# **Telecom Customer Churn – project report**

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

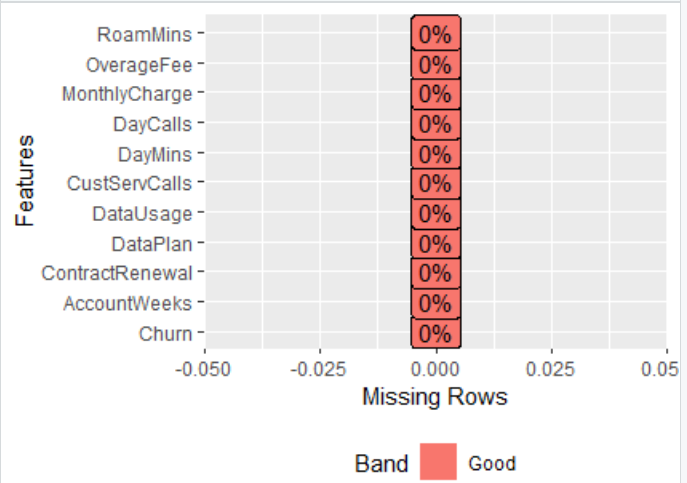
Data Preparation

1. Churn – 1 and 0 have been replaced by “Churned” and “NotChurned” respectively for easy understanding.
2. ContractRenewal – 1 and 0 have been replaced by “Yes” and “No” respectively for easy understanding.
3. DataPlan – 1 and 0 have been replaced by “Yes” and “No” respectively for easy understanding.

Also by doing so these variables have become categorical which makes more sense for this business problem.

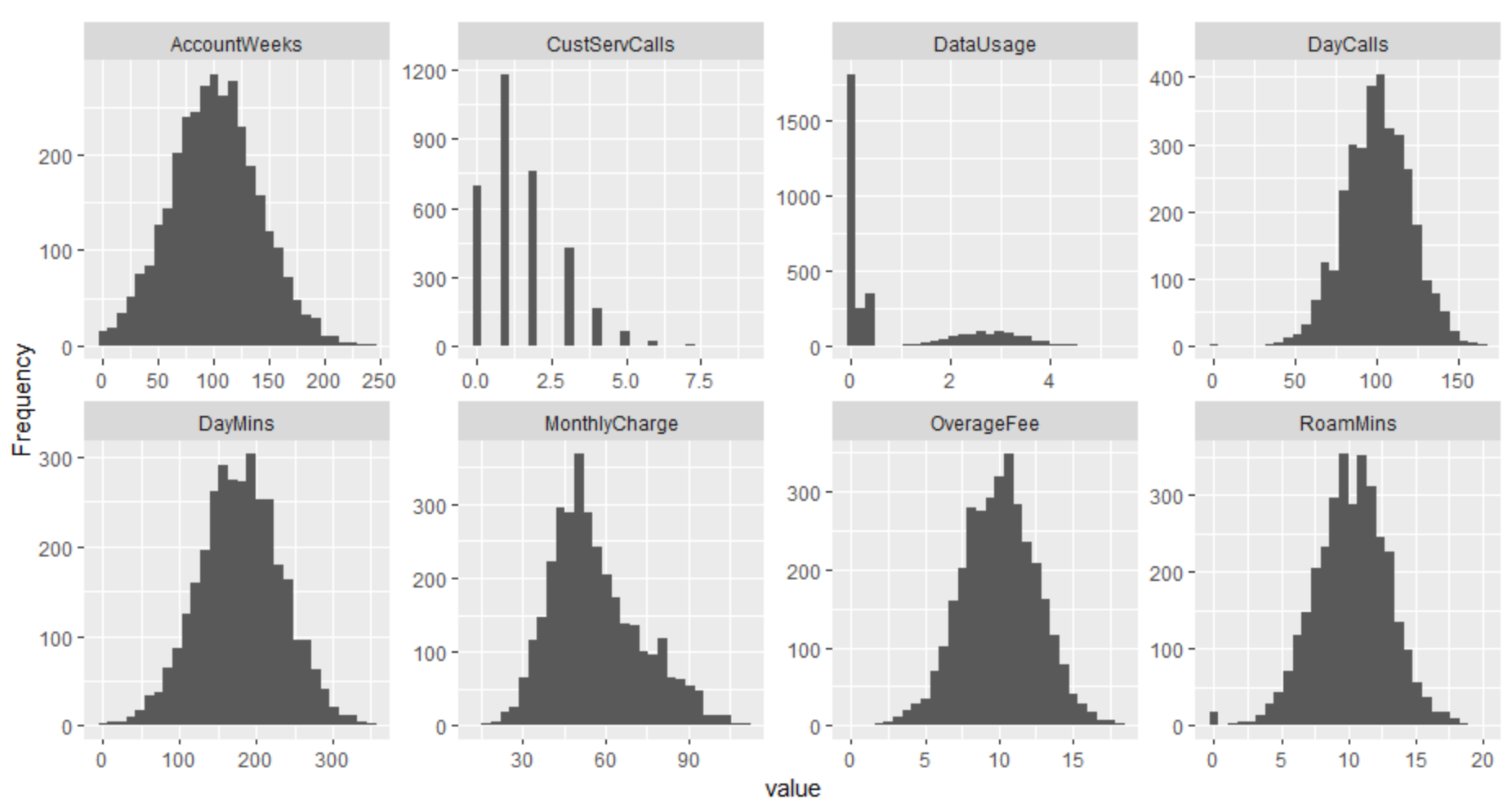
We check the summary of the dataset.

We check the missing value in the dataset and there is no missing data.

