CUSTOMER CHURN PREDICTION-TELECOM INDUSTRY

Abhishek Garg
PGP-BABI
September 2019

Table of Contents

BUSINESS CASE

Customer Churn is a burning problem for Telecom companies.

The dataset consists of information about postpaid customers of a company such customer usage behavior, contract details, as well as which were the customers who canceled their service. ***Based on this past data, the company is looking to build models that can help identify customers who are likely to cancel their service in the future or not.***

KEY THOUGHTS

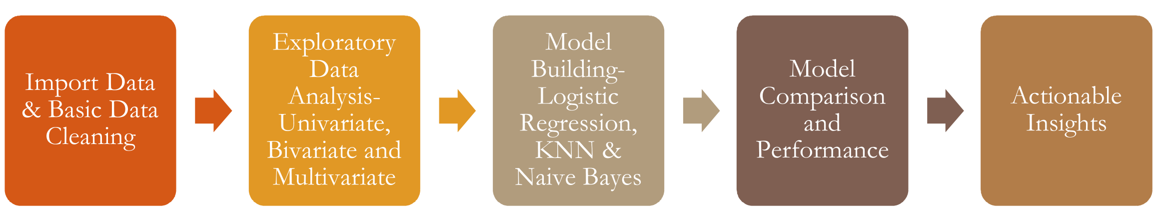
The key problem for the company is to correctly identify the customers which are likely to churn or cancel their service.

Since the telecom industry is quite competitive, ***any customer lost results in decreased revenue for the company*** and hence ***lower revenue and valuations*** (while keeping other factors constant)

Hence, while modelling it should be seen that misclassification of customers who have cancelled their service in the past data is minimized.

For this analysis classifier models such as Logistic Regression and KNN would be used to predict the customers who are likely to churn. Additionally, it would also be check if Naive Bayes classifier would also be implemented. If yes, the Naïve Bayes model would be built.

KEY STEPS



ENVIRONMENT SET-UP, IMPORT FILE AND BASIC DATA CLEANING

SET-UP WORKING DIRECTORY & IMPORT FILE

```
org_File= read.csv("D:/RProgramming/Predictive_Modelling/Project/DataSet/Cell
phone.csv",header=TRUE)
```

BASIC DATASET ANALYSIS

- The dataset contains 3333 observations with 11 variables
- All variables are stored as integer or numeric in the imported file. Relevant variables would be converted to other data types, wherever applicable
- Furthermore, variables such as data plan (whether a customer uses data plan or not) and data use (data usage by customers in GB) show that there might be a problem of collinearity. Since, multicollinearity is a concern for many machine learning algorithms. This would be checked if the variables are highly correlated to each other.

```
dim(org_File)

## [1] 3333   11

str(org_File)

## 'data.frame':    3333 obs. of  11 variables:
##  $ Churn          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AccountWeeks   : int  128 107 137 84 75 118 121 147 117 141 ...
##  $ ContractRenewal: int  1 1 1 0 0 0 1 0 1 0 ...
##  $ DataPlan       : int  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  : int  1 1 0 2 3 0 3 0 1 0 ...
##  $ DayMins        : num  265 162 243 299 167 ...
##  $ DayCalls       : int  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 ...
```

RENAMING COLUMNS FOR EASE OF UNDERSTANDING

VARIABLE UNDERSTANDING

- ***Customer_Churned_or_Not***- 1 if the customer has cancelled the service, 0 if not. This is out target variable.
- ***Relationship_in_Number_of_Weeks***-Number of weeks the customer has had an active account

- ***Contract_Renewed_or_Not***- 1 if the customer renewed the contract recently, 0 if not
- ***Use_Data_Plan_or_Not***- 1 if the customer uses data plan, 0 if not
- ***Data_Use_inGB***- Customer's monthly data usage in gigabytes
- ***Avg_Minutes***- Average daytime minutes per month.
- ***Avg_Calls-*** Average number of daytime calls
- ***Avg_Bill***- Average monthly bill
- ***Largest_Overage_Fee***- Largest overage fee in the last 12 months
- ***Avg_Roaming_Minutes***- Average number of roaming minutes

```
# Renaming coloums in original file for ease of understanding

names(org_File) = c("Customer_Churned_or_Not", "Relationship_in_Number_of_Wee
ks", "Contract_Renewed_or_Not","Use_Data_Plan_or_Not", "Data_Use_inGB", "Numb
er_Customer_Service_Calls", "Avg_Minutes", "Avg_Calls", "Avg_Bill", "Largest_
Overage_Fee", "Avg_Roaming_Minutes")
names(org_File)

##  [1] "Customer_Churned_or_Not"         "Relationship_in_Number_of_Weeks"
##  [3] "Contract_Renewed_or_Not"         "Use_Data_Plan_or_Not"
##  [5] "Data_Use_inGB"                   "Number_Customer_Service_Calls"
##  [7] "Avg_Minutes"                     "Avg_Calls"
##  [9] "Avg_Bill"                        "Largest_Overage_Fee"
## [11] "Avg_Roaming_Minutes"
```

CONVERT RELEVANT VARIABLE DATATYPE

- Since customer churn, contract renewed or not, use data plan or not are variables with yes/no type of options, they have been converted to factor variables.
- Number of customer services calls has been converted to ordered factor, since 1 call is less than 2 calls. This has specifically been done for the purposes of exploratory analysis

