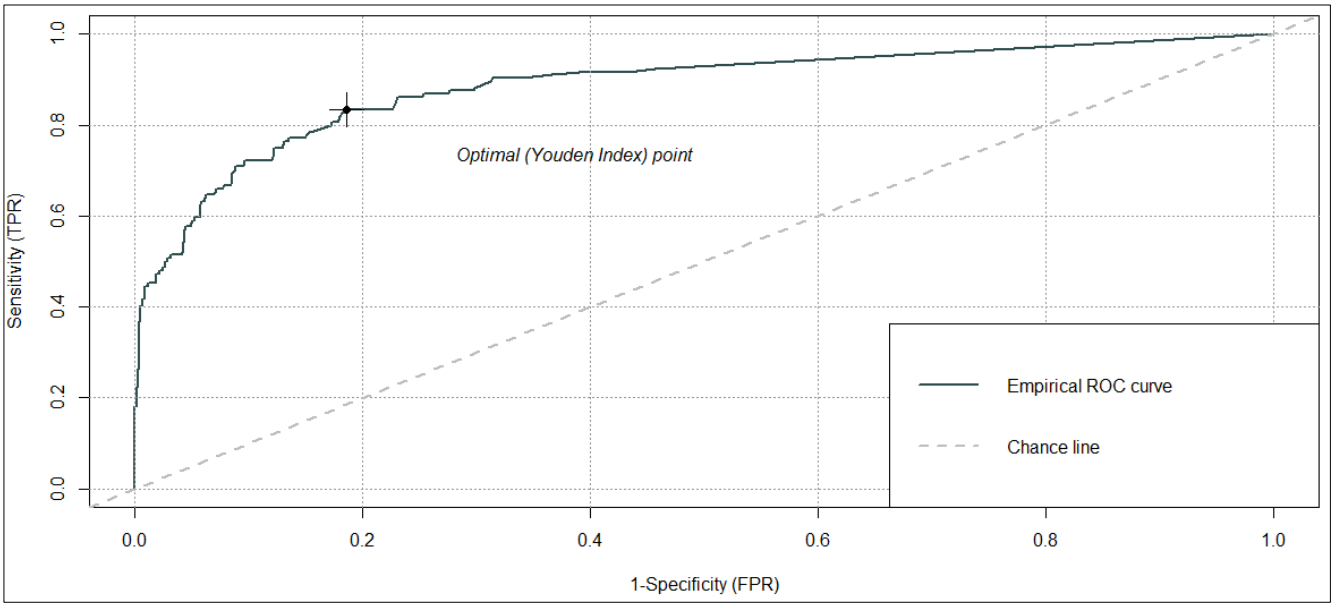
We could see that the gap between random and model curve is high at 57% of percentage response captured

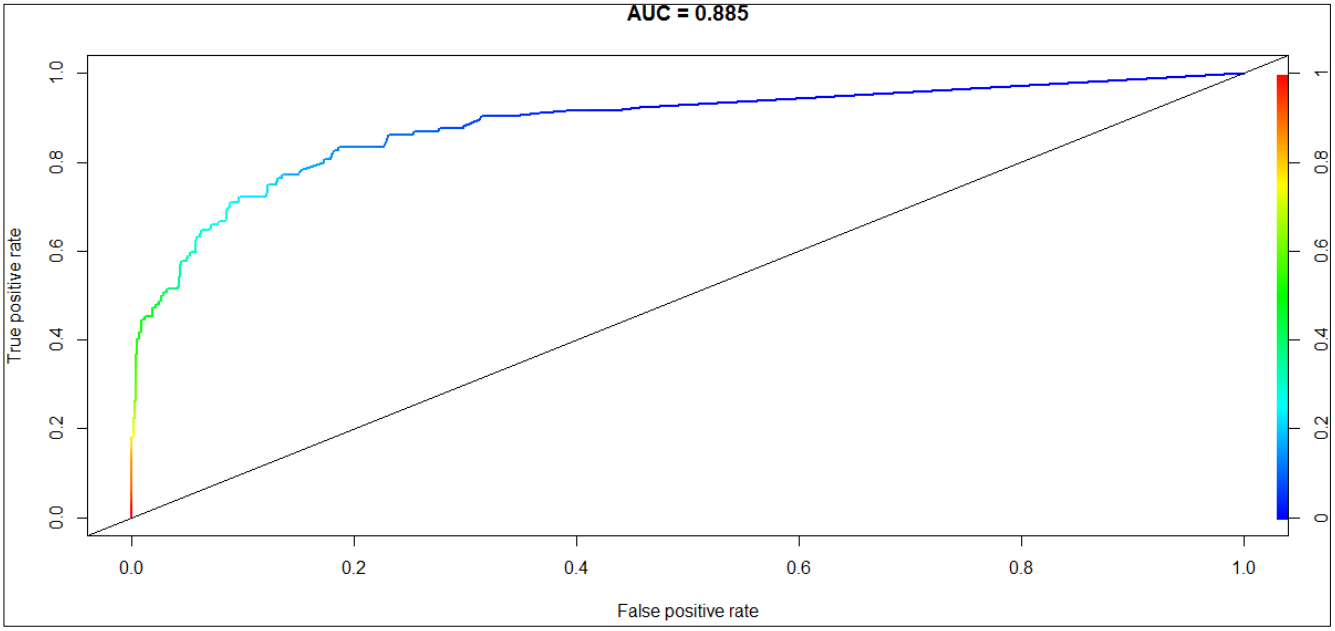
Gini coeff

```
[1] 0.8925492
```

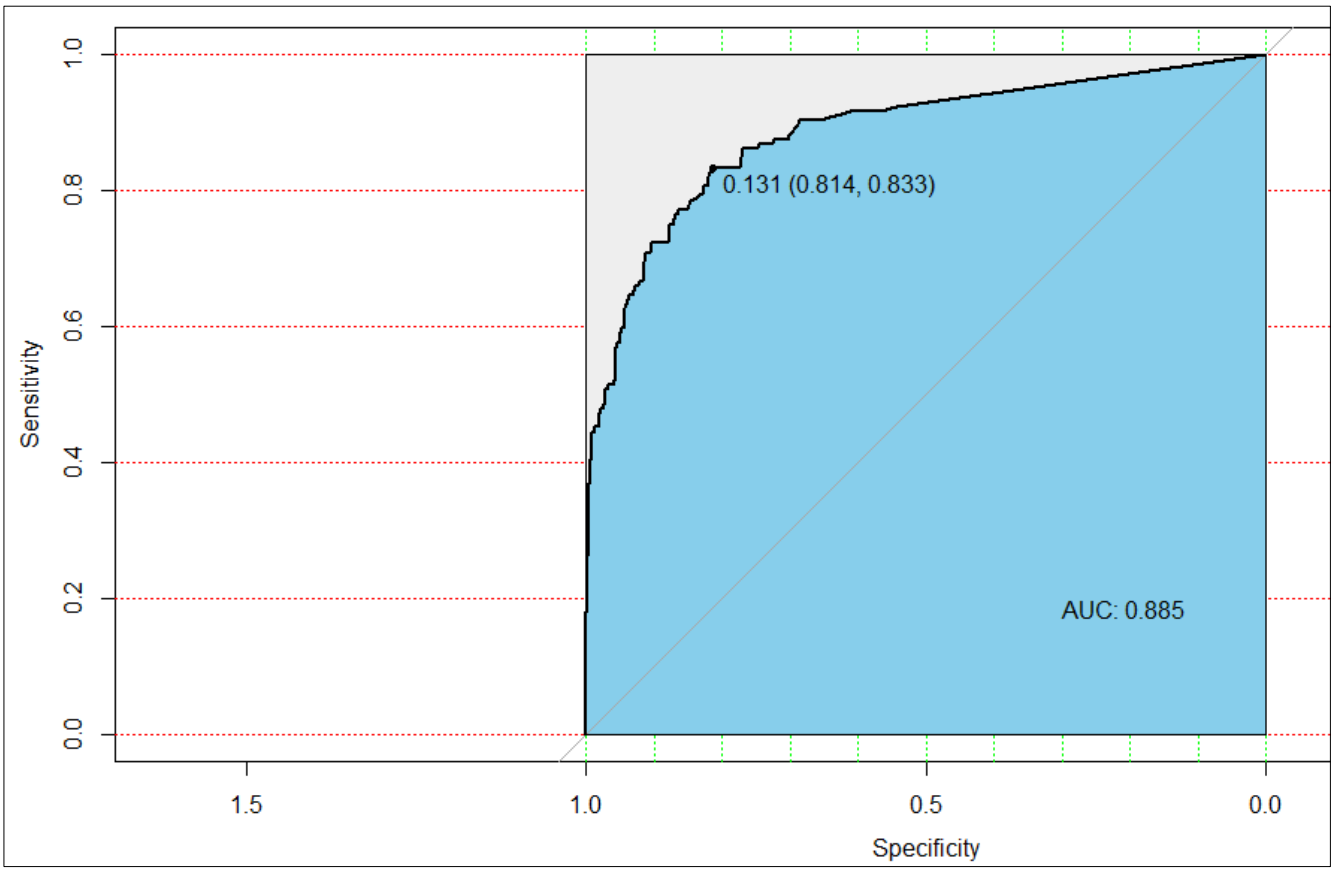
The Gini coefficient value is very high due to high overfit model. This is due to fact that the k – nn has been run on train set and then validated with same train set.

