This document forms the template for predictive analytics problem. Solution is divided into multiple phases – business understanding, data understanding, transformations of data, model building, model refining, and communicating results to stakeholders and/or decision makers.

Sample Solution for Predictive Modeling Project

Churn prediction for Telecom

Table of Contents

1. Discovery Phase

1.1. Domain Understanding

Customer attrition is an important issue for any industry. It is especially important in mature industries where the initial period of exponential growth has been left behind. Not surprisingly, attrition (or, to look on the bright side, retention) is a major application of data mining.

One of the first challenges in modeling attrition is deciding what it is and recognizing when it has occurred. This is harder in some industries than in others. At one extreme are businesses that deal in anonymous cash transactions.

When a once-loyal customer deserts his regular coffee bar for another down the block, the barista who knew the customer's order by heart may notice, but the fact will not be recorded in any corporate database. Even in cases where the customer is identified by name, telling the difference between a customer who has churned and one who just hasn't been around for a while may be hard. If a loyal Ford customer who buys a new F150 pickup every five years hasn't bought one for six years, has the customer defected to another brand?

Attrition is a bit easier to spot when a monthly billing relationship exists, as with credit cards. Even there, attrition might be silent. A customer may stop using the credit card, but not cancel it. Attrition is easiest to define in subscription-based businesses, and partly for that reason, attrition modeling is most popular in these businesses. Long-distance companies, mobile phone service providers, insurance companies, cable companies, financial services companies, Internet service providers, newspapers, magazines, and some retailers all share a subscription model where customers have a formal, contractual relationship that must be explicitly ended.

Lost customers must be replaced by new customers, and new customers are expensive to acquire. Often, new customers generate less revenue in the near term than established customers. This is especially true in mature industries where the market is fairly saturated — anyone likely to want the product or service probably already has it from somewhere, so the main source of new customers is people leaving a competitor.

Hence it is vital for such industries to understand the attrition.

This assignment focuses on one of such problems the Telecom Industry faces – Customer Churn.

1.2. Problem in hand

Students are given a Cell Phone Data file and are requested to build a Classification Model which can tell the parameters contributing (and not contributing) for Customer Churn (attrition), along with the intensity of each attribute.

The input file needs to divide into Training Dataset, which should contain 70% of the data and Testing Dataset, which would contain remaining 30% of the data.

The cell phone data file contains following attributes:

1.2.1. Data Attributes

Sr. No.	Variable	Description	Type
1	Churn **(Target Variable)**	1 if customer cancelled service, 0 if not	Categorical
2	AccountWeeks	number of weeks customer has had active account	Continuous
3	ContractRenewal	1 if customer recently renewed contract, 0 if not	Categorical
4	DataPlan	1 if customer has data plan, 0 if not	Categorical
5	DataUsage	gigabytes of monthly data usage	Continuous
6	CustServCalls	number of calls into customer service	Continuous
7	DayMins	average daytime minutes per month	Continuous
8	DayCalls	average number of daytime calls	Continuous
9	MonthlyCharge	average monthly bill	Continuous
10	OverageFee	largest overage fee in last 12 months	Continuous
11	RoamMins	average number of roaming minutes	Continuous

1.2.2. Hypothesis formation

The Cellphone File contains one Dependent and 10 Predictor variables.

The assignment aim is to identify the predictor variables which are significant for Customer Churn.

Null Hypothesis (Ho) – No predictor is able to predict the Churn

Alternate Hypothesis (Ha) – At least one of the predictors is able to predict the churn.

2. Data Understanding and transformation

2.1. Data Overview

- Check for Variables names, Five Point Summary, and their data type. This is the first opportunity to explore our data.

