greatlearning

Submitted to

GREAT LAKES
INSTITUTE OF MANAGEMENT
Global Mindset - Indian Roots

Predictive Modeling

ASSIGNMENT SUBMITTED BY

MOHAMED YUSUF S

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 customer usage behavior, contract details, and payment details. The data also indicates the customers who canceled 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.

Exploratory Data Analysis

Checking for the header, structure and summary of data

```
> summary(cellPhone)
     Churn         AccountWeeks   ContractRenewal     DataPlan        DataUsage
 Min.   :0.0000   Min.   :  1.0   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.: 74.0   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000   Median :101.0   Median :1.0000   Median :0.0000   Median :0.0000
 Mean   :0.1449   Mean   :101.1   Mean   :0.9031   Mean   :0.2766   Mean   :0.8165
 3rd Qu.:0.0000   3rd Qu.:127.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.7800
 Max.   :1.0000   Max.   :243.0   Max.   :1.0000   Max.   :1.0000   Max.   :5.4000
 CustServCalls      DayMins         DayCalls      MonthlyCharge      OverageFee
 Min.   :0.000   Min.   :  0.0   Min.   :  0.0   Min.   : 14.00   Min.   : 0.00
 1st Qu.:1.000   1st Qu.:143.7   1st Qu.: 87.0   1st Qu.: 45.00   1st Qu.: 8.33
 Median :1.000   Median :179.4   Median :101.0   Median : 53.50   Median :10.07
 Mean   :1.563   Mean   :179.8   Mean   :100.4   Mean   : 56.31   Mean   :10.05
 3rd Qu.:2.000   3rd Qu.:216.4   3rd Qu.:114.0   3rd Qu.: 66.20   3rd Qu.:11.77
 Max.   :9.000   Max.   :350.8   Max.   :165.0   Max.   :111.30   Max.   :18.19
    RoamMins
 Min.   : 0.00
 1st Qu.: 8.50
 Median :10.30
 Mean   :10.24
 3rd Qu.:12.10
 Max.   :20.00
> View(cellPhone)
> names(cellPhone)
 [1] "Churn"           "AccountWeeks"    "ContractRenewal" "DataPlan"
 [5] "DataUsage"       "CustServCalls"   "DayMins"         "DayCalls"
 [9] "MonthlyCharge"   "OverageFee"      "RoamMins"
> 
```

```
> str(cellPhone)
'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 ...
> 
```

```
> sum(is.na(cellPhone))
[1] 0
>
```

Observations:

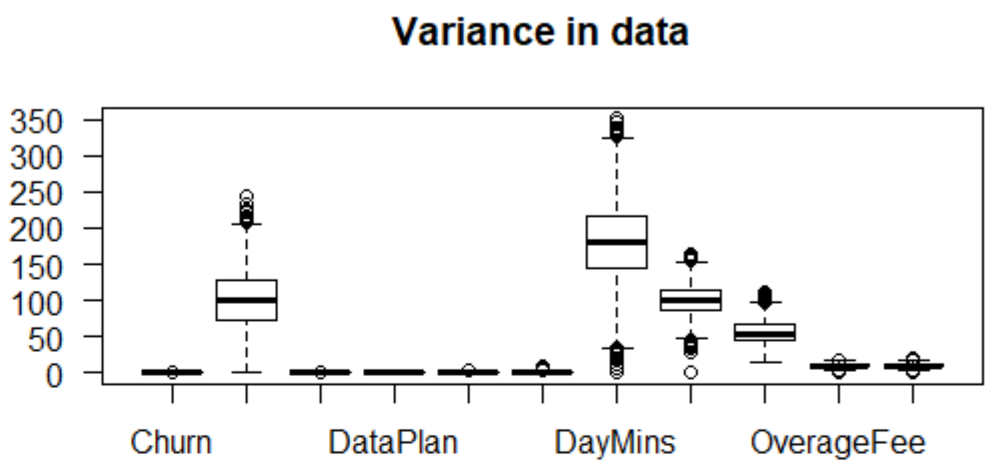
- There are 11 columns with 3333 observations for each of the columns. The data set has 10 independent variables which are the predictors and one dependent variable which is "churn".
- Summary gives the different quantiles of each column along with min and max values.
- There are no missing details in the dataset.
- Structure of data shows us all the data types of all the columns. Currently, some columns are considered numerical and some as integers. Since "Contract renewal" and "Data Plan" are currently considered as integers, we will execute the above commands once again after converting them to a factor. "Churn" though is a categorical variable, it is also the dependent variable. Hence we are not converting it as Logistic Regression limits the prediction using the sigmoid curve.

```
> str(cellPhone)
'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: 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 ...
>
```