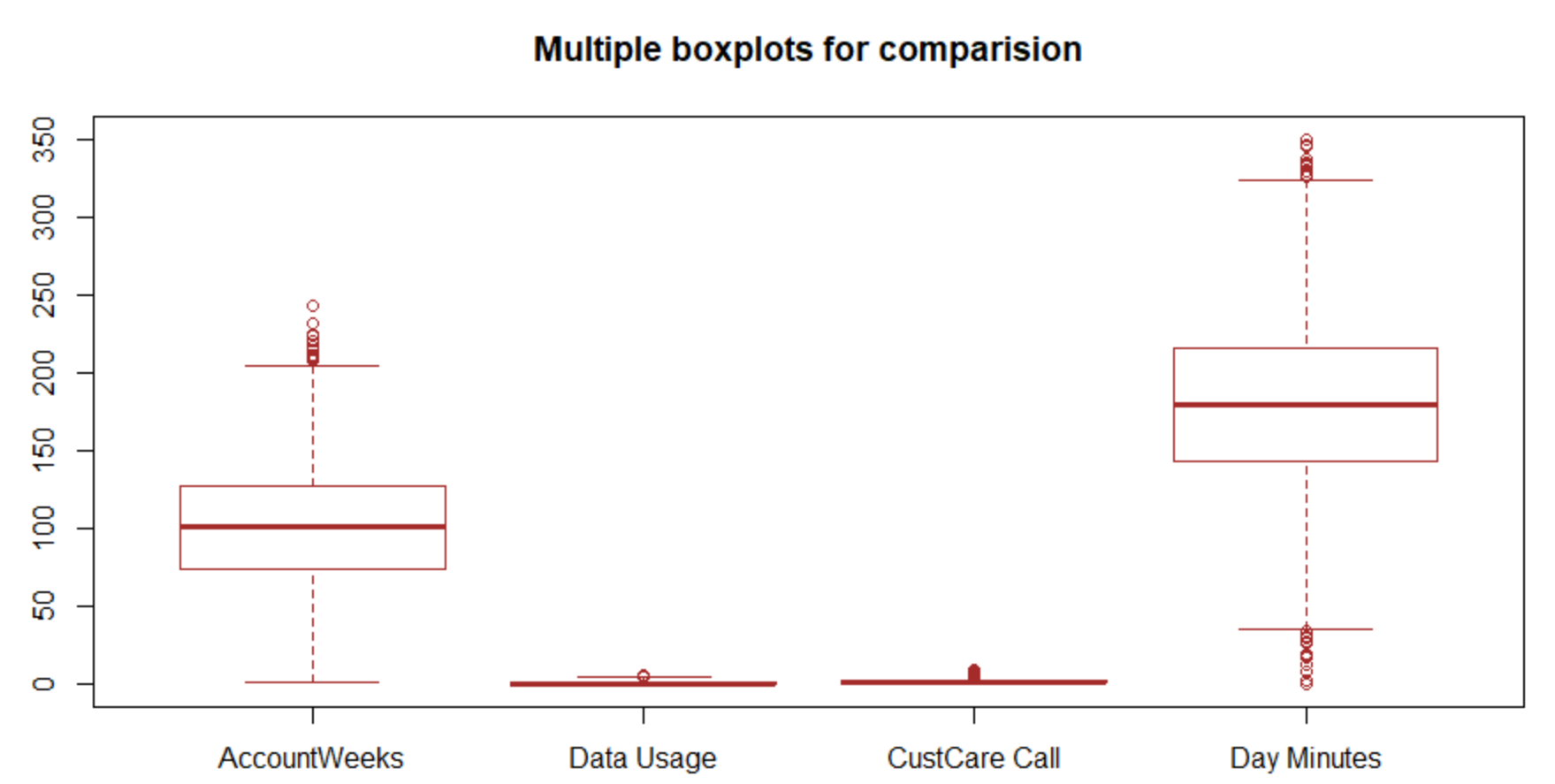


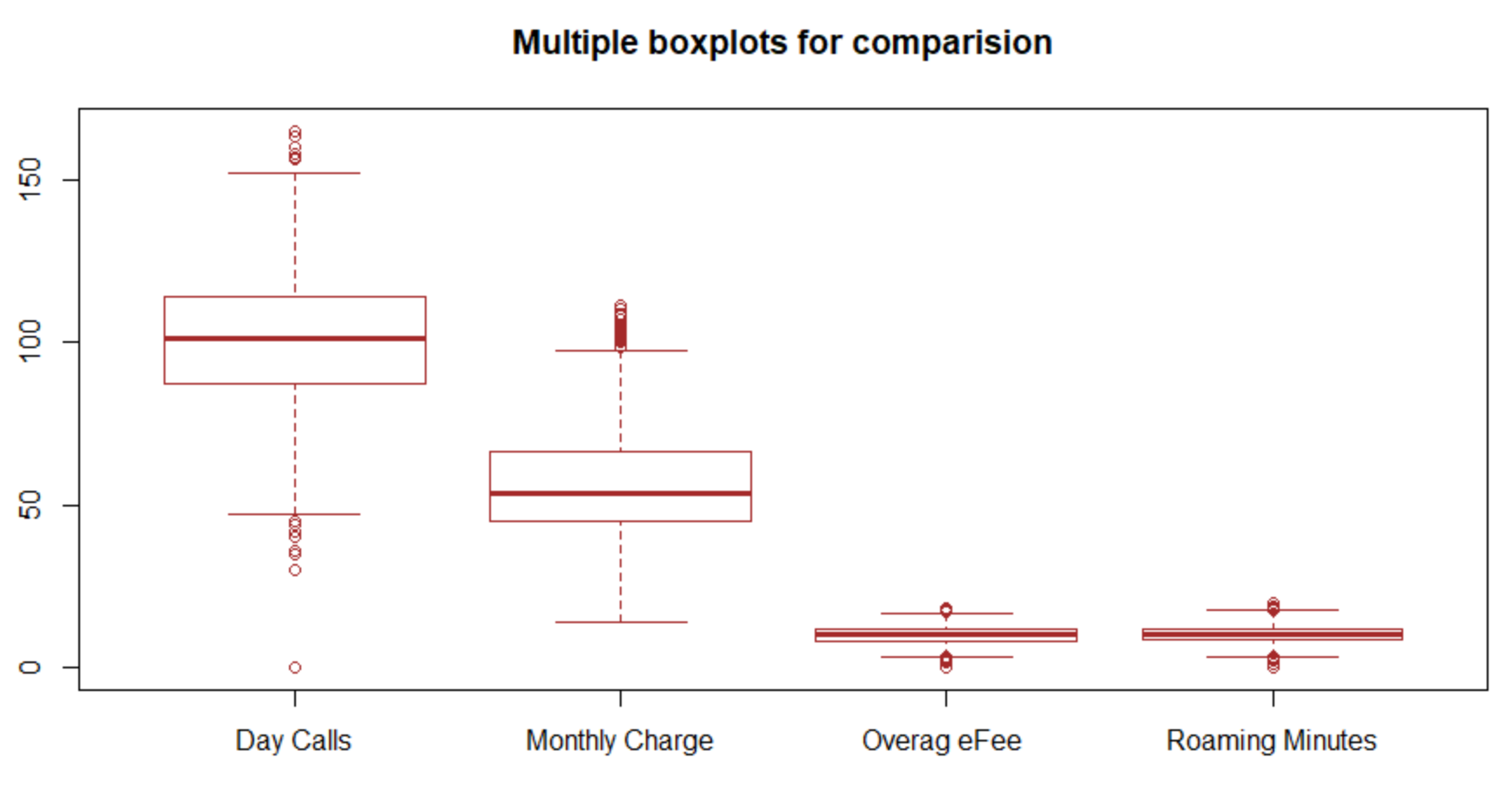
Let us take a quick look at the frequency distribution of the dataset. From the histogram it is evident the below columns have a normal distribution which will be useful for our further analysis.