There are 11 columns in the data with the names Churn, AccountWeeks, ContractRenewal, DataPlan, DataUsage, CustServCalls, DayMins, DayCalls, MonthlyCharge, OverageFee and RoamMins

There are 3333 records in the data and the data also contains a header giving us the required information about the data

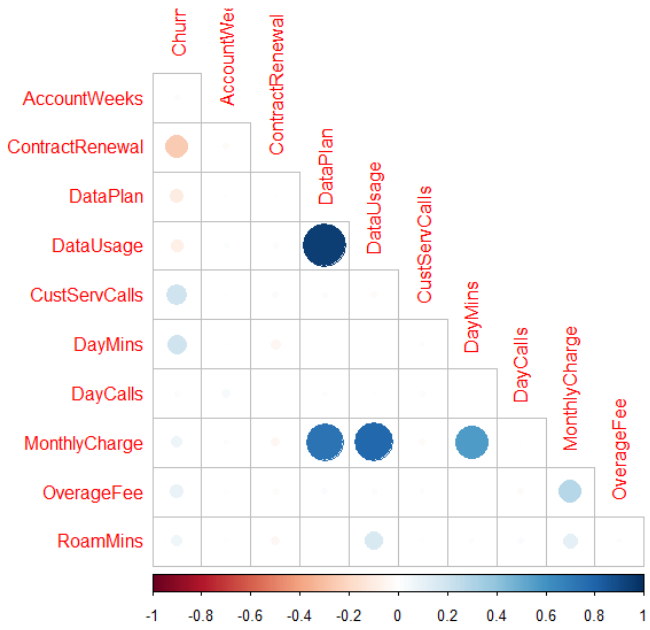
Since the data is given for the cellphone company that is seemingly interested in knowing if the user or the customer will churn or not that is will they cancel the service or retain it, also to check if the customer will renew the contract with the company giving them the assurance that customer is interested in staying with the company

It will be interesting to see how the users with different column values will be responding in the future to the company. Also, how the number of weeks that the user have been active on the network and the data plan as well as data usage, daily calls, monthly charge etc affects the churning or the contract renewal part.

In the given dataset, Churn, ContractRenewal and DataPlan are the variables that are dichotomous in nature hence only have values of either 0 or 1, depending upon the data.

The other given variables are all continuous where some are integer and others are numeric.

- Check for missing values & handle them (if required). Here we don't have any null values.
- Overview of unique levels of each column
- Identify the balance in the target variable. 14.5% of customers have churned.
- Find correlation among all the variables



From above plot, we can see that Contract Renewal is negatively correlated to the churn value, which is intuitive. Other variables that are negatively correlated are DataPlan and DataUsage that are again expected to show the such trend as higher these values, higher will be the chance to retain the customer.

Customer service calls, daily minutes and roaming minutes seem to have a positive correlation according to the data of churn. Here, more customer service calls make sense as the user is in more need of servicing regularly and hence can churn out of the company. Whereas, daily minutes cannot be that much effective as if a user is having more daily minutes, it seems to be having a good time with the service of the company.

Also, data plan is highly correlated with the data usage which makes complete sense, alongside that monthly charge is directly correlated with data usage and data plan. Here, we can have an issue regarding why the monthly charge is not affected by the Roaming Minutes and calling minutes. As can be seen, it is directly correlated to the daily minutes and slightly correlated to the roaming minutes.

- Convert binary variables into factors

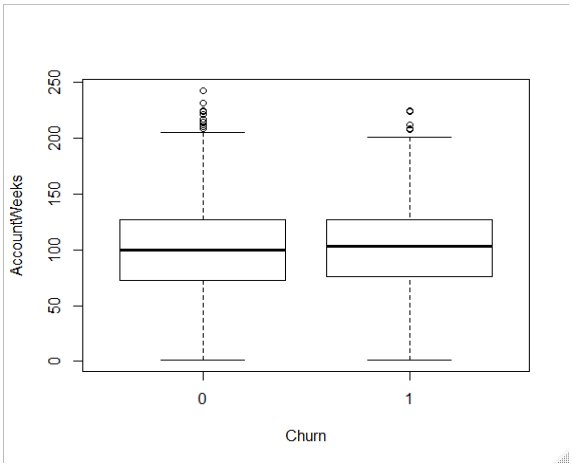
Here we have 3 variables, which need to be converted to factors – Churn, DataPlan, ContractRenewal.