We can see that 'Contract Renewal' and 'Data Plan' have been converted to a factor and it shows the number of 0's and 1's in the each of the columns. There is no change in any other column.

Boxplots

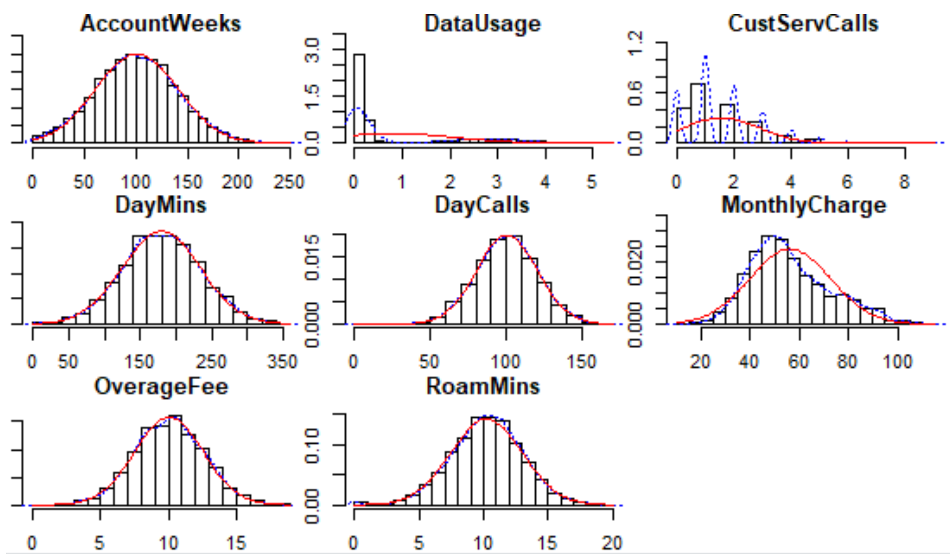
Let us now plot the columns to identify outliers using box plot



Observations: Looking at the box plots we conclude that 'CustServCalls' has the highest number of outliers, followed by 'RoamMins', 'MonthlyCharge', 'DayMins', 'OverageFee', 'DayCalls' and ''DataUsage'.

Histograms

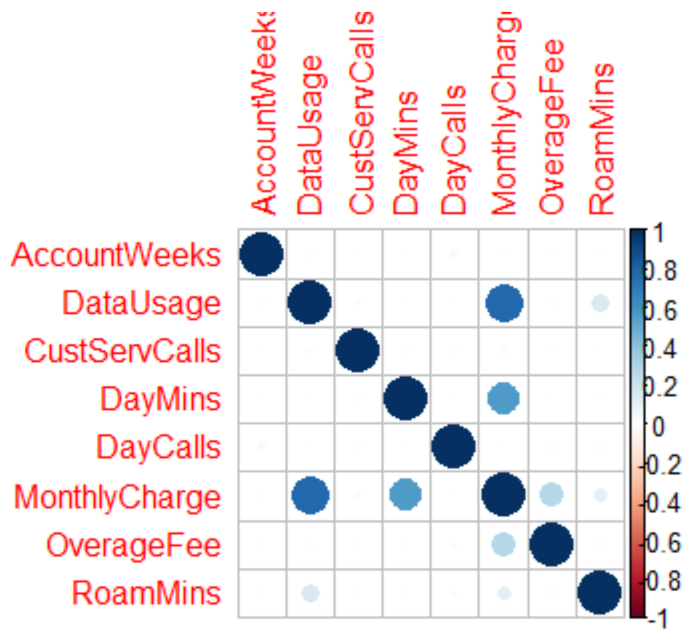
Plotting all the columns except 'Churn' (dependent variable), 'ContractRenewal' and 'DataPlan' (factors) in a single histogram.