* AccountWeeks
* DayCalls
* DayMins
* MonthlyCharge – a bit right skewed
* OverageFee
* RoamMins



Also it seems there are some outliers in the data usage value . Let us use a box plot to find out the outliers.



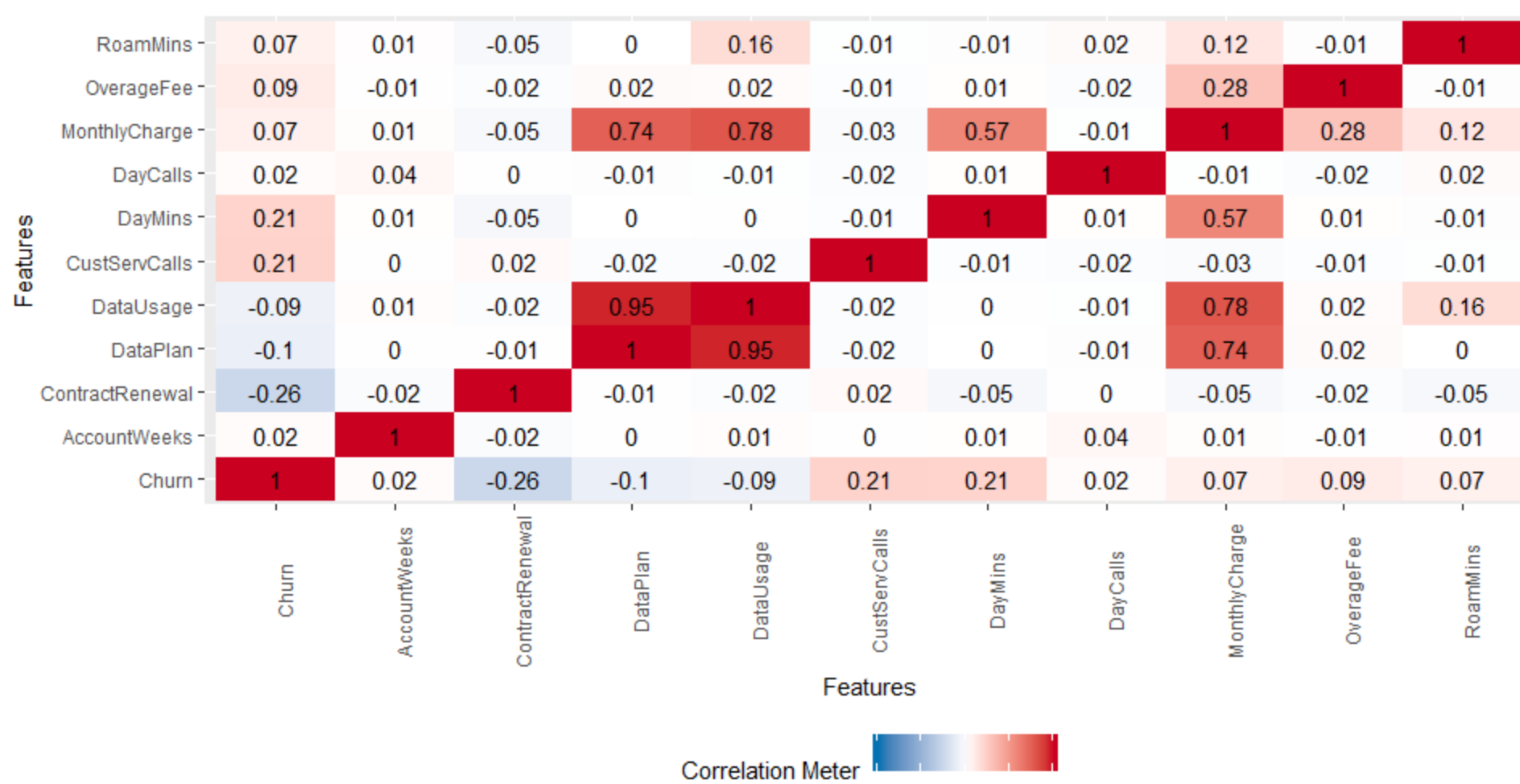


Even though there are some outliers but they look to be very valid . On a closer look at the dataset reveals that the rows with outliers in Monthly charge have either higher roaming or data usage or data call values which explains the customers’ overall high usage of cellular service. So we have decided not to modify any value in the dataset.

Verifying correlation - since the modified excel sheet has multiple factor column

we cannot perform correlation analysis since correlation in R works mainly on continuous

variable. So we load the original dataset for correlation and multicollinearity check



High correlation exists between

* dataPlan & data usage and
* monthly charge and data usage
* monthly charge and data usage

These are pretty obvious and makes sense too. So should I remove data plan and monthly charge altogether from my dataset .

Let us build a linear regression model with all dependant variable.