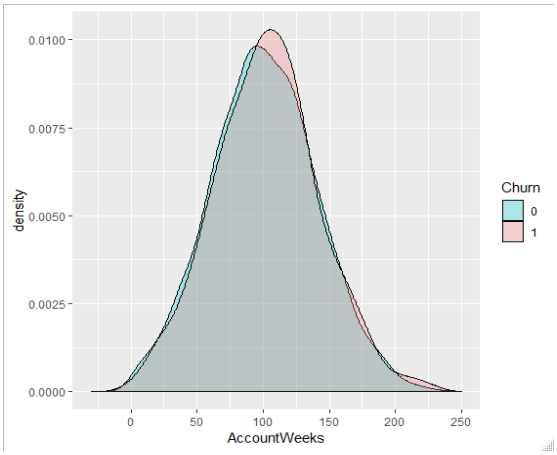
2.2.Exploratory data analysis

2.2.1. Continuous variables

- AccountWeeks –

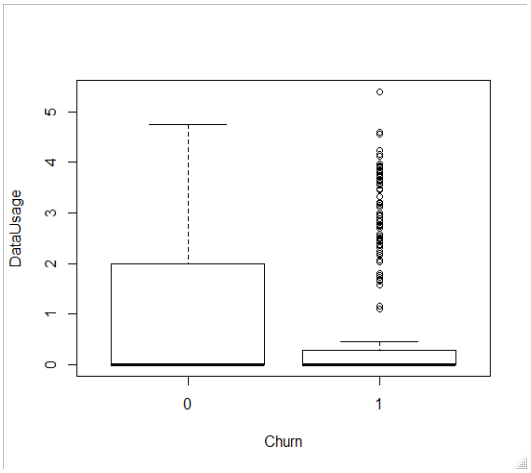
It is varying from 1.0 to 243.0 with a mean of 101.1, median of 101.0, range of 242.00 and a standard deviation of 6.94. Now, since account weeks are varying over a large range and have a greater standard deviation, we are going to consider it as a continuous variable in the whole upcoming analysis.

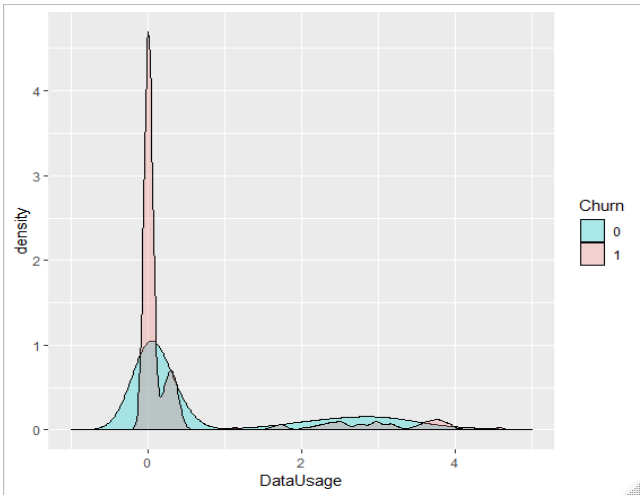




- DataUsage –

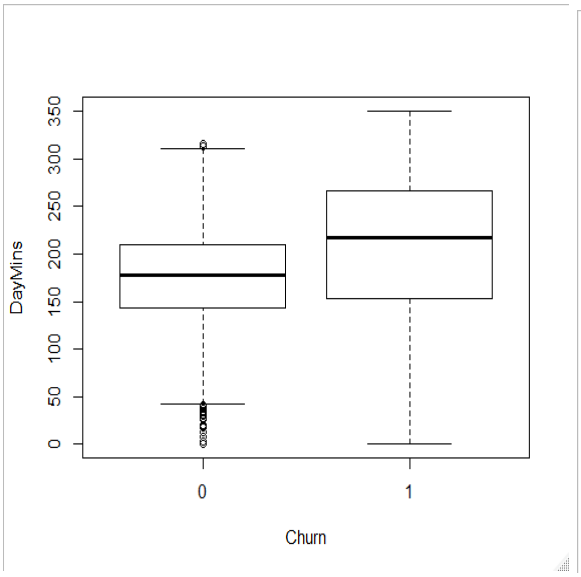
It is significantly different for 2 classes of customers – who churn and who don't. It is varying from 0 to 5.40 with a mean of 0.8, median of 00, range of 5.40 and a standard deviation of 1.6. Here, since the data usage is varying over a range of 9.00, we can make an inference for data usage to be a continuous variable, considering that these are not levels of some sort.