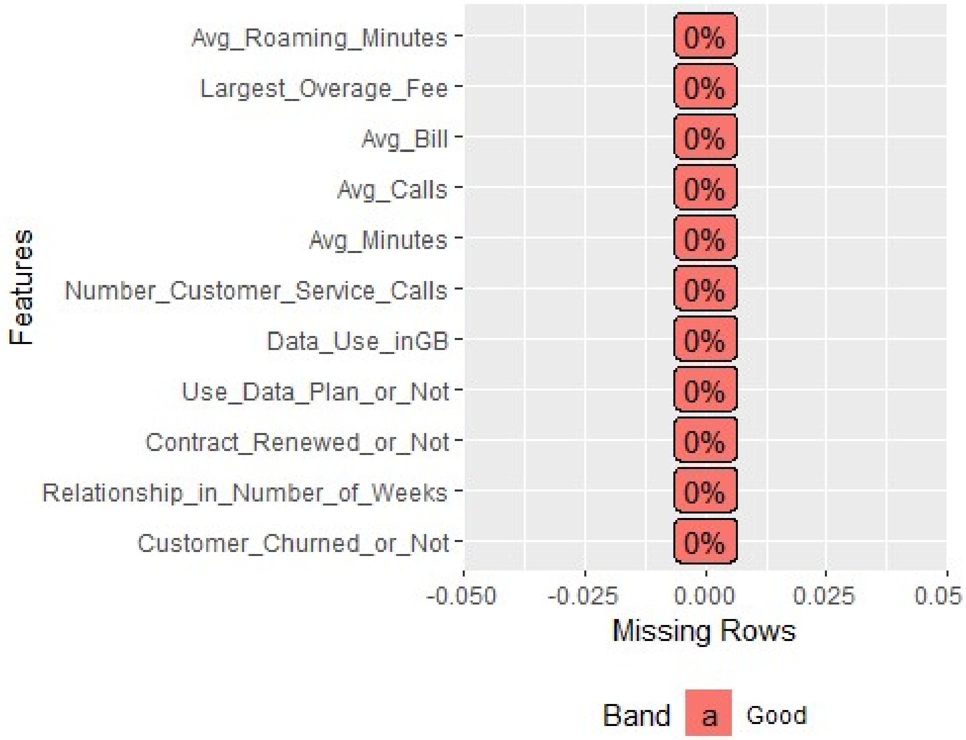
- ```
  #
  cat_variables=  c("Customer_Churned_or_Not", "Contract_Renewed_or_Not",
  "Use_Data_Plan_or_Not")
  org_File[,cat_variables] = lapply(org_File[,cat_variables] , factor)


  #Number of customer calls is converted to ordinal variable;This has bee
  n done for data analysis, bit for the models it would be converted to i
  ntegers

  org_File$Number_Customer_Service_Calls= factor(org_File$Number_Customer
  _Service_Calls, ordere=TRUE, levels = c(0,1,2,3,4,5,6,7,8,9))
  str(org_File)
  ```
  ```

CHECK THE FILE FOR MISSING VALUES

The dataset contain no missing values



EXPLORATORY DATA ANALYSIS

UNIVARIATE ANALYSIS

SUMMARY OF DATA

Summary function shows that out of 3333 observations, 483 customers have churned in the past data. ***Although this imbalance might affect the model performance, this would be used as is for the current analysis.***

Furthermore, most of the numeric variables seem to be normally distributed as the man and median values are close