***model0 = lm(Churn~., mydata1)***

The model summary is as below :

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.433e-01 5.363e-02 -2.672 0.007580 \*\*

AccountWeeks 8.888e-05 1.396e-04 0.637 0.524402

ContractRenewal -2.993e-01 1.882e-02 -15.904 < 2e-16 \*\*\*

DataPlan -4.175e-02 4.381e-02 -0.953 0.340650

DataUsage -2.835e-02 1.933e-01 -0.147 0.883401

CustServCalls 5.829e-02 4.222e-03 13.804 < 2e-16 \*\*\*

DayMins 1.021e-03 3.272e-03 0.312 0.754936

DayCalls 3.409e-04 2.769e-04 1.231 0.218433

MonthlyCharge 1.428e-03 1.924e-02 0.074 0.940838

OverageFee 1.046e-02 3.280e-02 0.319 0.749780

RoamMins 8.765e-03 2.307e-03 3.800 0.000147 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3203 on 3322 degrees of freedom

Multiple R-squared: 0.1747, Adjusted R-squared: 0.1722

F-statistic: 70.31 on 10 and 3322 DF, p-value: < 2.2e-16

This model gives adjusted R squared value of 0.17 only and only 3 of the variables seems to be significant . This is happening due to the existence of the collinearity.

We should examine the VIF value :

AccountWeeks ContractRenewal DataPlan DataUsage CustServCalls

1.003791 1.007216 12.473470 1964.800207 1.001945

DayMins DayCalls MonthlyCharge OverageFee RoamMins

1031.490608 1.002935 3243.300555 224.639750 1.346583

Very high VIF value( lot higher than 10) is observed for many variables, hence confirming the multicollinearity .

Let us perform a quick step regression to find only the significant variables.

|  |
| --- |
| Stepwise Selection Summary  ----------------------------------------------------------------------------------------------  Added/ Adj.  Step Variable Removed R-Square R-Square C(p) AIC RMSE  ----------------------------------------------------------------------------------------------  1 ContractRenewal addition 0.068 0.067 424.3540 2271.7769 0.3400  2 CustServCalls addition 0.114 0.113 239.9680 2104.0145 0.3315  3 DayMins addition 0.152 0.151 88.6030 1959.5371 0.3244  4 DataPlan addition 0.162 0.161 50.7590 1922.4054 0.3225  5 OverageFee addition 0.170 0.169 19.0360 1890.9218 0.3209  6 RoamMins addition 0.174 0.173 5.7670 1877.6507 0.3203  ---------------------------------------------------------------------------------------------- |
|  |
| |  | | --- | | > | |

So we have the list of significant variables that we need to consider.

We have created a new dataset called ”cleandata” with only these data set .

We now build a model with these new dataset .

#new model with reduced variables

model1 = lm(Churn~ContractRenewal+CustServCalls+

DayMins+OverageFee+DataPlan+RoamMins, mydata1)

summary(model1)

Model summary :

coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.089256 0.041288 -2.162 0.0307 \*

ContractRenewal -0.299318 0.018804 -15.918 < 2e-16 \*\*\*

CustServCalls 0.058217 0.004220 13.794 < 2e-16 \*\*\*

DayMins 0.001263 0.000102 12.385 < 2e-16 \*\*\*

OverageFee 0.012817 0.002189 5.854 5.25e-09 \*\*\*

DataPlan -0.079761 0.012406 -6.429 1.47e-10 \*\*\*

RoamMins 0.007776 0.001990 3.908 9.48e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3203 on 3326 degrees of freedom

Multiple R-squared: 0.174, Adjusted R-squared: 0.1725

F-statistic: 116.8 on 6 and 3326 DF, p-value: < 2.2e-16

While we do not have much improvement on the Adjusted R-squared , but we now see that all the variables are significant with low p values.

Lets quickly check the VIF for the new model ,

***> vif(model1) # VIF is around 1 for all variables***

ContractRenewal CustServCalls DayMins OverageFee DataPlan

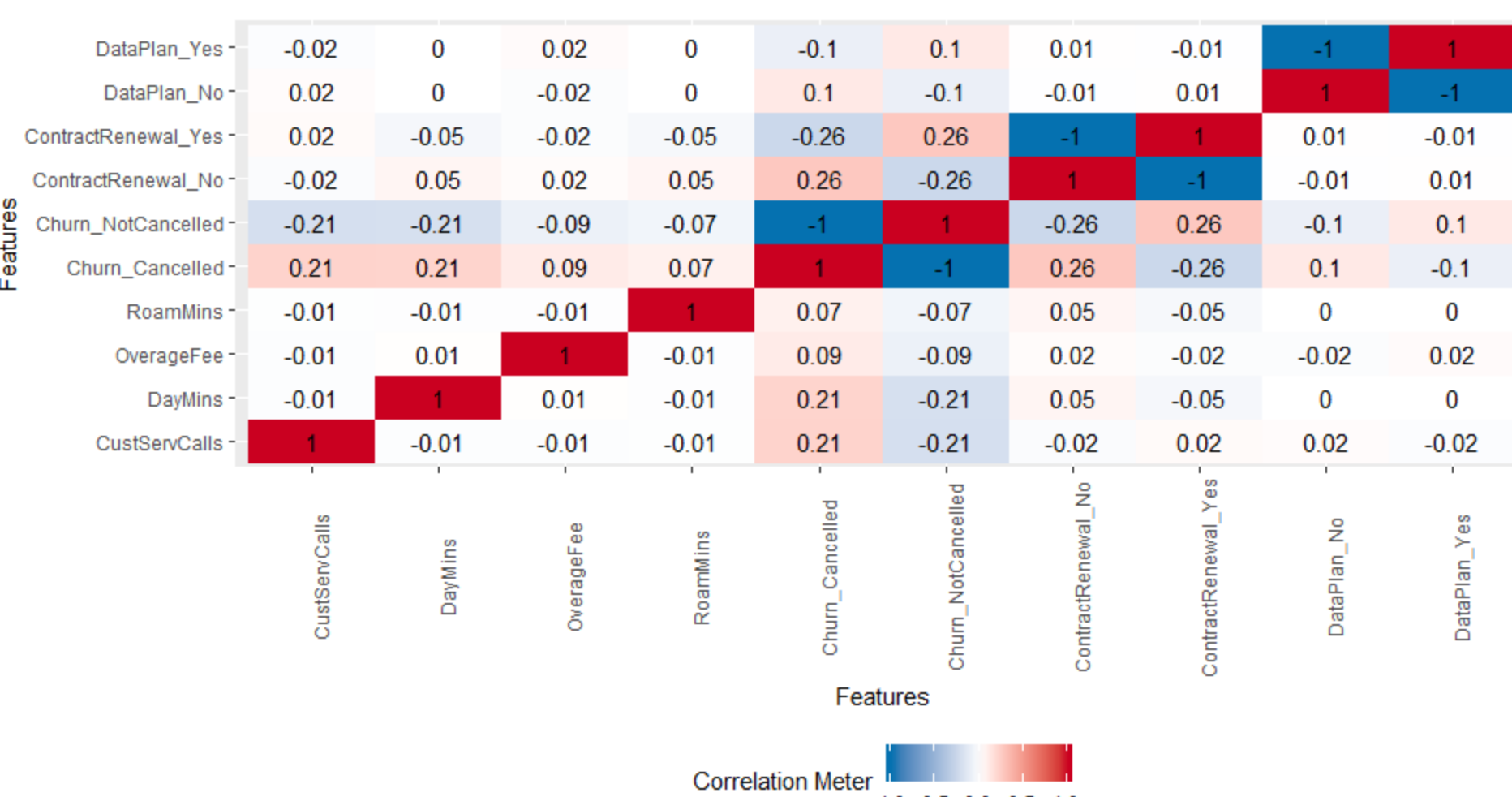
1.005557 1.001289 1.002793 1.001149 1.000806