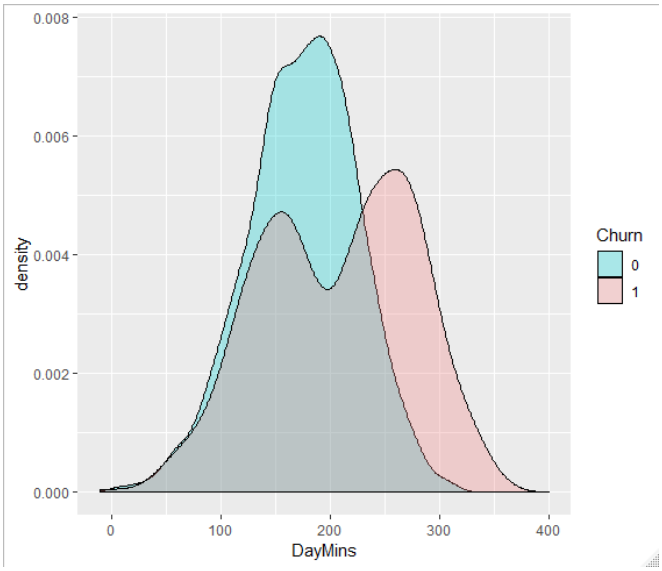




- DayMins –

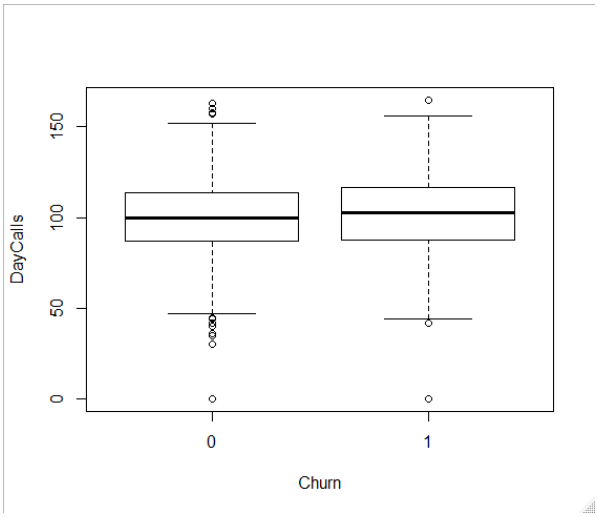
It is varying from 0.00 to 350.80 with a mean of 179.8, median of 178.4, range of 4.00 and a standard deviation of 0.958. Here, for the case of cellphone data, we can clearly see that its range is quite big. Also, we can make the inference from seeing the data that it is the minutes of the day that are spent by the user on an average using the cell phone, let's say particularly the minutes spent on the calls.