Observations: From the plot we can see that other than data usage and 'CustServiceCalls', all the other columns follow a normal distribution curve.

Correlation

Viewing the correlation between variables using corrplot.




```
> corMatCellPhone
              AccountWeeks    DataUsage CustServCalls      DayMins     DayCalls
AccountWeeks   1.000000000  0.014390757  -0.003795939  0.006216021  0.038469882
DataUsage      0.014390757  1.000000000  -0.021722518  0.003175951 -0.007962079
CustServCalls -0.003795939 -0.021722518   1.000000000 -0.013423186 -0.018941930
DayMins        0.006216021  0.003175951  -0.013423186  1.000000000  0.006750414
DayCalls       0.038469882 -0.007962079  -0.018941930  0.006750414  1.000000000
MonthlyCharge  0.012580670  0.781660429  -0.028016853  0.567967924 -0.007963218
OverageFee    -0.006749462  0.019637372  -0.012964219  0.007038214 -0.021448602
RoamMins       0.009513902  0.162745576  -0.009639680 -0.010154586  0.021564794
              MonthlyCharge   OverageFee     RoamMins
AccountWeeks    0.012580670 -0.006749462  0.009513902
DataUsage       0.781660429  0.019637372  0.162745576
CustServCalls  -0.028016853 -0.012964219 -0.009639680
DayMins         0.567967924  0.007038214 -0.010154586
DayCalls       -0.007963218 -0.021448602  0.021564794
MonthlyCharge   1.000000000  0.281766048  0.117432607
OverageFee      0.281766048  1.000000000 -0.011023336
RoamMins        0.117432607 -0.011023336  1.000000000
>
```

Observations: From the correlation plot we see that the only column which shows significant correlation is “Monthly Charge”. It is correlated to DataUsage, DayMins and OverageFee in the same order.

Logistic Regression

The data has a categorical variable “Churn” with a binary response (0 for No & 1 for Yes). We can run logistic regression on this. So as a first step, we create dummy variables required for the regression so that we can look at the exact influence of all the variables including the categorical variable.

Split the data into train and test sets based on random split with 70:30 ratio.

```
> ##Logictic Regression
> cellPhoneIntermediate<-dummyVars("~ .",data = cellPhone,fullRank = T)
> cellPhoneForRegression<-data.frame(predict(cellPhoneIntermediate, newdata =  cellPhone))
> set.seed(seedValue)
> sample <- sample.split(cellPhoneForRegression,SplitRatio = 0.70)
> cellPhoneTrainDS <- as.data.frame(subset(cellPhoneForRegression,sample ==TRUE))
> cellPhoneTestDS <- as.data.frame(subset(cellPhoneForRegression,sample ==FALSE))
> 
```

Executing logistic regression for Train dataset.

```
> summary(logrmForAll)

Call:
glm(formula = Churn ~ ., family = binomial(), data = cellPhoneTrainDS)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.0270  -0.5113  -0.3461  -0.2097   2.9317  

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)    
(Intercept)       -5.445595   0.681342  -7.992 1.32e-15 ***
AccountWeeks       0.001432   0.001755   0.816   0.4145    
ContractRenewal.1 -2.087243   0.175320 -11.905  < 2e-16 ***
DataPlan.1        -1.543426   0.665447  -2.319   0.0204 *  
DataUsage          1.103027   2.418118   0.456   0.6483    
CustServCalls      0.544910   0.049580  10.991  < 2e-16 ***
DayMins            0.025906   0.040773   0.635   0.5252    
DayCalls          -0.001007   0.003409  -0.295   0.7677    
MonthlyCharge     -0.081323   0.239670  -0.339   0.7344    
OverageFee         0.288225   0.409452   0.704   0.4815    
RoamMins           0.070857   0.027719   2.556   0.0106 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1785.7  on 2120  degrees of freedom
Residual deviance: 1390.6  on 2110  degrees of freedom
AIC: 1412.6

Number of Fisher Scoring iterations: 5
```