Test





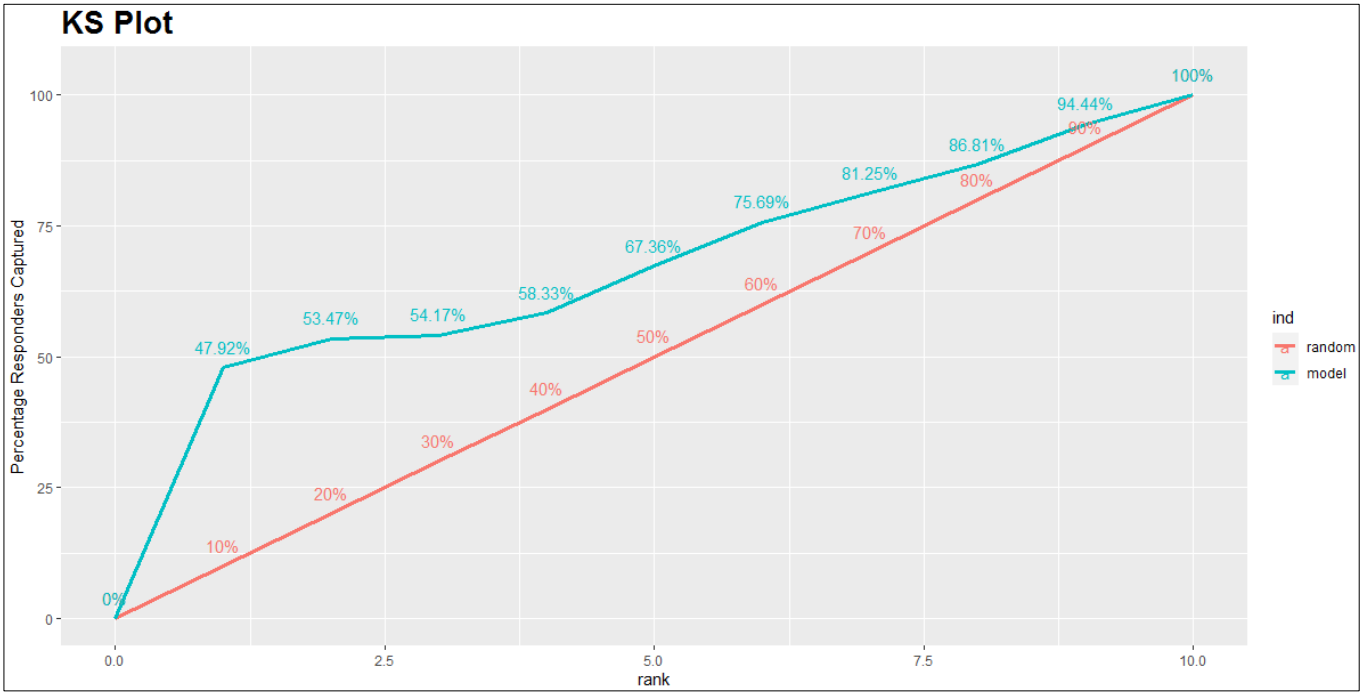
In case of K-nn test set, the model has performed better with auc value of 88.5 % which is higher compared to the other two models hence the auc and roc for k-nn test set has highest accuracy.



KS

rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders
1	100	31	69	14.41441	0.479166667	0.03625731
2	100	92	8	14.41441	0.055555556	0.10760234
3	100	99	1	14.41441	0.006944444	0.11578947
4	100	94	6	14.41441	0.041666667	0.10994152
5	100	87	13	14.41441	0.090277778	0.10175439
6	100	88	12	14.41441	0.083333333	0.10292398
7	100	92	8	14.41441	0.055555556	0.10760234
8	100	92	8	14.41441	0.055555556	0.10760234
9	100	89	11	14.41441	0.076388889	0.10409357
10	99	91	8	14.27027	0.055555556	0.10643275

cum_perc_responders	cum_perc_non_responders	difference
0.4791667	0.03625731	0.44290936
0.5347222	0.14385965	0.39086257
0.5416667	0.25964912	0.28201754
0.5833333	0.36959064	0.21374269
0.6736111	0.47134503	0.20226608
0.7569444	0.57426901	0.18267544
0.8125000	0.68187135	0.13062865
0.8680556	0.78947368	0.07858187
0.9444444	0.89356725	0.05087719
1.0000000	1.00000000	0.00000000



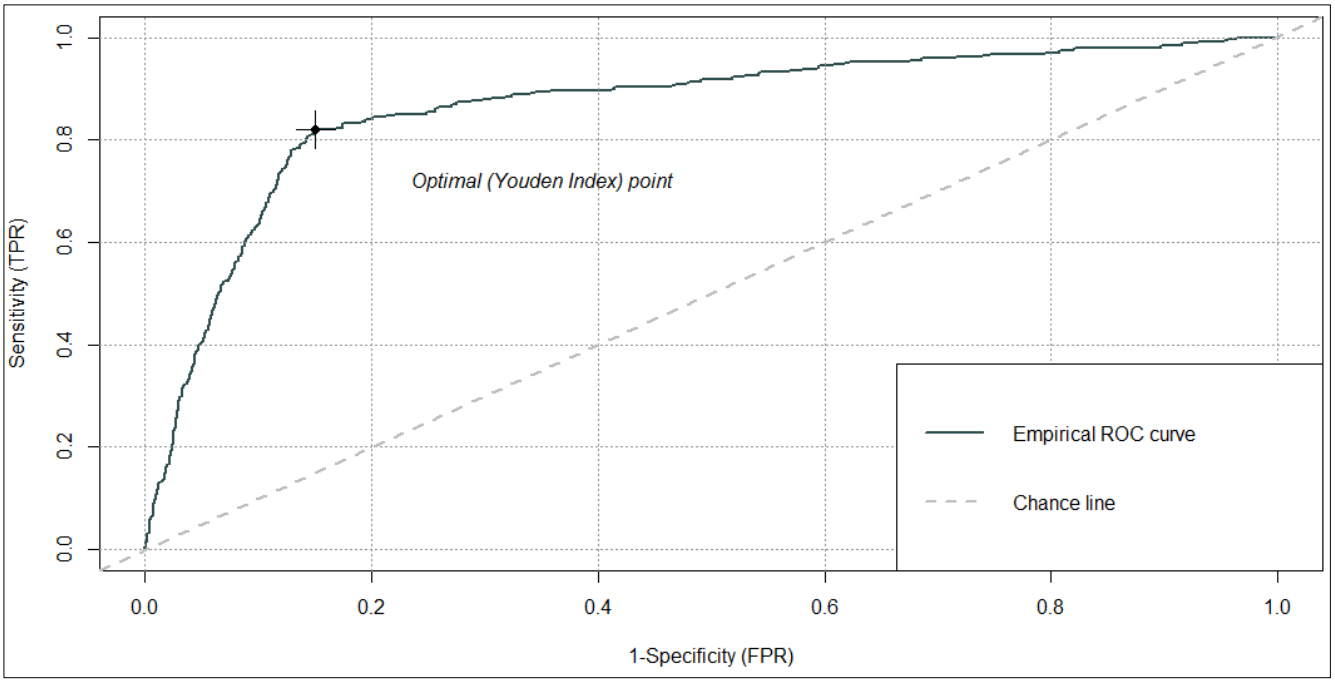
Gini Coefficient

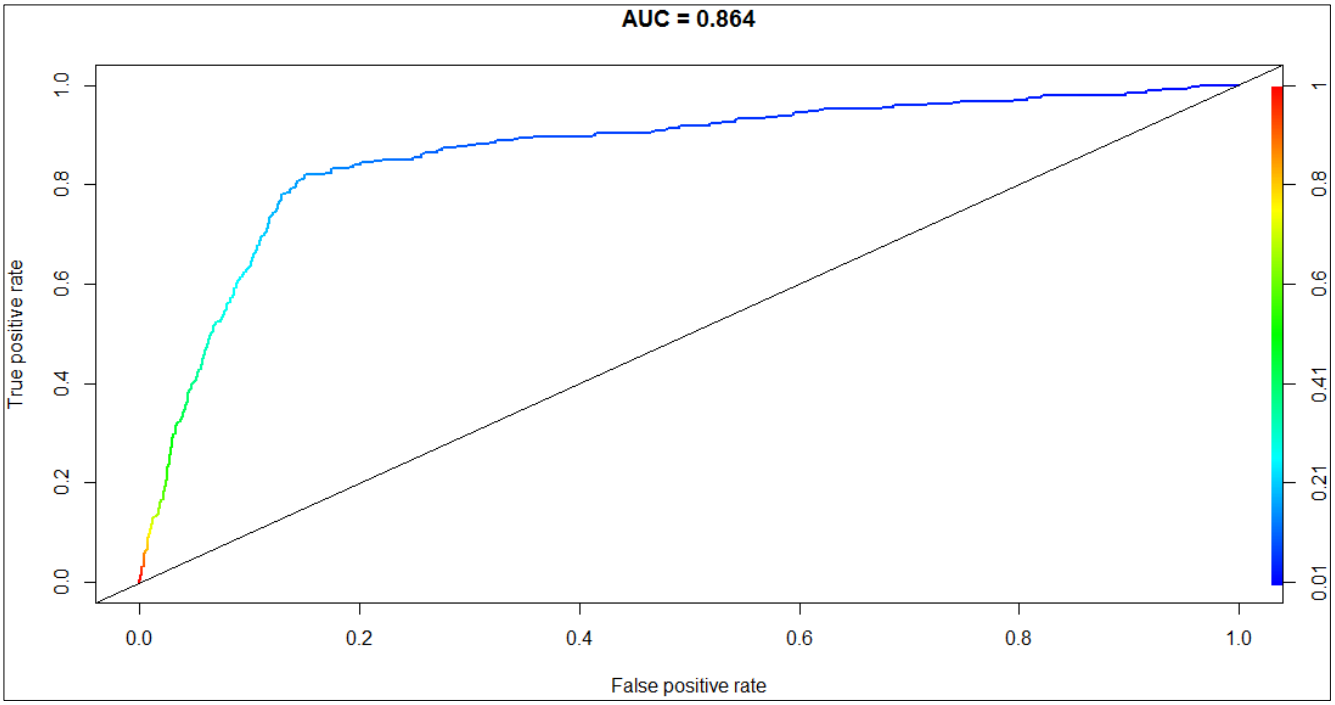
The gini coefficient is high compared to the other two models which shows that this model is very robust in performance.

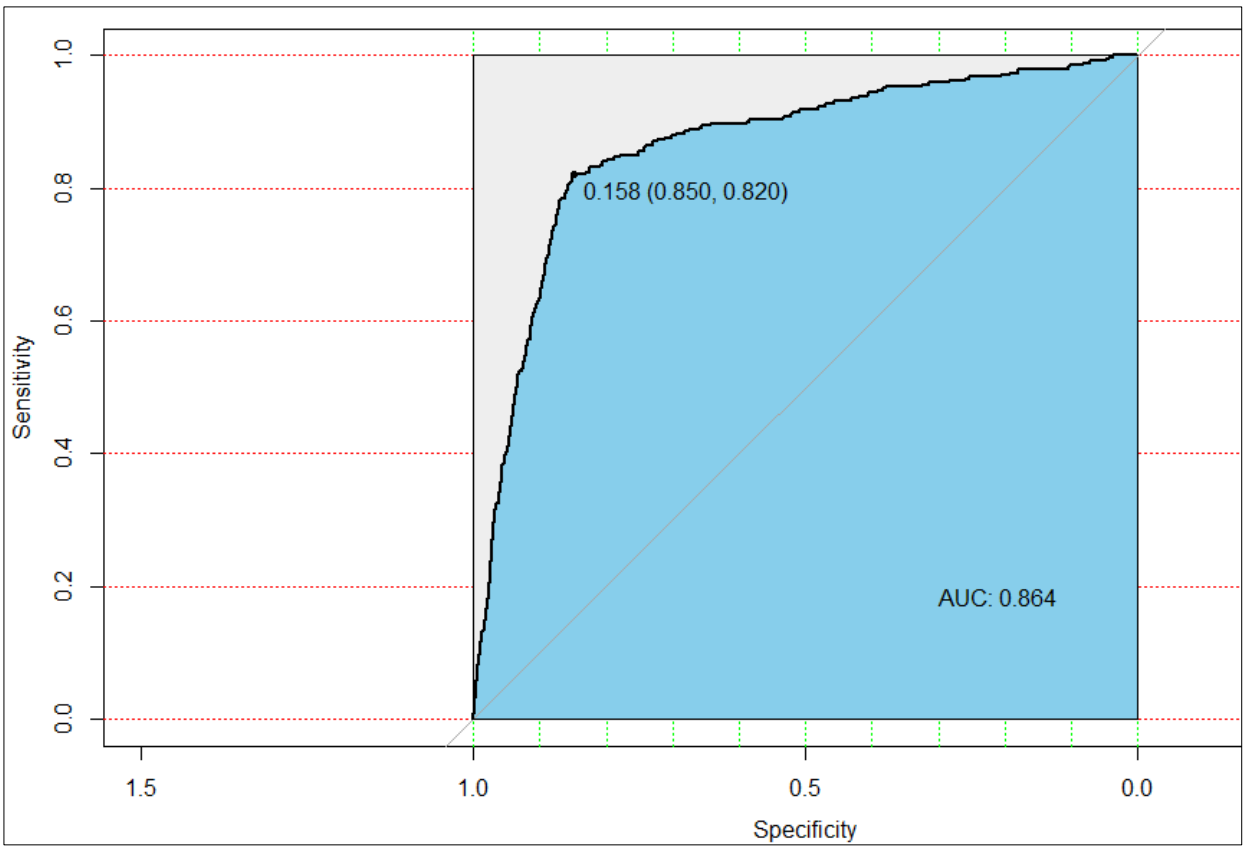
```
[1] 0.8052453
```

Naïve Bayes :

