6/21/2020

MINI PROJECT- CELLPHONE PROJECT-Telecom Customer Churn Prediction Assessment

PREDICTIVE MODELLING

Rithu A M

Post Graduate Program in Business Analytics and Business Intelligence

Contents

Project Objective:

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 post-paid 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.

You are expected to do the following:

1. **EDA**
    - How does the data looks like, Univariate and bivariate analysis? Plots and charts which illustrate the relationships between variables
    - Look out for outliers and missing values
    - Check for multicollinearity & treat it
    - Summarize the insights you get from EDA

2. **Build Models and compare them to get to the best one**
    - Logistic Regression
    - KNN
    - Naive Bayes (is it applicable here? comment and if it is not applicable, how can you build an NB model in this case?)
    - Model Comparison using Model Performance metrics & Interpretation

3. **Actionable Insights**
    - Interpretation & Recommendations from the best model

Project Approach

1.1EDA - Basic data summary, Univariate, Bivariate analysis, graphs

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

1.3 EDA - Check for Multicollinearity - Plot the graph based on Multicollinearity & treat it.

1.4 EDA - Summarize the insights you get from EDA

2.1 Applying Logistic Regression

2.2 Interpret Logistic Regression

2.3 Applying KNN Model

2.4 Interpret KNN Model

2.5 - Applying Naive Bayes Model

2.6 Interpret Naive Bayes Model

2.7 Confusion matrix interpretation for all models

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

2.9 Remarks on Model validation exercise <Which model performed the best>
3. Actionable Insights and Recommendations

Assumptions

- Churn
- Account Weeks
- Contract Renewal
- Data Plan
- Data Usage
- Customer Service Calls
- Day Mins
- Day Calls
- Monthly Charge
- Overage Fee
- Roam Mins

Environment Setup and Data Import

Install necessary Packages and Invoke Libraries
Use this section to install necessary packages and invoke associated libraries. Having all the packages at the same places increases code readability.
Below are the Packages used in this project:

- library(readr)
- library(readxl)
- library(ggplot2)
- library(gridExtra)
- library(DataExplorer)
- library(dplyr)
- library(corrplot)
- library(car)
- library(caret)
- library(lattice)
- library(e1071)
- library(caTools)
- library(ROCR)
- library(pROC)
- library(blorr)
- library(kableExtra)

Set up working Directory

Setting a working directory on starting of the R session makes importing and exporting data files and code files easier. Basically, working directory is the location/ folder on the PC where you have the data, codes etc. related to the project.

```
setwd("G:/Projects/Predictive Modelling Project")
> getwd()
[1] "G:/Projects/Predictive Modelling Project"
```

Read & Import Dataset

The given dataset is in .xlsx format. Hence, the command 'read.csv' is used for importing the file.

```
cell=read.csv("Cellphone.csv",header = TRUE)
```

Dataset has 3333Observations divided amongst 11 variables

Dimension of Dataset

```
##Checking the dimensions of dataset##
> dim(cell)
[1] 3333   11
```

```
cell$Churn=as.factor(cell$Churn)
> cell$ContractRenewal=as.factor(cell$ContractRenewal)
> cell$DataPlan=as.factor(cell$DataPlan)
```

Converting Churn, Contract Renewal and Data plan as factored variables as they have valve as 0 or 1

Structure of Dataset

```
str(cell)
'data.frame':   3333 obs. of  11 variables:
 $ Churn          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ AccountWeeks   : int  128 107 137 84 75 118 121 147 117 141 ...
 $ ContractRenewal: Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 1 ...
 $ DataPlan       : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 2 1 1 2 ...
 $ 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 ...
```

Responsible Variable Churn is an imbalanced class with 2850 as No or '0' and 483 as Yes or '1'