Observations: Looking at the results of the model, statistically the variables ‘ContractRenewal’, ‘CustServCalls’, ‘RoamMins’, and ‘DataPlan’ seem to be significant. From the Null deviance 1785.7 and Residual deviance 1390.6 i.e. the error rate without any independent variables and the error rate with the independent variables, respectively we can see that the independent variables does decrease the error rate. The AIC for this model is 1412.6.

We will run the log likelihood test to understand the model better.

```
Likelihood ratio test

Model 1: Churn ~ AccountWeeks + ContractRenewal.1 + DataPlan.1 + DataUsage +
    CustServCalls + DayMins + DayCalls + MonthlyCharge + OverageFee +
    RoamMins
Model 2: Churn ~ 1
  #Df  LogLik  Df  Chisq Pr(>Chisq)
1  11 -695.29
2   1 -892.83 -10 395.09  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Observations: Based on the Chi-Sqaure value 395.09 of the Log Likelihood test above, the Churn rate depends on 'ContractRenewal', 'AccountWeeks', 'DataPlan', 'DataUsage', 'CustServCalls', 'DayMins', 'DayCalls', 'MonthlyCharge', 'OverageFee' & 'RoamMins'.

Further, we will check pseudo R-Square values.

```
> pR2(logrmForAll)
         llh      llhNull           G2     McFadden         r2ML         r2CU
-695.2852576 -892.8280204  395.0855255    0.2212551    0.1699532    0.2986319
>
```

Observations: The McFadden pseudo R2 value is ~22% indicating that 22% of the uncertainty of intercept is only explained by this model which is average.

We will now execute logistic regression based on the vif of the columns.

```
> vif(logrmForAll)
     AccountWeeks ContractRenewal.1        DataPlan.1         DataUsage     CustServCalls
         1.002921          1.058602         14.943986       1812.689688          1.080908
          DayMins          DayCalls     MonthlyCharge        OverageFee          RoamMins
       961.339933          1.001792       3063.000625        207.898696          1.213243
>
```

Observations: As the Variance Inflation Factor (VIF) from the initial "logrmForAll" model shows a high value for 'MonthlyCharge' and 'DataUsage' we are dropping these 2 and execute logistic regression.

Logistic Regression without 'MonthlyCharge' and 'DataUsage'

```
> summary(logrdrop2)

Call:
glm(formula = Churn ~ ., family = binomial(), data = cellPhoneTrainDS[,
    -c(5, 9)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0494  -0.5083  -0.3480  -0.2119   2.9302

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -5.595272   0.667436  -8.383  < 2e-16 ***
AccountWeeks       0.001463   0.001757   0.833  0.40493
ContractRenewal.1 -2.103562   0.174959 -12.023  < 2e-16 ***
DataPlan.1        -0.717344   0.172398  -4.161 3.17e-05 ***
CustServCalls      0.542321   0.049408  10.976  < 2e-16 ***
DayMins            0.012096   0.001339   9.032  < 2e-16 ***
DayCalls          -0.001072   0.003403  -0.315  0.75289
OverageFee         0.149346   0.028625   5.217 1.82e-07 ***
RoamMins           0.085758   0.025387   3.378  0.00073 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1785.7  on 2120  degrees of freedom
Residual deviance: 1392.3  on 2112  degrees of freedom
AIC: 1410.3