```
summary(org_File)
```

```
##  Customer_Churned_or_Not Relationship_in_Number_of_Weeks
##  0:2850                  Min.   :  1.0
##  1: 483                  1st Qu.: 74.0
##                          Median :101.0
```

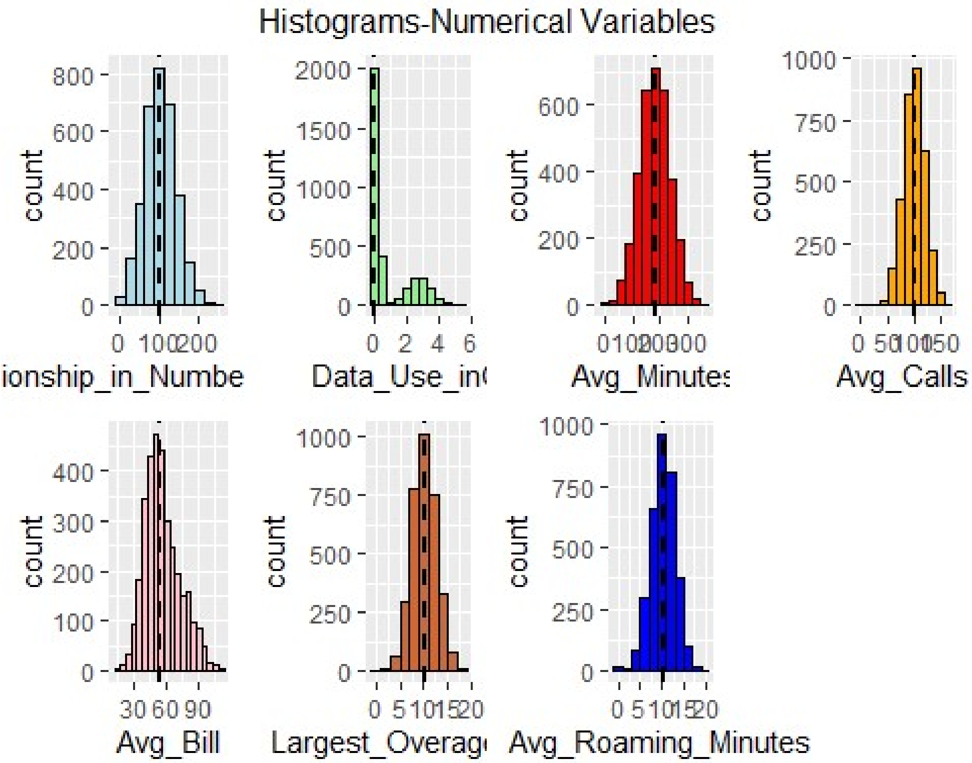
```
##                               Mean   :101.1
##                               3rd Qu.:127.0
##                               Max.   :243.0
##
##  Contract_Renewed_or_Not Use_Data_Plan_or_Not Data_Use_inGB
##  0: 323                  0:2411               Min.   :0.0000
##  1:3010                  1: 922               1st Qu.:0.0000
##                                               Median :0.0000
##                                               Mean   :0.8165
##                                               3rd Qu.:1.7800
##                                               Max.   :5.4000
##
##  Number_Customer_Service_Calls  Avg_Minutes      Avg_Calls
##  1      :1181                  Min.   :  0.0   Min.   :  0.0
##  2      : 759                  1st Qu.:143.7   1st Qu.: 87.0
##  0      : 697                  Median :179.4   Median :101.0
##  3      : 429                  Mean   :179.8   Mean   :100.4
##  4      : 166                  3rd Qu.:216.4   3rd Qu.:114.0
##  5      :  66                  Max.   :350.8   Max.   :165.0
##  (Other):  35
##     Avg_Bill       Largest_Overage_Fee Avg_Roaming_Minutes
##  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
```

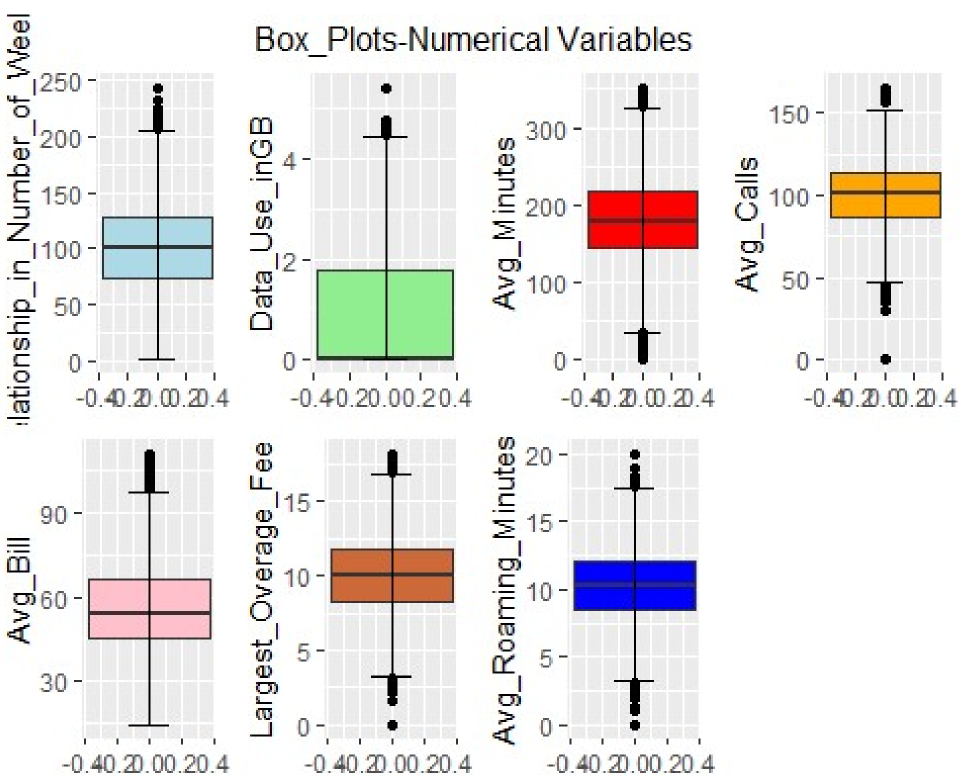
HISTOGRAMS & BOXPLOTS

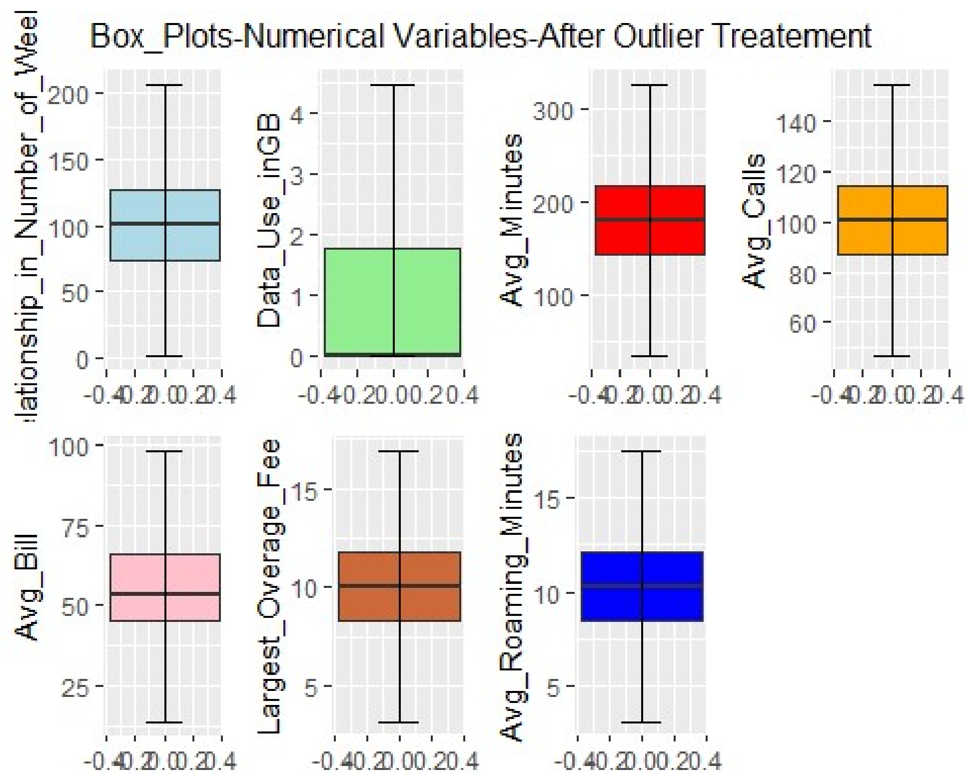
As observed, most of the numeric variables are nearly normally distributed.

However, the variable data use in GB's is right skewed, with a majority of customers not using any data.

Boxplots show the presence of outliers in all the numeric variables. ***These have been treated using the IQR rule***

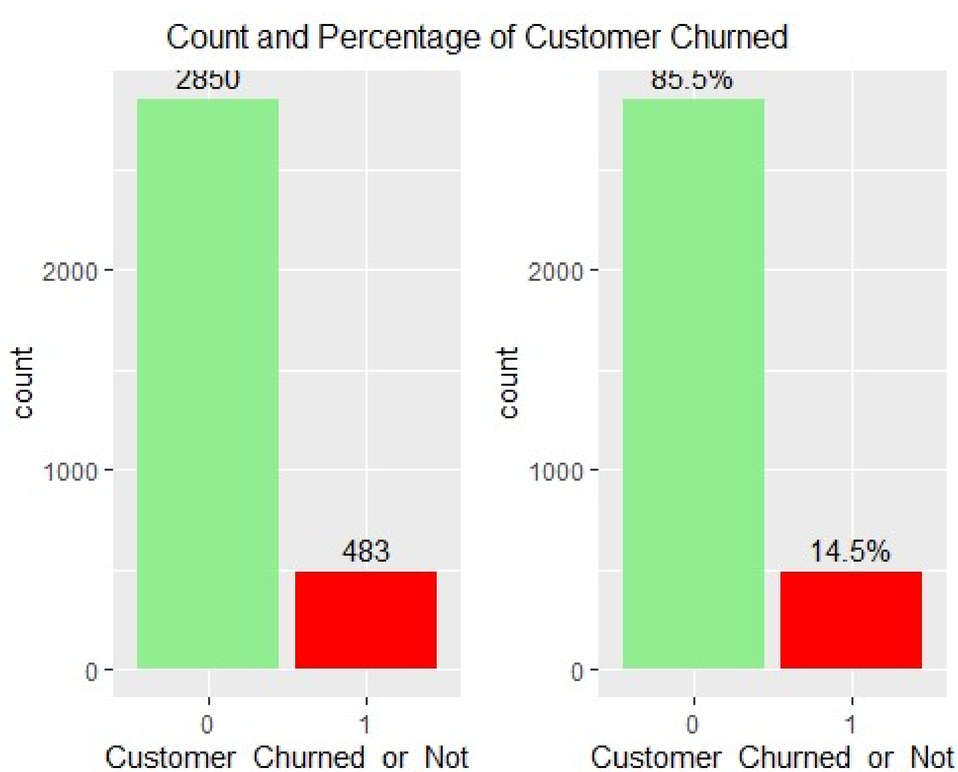

Histograms-Numerical Variables
count
Avg_Minutes
Avg_Calls
Avg_Bill
Avg_Roaming_Minutes


Box_Plots-Numerical Variables
Data_Use_inGB
Avg_Minutes
Avg_Calls
Avg_Bill
Largest_Overage_Fee
Avg_Roaming_Minutes



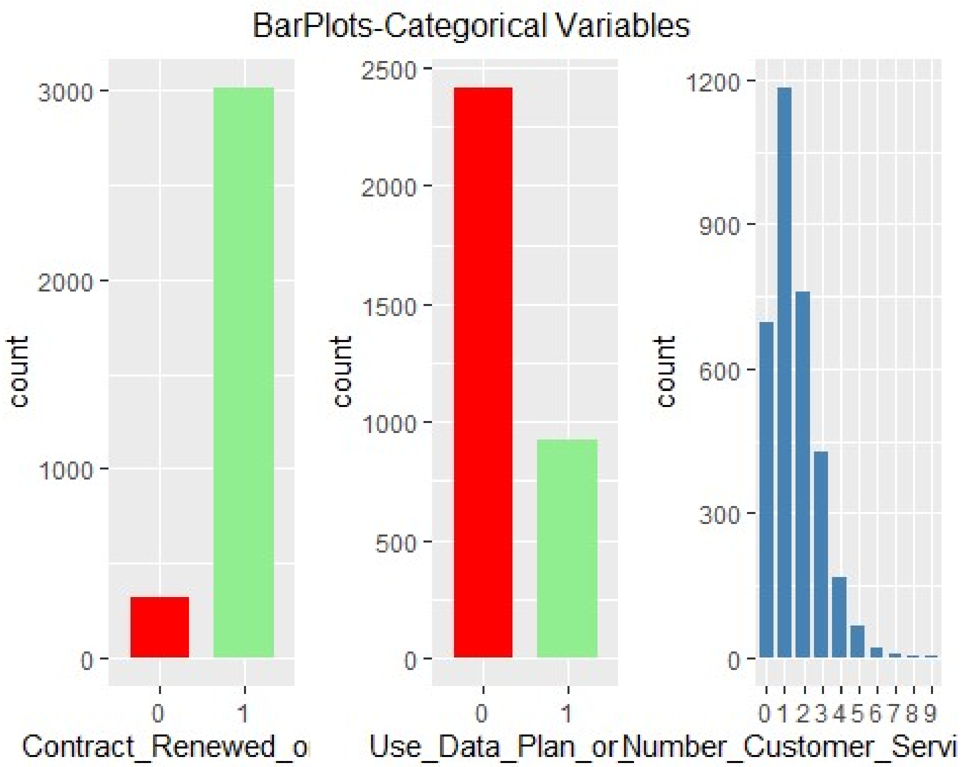
ANALYSIS IF CUSTOMER CHURN VARIABLE

Seeing the number and percentage of customers churned it can be seen that nealy 15% of customers have churned recently.



CATEGORICAL VARIABLE ANALYSIS-BAR PLOTS

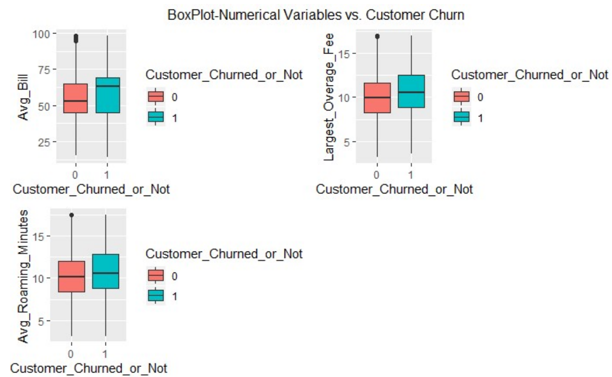