RoamMins

1.002485

All the variables showing VIF around 1 , we also recheck the correlation matrix and as expected we see the multicolienarity has been completely removed from the dataset.

The correlation plot for cleandata set:



So we are ready with our dataset. Our new dataset contains only the below variables. Please note CHURN is the dependant variable in this dataset.

* "Churn"
* "ContractRenewal"
* "DataPlan"
* "CustServCalls"
* "DayMins"
* "OverageFee"
* "RoamMins"

As a best practice we have divided the cleandata in 2 subsets as below:

1. TrainData – contains 75% of the total data row
2. TestData - contains 25% of the total data row

**1.4 EDA - Summarize the insights you get from EDA**

* The dataset has total 3333 rows and 11 columns
* Here the dependant variable is Churn – it indicates whether a customer has cancelled his/her subscription or not
* There is no missing value in the data set
* There are outliers in the below variables but they all seems

to be valid scenario

* + DayCall
  + MonthlyCharge
  + AccountsinWeeks
  + DayMin
* Most of the variables in the dataset follow a normal

Distribution.

* Following pair of variables in the dataset have high co-relation:
  + Data Plan & data usage and
  + Monthly charge and data usage
  + Monthly charge and data usage
* Due to this, we have multicollinearity in the dataset. This has been proven by high VIF values after building a linear regression model.
* We have treated multicollinearity and the final dataset has only below variables
  + Churn"
  + "ContractRenewal"
  + "DataPlan"
  + "CustServCalls"
  + "DayMins"
  + "OverageFee"
  + "RoamMins"
  1. **Applying Logistic Regression**

Logistic regression, also called a logit model, is used to model dichotomous outcome variables. In the logit model the log odds of the outcome is modeled as a linear combination of the predictor variables.

We will use glm() here to perform Logistic regression.

***LogitModel1 <- glm(Churn ~ ., data = TrainData,* family = binomial(link = 'logit'))**

***print(LogitModel1)***

Coefficients:

(Intercept) ContractRenewal1 DataPlan CustServCalls

-5.49014 -1.97468 -1.00127 0.48332

DayMins OverageFee RoamMins

0.01310 0.13125 0.07985

Degrees of Freedom: 2498 Total (i.e. Null); 2492 Residual

Null Deviance: 2010

Residual Deviance: 1613 AIC: 1627

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9577 -0.5078 -0.3487 -0.2140 2.9912

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.490140 0.506357 -10.842 < 2e-16 \*\*\*

ContractRenewal1 -1.974680 0.166330 -11.872 < 2e-16 \*\*\*

DataPlan -1.001274 0.173068 -5.785 7.23e-09 \*\*\*

CustServCalls 0.483324 0.045580 10.604 < 2e-16 \*\*\*

DayMins 0.013099 0.001279 10.242 < 2e-16 \*\*\*

OverageFee 0.131253 0.026458 4.961 7.02e-07 \*\*\*

RoamMins 0.079855 0.023550 3.391 0.000697 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2009.9 on 2498 degrees of freedom

Residual deviance: 1612.5 on 2492 degrees of freedom

AIC: 1626.5

Number of Fisher Scoring iterations: 5

Now lets try to apply the same model on test data set and we get below result:

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9668 -0.5209 -0.3504 -0.1982 2.9412

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.689621 0.850777 -6.688 2.27e-11 \*\*\*

ContractRenewal1 -2.029851 0.286540 -7.084 1.40e-12 \*\*\*

DataPlan -0.795180 0.264758 -3.003 0.002670 \*\*

CustServCalls 0.560178 0.075300 7.439 1.01e-13 \*\*\*

DayMins 0.011801 0.001984 5.947 2.73e-09 \*\*\*

OverageFee 0.158652 0.044535 3.562 0.000367 \*\*\*

RoamMins 0.092434 0.040674 2.273 0.023054 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 745.06 on 833 degrees of freedom

Residual deviance: 575.18 on 827 degrees of freedom

AIC: 589.18

Number of Fisher Scoring iterations: 5

Now we use the model on test data set and evaluate different model performance parameter. We will see this in detail in the later section of this project report.

**2.2 Interpret Logistic Regression**

The output shows the coefficients, All the variables here are statistically significant, The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.