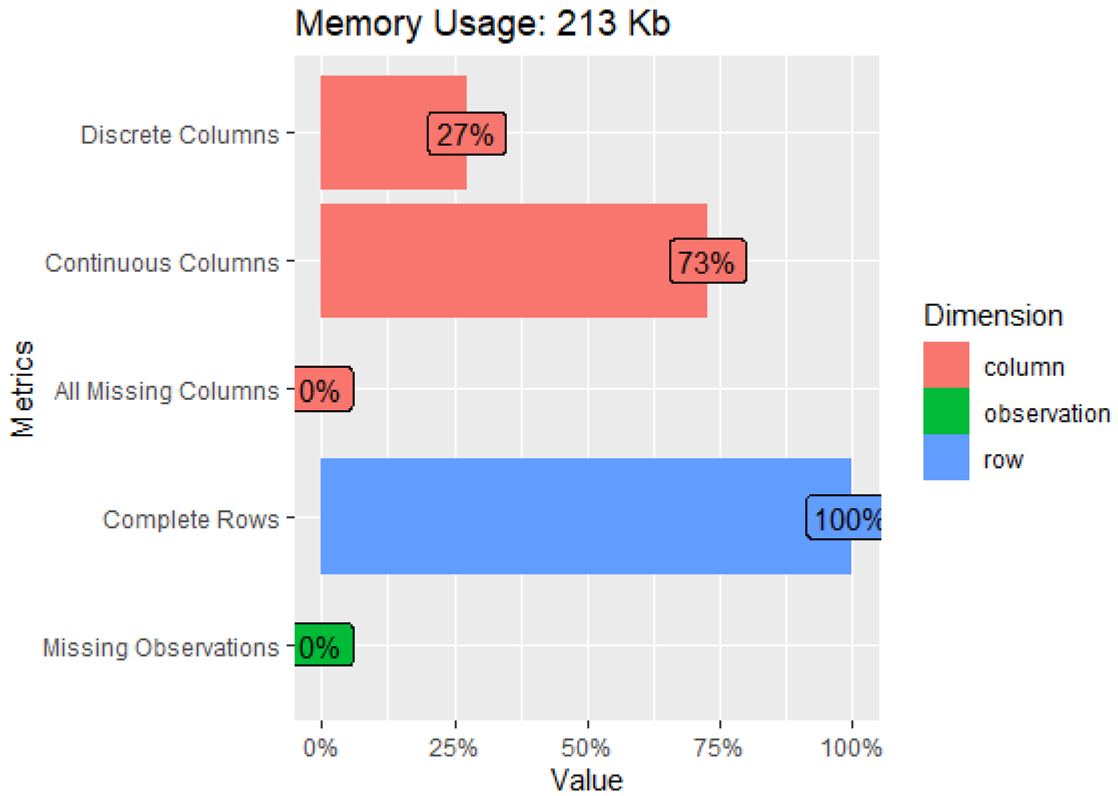
Summary of Dataset

```
summary(cell)
 Churn     AccountWeeks    ContractRenewal DataPlan   DataUsage       CustSe
rvCalls       DayMins
 0:2850   Min.   :  1.0   0: 323          0:2411   Min.   :0.0000   Min.   
:0.000   Min.   :  0.0  
 1: 483   1st Qu.: 74.0   1:3010          1: 922   1st Qu.:0.0000   1st Qu
.:1.000   1st Qu.:143.7  
          Median :101.0                            Median :0.0000   Median 
:1.000   Median :179.4  
          Mean   :101.1                            Mean   :0.8165   Mean   
:1.563   Mean   :179.8  
          3rd Qu.:127.0                            3rd Qu.:1.7800   3rd Qu
.:2.000   3rd Qu.:216.4  
          Max.   :243.0                            Max.   :5.4000   Max.   
:9.000   Max.   :350.8  
    DayCalls      MonthlyCharge      OverageFee       RoamMins    
 Min.   :  0.0   Min.   : 14.00   Min.   : 0.00   Min.   : 0.00  
 1st Qu.: 87.0   1st Qu.: 45.00   1st Qu.: 8.33   1st Qu.: 8.50  
 Median :101.0   Median : 53.50   Median :10.07   Median :10.30  
 Mean   :100.4   Mean   : 56.31   Mean   :10.05   Mean   :10.24  
 3rd Qu.:114.0   3rd Qu.: 66.20   3rd Qu.:11.77   3rd Qu.:12.10  
 Max.   :165.0   Max.   :111.30   Max.   :18.19   Max.   :20.00  
```

Exploratory Data Analysis

```
##Introductory plot of dataset##
> plot_intro(cell)
```

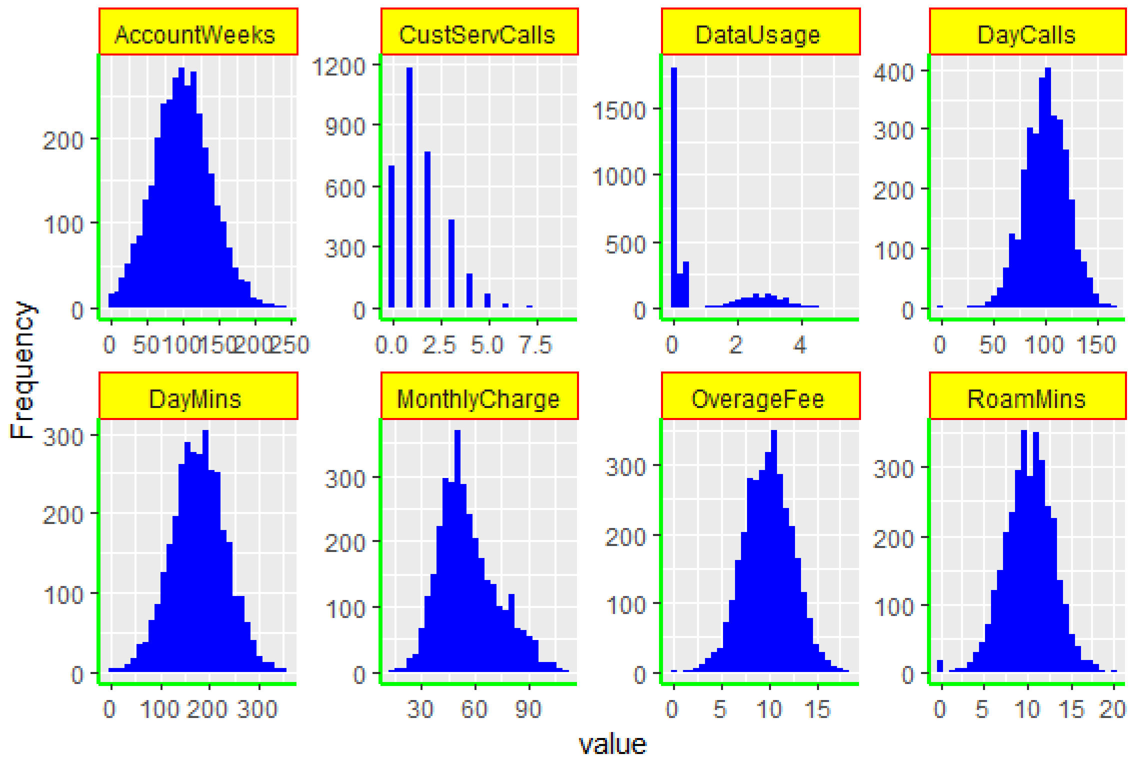
Introductory Plot



```
##Histogram of Variables##
> plot_histogram(cell,geom_histogram_args = list(fill="blue"),
+                theme_config = list(axis.line=element_line(size=1,colour=
"green"),
+                                    strip.background=element_rect(color="
red",fill="yellow")))
```