Number of Fisher Scoring iterations: 5


> lrtest(logrdrop2)
Likelihood ratio test

Model 1: Churn ~ AccountWeeks + ContractRenewal.1 + DataPlan.1 + CustServCalls +
    DayMins + DayCalls + OverageFee + RoamMins
Model 2: Churn ~ 1
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   9 -696.17
2   1 -892.83 -8 393.31  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> pR2(logrdrop2)
         llh      llhNull           G2     McFadden         r2ML         r2CU
-696.1743850 -892.8280204  393.3072708    0.2202593    0.1692570    0.2974085
> vif(logrdrop2)
     AccountWeeks ContractRenewal.1        DataPlan.1     CustServCalls           DayMins
         1.002525          1.055392          1.012183          1.076847          1.040994
         DayCalls        OverageFee          RoamMins
         1.001772          1.018473          1.010674
> |
```

Observations: After removing 2 variables ‘MonthlyCharge’ and ‘DataUsage’ we run the same tests and see that null deviance score is 1785.7 which is same as first mode, Residual deviance is 1392.3 against 1390.6 in the first model, AIC is 1410.3 against 1412.6, Chi Sq value is 393.31 against 395.09, McFadden’s error rate is 22% which is same as in the first model and VIF doesn’t show any variable with significant value. There are very minor changes in few parameters and no significant change from the first model.

Model measurement metrics

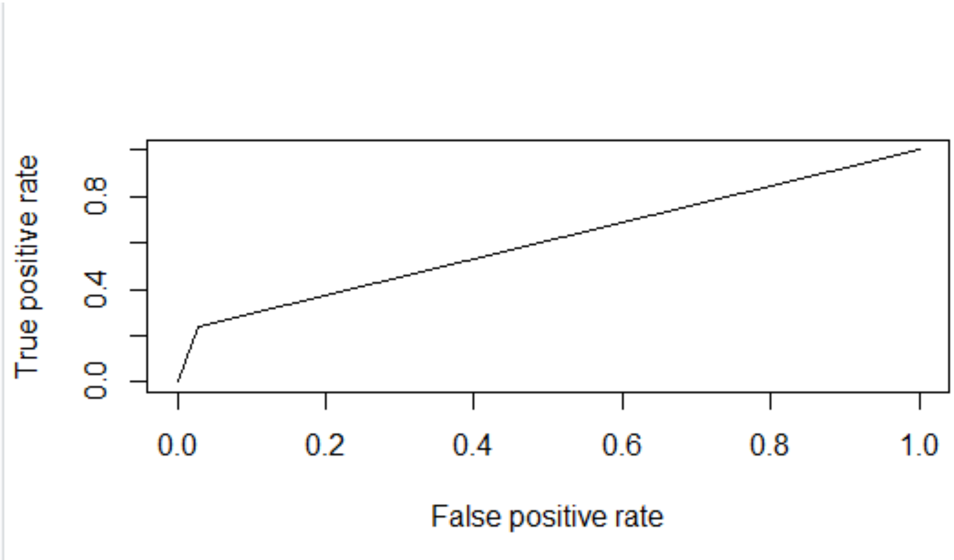
The confusion matrix for the initial model is shown below

```
> confusionMatrix(confmatpredLogRmForAll,mode="everything" )
Confusion Matrix and Statistics

         Actual
Predicted    0    1
        0 1754  240
        1   51   76

               Accuracy : 0.8628
                 95% CI : (0.8474, 0.8772)
    No Information Rate : 0.851
    P-Value [Acc > NIR] : 0.06639

                  Kappa : 0.2818

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

            Sensitivity : 0.9717
            Specificity : 0.2405
         Pos Pred Value : 0.8796
         Neg Pred Value : 0.5984
              Precision : 0.8796
                 Recall : 0.9717
                     F1 : 0.9234
             Prevalence : 0.8510
         Detection Rate : 0.8270
   Detection Prevalence : 0.9401
      Balanced Accuracy : 0.6061

       'Positive' Class : 0
```

Observation Metrics: The overall accuracy is 86.3%, with sensitivity (prediction of true positives) at 97.2% and Specificiity (prediction of true negatives) at 24%. The Positive prediction value is 87.9% and Negative prediction value is 59.8%. These show that the model is moderately good.

Accuracy	0.8628
Sensitivity	0.9717
Specificity	0.2405
F1 Score	0.9234

Area Under Curve (AUC), KS Statistic and GINI Score



Observation: Performance chart shows gradual increase after a certain point, which means the model is performing well.