* + In this model , all the variables are quite significant with p value <0.05
  + For every one unit change in ContractRenewal1 , the log odds of Churning (versus non-churning) **decreases** by 1.97468 .
  + Logically if a customer renews the contract, then chances of his or her to stay in the contract increases
  + For a one unit increase inCustServCalls , the log odds of being churning increases by 0.48332 .
  + For a one unit increase inOverageFee , the log odds of being churning increases by 0.13125 .
  + This tells us that if a customer has to make more customer care calls(assuming the customer has some kind of complaints) or there is a hike in overage fee , the probability of the customer to churn increase

**2.5 - Applying KNN Model/ Interpret KNN Model**

1. Before we apply KNN model , it is very important to normalize the dataset , otherwise the distance calculation will often produce misleading results.
2. We will also create a new dataset without the response variable Churn since we are trying to predict the Churn variable.

After normalizing the data , lets us examine how the data looks like :

|  |
| --- |
| head(KNNData)  ContractRenewal DataPlan CustServCalls DayMins OverageFee RoamMins  1 1 1 0.1111111 0.7557013 0.5426058 0.500  2 1 1 0.1111111 0.4606613 0.5376581 0.685  3 1 0 0.0000000 0.6938426 0.3331501 0.610  4 0 0 0.2222222 0.8534778 0.1704233 0.330  5 0 0 0.3333333 0.4751995 0.4079164 0.505  6 0 0 0.0000000 0.6368301 0.6063771 0.315 |
|  |
| |  | | --- | | > | |

Let us split the KNN Data set for train and test data set :

set.seed(1234)

TRAIN\_INDEX <- sample(1:nrow(KNNData),0.75\*nrow(KNNData))

TrainKNN <- KNNData[TRAIN\_INDEX,]

TestKNN <- KNNData[-TRAIN\_INDEX,]

dim(TrainKNN)

dim(TestKNN)

> dim(TrainKNN)

[1] 2499 6

> dim(TestKNN)

[1] 834 6

## Split the dataset into train and test for development and out of sample testing respectively.

Find the K Value – lets assume square root for 2499( no of values in the train dataset ) which is 49.98.

So let us build 2 KNN model with values of K as 49 and 50.

KNN.49<- knn(train=train.churn, test=test.churn, cl=train.churn$Churn, k=5)

KNN.50<- knn(train=train.churn, test=test.churn, cl=train.churn\_labels, k=50)

summary(KNN.49)

summary(KNN.50)

tb1 <- table(KNN.49,testtarget)

tb2<-table(KNN.50,testtarget)

tb1

tb2

Confusion matrix:

testtarget

KNN.49 0 1

0 857 0

1 0 143

##check the accuracy

accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) \* 100}

accuracy(tb1)

accuracy(tb2)

For both the models, we get accuracy 100%

So we can conclude that with the help of KNN process, we are able to predict the result with 100% accuracy for this data set.

Please note , here we are using all predictors to predict the value , that is why we may be getting 100% accuracy .

Let us try to predict based on only DataPlan , CustomerServiceCall ,OverageFee and Roaming.

tb1\_A <- table(KNN.49\_,testtarget)

accuracy(tb1\_A)

Accuracy : 86.3%

**2.5 - Applying Naive Bayes Model/2.6 Interpret Naive Bayes Model**

Naive Bayes is a Supervised Machine Learning algorithm based on the Bayes Theorem that is used to solve classification problems by following a probabilistic approach. It is based on the idea that the predictor variables in a Machine Learning model are independent of each other. Meaning that the outcome of a model depends on a set of independent variables that have nothing to do with each other.

First , we create train and test data set .

Let us build a model for Naïve Bayes on Traindata set.

NBModel=naiveBayes(Churn~.,data=TrainData)

We get below result :

|  |
| --- |
| A-priori probabilities:  Y  0 1  0.8615446 0.1384554  Conditional probabilities:  ContractRenewal  Y 0 1  0 0.06827682 0.93172318  1 0.28612717 0.71387283  DataPlan  Y [,1] [,2]  0 0.2902926 0.4540028  1 0.1502890 0.3578720  CustServCalls  Y [,1] [,2]  0 1.437993 1.165244  1 2.164740 1.844388  DayMins  Y [,1] [,2]  0 175.0848 49.58496  1 205.6280 68.89138  OverageFee  Y [,1] [,2]  0 9.927065 2.510701  1 10.559566 2.593533  RoamMins  Y [,1] [,2]  0 10.17120 2.778808  1 10.63584 2.876832 |
|  |
| |  | | --- | |  | |

This model gives individual condition probability distribution . We can say

* + 1. The prior probability of a customer churn =0 ( not churned) is 86%
    2. The prior probability of a customer getting churned is 14%