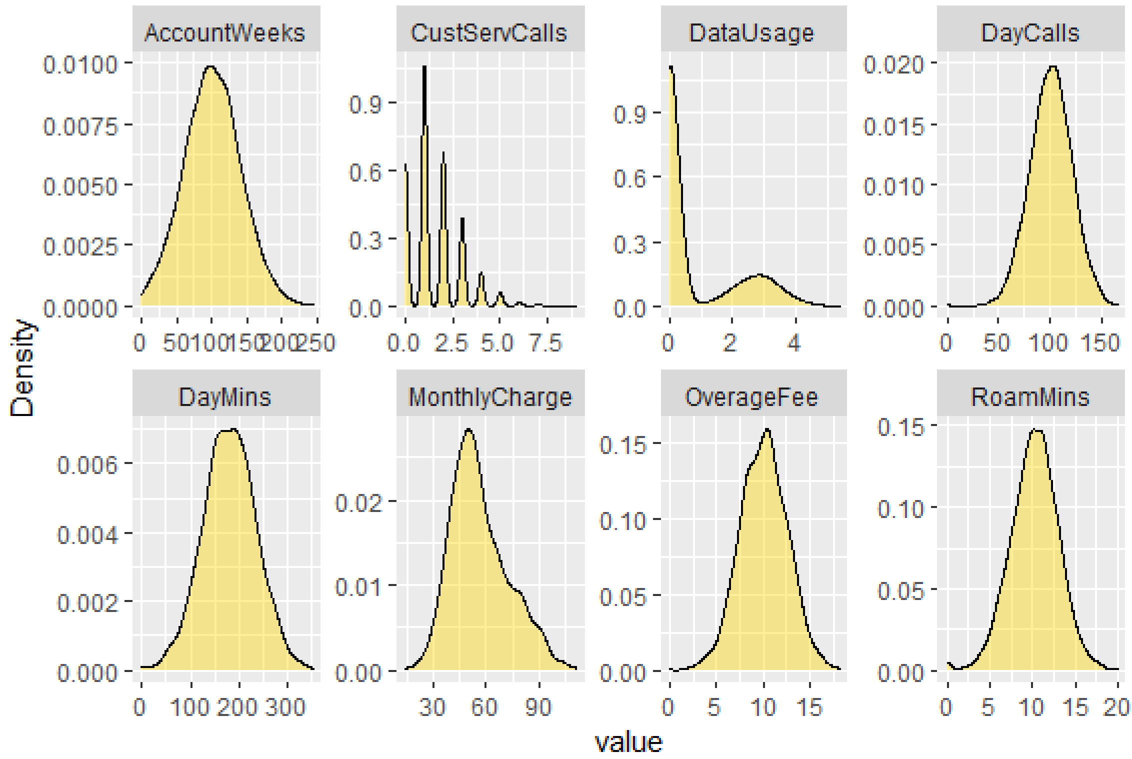
Checking the Distribution of Variables

<u>Histogram Plot</u>



```
##Density Plots of variables##
> plot_density(cell,geom_density_args = list(fill="gold",alpha=0.4))
```