A relatively small number of customers have not renewed their contract, but majority of customers do not use data plan. It can also be seen that while a majority of customers call the customer service at least once, the count of customers decreases as the calls increase.



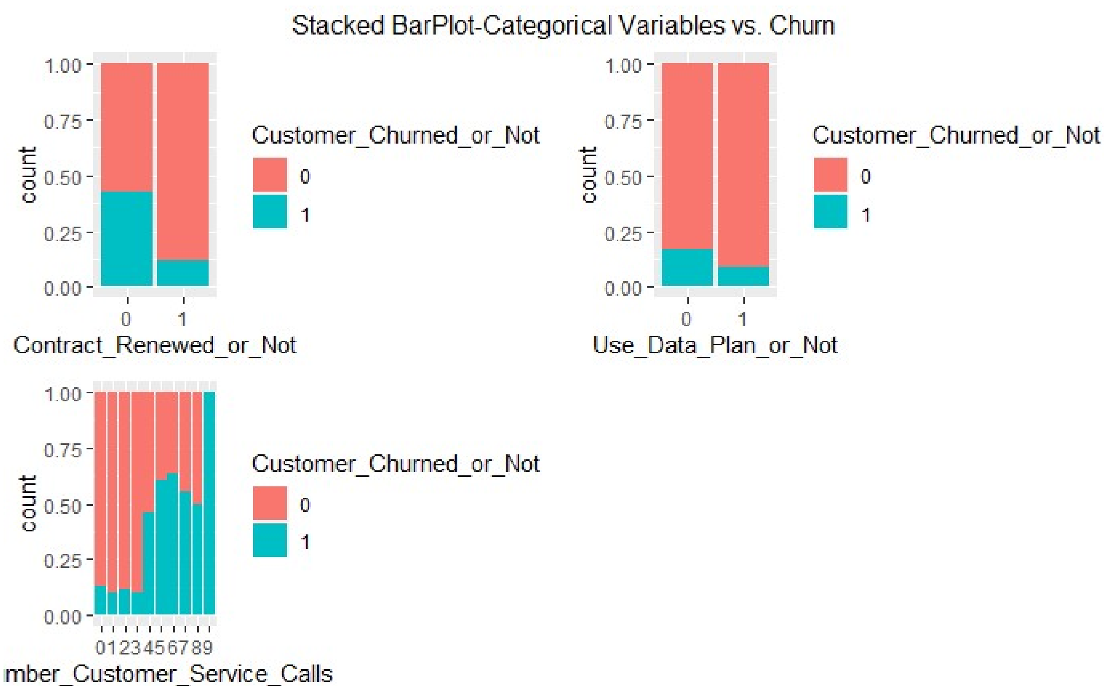
BIVARIATE ANALYSIS

Seeing the variables by customer churn indicates:

- ***Numerical Variables***
  - Customers with higher number of minutes and higher bill are more likely to churn
  - There is no significant difference in the whether a customer churned or not on the basis of relationship in number of weeks, average calls, largest overage fee or average roaming minutes
  - Customers who have churned have a lower range of data usage than the customers who have not.



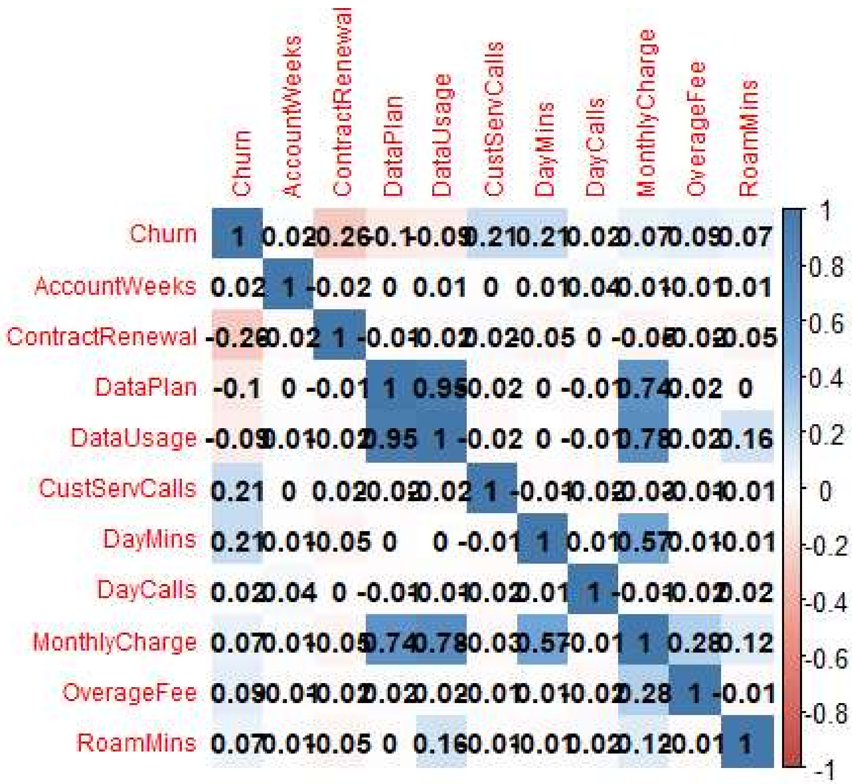
- ***Categorical Variables:***
    - Customers who have not renewed their contract recently as well as customers who do not use data plan show a propensity to churn more.
    - Furthermore, it can also be seen that as the number of calls to customer service increases, the more likely it becomes that the customer would cancel service



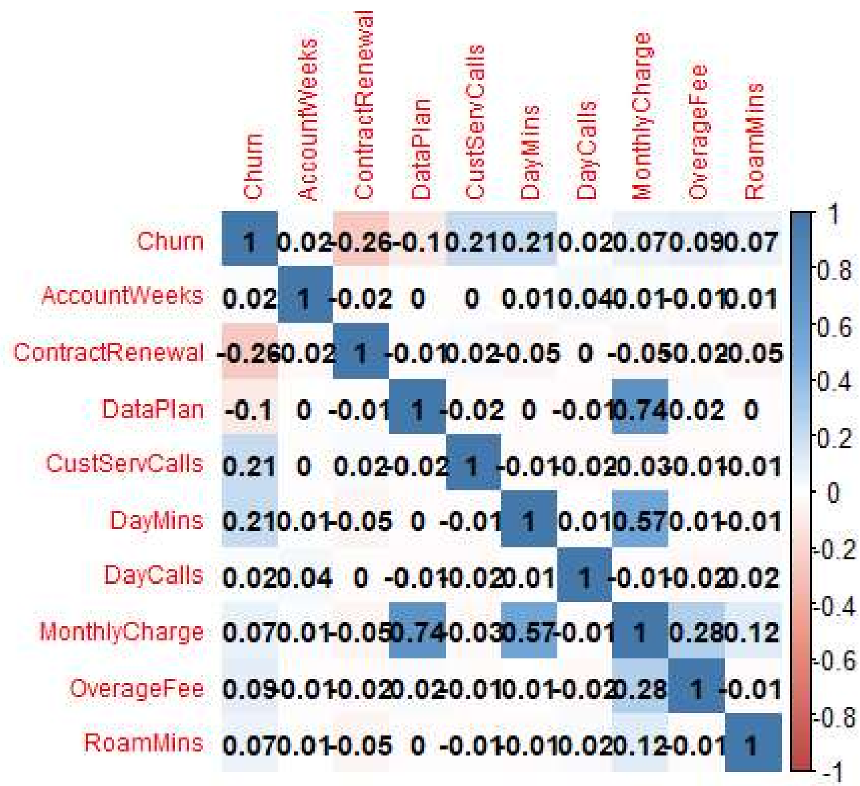
MULTIVARIATE ANALYSIS

From the correlation plot the variables use data plan or not as well as data use in GB are significantly correlated (0.95). ***The variable data use in GB would be removed for modelling purpose.***

Although monthly charge also seems to be correlated with data plan and dayMins, it has been retained currently for modelling purposes.



- Correlation Plot after removal of data use in GB variable



MODEL BUILDING

LOGISTIC REGRESSION MODEL

- Copy of test and training dataset are created
  - Training dataset has 2270 observations, while test set has 1063 observations
    - In the training data there 323 customers or 14.2% customers who have churned
    - In the test data there 160 customers or 15.1% customers who have churned
  - Logistic Model was built using all the variables, then features were removed with high VIF values. Further all insignificant variables were removed from the model to achieve the final mode.
- ***Model Evaluation***
  - ***Upon using the default 0.5 cutoff*** the training set gave an accuracy of 0.867 and the AUC of 0.81. For the test set, the accuracy is 0.85 and the AUC is 0.81. ***The final model seems to be good because the accuracy and AUC do not have big difference between the training and test sets.***
  - But the Specificities (customers likely to churn correctly classified)for two sets are as low as 0.17.

Evaluation Parameters-Training