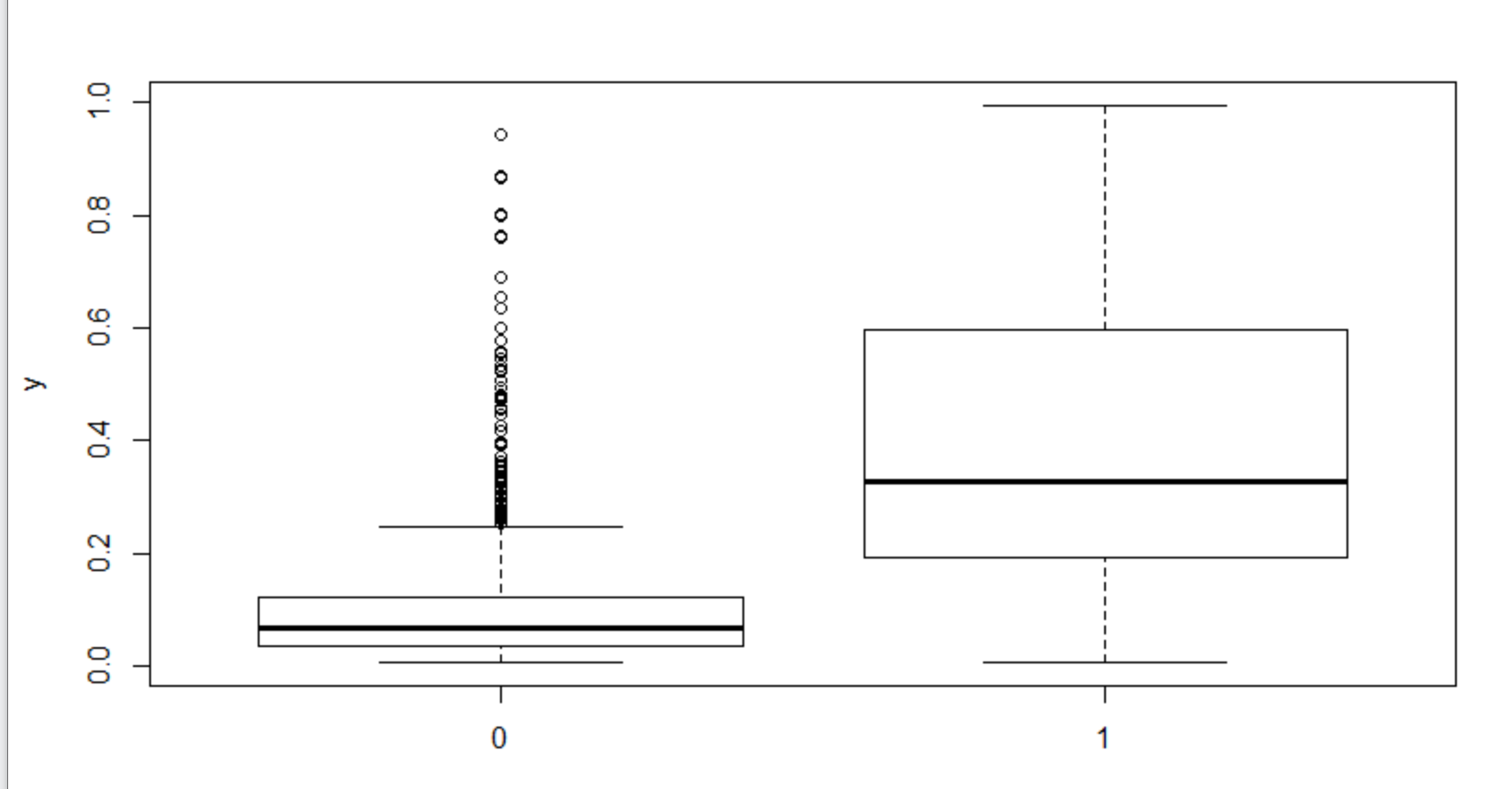
It also gives the specification of other conditional probabilities.

* + 1. The distribution of ContractRenewal given customer has churned is having a mean value 28612717 with a standard deviation of 71387283
    2. Similarly we can conclude the distribution of all other independent variables.

Once we build this model, we use this model on test data set to see how it performs and try to plot the prediction.

predict1 <-predict(NBModel,type="raw",newdata=TestData)

plot(TestData$Churn,predict1[,2])



We see huge number of prediction error in the model. Let us see the accuracy of the model.

***NBAccuracy = table(TestData$Churn,predict1[,2])***

***accuracy(NBAccuracy)***

Accuracy turns out to be only 11% for this model.

**Confusion matrix interpretation for all models**

Logistic Regression Model

LogitAccuracy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CONFUSION MATRIX | | | |  |  |
| **Logistic Regression** | | | | | | |
| Training Dataset |  |  |  | Testing Dataset |  |  |
|  |  |  |  |  |  |  |
| Total = 2499 | Actual Positive | Actual Negative |  | Total = 834 | Actual Positive | Actual Negative |
|  |  |  |  |  |  |  |
| Predicted Positive | 58 | 46 (Type 1 Error) |  | Predicted Positive | 25 | 18(Type 1 Error) |
| Predicted Negative | 288 (Type 2 Error) | 2107 |  | Predicted Negative | 112 (Type 2 Error) | 679 |

* Type 1 error: Customer is not churned but the model predicted the customer will churn
* Type 2 error: The customer has churned out but the model predicted that this set of customers will not churn
* Model Accuracy on Train data set = [(58+2107)/2499]\*100 = 86.63%
* Model Accuracy on Test data set = [(25+679)/834]\*100 = 84.41%
* Sensitivity on Train = 16%
* Sensitivity on test = 18%

**Confusion Matrix for KNN**

|  |  |  |
| --- | --- | --- |
| Test Dataset |  |  |
|  |  |  |
| Total = 1000 | Actual Positive | Actual Negative |
|  |  |  |
| Predicted Positive | 857 | 0 (Type 1 Error) |
| Predicted Negative | 0 (Type 2 Error) | 143 |

Here all the outcomes have been correctly predicted.

**Confusion Matrix for Naïve Bayes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Naïve Nayes** | | | | | | |
| Training Dataset |  |  |  | Testing Dataset |  |  |
|  |  |  |  |  |  |  |
| Total = 2499 | Actual Positive | Actual Negative |  | Total = 834 | Actual Positive | Actual Negative |
|  |  |  |  |  |  |  |
| Predicted Positive | 158 | 166 (Type 1 Error) |  | Predicted Positive | 54 | 57 (Type 1 Error) |
| Predicted Negative | 199 (Type 2 Error) | 1976 |  | Predicted Negative | 72 (Type 2 Error) | 651 |

* Model Accuracy on Train data set 85.39%
* Model Accuracy on Test data set = 84.34%
* Sensitivity on Train = 44%
* Sensitivity on test = 46%

**Interpretation of other Model Performance Measures for logistic <KS, AUC, GINI>**

Let us try to evaluate the performance for our Logit model .