Density Plot

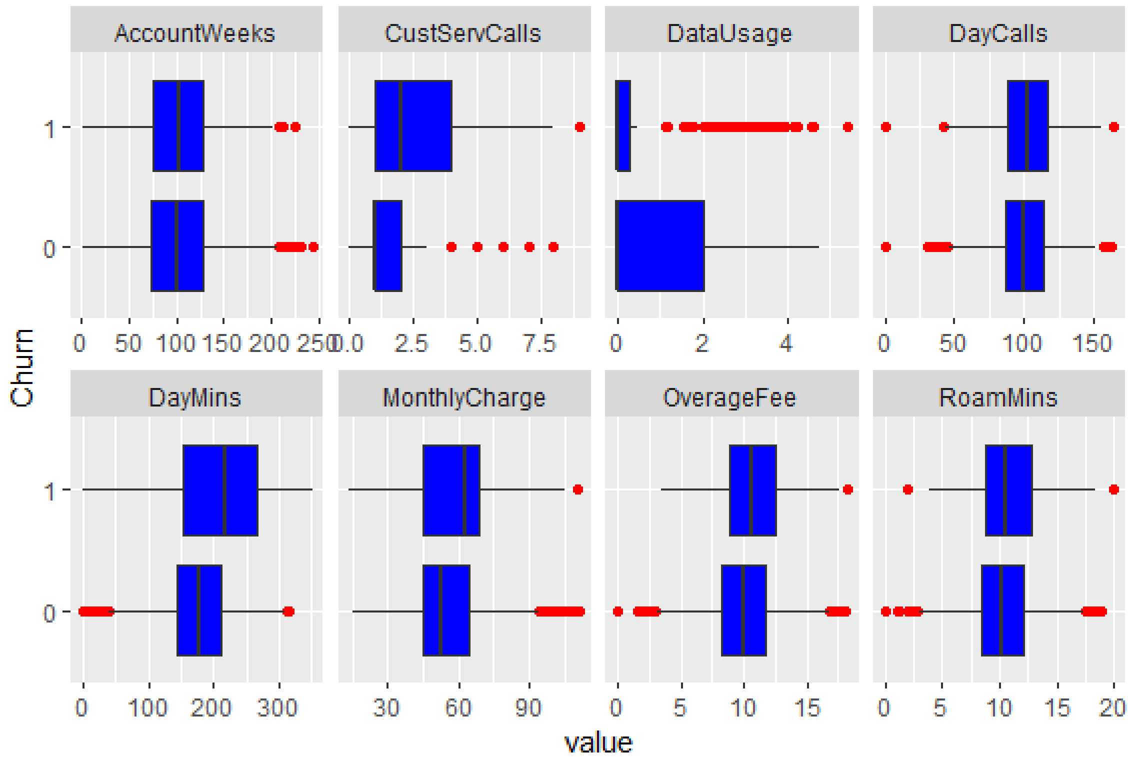


Checking for Outliers

Checking outliers with respect to Churn response Variable

```
plot_boxplot(cell,by ="Churn", geom_boxplot_args = list("outlier.color" =
"red", fill="blue"))
```

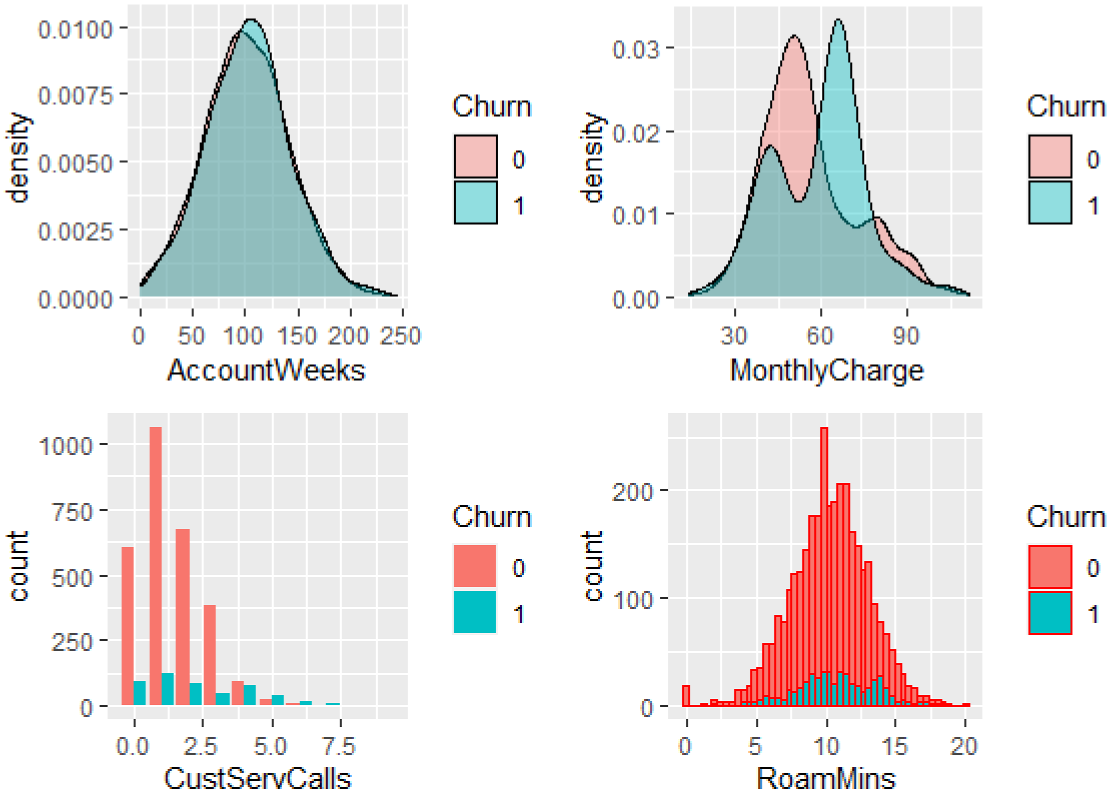
Box Plot



- Data Usage has many outliers for the Churners (Class 1)
- Day Mins and Monthly Charge has many outliers in the Non - Churner Category (Class 0)

Bivariate Analysis

```
p1 = ggplot(cell, aes(AccountWeeks, fill=Churn)) + geom_density(alpha=0.4)
> p2 = ggplot(cell, aes(MonthlyCharge, fill=Churn)) + geom_density(alpha=0
.4)
> p3 = ggplot(cell, aes(CustServCalls, fill=Churn))+geom_bar(position = "d
odge")
> p4 = ggplot(cell, aes(RoamMins, fill=Churn)) + geom_histogram(bins = 50,
color=c("red"))
> grid.arrange(p1, p2, p3, p4, ncol = 2, nrow = 2)
```



Insights

- From Histograms almost all continuous predictors like Account Weeks, Day Calls/Mins, OverageFee, Roam mins have normal distributions