```
# For training set
confusionMatrix(data = train_pred, reference = train_actual)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1905  258
##        Yes   42   65
##
##                Accuracy : 0.8678
##                  95% CI : (0.8532, 0.8815)
##     No Information Rate : 0.8577
##     P-Value [Acc > NIR] : 0.08723
##
##                   Kappa : 0.2492
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.9784
##             Specificity : 0.2012
##          Pos Pred Value : 0.8807
##          Neg Pred Value : 0.6075
##              Prevalence : 0.8577
##          Detection Rate : 0.8392
##    Detection Prevalence : 0.9529
##       Balanced Accuracy : 0.5898
##
##        'Positive' Class : No
##
```

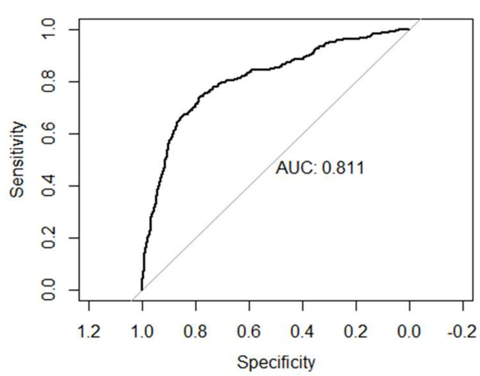
Evaluation Parameters-Testing