```
> ksLogRmForAll
[1] 0.2122515
> aucLogRmForAll
[1] 0.6061257
> giniLogRmForAll
[1] 0.9401226
>
```

Observation Metrics:

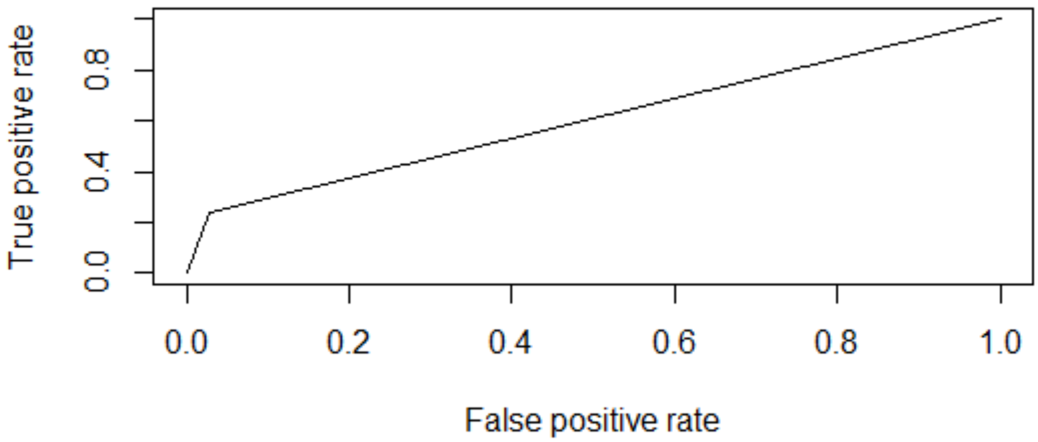
KS	0.212252
AUC	0.606126
GINI	0.940123

Alternate model with lesser AIC value (without ‘MonthlyCharge’ and ‘DataUsage’)

```
> confusionMatrix(confmatpredLogrdrop2,mode="everything" )
Confusion Matrix and Statistics

          Actual
Predicted    0    1
        0 1755  239
        1   50   77

               Accuracy : 0.8637
                 95% CI : (0.8484, 0.8781)
    No Information Rate : 0.851
    P-Value [Acc > NIR] : 0.05176

                  Kappa : 0.2867

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

            Sensitivity : 0.9723
            Specificity : 0.2437
         Pos Pred Value : 0.8801
         Neg Pred Value : 0.6063
              Precision : 0.8801
                 Recall : 0.9723
                     F1 : 0.9239
             Prevalence : 0.8510
         Detection Rate : 0.8274
   Detection Prevalence : 0.9401
      Balanced Accuracy : 0.6080

       'Positive' Class : 0

>
```

Observation Metrics: The overall accuracy is 86.3%, with sensitivity (prediction of true positives) at 97.2% and Specificiity (prediction of true negatives) at 24%. The Positive prediction value is 88% and Negative prediction value is 60.6%. These show that the model is moderately good and a little better than above.

Accuracy	0.8637
Sensitivity	0.9723
Specificity	0.2437
F1 Score	0.9239



Observation: Performance chart shows gradual increase after a certain point, which means the model is performing well.