- Monthly charge has its distribution skewed to a bit left which can be ignored

- Customers who churn vs who dont are mostly have similar distribution for the Account weeks with mean of Churn(1) = 103 Weeks and Not Churn(0) ~ 101 Weeks

- On an Average Customers who Churn are utilizing more Day Minutes(207 mins) than who don’t (175 mins)

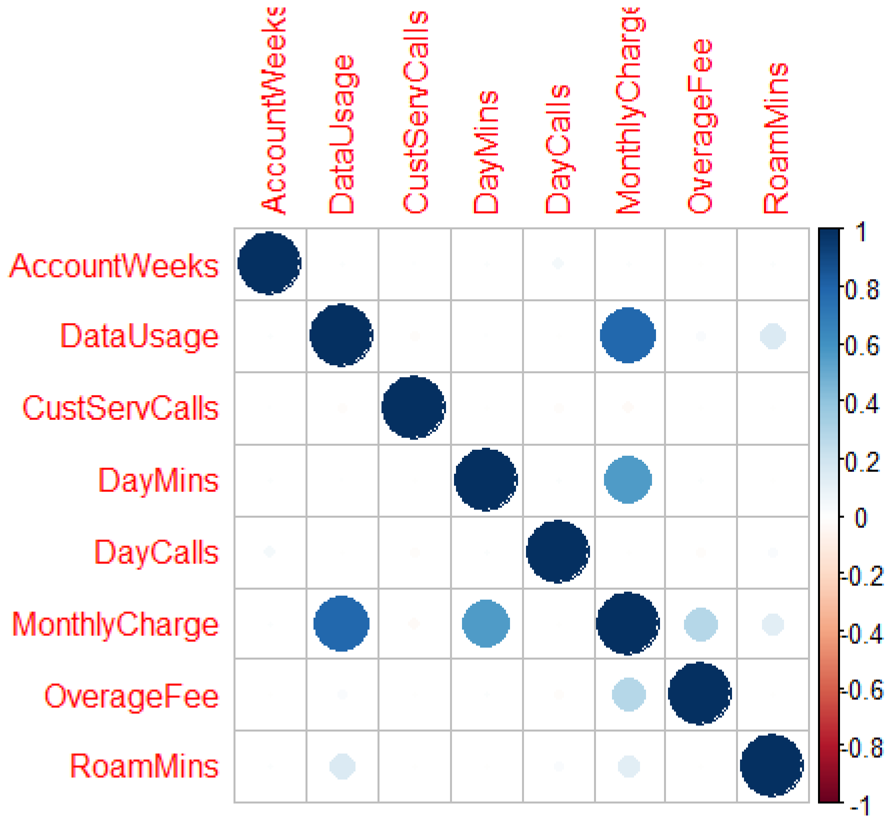
- On the other hand Churning customers data usage (0.54 GB)on an average is less compared to Non-Churning ones (0.86 GB)

- Churning Customers call Customer Service more in the bracket of (5 - 10 calls) v/s the bracket of (0-5 Calls)

- Monthly Charges are also more for Churn customers compared to Non-Churn in the 60 - 75 monetary amounts

Check for Multicollinearity

```
cell.numeric=cell%>% select_if(is.numeric)
> a=round(cor(cell.numeric),2)
> corrplot(a)
```



Insight On Collinearity

Data Suggests there is very strong correlation between Monthly charges and data usage which is quite obvious .So we can replace one variable with another after evaluation.

Apart from that no predictor has shown VIF (Variance inflation factor of beyond 2) so doing a principal component Analysis to cure Multicollinearity can be ignored for this dataset.

Classifier Models

Splitting dataset -Train and Test

```
set.seed(233)
> split = createDataPartition(cell$Churn , p=0.7, list = FALSE)
> train.cell = cell[split,]
> test.cell = cell[-split,]
```

Checking Dimensions of Train and Test Splits of Dataset

Train Dataset

```
dim(train.cell)
[1] 2334   11
```

Test Dataset

```
dim(test.cell)
[1] 999  11
```

Matrix for check of split of Response var in Train and Test Datasets

Train Dataset

```
table(train.cell$Churn)

   0    1 
1995  339 
```

Test Dataset

```
table(test.cell$Churn)

  0   1 
855 144 
```

Split	Class0(No-Churn)	Class1(Churn)
Train Cell	1995	339
Test Cell	855	144

Glaring Imbalance of Classes in Train and Test sets can be cured through up or down sample

K-Nearest Neighbour Classifier

```
trctl = trainControl(method = "repeatedcv", number = 10, repeats = 3)
> set.seed(1111)
> knn.fit = train(Churn~., data = train.cell, method="knn",
+                 trControl= trctl, preProcess = c("center", "scale"),
+                 tuneLength= 10)
> knn.fit
```

```
k-Nearest Neighbors

2334 samples
  10 predictor
   2 classes: '0', '1'

Pre-processing: centered (10), scaled (10)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 2101, 2101, 2100, 2100, 2101, 2101, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.8930363  0.4698420
   7  0.8944633  0.4633050
   9  0.8924671  0.4464257
  11  0.8916063  0.4307874
  13  0.8923106  0.4335727
  15  0.8905981  0.4144264
  17  0.8901714  0.3960884
  19  0.8901665  0.3839394
  21  0.8911704  0.3871440
  23  0.8891712  0.3687066

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 7.
```