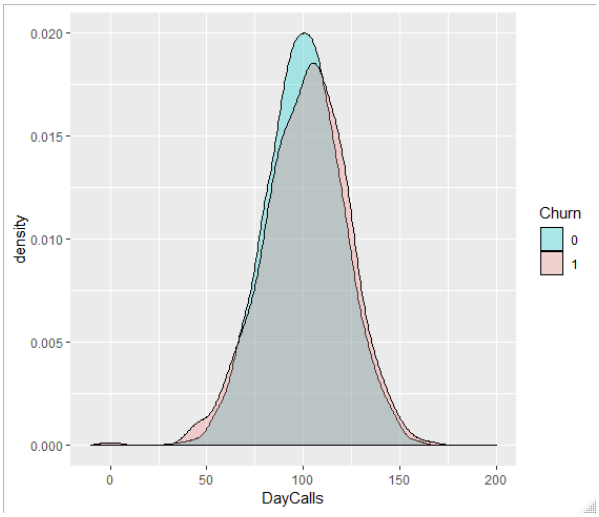




- DayCalls –

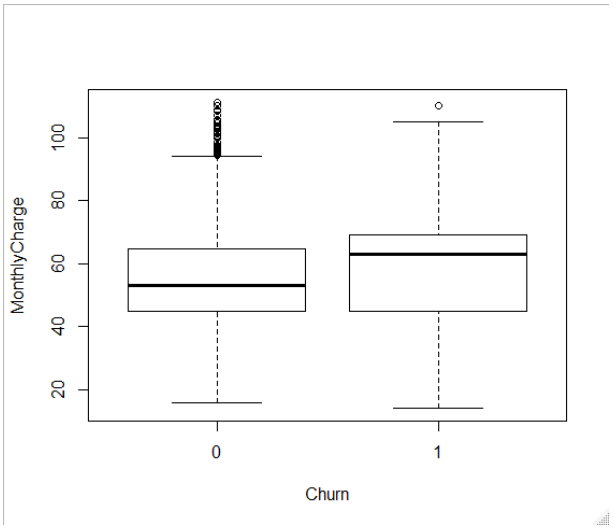
It is varying from 1.00 to 5.00 with a mean of 3.31, median of 3.00, range of 4.00 and a standard deviation of 0.958. Here, for the case of fitness, we can clearly see that its range is quite small and has a very low standard deviation.

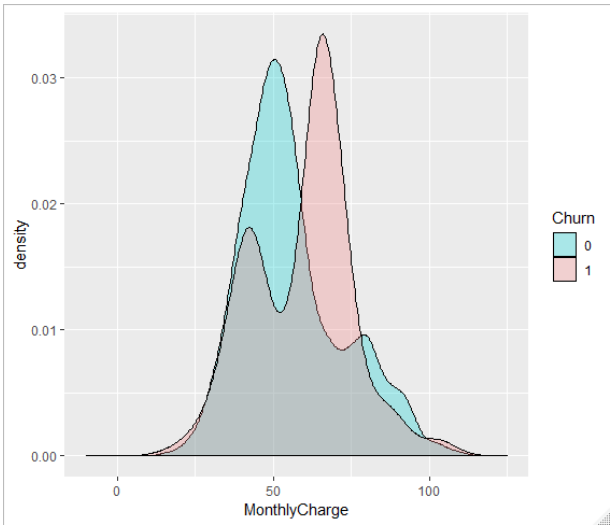




- MonthlyCharge –

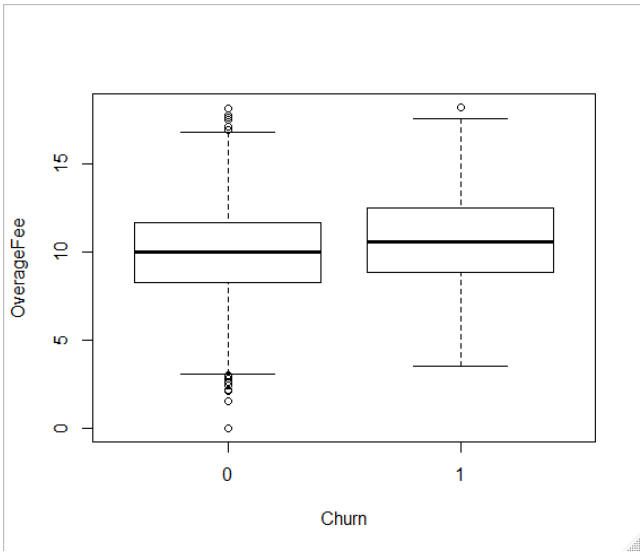
It is varying from 0 to 165 with a mean of 100.2, median of 101.2, range of 165. Here, the monthly charge is the value of money or the bill that is paid by the user in the given contract monthly.