```
# For the test set
confusionMatrix(data = test_pred, reference = test_actual)

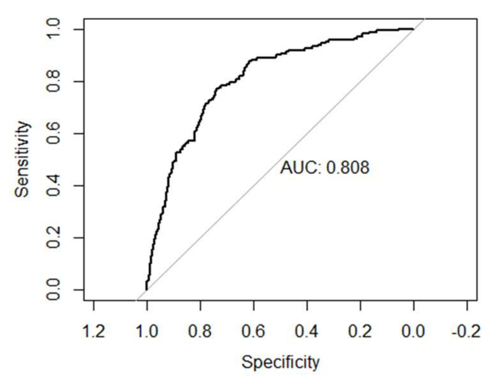
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  880 132
##        Yes  23  28
##
##                Accuracy : 0.8542
##                  95% CI : (0.8315, 0.8749)
##     No Information Rate : 0.8495
##     P-Value [Acc > NIR] : 0.3529
##
##                   Kappa : 0.2078
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9745
##             Specificity : 0.1750
##          Pos Pred Value : 0.8696
##          Neg Pred Value : 0.5490
##              Prevalence : 0.8495
##          Detection Rate : 0.8278
##    Detection Prevalence : 0.9520
##       Balanced Accuracy : 0.5748
##
##        'Positive' Class : No
##
```

ROC of Training and Test Sets

TRAINING ROC

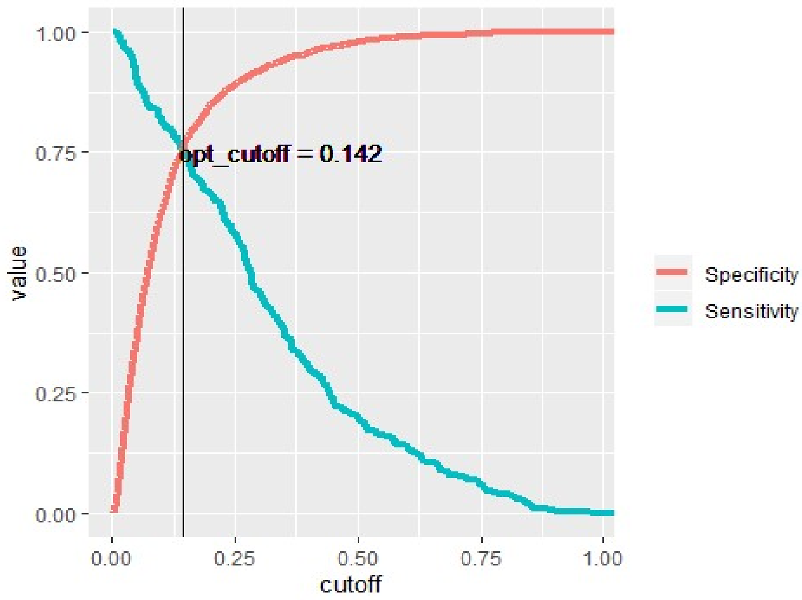


TEST ROC



However, since the objective of the analysis is to ***identify the maximum number of customers who are likely to churn***, the goal here should be maximize specificity, while not compromising much on the accuracy an sensitivity.

Hence an optimal cutoff was identified at 0.142 and implemented on both the testing and training sets



Upon using the optimal cut-off the following results were achieved:

For the training set, the Accuracy is 0.75, and the Sensitivity and Specificity are both about 0.73. For the test set, the Accuracy is 0.75, and the Sensitivity and Specificity are 0.76 and 0.73 respectively. Overall, this model with adjusted cutoff works well.

Evaluation Parameters-Training

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1459   77
##        Yes  488  246
##
##                Accuracy : 0.7511
##                  95% CI : (0.7328, 0.7688)
##     No Information Rate : 0.8577
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3338
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7494
##             Specificity : 0.7616
##          Pos Pred Value : 0.9499
##          Neg Pred Value : 0.3351
##              Prevalence : 0.8577
##          Detection Rate : 0.6427
##    Detection Prevalence : 0.6767
##       Balanced Accuracy : 0.7555
##
```