Interpretation of K-NN

```
knn.pred=predict(knn.fit,test.cell)
> mean(knn.pred==test.cell$Churn)
[1] 0.9079079
```

Confusion Matrix for K-NN

```
knn.CM=confusionMatrix(knn.pred,test.cell$Churn,positive = "1")
> knn.CM
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 840  77
         1  15  67

               Accuracy : 0.9079
                 95% CI : (0.8883, 0.9251)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 4.708e-07

                  Kappa : 0.5454

 Mcnemar's Test P-Value : 2.022e-10

            Sensitivity : 0.46528
            Specificity : 0.98246
         Pos Pred Value : 0.81707
         Neg Pred Value : 0.91603
             Prevalence : 0.14414
         Detection Rate : 0.06707
   Detection Prevalence : 0.08208
      Balanced Accuracy : 0.72387

       'Positive' Class : 1
```

- Trained tuned model for K-NN gives 7 as the optimal valve for the accuracy of 89.23%

- Repeated Cross validation method was used to get the optimal valve of"K"

- Confusion Matrix suggests that model has very high accuracy of 90% but its positive class prediction rate (Churning Rate) is around 81% which is good enough in real time scenarios but can be improved with further tuning

Naïve Bayes Classifier

```
NB.fit=naiveBayes(Churn~.,data = train.cell)
> NB.fit
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.8547558 0.1452442 

Conditional probabilities:
   AccountWeeks
Y       [,1]     [,2]
  0 100.7193 40.56308
  1 102.8142 39.02356

   ContractRenewal
Y            0          1
  0 0.06015038 0.93984962
  1 0.28613569 0.71386431

   DataPlan
Y           0         1
  0 0.6942356 0.3057644
  1 0.8466077 0.1533923

   DataUsage
Y        [,1]     [,2]
  0 0.8966065 1.312729
  1 0.5113569 1.116933

   CustServCalls
Y       [,1]     [,2]
  0 1.432080 1.154023
  1 2.147493 1.809431

   DayMins
Y       [,1]     [,2]
  0 174.9146 49.91691
  1 210.7254 68.31656

   DayCalls
Y       [,1]     [,2]
  0 100.3484 19.65042
  1 102.5428 20.07413

   MonthlyCharge
Y       [,1]     [,2]
  0 56.04576 16.67845
  1 59.73304 15.55125

   OverageFee
Y        [,1]     [,2]
  0  9.912381 2.511852
  1 10.764513 2.510468

   RoamMins
Y       [,1]     [,2]
  0 10.20206 2.798420
  1 10.57788 2.737805
```

- Navie Bayes works best with Categorical values but can be made to work on mix datasets having continuous as well as categorical variables as predictors like in cellphone dataset

- Since this algo runs on Conditional Probabilities it becomes very hard to silo the continous variables as they have no frequency but a continuum scale

- For continous variables what NB does is takes their mean and standard deviation or variability and treats it as cut off thresholds ; say anything less than mean of distributed predictor values is 0 and more than mean is 1

- Above law suits binary classifier ; however if we have multinomial Response categories than it will have to go for quantiles, deciles n-iles partitioning the data accordingly and assigning them the probabilities

- Based on above NB's working on mixed dataset and its accuracy is always questionable.Its findings and predictions need to be supported by other Classifiers before any actionable operations

- The Output for the NB model displays in the matrix format for each predictor its mean [,1] and std deviation [,2] for class 1 and class 0

- The independence of predictors (no-multicollinearity) has been assumed for sake of simplicity

Intepretation of Naïve Bayes

```
NB.pred=predict(NB.fit,test.cell,type = "class")
> mean(NB.pred==test.cell$Churn)
[1] 0.8658659
```

```
NB.CM=confusionMatrix(NB.pred,test.cell$Churn,positive = "1")
> NB.CM
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 814  93
         1  41  51

               Accuracy : 0.8659
                 95% CI : (0.8432, 0.8864)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.1968

                  Kappa : 0.3603

 Mcnemar's Test P-Value : 1.054e-05

            Sensitivity : 0.35417
            Specificity : 0.95205
         Pos Pred Value : 0.55435
         Neg Pred Value : 0.89746
             Prevalence : 0.14414
         Detection Rate : 0.05105
   Detection Prevalence : 0.09209
      Balanced Accuracy : 0.65311

       'Positive' Class : 1
```

- Definitely its accuracy is 86% but its positive prediction rate is 55% which is quite low

- Also, the method of assigning probabilities is not dependable for continuous predictors

- Sensitivity which is TP/ TP + FN is just 35 % another hallmark of untrustworthiness

Logistic Regression Classifier

Running a Logit R through GLM

```
logitR.fit=glm(Churn~.,data = train.cell,family="binomial")
> summary(logitR.fit)

Call:
glm(formula = Churn ~ ., family = "binomial", data = train.cell)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0721  -0.5086  -0.3378  -0.1896   3.0096

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.705390   0.669799 -10.011   <2e-16 ***
AccountWeeks      0.001255   0.001672   0.751   0.4529
ContractRenewal1 -2.082526   0.177252 -11.749   <2e-16 ***
DataPlan1        -1.478519   0.660252  -2.239   0.0251 *
DataUsage        -1.171521   2.335388  -0.502   0.6159
CustServCalls     0.493564   0.048086  10.264   <2e-16 ***
DayMins          -0.008078   0.039429  -0.205   0.8377
DayCalls          0.006872   0.003394   2.025   0.0429 *
MonthlyCharge     0.130922   0.231817   0.565   0.5722
OverageFee       -0.041998   0.395340  -0.106   0.9154
RoamMins          0.053539   0.026377   2.030   0.0424 *
---
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: 1492.2  on 2323  degrees of freedom
AIC: 1514.2

Number of Fisher Scoring iterations: 6
```