Train set :



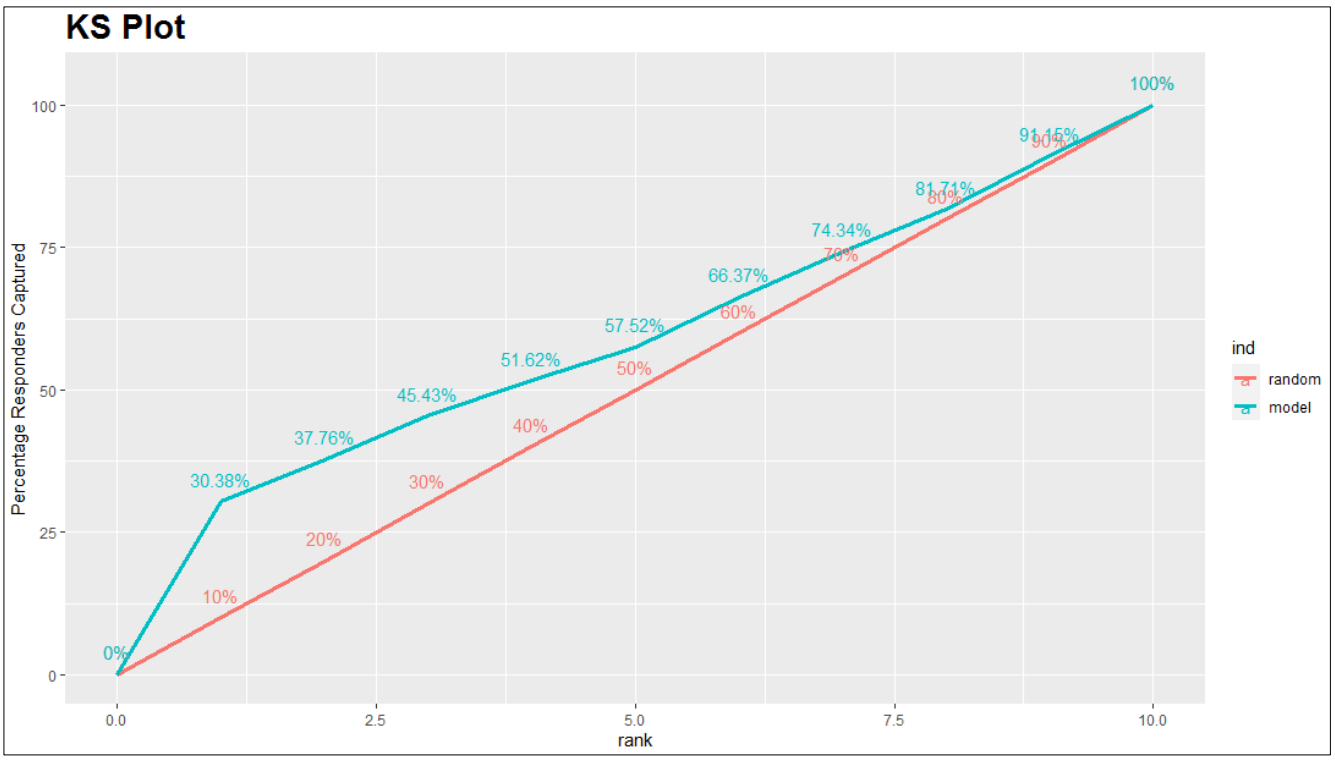




KS Statistics :

	rank	total_pop	non_responders	responders e	xpected_responders_by_random	perc_responders	perc_non_responders
1	1	233	130	103	33.84190	0.30383481	0.06516291
2	2	233	208	25	33.84190	0.07374631	0.10426065
3	3	233	207	26	33.84190	0.07669617	0.10375940
4	4	233	212	21	33.84190	0.06194690	0.10626566
5	5	233	213	20	33.84190	0.05899705	0.10676692
6	6	233	203	30	33.84190	0.08849558	0.10175439
7	7	233	206	27	33.84190	0.07964602	0.10325815
8	8	233	208	25	33.84190	0.07374631	0.10426065
9	9	233	201	32	33.84190	0.09439528	0.10075188
10	10	237	207	30	34.42288	0.08849558	0.10375940

	cum_perc_responders	cum_perc_non_responders	difference
1	0.3038348	0.06516291	0.23867190
2	0.3775811	0.16942356	0.20815756
3	0.4542773	0.27318296	0.18109433
4	0.5162242	0.37944862	0.13677557
5	0.5752212	0.48621554	0.08900570
6	0.6637168	0.58796992	0.07574689
7	0.7433628	0.69122807	0.05213476
8	0.8171091	0.79548872	0.02162042
9	0.9115044	0.89624060	0.01526382
10	1.0000000	1.00000000	0.00000000

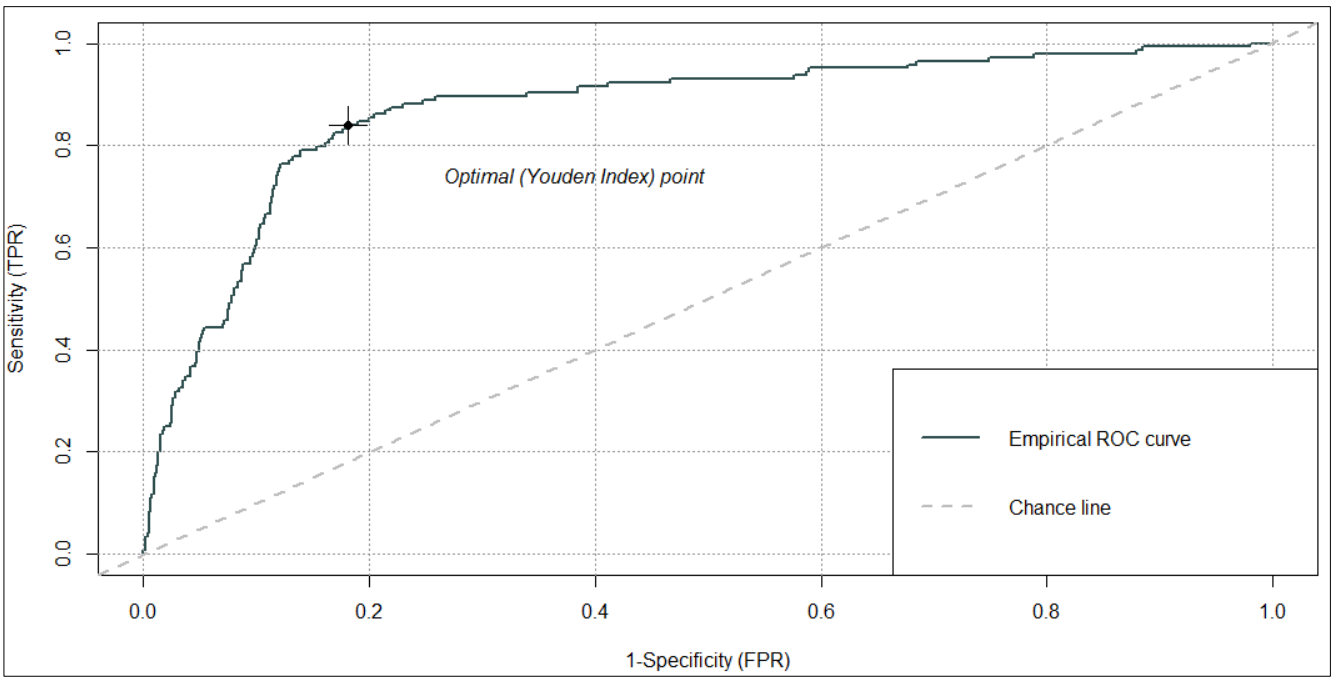


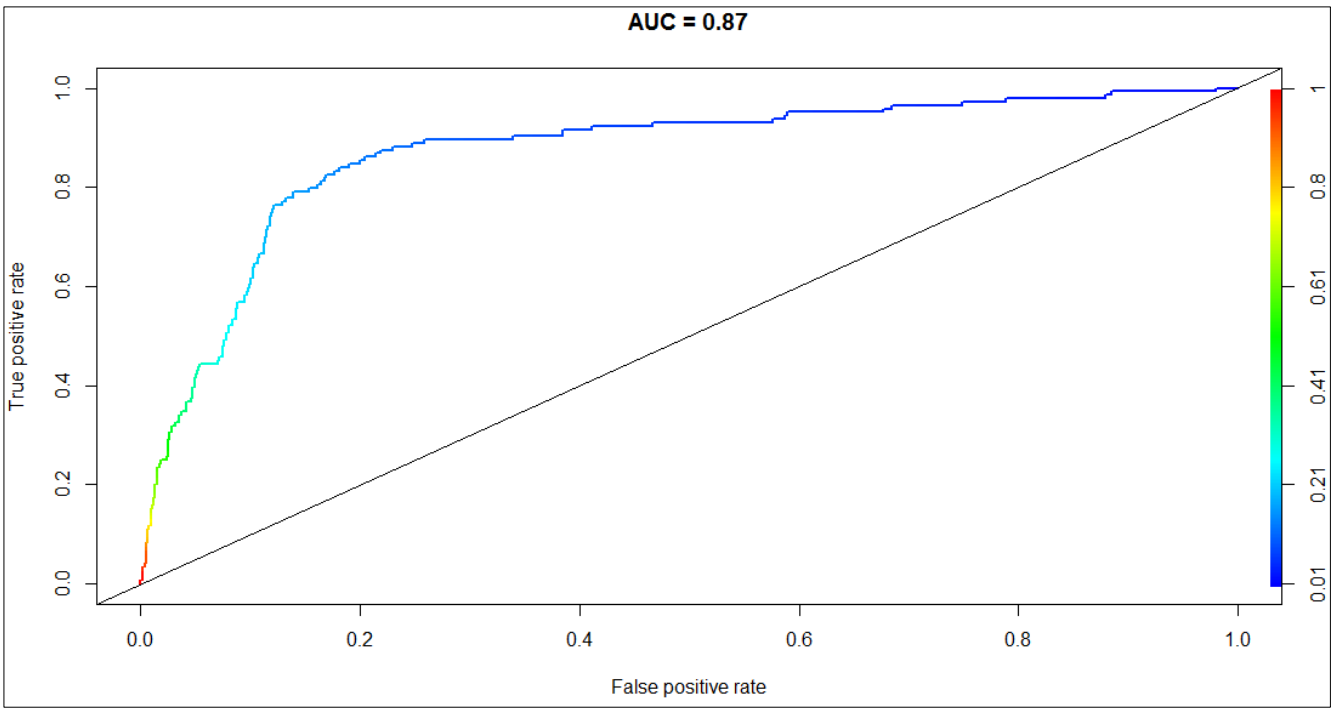
Gini Coefficient :