Evaluation Parameters-Testing

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  686  43
##        Yes 217 117
##
##                Accuracy : 0.7554
##                  95% CI : (0.7284, 0.781)
##     No Information Rate : 0.8495
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3392
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7597
##             Specificity : 0.7312
##          Pos Pred Value : 0.9410
##          Neg Pred Value : 0.3503
##              Prevalence : 0.8495
##          Detection Rate : 0.6453
##    Detection Prevalence : 0.6858
##       Balanced Accuracy : 0.7455
##
##        'Positive' Class : No
##
```

Furthermore 10-Fold cross validation was implemented on the model using the optimal cut-off and an accuracy of 74% was achieved

```
#10 Fold validation with Logistic Regression

set.seed(3000)
folds_logit = createFolds(logit_train$Customer_Churned_or_Not, k=10)
str(folds_logit)
```

```
## List of 10
##  $ Fold01: int [1:228] 6 9 12 22 25 42 44 55 56 79 ...
##  $ Fold02: int [1:227] 15 20 23 30 58 78 87 90 96 111 ...
##  $ Fold03: int [1:228] 19 26 32 37 38 57 59 66 68 69 ...
##  $ Fold04: int [1:226] 40 53 62 70 74 85 91 92 100 109 ...
##  $ Fold05: int [1:226] 2 16 24 31 35 61 77 102 113 121 ...
##  $ Fold06: int [1:227] 7 14 27 29 39 45 46 49 64 65 ...
##  $ Fold07: int [1:227] 1 5 11 13 18 28 34 36 82 108 ...
##  $ Fold08: int [1:227] 8 10 21 33 50 76 86 98 101 103 ...
##  $ Fold09: int [1:227] 3 4 47 54 72 73 80 83 84 94 ...
##  $ Fold10: int [1:227] 17 41 43 48 51 52 60 63 67 81 ...
```

```
cv_logit=lapply(folds_logit,function(x){
  train.logit.kval=logit_train[x,]
  test.logit.kval=logit_test[-x,]
  logit.kval=glm(logit_Eq_3, train.logit.kval, family = binomial)
  logit.kval.pred=predict(logit.kval, test.logit.kval, type = "response")
  tab.logit.kval=table(test.logit.kval$Customer_Churned_or_Not, logit.kval.pr
ed>0.14)

  sum(diag(tab.logit.kval))/sum(tab.logit.kval)
})

str(cv_logit)
```

```
## List of 10
##  $ Fold01: num 0.715
##  $ Fold02: num 0.714
##  $ Fold03: num 0.749
##  $ Fold04: num 0.725
##  $ Fold05: num 0.729
##  $ Fold06: num 0.719
##  $ Fold07: num 0.773
##  $ Fold08: num 0.734
##  $ Fold09: num 0.786
##  $ Fold10: num 0.754
```

```
fit.logit<-mean(unlist(cv_logit))
fit.logit
```

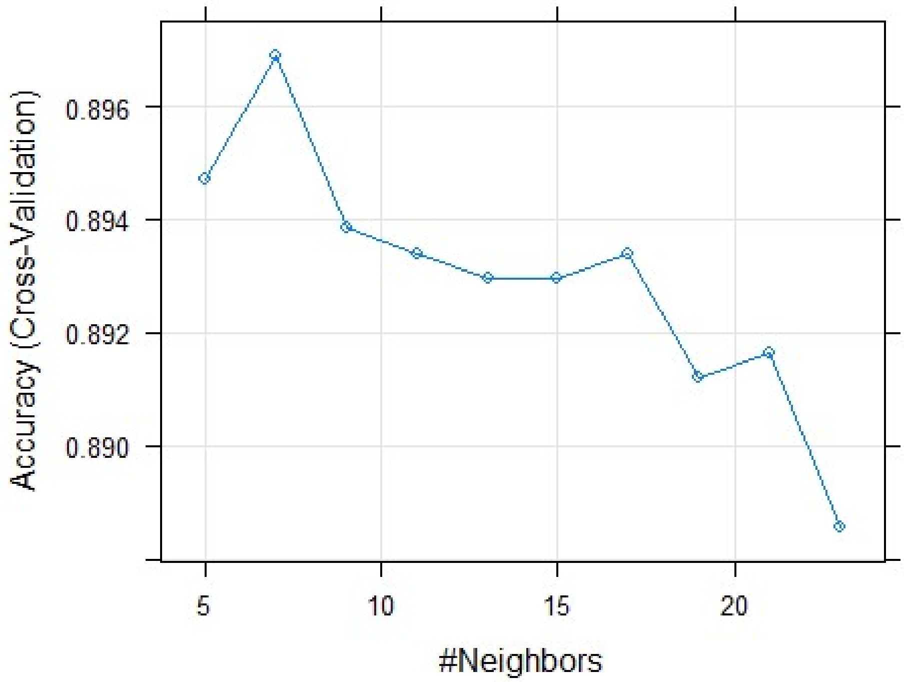
```
## [1] 0.7396284
```

OTHER PERFORMANCE MEASURES FOR TEST AND TRAINING SET

	Training	Test
KS	0.524	0.52
AUC	0.81	0.80
Gini	0.17	0.16

K-NEAREST NEIGHBOR MODEL

- The model development using the caret package indicated that with k=7 a maximum accuracy could be achieved (89.69)



- ***However, since the primary objective was to identify the maximum number of customers who are likely to churn, models with other k values were tried*** with a goal to maximize specificity and not compromise much on accuracy.
  - It was discovered that with k=3, accuracy was not compromised much, but specificity increased to 0.49. Hence. K=3 was chosen in the final model.

Evaluation Parameters-k=7