Checking for Variance Inflation Factor

```
vif(logitR.fit)
   AccountWeeks ContractRenewal        DataPlan       DataUsage    CustServ
Calls         DayMins
       1.003595        1.063610       14.023825     1561.023424         1.0
79516      939.760400
       DayCalls   MonthlyCharge      OverageFee        RoamMins
       1.010048     2742.930648      206.421768        1.176026
```

Dataplan, DataUsage,Daymin,Monthlycharge and overagefees will need to be cured through PCA before building a letter Logit Regression Model.

Chi Square Test to check the significant predictors with varying sig levels

```
anova(logitR.fit, test = "Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Churn

Terms added sequentially (first to last)


                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
NULL                             2333     1934.3              
AccountWeeks     1    0.780      2332     1933.5   0.37699    
ContractRenewal  1  130.640      2331     1802.9 < 2.2e-16 ***
DataPlan         1   39.983      2330     1762.9 2.562e-10 ***
DataUsage        1    1.420      2329     1761.5   0.23339    
CustServCalls    1   88.656      2328     1672.8 < 2.2e-16 ***
DayMins          1  129.359      2327     1543.4 < 2.2e-16 ***
DayCalls         1    4.173      2326     1539.3   0.04107 *  
MonthlyCharge    1   42.920      2325     1496.3 5.703e-11 ***
OverageFee       1    0.007      2324     1496.3   0.93479    
RoamMins         1    4.168      2323     1492.2   0.04121 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

ANOVA test on Predictors suggest we can leave out Overage fees, Data usage and Account Weeks from the list and proceed to build a model without these predictor variables for Logit regression.

Interpretation of Logit Regression

```
logitR.pred = predict(logitR.fit, newdata = test.cell, type = "response")
> logitR.predicted = ifelse(logitR.pred > 0.5 , 1, 0)
> logitR.predF = factor(logitR.predicted, levels = c(0,1))
> mean(logitR.predF == test.cell$Churn)
[1] 0.8478478
```

Confusion Matrix for LogitR Model

```
logitR.CM=confusionMatrix(logitR.predF,test.cell$Churn,positive = "1")
> ##Confusion Matrix for LogitR model##
> logitR.CM
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 821 118
         1  34  26

               Accuracy : 0.8478
                 95% CI : (0.8241, 0.8696)
    No Information Rate : 0.8559
    P-Value [Acc > NIR] : 0.7794

                  Kappa : 0.1859

 Mcnemar's Test P-Value : 1.671e-11

            Sensitivity : 0.18056
            Specificity : 0.96023
         Pos Pred Value : 0.43333
         Neg Pred Value : 0.87433
             Prevalence : 0.14414
         Detection Rate : 0.02603
   Detection Prevalence : 0.06006
      Balanced Accuracy : 0.57039

       'Positive' Class : 1
```

- Logistic Regression also performs poorly in case of general model with positive pred rate of 43% and Sensitivity of just 18%

- Ofcourse this model can be improved through better selection of predictors and their interaction effects but the general case is worst performer

- This LR model also suffers from accuracy paradox such that if threshold probability is decreses from 0.5 to say 0.2 or 0.1 then more cases will fall in Churner category (1)

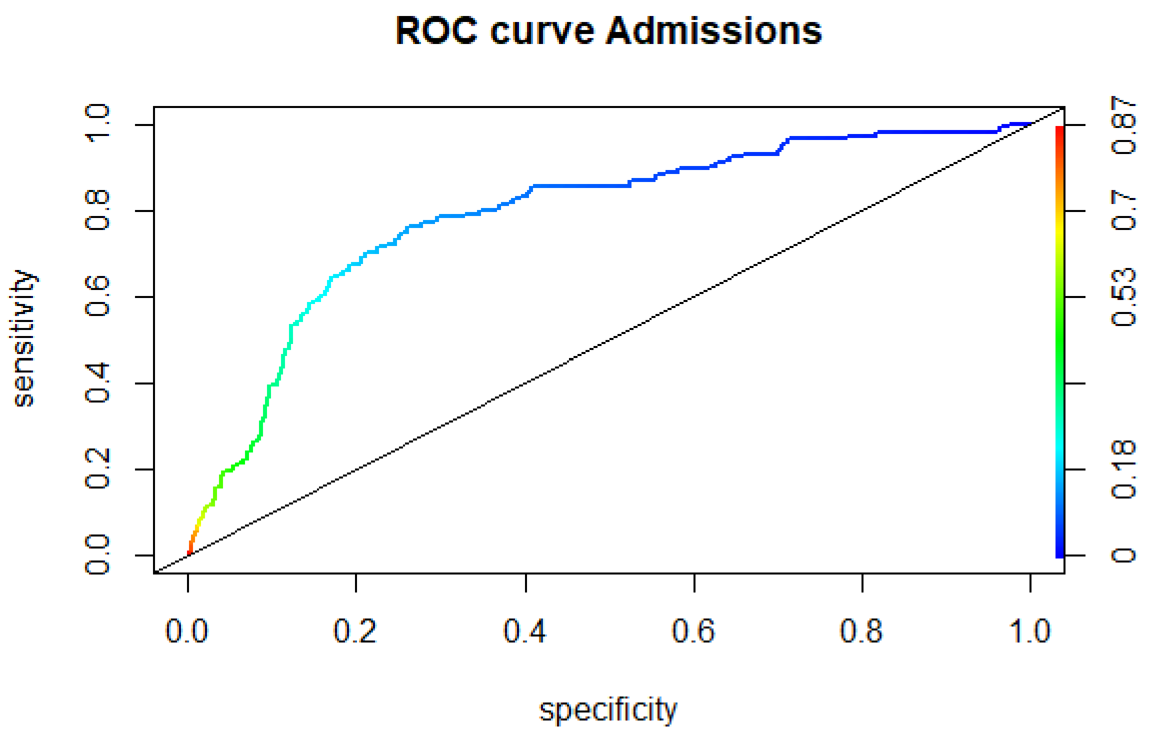
ROC Curve for LR Model

AUC or Area under the curve is 78% ie dataset has 78.6% concordant pairs