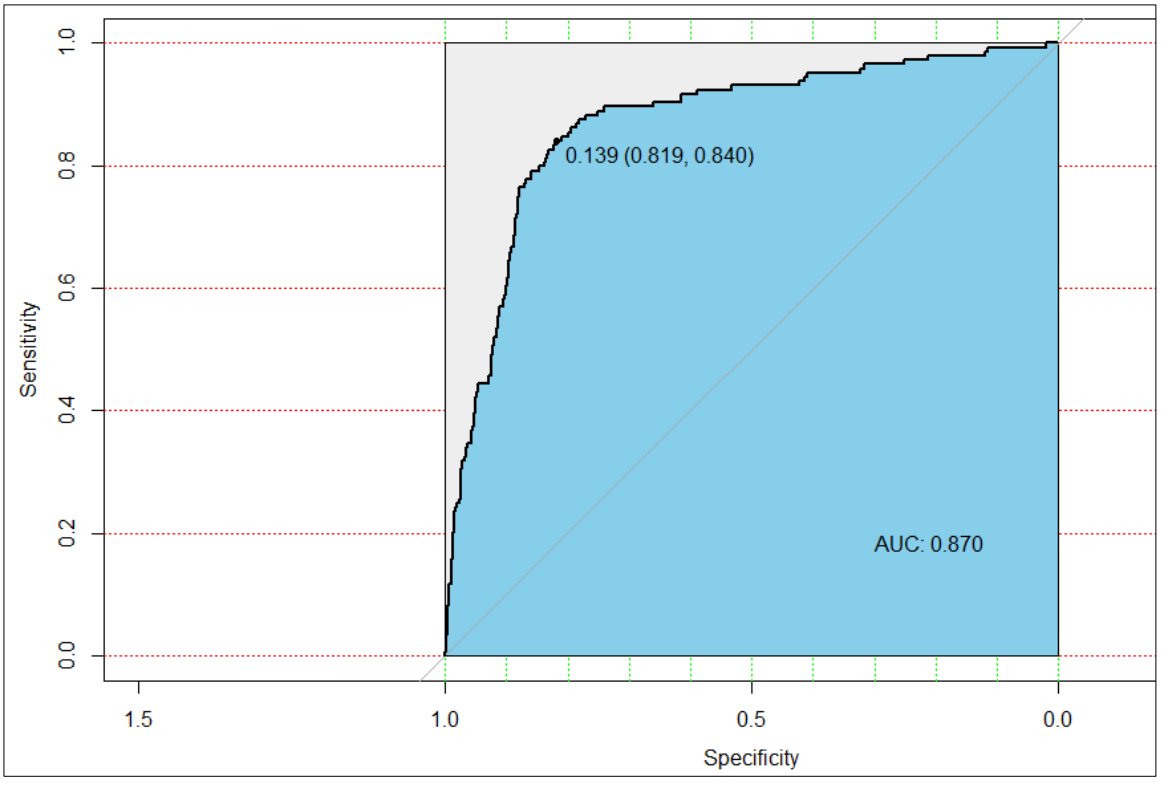
The gini coefficient and KS statistics is good for naïve bayes model

```
[1] 0.4931798
```

Test set :



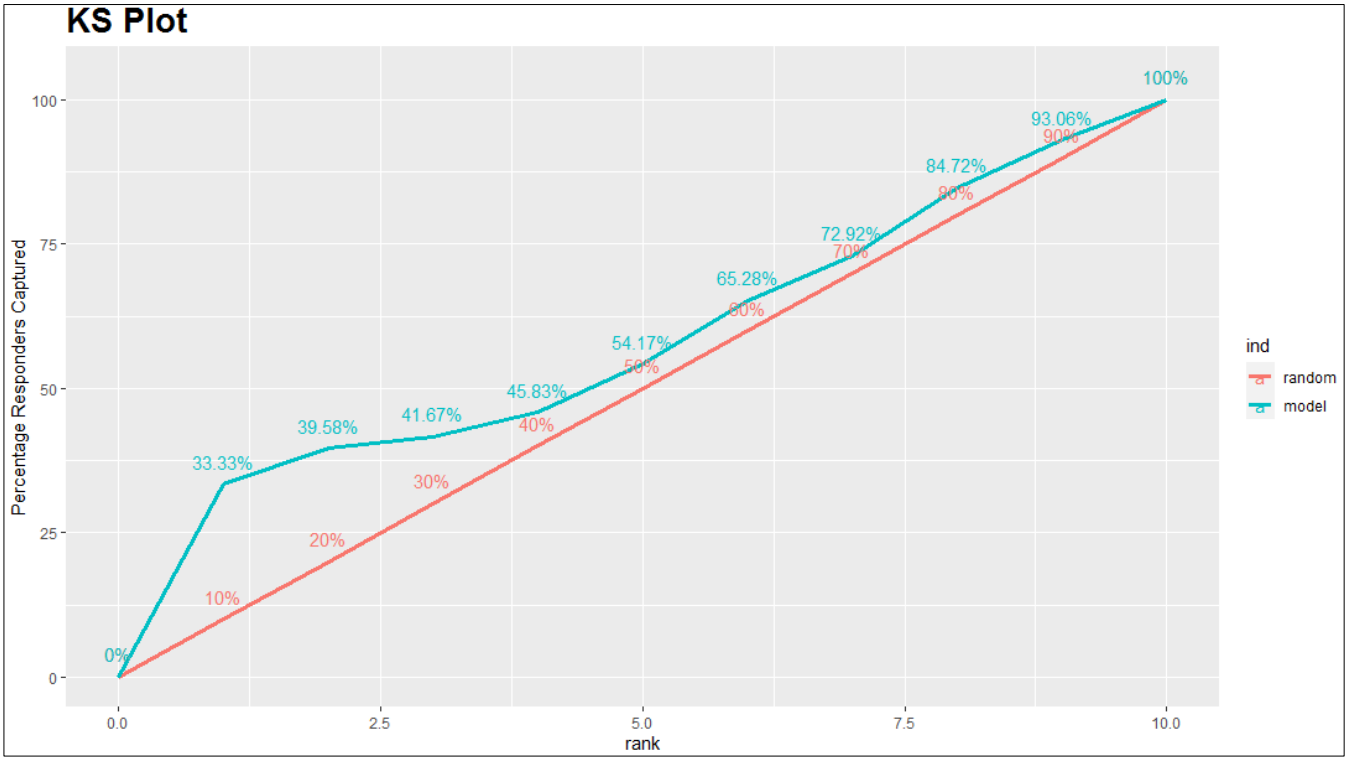




KS Statistics :

rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders
1	100	52	48	14.41441	0.33333333	0.06081871
2	100	91	9	14.41441	0.06250000	0.10643275
3	100	97	3	14.41441	0.02083333	0.11345029
4	100	94	6	14.41441	0.04166667	0.10994152
5	100	88	12	14.41441	0.08333333	0.10292398
6	100	84	16	14.41441	0.11111111	0.09824561
7	100	89	11	14.41441	0.07638889	0.10409357
8	100	83	17	14.41441	0.11805556	0.09707602
9	100	88	12	14.41441	0.08333333	0.10292398
10	99	89	10	14.27027	0.06944444	0.10409357

cum_perc_responders	cum_perc_non_responders	difference
0.3333333	0.06081871	0.27251462
0.3958333	0.16725146	0.22858187
0.4166667	0.28070175	0.13596491
0.4583333	0.39064327	0.06769006
0.5416667	0.49356725	0.04809942
0.6527778	0.59181287	0.06096491
0.7291667	0.69590643	0.03326023
0.8472222	0.79298246	0.05423977
0.9305556	0.89590643	0.03464912
1.0000000	1.00000000	0.00000000