```
> ksLogrdrop2
[1] 0.2159701
> aucLogrdrop2
[1] 0.607985
> giniLogrdrop2
[1] 0.9401226
> 
```

Observation Metrics:

KS	0.215970
AUC	0.607985
GINI	0.940123

Both the models above show more or less similar results as observed in the metrics below.

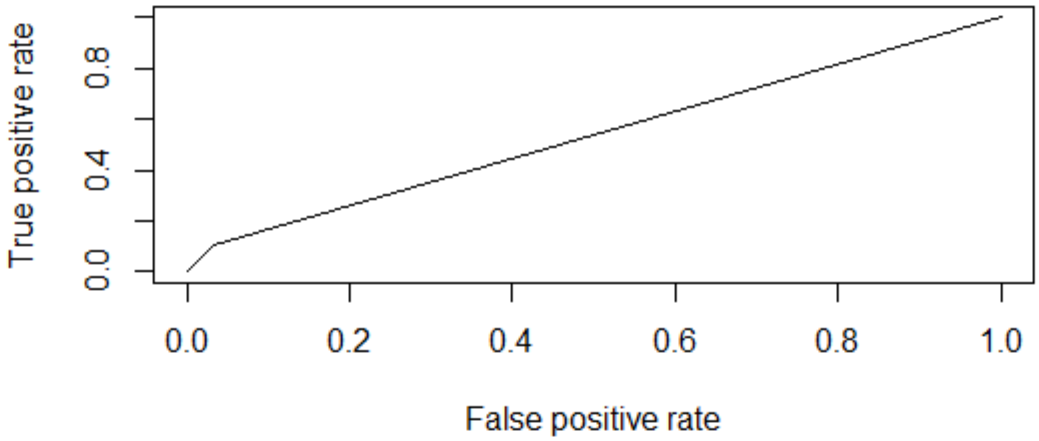
Metric	Model 1	Model 2
Accuracy	0.8628	0.8637
Sensitivity	0.9717	0.9723
Specificity	0.2405	0.2437
F1 Score	0.9234	0.9239
KS	0.2123	0.2160
AUC	0.6061	0.6080
GINI	0.9401	0.9401

Executing Initial model for Test dataset

```
> confusionMatrix( confmatpredLogRmForAll,mode="everything" )
Confusion Matrix and Statistics

         Actual
Predicted    0    1
        0 1011  149
        1   34   18

               Accuracy : 0.849
                 95% CI : (0.8276, 0.8687)
    No Information Rate : 0.8622
    P-Value [Acc > NIR] : 0.9141

                  Kappa : 0.1059

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

            Sensitivity : 0.9675
            Specificity : 0.1078
         Pos Pred Value : 0.8716
         Neg Pred Value : 0.3462
              Precision : 0.8716
                 Recall : 0.9675
                     F1 : 0.9170
             Prevalence : 0.8622
         Detection Rate : 0.8342
   Detection Prevalence : 0.9571
      Balanced Accuracy : 0.5376

       'Positive' Class : 0
```

Observation Metrics: The overall accuracy is 84.9%, with sensitivity (prediction of true positives) at 96.7% and Specificiity (prediction of true negatives) at 10%. The Positive prediction value is 87.1% and Negative prediction value is 34.6%. These show that the model is moderately good.

Accuracy	0.8490
Sensitivity	0.9675
Specificity	0.1078
F1 Score	0.9170



Observation: Performance chart shows gradual increase after a certain point, which means the model is performing well.