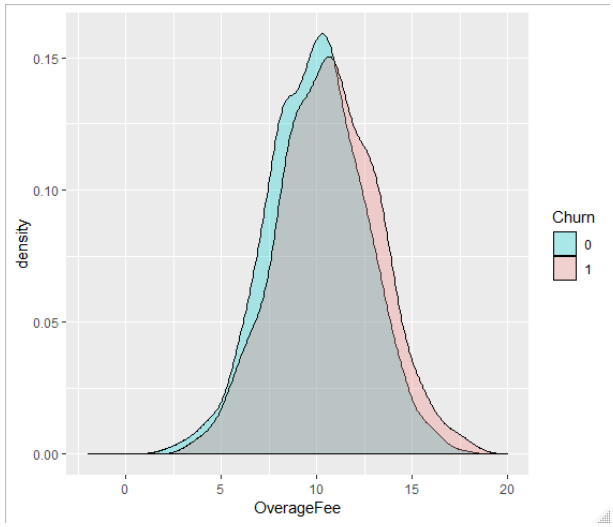




- OverageFees –

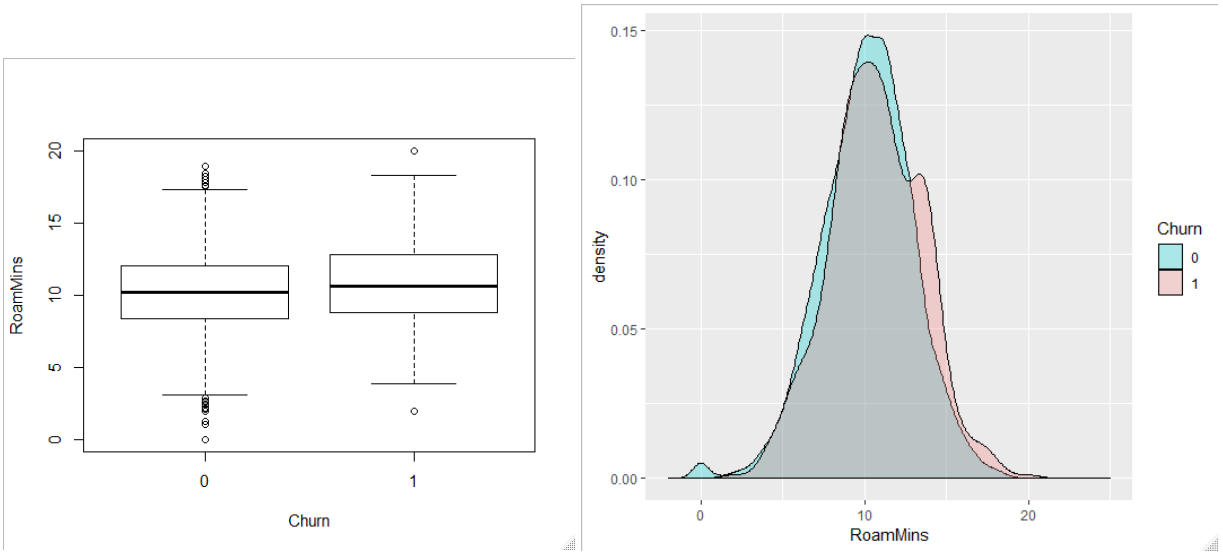
It is varying from 0.0 to 18.19 with a mean of 10.07, median of 10.10, range of 18.19. Here, the overage fee is the fees paid by the user in order to penalty the late paid bills.