Gini

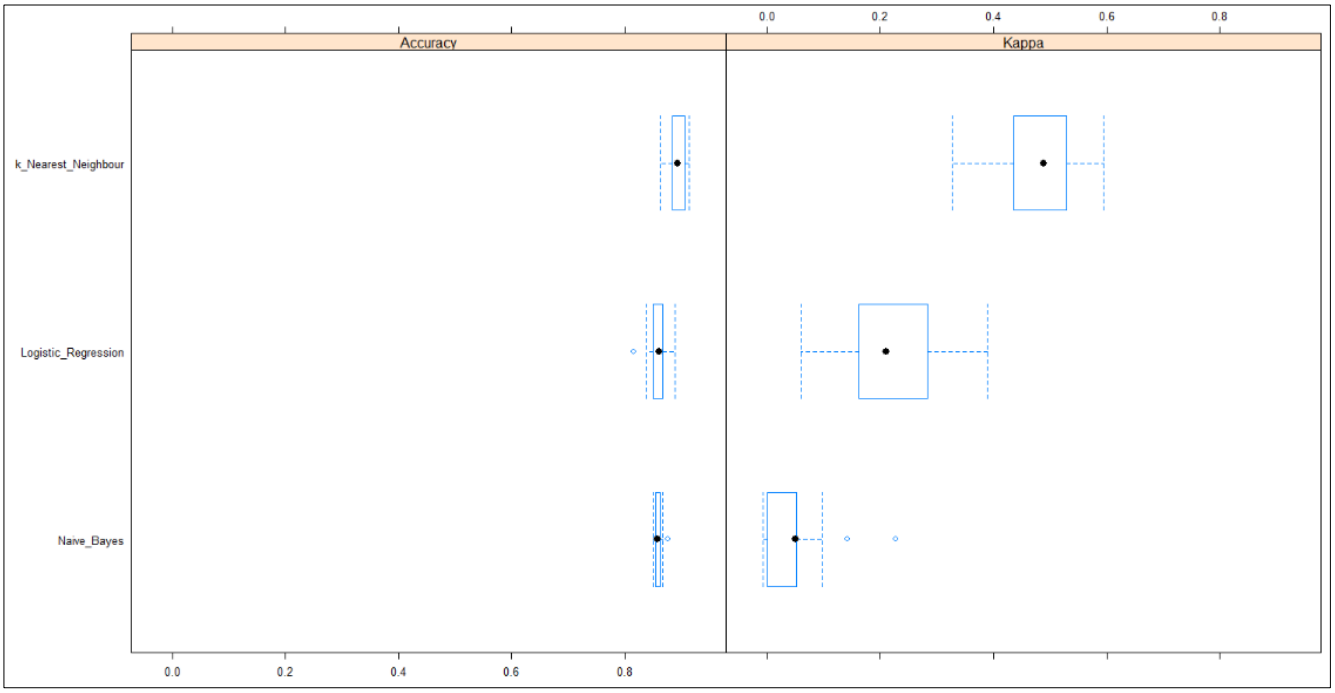
```
[1] 0.5731401
```

9 Remarks on Model validation exercise

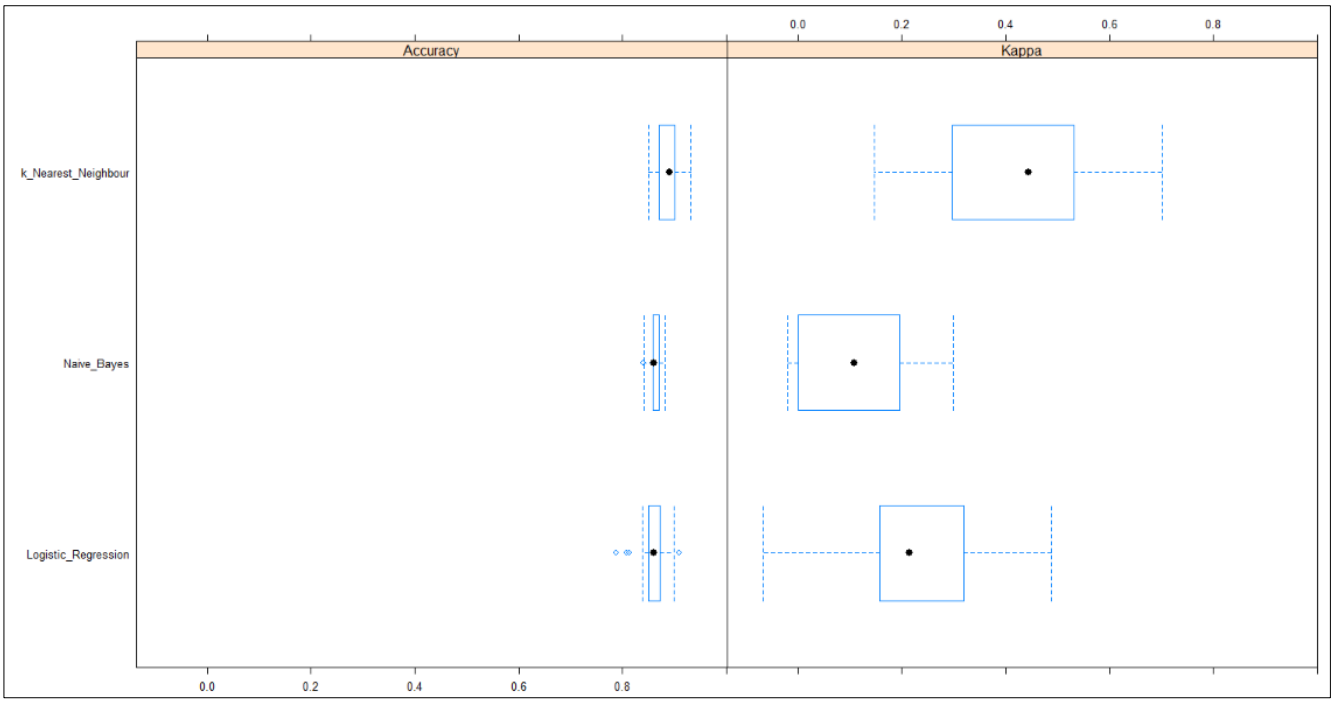
Comparing all the three models the K-nn seems to be best optimized and has high accuracy rate for training dataset compared to the other two. Also other values such as precision, F1 score and kappa coefficients are better for the k-nn model. Hence we can say that the k-nn is best suitable for our classification analysis.

Cross validated models:

Training set



Test set



Train

	Logistic Regression	K – Nearest Neighbour	Naïve Bayes	Comments
Accuracy	86.42	**94.73**	85.95	
Sensitivity	20.06	**65.782**	3.5	Low Type II Error
Specificity	97.69	99.65	**99.99**	Higher value of true negative and lower false positive rate
Precision	59.65	**96.95**	92.3	Low Type I Error
F1 score	30	**78.383**	6.8	
Kappa	24.5	**75.51**	5.8	

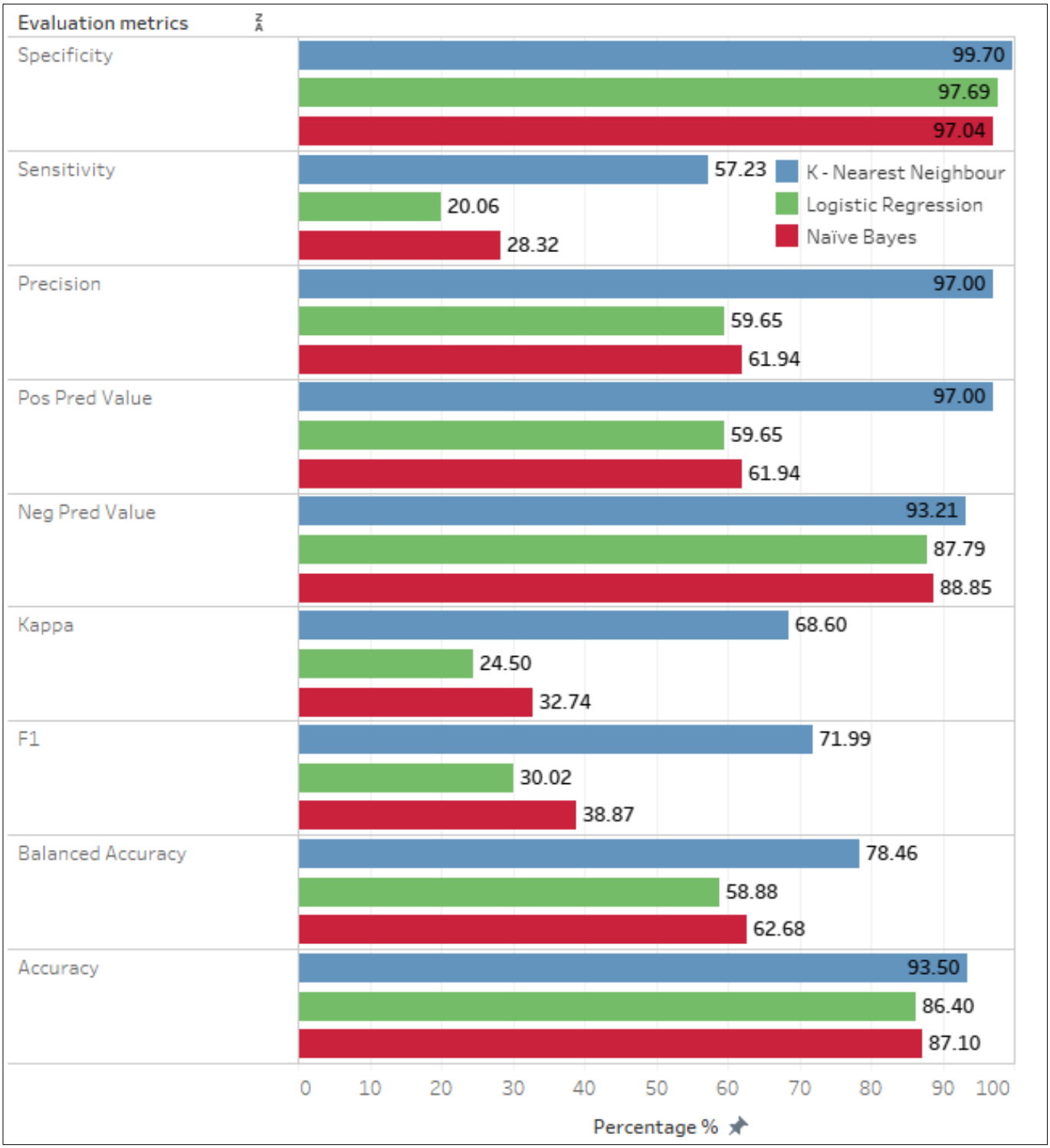
Test

	Logistic Regression	K – Nearest Neighbour	Naïve Bayes	Comments
Accuracy	85.89	**94.99**	86.79	
Sensitivity	20.14	**68**	8.33	Low Type II Error
Specificity	96.95	99.53	**100**	Higher value of true negative and lower false positive rate
Precision	52.72	96	**100**	Low Type I Error
F1 score	29.14	**79.6**	15.38	
Kappa	23	**76.92**	13.47	

For the Train dataset, k - nn still maintains high accuracy but specificity is slightly higher in Naïve bayes model compared to k-nn. Logistic regression on the other hand has very slight less accuracy compared to other two. Comparitively all the three models have nearly same difference in accuracy and performed well with few percentage differences. In case of Naïve Bayes Specificity and precision is high for test data set for cross validated models.