```
> ksLogRmForAll
[1] 0.07524855
> aucLogRmForAll
[1] 0.5376243
> giniLogRmForAll
[1] 0.9570957
>
```

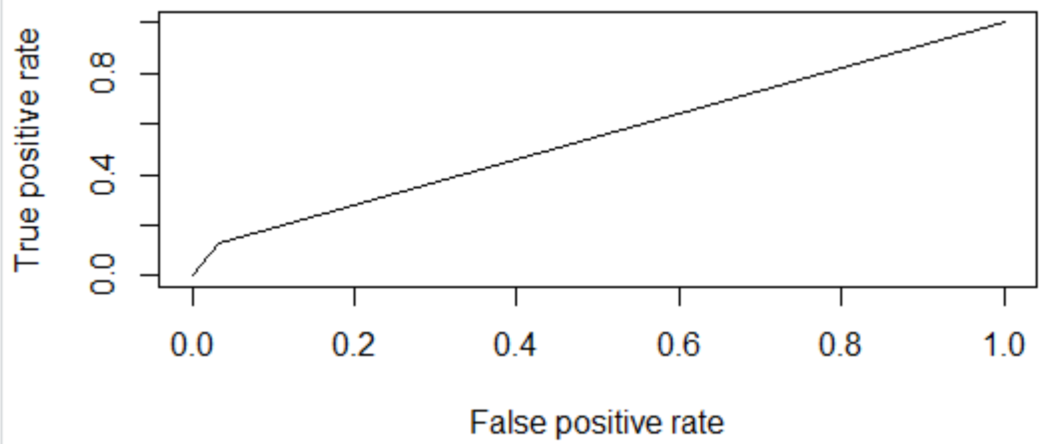
KS	0.0752
AUC	0.5376
GINI	0.9571

Alternate model with lesser AIC value (without 'MonthlyCharge' and 'DataUsage')

```
> confusionMatrix(confmatpredLogrdrop2,mode="everything" )
Confusion Matrix and Statistics

         Actual
Predicted    0    1
        0 1011  145
        1   34   22

               Accuracy : 0.8523
                 95% CI : (0.8311, 0.8718)
    No Information Rate : 0.8622
    P-Value [Acc > NIR] : 0.8511

                  Kappa : 0.1376

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

            Sensitivity : 0.9675
            Specificity : 0.1317
         Pos Pred Value : 0.8746
         Neg Pred Value : 0.3929
              Precision : 0.8746
                 Recall : 0.9675
                     F1 : 0.9187
             Prevalence : 0.8622
         Detection Rate : 0.8342
   Detection Prevalence : 0.9538
      Balanced Accuracy : 0.5496

       'Positive' Class : 0
```

Observation Metrics: The overall accuracy is 85.2%, with sensitivity (prediction of true positives) at 96.7% and Specificiity (prediction of true negatives) at 13%. The Positive prediction value is 87.4% and Negative prediction value is 39.29%. These show that the model is moderately good.

Accuracy	0.8523
Sensitivity	0.9675
Specificity	0.1317
F1 Score	0.9187



Observation: Performance chart shows gradual increase after a certain point, which means the model is performing well.

```
> ksLogrdrop2
[1] 0.09920064
> aucLogrdrop2
[1] 0.5496003
> giniLogrdrop2
[1] 0.9537954
```

Observation Metrics:

KS	0.0992
AUC	0.5496
GINI	0.9538

Results from the second model are better than the initial model with minimal differences

Metric	Model 1	Model 2
Accuracy	0.8490	0.8523
Sensitivity	0.9675	0.9675
Specificity	0.1078	0.1317
F1 Score	0.9170	0.9187
KS	0.0752	0.0992
AUC	0.5376	0.5496
GINI	0.9571	0.9538

Conclusion

From the different models we can see that the second model has better metrics for both test and train and we can use that model. However using the odds function on the initial full model,, we can see that “DataUsage” and “Customer Service Calls” have a high influence on the churn rate. Customers could have been overcharged with Data Usage or might be needing more details on why DataUsage is high and probably its associated charges are concerning. The company needs to concentrate on the users who call Customer service frequently as well. A combination of these two could help reduce the churn.

```
> oddModel<-exp(coef(logrmForAll))
> print(oddModel)
  (Intercept)  AccountWeeks ContractRenewal.1    DataPlan.1     DataUsage
  0.004315271   1.001433239       0.124028588   0.213647888   3.013274050
CustServCalls       DayMins          DayCalls MonthlyCharge    OverageFee
  1.724452699   1.026244176       0.998993628   0.921895829   1.334057315
     RoamMins
  1.073427900
> 
```