```
#knn with k-7
KNN_7<-knn(train=KNN_train_New[,-1],test=KNN_test_New[-1], cl=KNN_train[,1
],k=7)
confusionMatrix(data = KNN_7, reference = KNN_test_New[,1])

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 889  96
##          1  14  64
##
##                Accuracy : 0.8965
##                  95% CI : (0.8766, 0.9142)
##     No Information Rate : 0.8495
##     P-Value [Acc > NIR] : 4.382e-06
##
##                   Kappa : 0.4872
##
##  Mcnemar's Test P-Value : 1.136e-14
##
##             Sensitivity : 0.9845
##             Specificity : 0.4000
##          Pos Pred Value : 0.9025
##          Neg Pred Value : 0.8205
##              Prevalence : 0.8495
##          Detection Rate : 0.8363
##    Detection Prevalence : 0.9266
##       Balanced Accuracy : 0.6922
##
##        'Positive' Class : 0
```

Evaluation Parameters-k=3

```
#knn with k-3
KNN_3<-knn(train=KNN_train_New[,-1],test=KNN_test_New[-1],
cl=KNN_train[,1],k=3)
confusionMatrix(data = KNN_3, reference = KNN_test_New[,1])

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 867  82
##          1  36  78
##
##                Accuracy : 0.889
##                  95% CI : (0.8686, 0.9073)
##     No Information Rate : 0.8495
##     P-Value [Acc > NIR] : 0.0001109
##
##                   Kappa : 0.5077
##
##  Mcnemar's Test P-Value : 3.434e-05
##
##             Sensitivity : 0.9601
##             Specificity : 0.4875
##          Pos Pred Value : 0.9136
##          Neg Pred Value : 0.6842
##              Prevalence : 0.8495
##          Detection Rate : 0.8156
##    Detection Prevalence : 0.8928
##       Balanced Accuracy : 0.7238
##
##        'Positive' Class : 0
##
```

Finally 10-fold cross validation was performed to achieve an accuracy level of 86.39

NAÏVE BAYES MODEL

Although the base version of the Naive bayes model requires categorical predictors, the given dataset contains mostly numeric variables.

- One of the ways to deal with this situation is to discretize numeric variable using bins, however since the number of variables concerned is large, it might result in information loss.
- ***Other option for numerical variable is that normal distribution is assumed (bell curve, which is a strong assumption)***.
    - In our case all the numeric variables follow nearly normal distribution, hence naive bayes can be applied.
        - The variable data use in GB, which was skewed has been removed for Naive bayes
    - Further, normalized values of numerical variables have been considered to build the model

Evaluation Parameters Naïve Bayes

```
#predictions using Naive Bayes
NB_Model_1_Predict= predict(NB_Model_1,newdata=NB_Model_test[-1])

# Checking Confusion matrix
confusionMatrix(data = NB_Model_1_Predict, reference = NB_Model_test[,1])
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 867 103
##          1  36  57
##
##                Accuracy : 0.8692
##                  95% CI : (0.8475, 0.8889)
##     No Information Rate : 0.8495
##     P-Value [Acc > NIR] : 0.03747
##
##                   Kappa : 0.3822
##
##  Mcnemar's Test P-Value : 2.168e-08
##
##             Sensitivity : 0.9601
##             Specificity : 0.3563
##          Pos Pred Value : 0.8938
##          Neg Pred Value : 0.6129
##              Prevalence : 0.8495
##          Detection Rate : 0.8156
##    Detection Prevalence : 0.9125
##       Balanced Accuracy : 0.6582
##
##        'Positive' Class : 0
```

Finally 10-fold cross validation was performed to achieve accuracy rate of 85.7

MODEL COMPARISON AND INTERPRETATION

Parameters	Logistic Regression	KNN Model	NAÏVE BAYES
Model Accuracy-Test Set	75.5	88.9	86.9
Accuracy on 10-Fold Validation	73.9	86.4	85.7
Sensitivity	75.9	96.0	96.0
Specificity	73.1	48.7	35.6

FINAL INSIGHTS

MODEL PERFORMANCE INSIGHTS

The performance of all the models showed good accuracy and other performance measures. The high accuracy could be a result of imbalanced nature of the data.

Upon looking at the performance of various models on the given dataset following could be deduced:

Although the KNN model performed the best in terms of overall accuracy levels (followed by Naïve Bayes and Logistic Regression), ***since the primary objective of this exercise is to correctly identify customers who are likely to leave, it is the logistic regression model that performed best in identifying the potential churns (highest specificity of 73.1)***.

The logistic regression model is correctly able to classify ~73-75% and based on the analysis the company could take actions to reduce the churn rate. However it would strongly be suggested to test on more data or treat the imbalance as seen in the current database to get more reliable results.

BUSINESS INSIGHTS

- From the initial data analysis some observations were made which if implemented could help the company reduce churn. These include among others:
  - High Churn rate was seen in customers as the calls to customer service increased. This could most probably due to complaints not being resolved. The company could take proactive steps to ensure that number of unresolved complaints are minimized so as to reduce customer dissatisfaction

- It was seen that the customers with higher minutes and bills showed more propensity to churn. The company could send out some offers to these customers so as to increase loyalty.