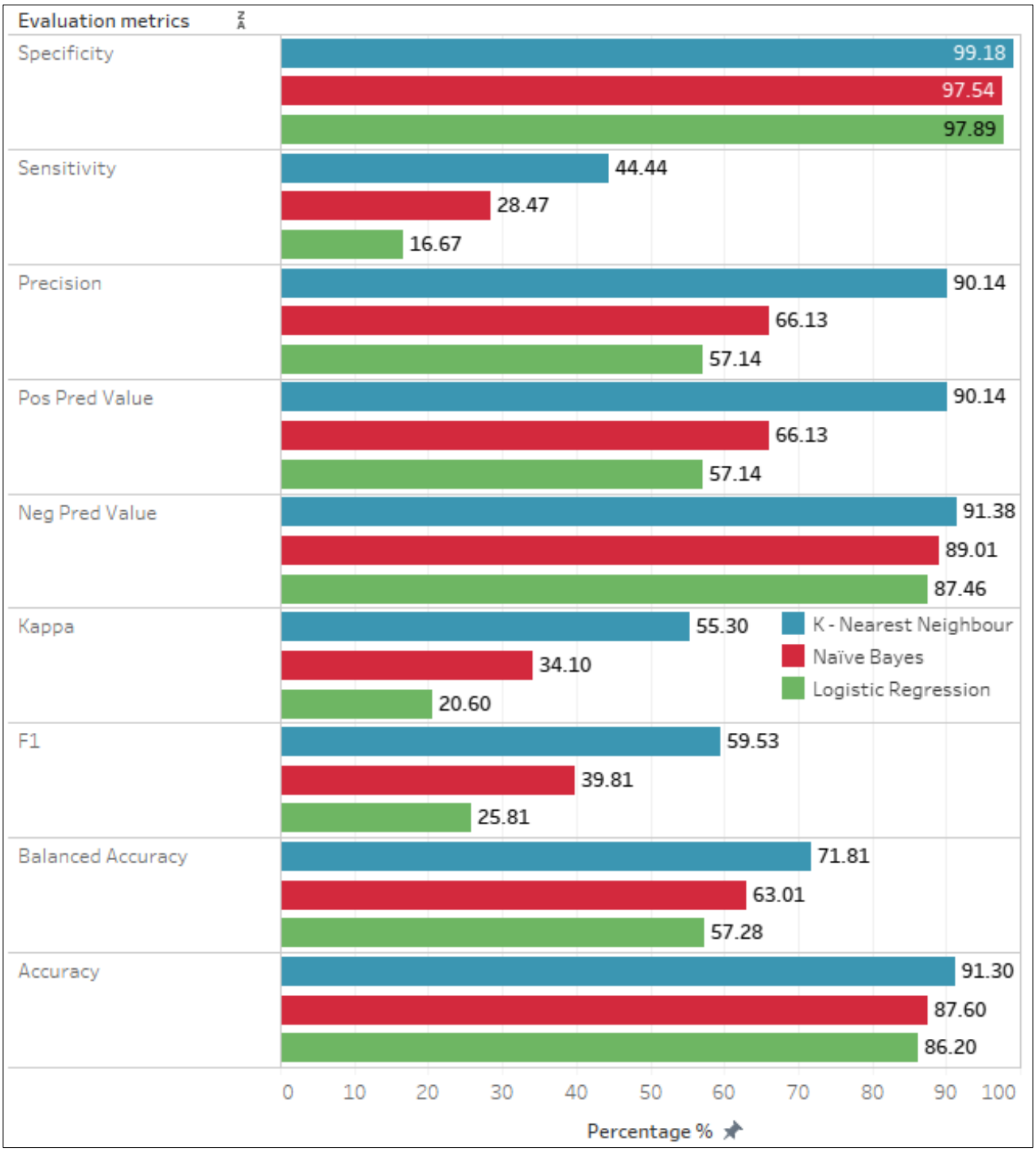
Next we are comparing for the models which are not cross validated.

Train set



K – nn has highest accuracy , F1 m, kappa , Precision , sensitivity and specificity . While logistic regression and naïve bayes performed nearly the same with few percentage difference in the train data sets.

For Testing datasets :



Hence on overall comparison, we can say that the K-NN model has high accuracy and best suited for the classification model. However k – nn is a lazy algorithm with less assumptions and also its non – parametric hence if different dataset is provided its accuracy can reduce. Hence in parametric models compared to logistic and Naïve bayes , Naïve bayes does better job and has good accuracy. Overall all the models performed well with very few percentage differences among the models.

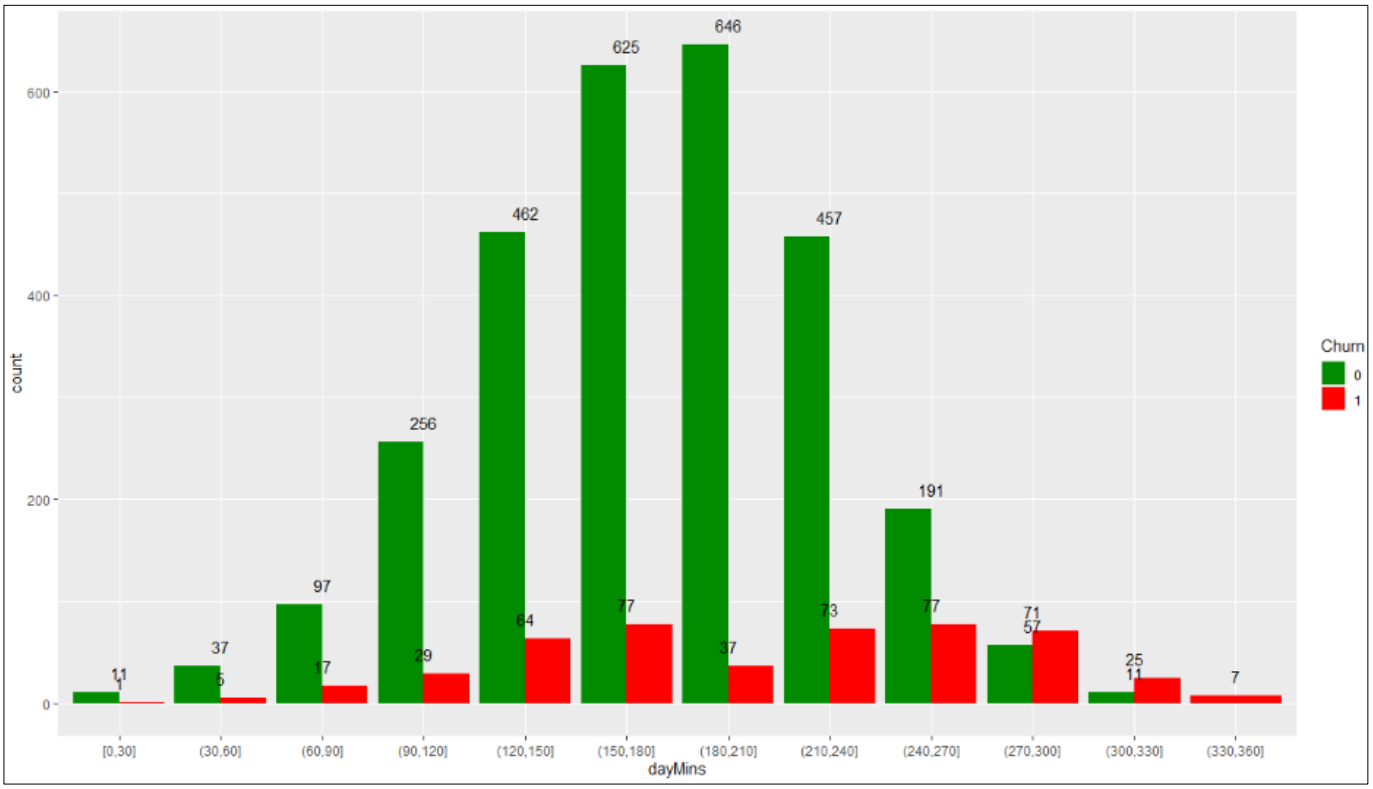
10 Inference, Actionable Insights and Recommendations

From the three machine learning models we have derived the variables or factors that are leading to higher churn or cancellation of service by customers. We could see that the DayMins, Contract renewal, Customer service calls are common for all three models which are contributing to maximum churn. Others like overage fee, DataPlan , monthly bill and roam mins are moderate contributing factor. While accountweeks and DayCall are least contributing to churn.

By plotting the original dataset with churn and non – churn classes ,below is the analysis visualized.

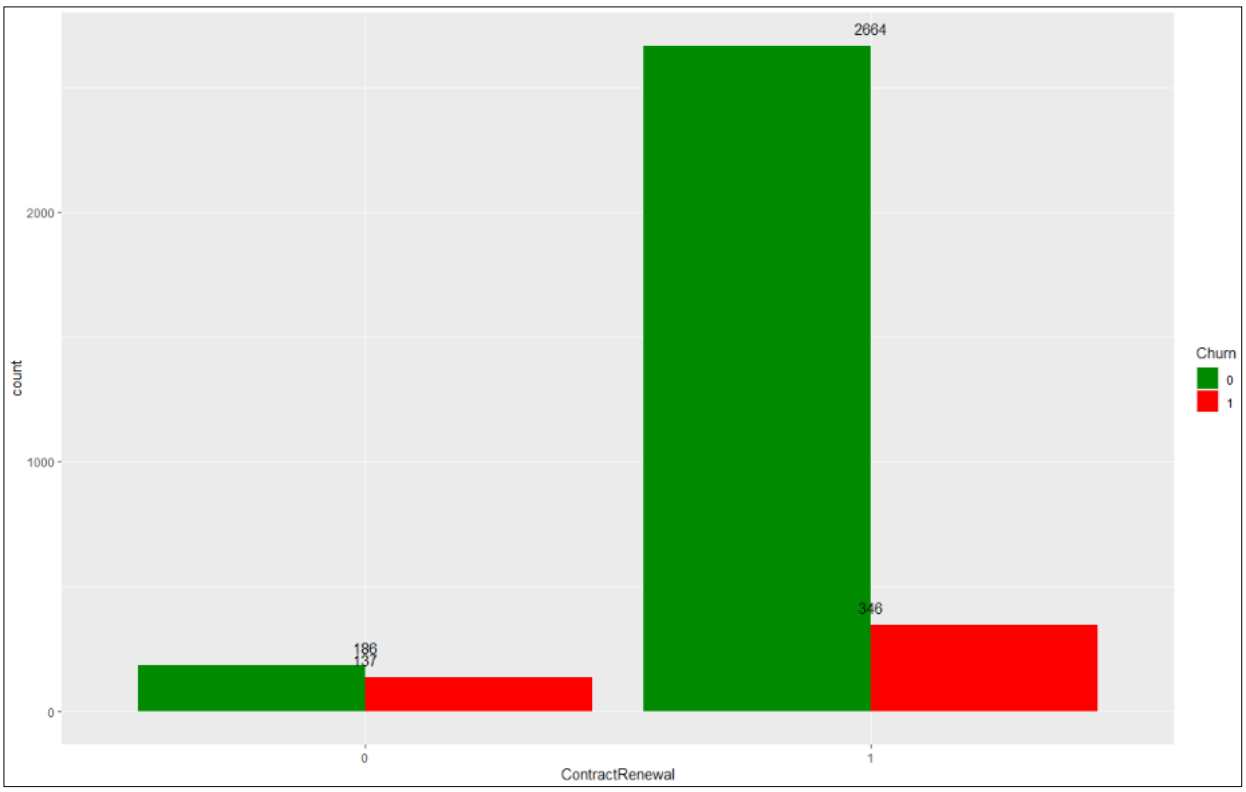
DayMins :

As per the ML models, the average daytime minutes per month held highest importance for churn hence from below graph we can see that from 210 day minutes to 360 minutes (i.e from 3.5 hours to 6 hours), the percentage of churn compared with total is higher. Also we saw earlier in EDA (See section 2 bivariate analysis) that the Daymins and monthly bill is highly correlated. Hence customers who have higher daytime minutes of call per month have higher bill amount hence they are likely to cancel due to higher billing amount due to dissatisfaction.