```
ROCRpred = prediction(logitR.pred, test.cell$Churn)
> AUC=as.numeric(performance(ROCRpred, "auc")@y.values)
> ## Area under the curve for LR model
> AUC
[1] 0.7868259
```

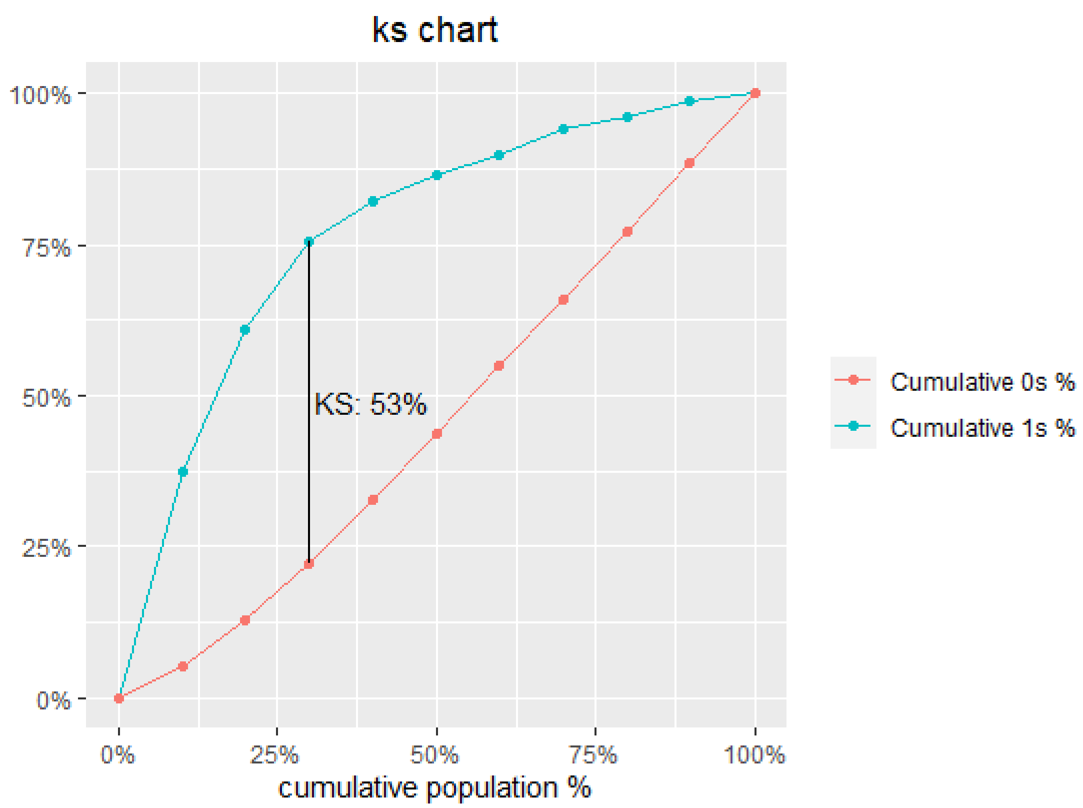
ROC Curve for the Model

```
perf=performance(ROCRpred,"tpr","fpr")
> plot(perf,col="black",lty=2,lwd=2,colorize=T,main="ROC curve Admissions"
,xlab="specificity",
+       ylab="sensitivity")
> abline(0,1)
```



KS Curve for the Model

```
ks=blr_gains_table(logitR.fit)
> blr_ks_chart(ks,title="ks chart",
+              yaxis_title = "",xaxis_title = "cumulative population %",
+              ks_line_color = "black")
```



Conclusion

Model Name	Positive Pred %	Accuracy %
k-NN	**81**	90
Naive Bayes	55	86
Logit Regression	43	84

k-NN performs the best with Positive pred rate of 81% in the general case model where the formula intends to take all the 10 predictors irrespective of their type whether continuous or categorical

The intended or any refined / tuned target model should be able to catch the Churners based on the data provided. Of course the dataset is lopsided in favour of more-No Churners rather than our intended target of finding Churners based on their behaviour hidden in the dataset.

Naive Bayes has no parameters to tune, but k-NN and Logit Regression can be improved by fine tuning the train control parameters and also deploying the up/down sampling approach for Logistic regression to counteract the class imbalance

THANK YOU