We start by calculating the error rate in the model:

## a function for error rate

get\_Error\_Rate=function(trues, predicted\_prb, cutoff){

preds=ifelse(predicted\_prb<cutoff,0,1)

tab=table(preds, trues)

round((tab[1,2]+tab[2,1])/sum(tab), 4)

}

get\_Error\_Rate(TrainData$Churn,LogitModel1$fitted.values, 0.5)

get\_Error\_Rate(TestData$Churn,LogitModel2$fitted.values, 0.5)

So for train data set we get 13.4% and test data set we have 15.4% error rate. So this gives a low error rate for both the train and test data set

ROC:

#plot ROC

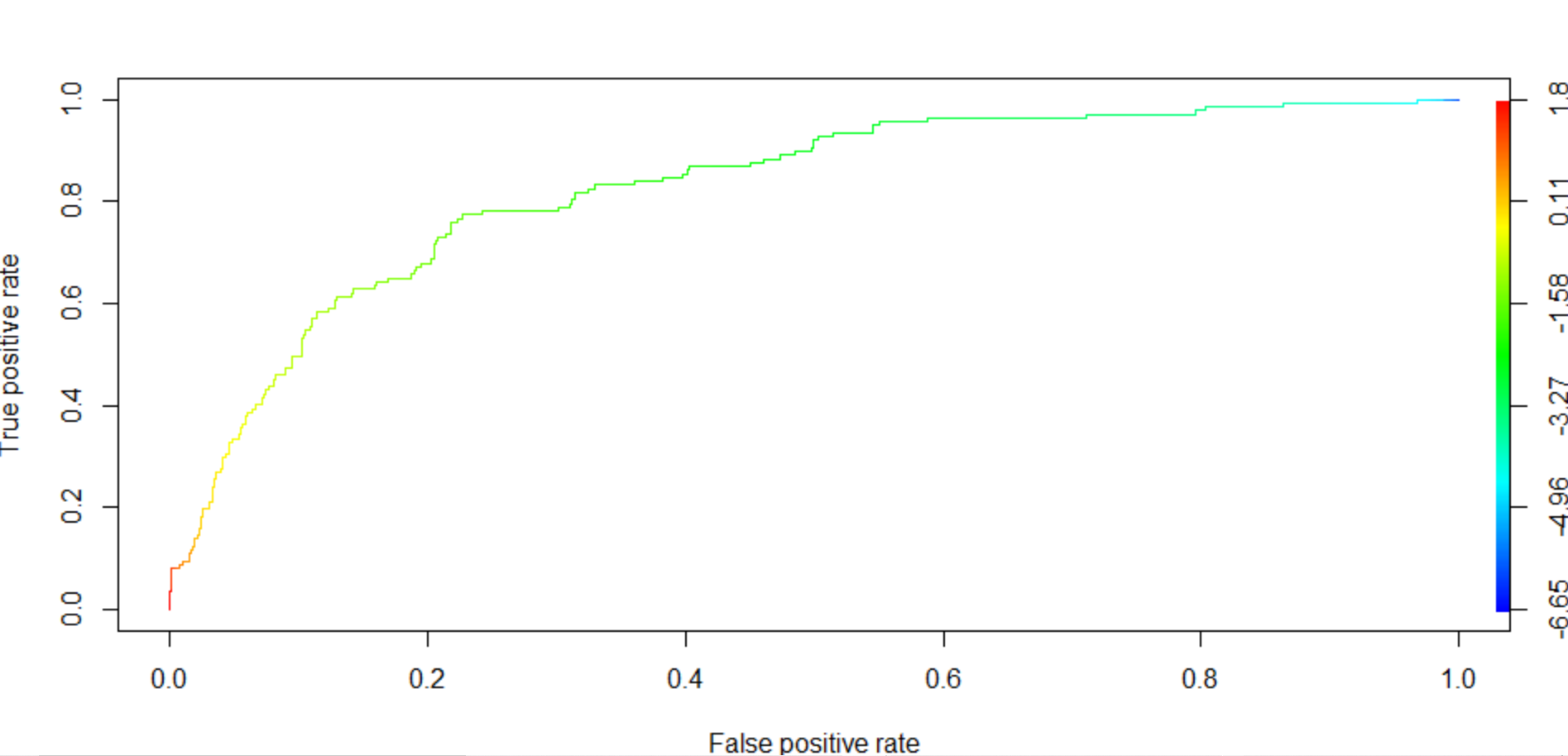
***predictROC1 = predict(LogitModel1, newdata = TrainData)***

***pred1 = prediction(predictROC1, TrainData$Churn)***

***perf1 = performance(pred1, "tpr", "fpr")***

***plot(perf1,colorize =T)***

We can also try to plot a curve for Sensitivity (True Postive Rate (TPR)) vs Specificity (True Negative Rate (TNR))



**AUC:**

For our model , we have Area under curve is as below :

* + 1. Train Data Set : 81.19%
    2. Test Data Set : 82.58%

So this tells us that we can classify customers who will be churned or not with more than 80% probability.

**KS Value**

Along with AUROC, another measure is the KS statistic. It is the maximum difference between TPR and FPR.

We get following KS value :

1. Train Data Set : 0.5380501
2. Test Data Set : 0.5470368

So we get a significant KS value which further ensures that we have a good performing model.

**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=x line, passing through (0,0) and (1,1))
    2. The top left triangle above the random model line (y=x line).

For our model we have GINI coefficient valu as below :

1. Train Data Set : 0.5187371
2. Test Data Set 0.5360111

**2.9** **Remarks on Model validation exercise <Which model performed the best>**

We have built 3 models here and out of these 3 ,

1. KNN model has the best performance with 100% accuracy, but looks like it has over fitted.
2. The logistic model gives more than 85% accuracy. It also exhibits a very significant AUC and KS and GINI value .
3. The Naïve Bayes model provided 82% accuracy. So this is close to our logistic regression model too.
4. However if we focus on sensitivity , we see Naïve Bayes gives a far better value than the logistic model

So I would recommend to use **Naïve Bayes** model for our business scenario.

Now if we talk about the best model validation technique, we need to understand that we are trying to predict which all customer will be churned based on the given dataset.In other term , we can say we need to focus more on our response variable being TRUE ( customer churn happens – True Positive ).

So we should focus more on the below parameters since they deal with TPR:

1. Accuracy
2. Precision
3. Sensitivity

**Actionable Insights and Recommendations**

After evaluating all kinds of models in our data set , we understand that the most important factors behind customer churn is the following :

* + Whether customer has renewed the Contract
  + If a customer is using Data Plan or not
  + No of calls to service desk
  + Time spent on calls each day
  + Overage Fee
  + Roaming Minutes

So few of the recommendation would be:

1. Sales team should focus more to convince customers to renew their contracts. They might provide some promotional offer for early or long term contract renewal.
2. If a customer is making more number of calls to customer care , special attention should be given to solve the issue .
3. Customers who are using more time on phone or in roaming , needs to be offered some free talk time and extra data plan so that their overall cost does not become too huge.