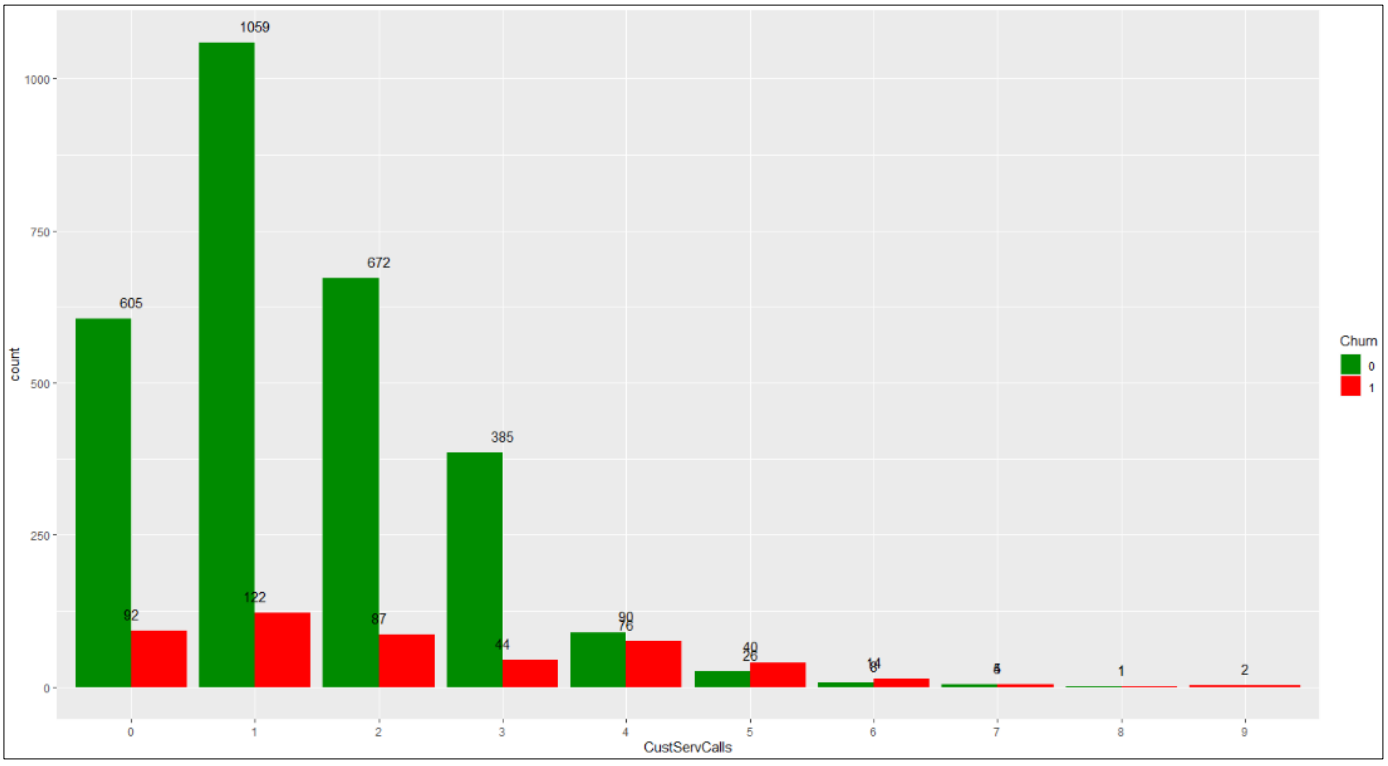
Contract Renewal :

Next most important and highly significant variable is contract renewal which is if the customer recently renewed the contract or not (‘1’ if recently renewed and ‘0’ if not). As per the below visuals we can see that out of the total customers who did not renew the contract (323 total) 42% have churned. Hence we can say that the maximum churn is from the customers who have not renewed the contract.



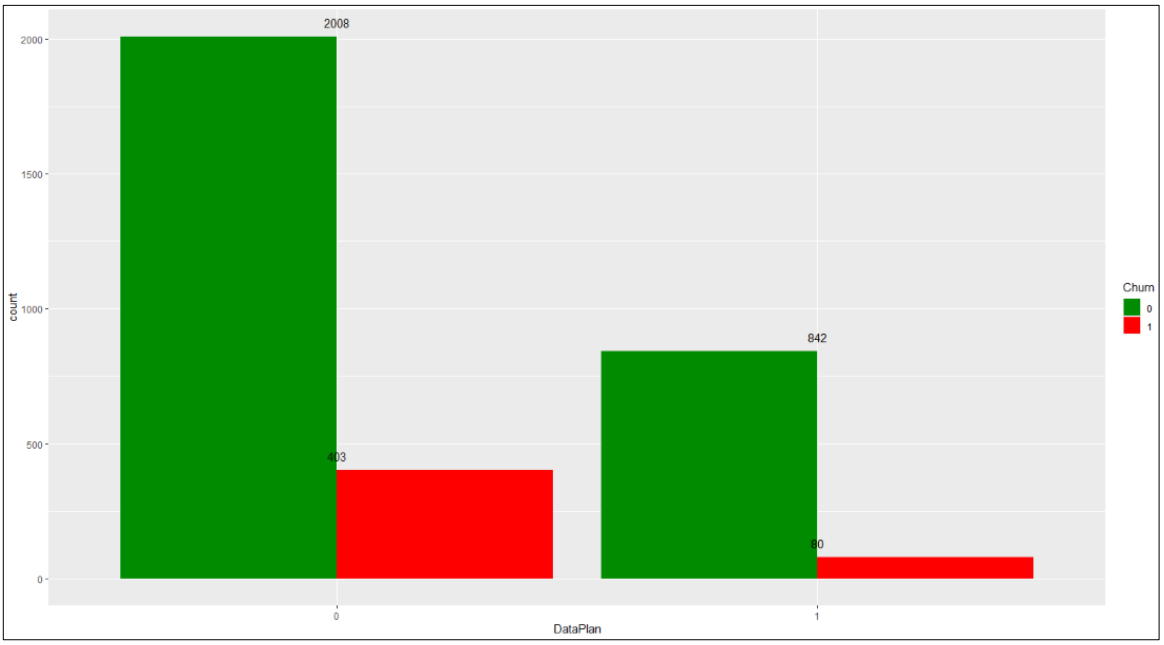
Customer Service:

Next important factor is the number of calls into the customer service. We could see from the visuals that customer service calls beyond three calls have higher churn rate. Hence it is evident that if the customer has made calls to the customer service more than three times then the probability of churning is higher.



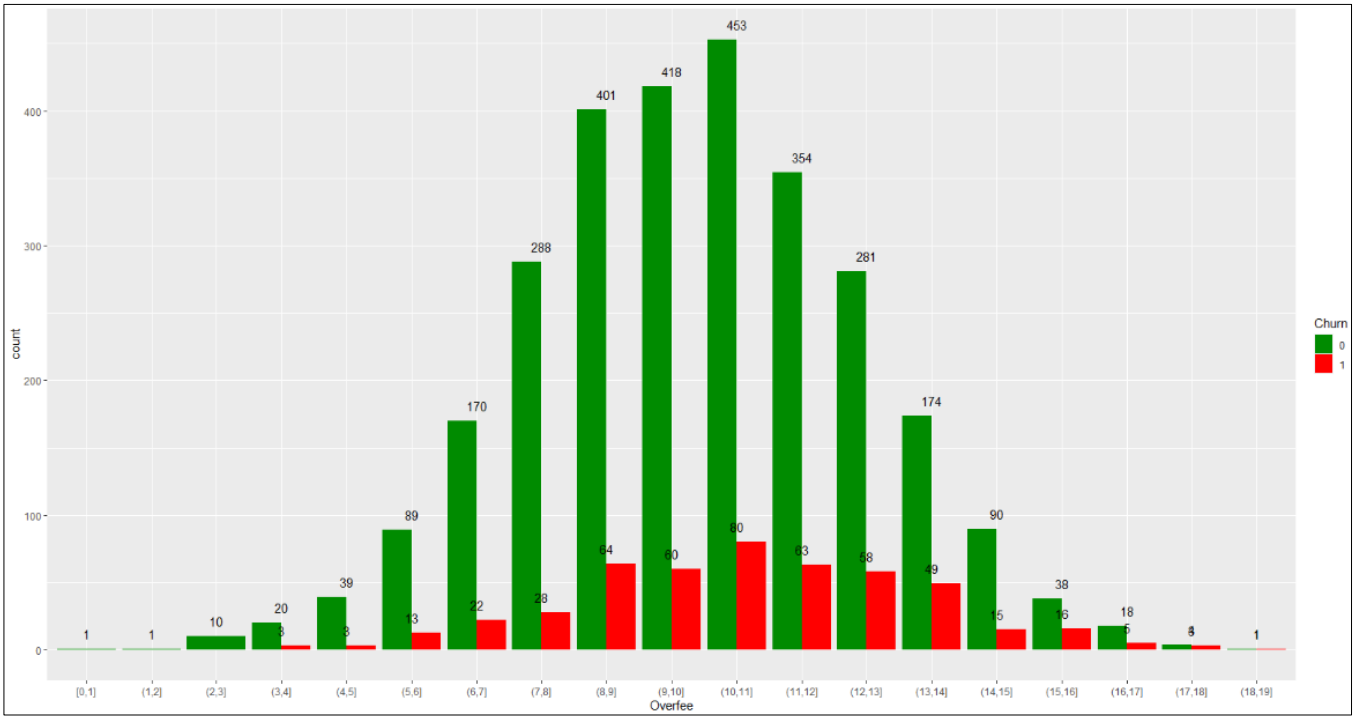
Data Plan :

Next factor is the data plan, we could see that the customers without data plan has high churn rate compared to customers with data plan. This may be due to charges or bill connected with the data usage when the customer has not enrolled for data plan. Customer may be dissatisfied with the extra charge associated with data usage on non – data plan account and would have led to churn .



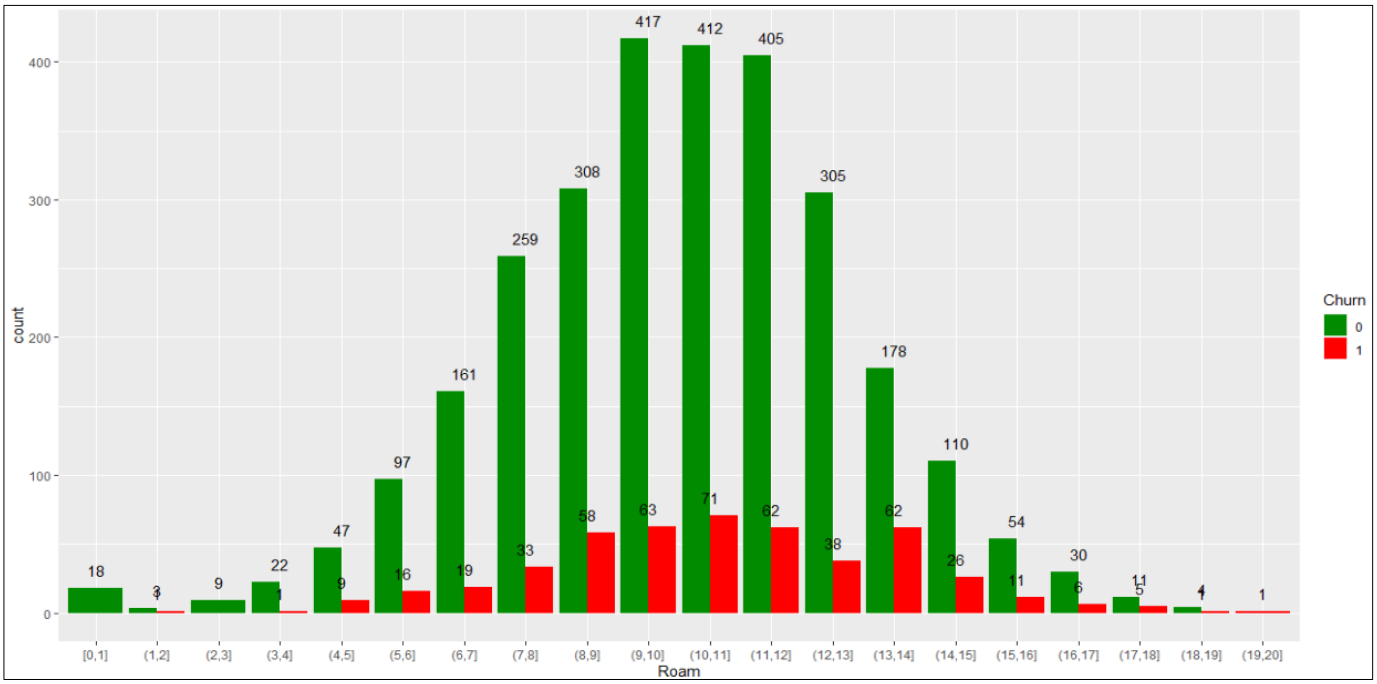
Overage Fee :

From below we can see that beyond overage fee of 12 Rupees, the churn of customers is slightly high and this fee may be associated with datausage, daymins or roaming charges. Though this variable is moderately significant. It can give insights on the charges beyond which churn rate is increasing.



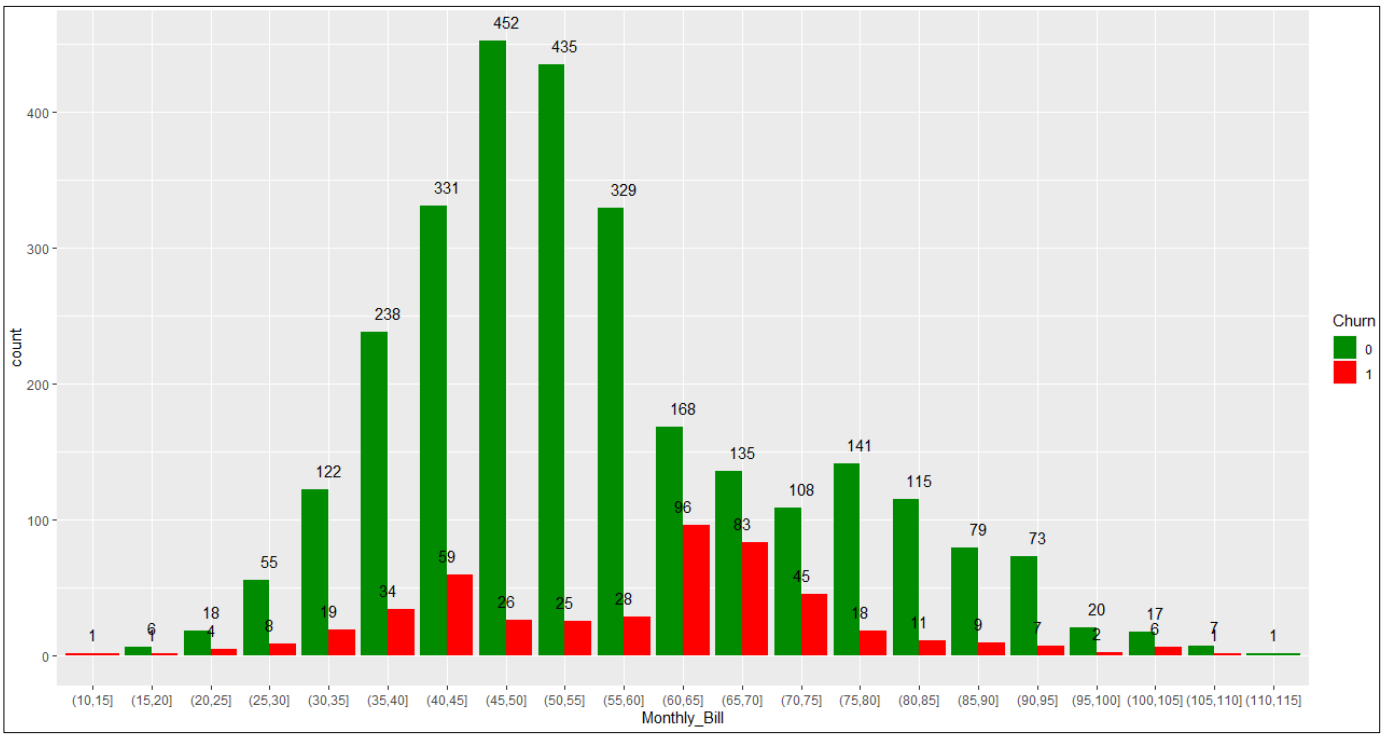
Roaming minutes :

Beyond roaming minutes of 13, the churn rate is increasing. This may be due to charges associated with roaming. Beyond 13 minutes the percentage of churn is high compared to total.

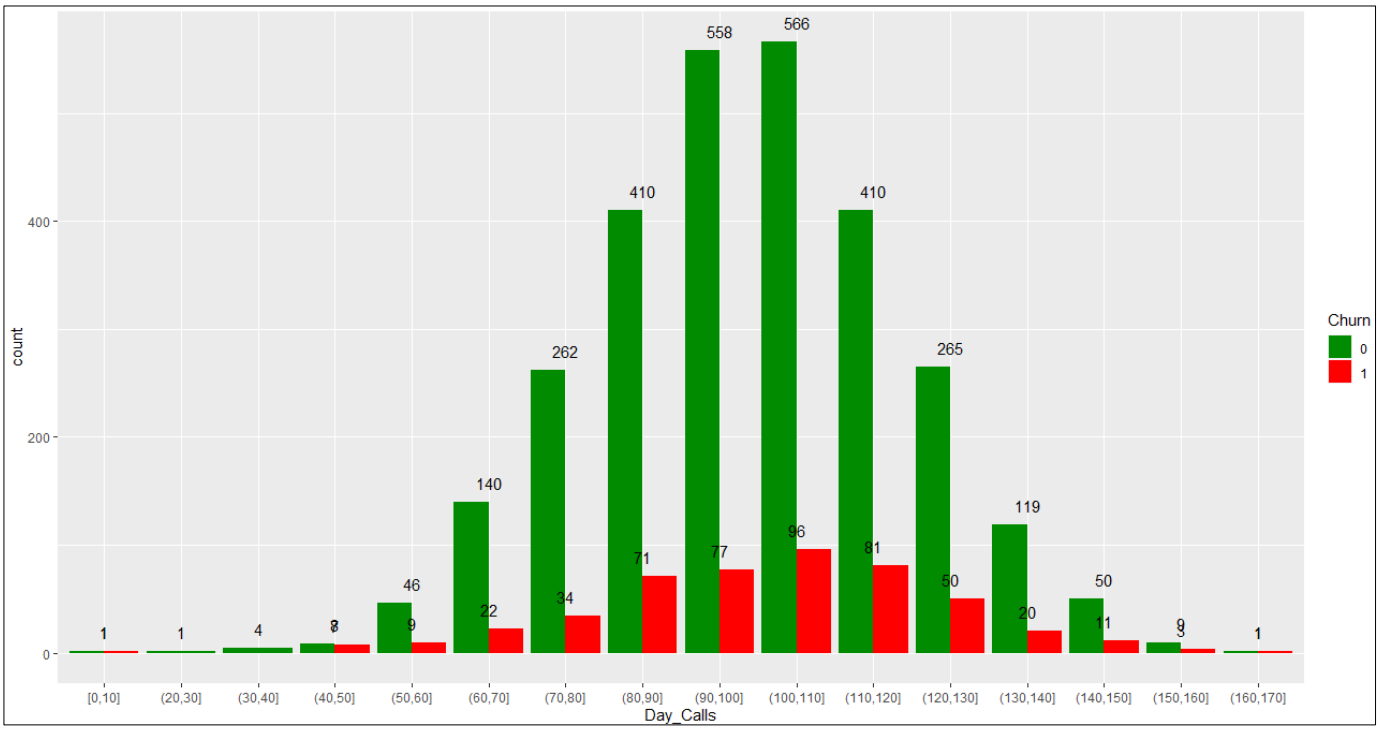


Monthly Bill Amount :

The monthly bill amount between 60 to 75 witnessed highest churn rate compared to other ranges.



Average number of daytime calls :



Recommendations and Insights:

From the above analysis we have seen the three factors contributing to customer churn. On dayMins factor, we can get feedback from customers on reason for churn owning to daytime call charges and the company can develop a flexible and low cost day call plans or introduce rate cutter to avoid overage fee and high monthly bill for the customers . If the higher daytime calls are from business executives, salespersons etc then the company needs to segment those customer and develop new calling or subscription plans and promote them with marketing campaign and promotions. The second factor is the 'contract renewal' , in which the company can study on customers recency of contract renewal and collect data on those customers who did not renew recently and study the customer demographics, billing information, call and data usage to seek reasons for not renewing. The third factor is on customer service calls, customers who call into customer service beyond three times have high churn rate , hence the company needs to analyze the calls via recordings and check if the customers concerns were addressed via the calls and why customers are reaching to call centre beyond three times, whether it's for same issue or different, the customer service provided quick response with less turnaround time etc. The company can provide feedback mail or sms system after each customer call to check if the customer calls were satisfied and resolved the issue or it led to customer dissatisfaction. The other factors like roaming charges , overage fee and data usage charges need to be revisited by the company to check if these charges are designed as per customer needs or its nearly for company profit. Hence the company needs to value customer feedback and provide right data and telecom that meets the customer needs for various customer segments.

11 Conclusion

We studied how the three Machine learning models can be effective in predicting the target variable and evaluated all the three models to check which the best model was. We also saw how same dataset produced nearly same prediction results and evaluated the accuracy of all the three models. We solved the business problem of a telecom firm facing higher customer churn and we proved that the three factors of DayMins, Contract_renewal , CustServCalls (number of calls into customer service) were the top contributors to the higher churn and via machine learning we assisted the firm to investigate further on the factors contributing to claims. We used the classification model to predict the churn for the training data and build the best models, we have seen that all the three models has performed well for future predictions based on test data and the robustness of the model has been seen by percentage of accuracy varying between 85 % to 94 % for all the models.

12 Appendix A – Source Code

Attached separately as file name “Final Project . R” and Python files.