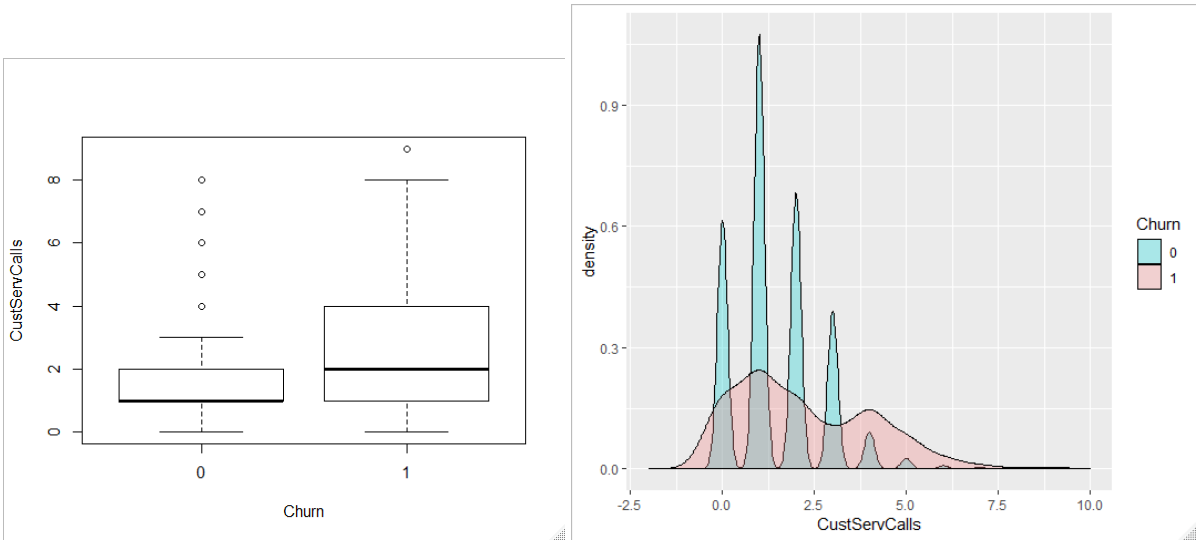
- RoamMins –

It is varying from 1.00 to 20.00 with a mean of 10.30, median of 10.24, range of 19.00. Here, for the case of cellphone data, we can clearly see that its range is quite small and has a very low standard deviation. Also, we can make the inference from seeing the data that such a small range for fitness and all values being integers, means that fitness is actually a factor minutes spent in roaming calls.



- CustServCalls –

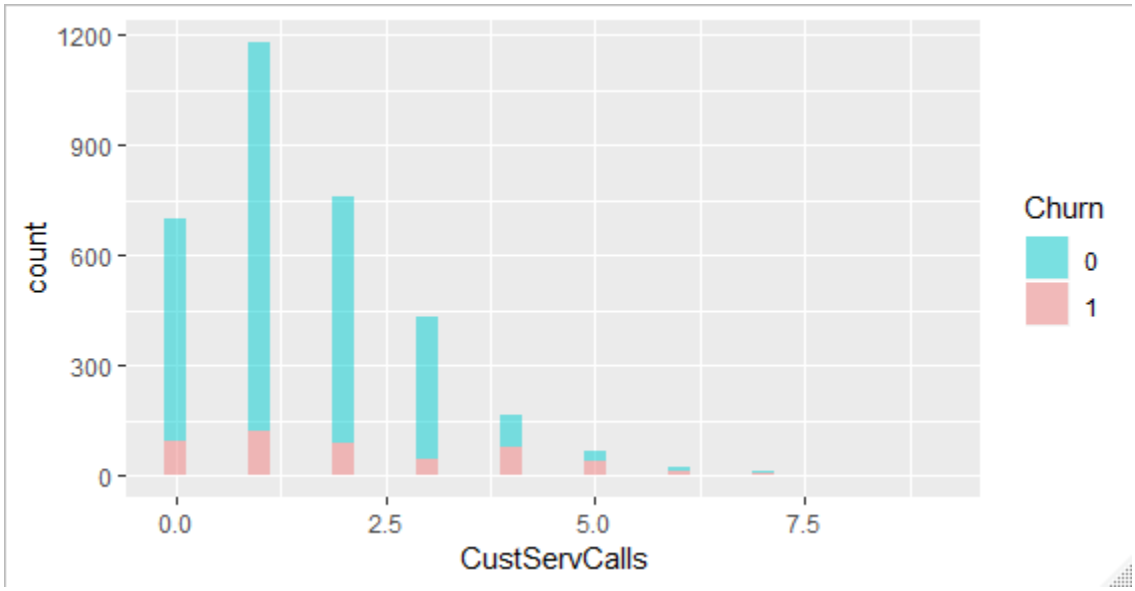
It is varying from 0.00 to 9.00 with a mean of 1.50, median of 1.00, range of 9.00 and a standard deviation of 1.08. For the case of customer service calls, we can tell that it is the number of times the user has called the customer service for any sort of help or recommendations.



2.2.2. Categorical variables

- CustServCalls –

Above graph shows, though this variable is considered as numeric feature, it has specific levels. So, let's explore its influence on churn using proportion table too.




```
> prop.table(table(CustServCalls,Churn),1)*100
             Churn
CustServCalls         0         1
            0  86.80057  13.19943
            1  89.66977  10.33023
            2  88.53755  11.46245
            3  89.74359  10.25641
            4  54.21687  45.78313
            5  39.39394  60.60606
            6  36.36364  63.63636
            7  44.44444  55.55556
            8  50.00000  50.00000
            9   0.00000 100.00000
```

- ContractRenewal –

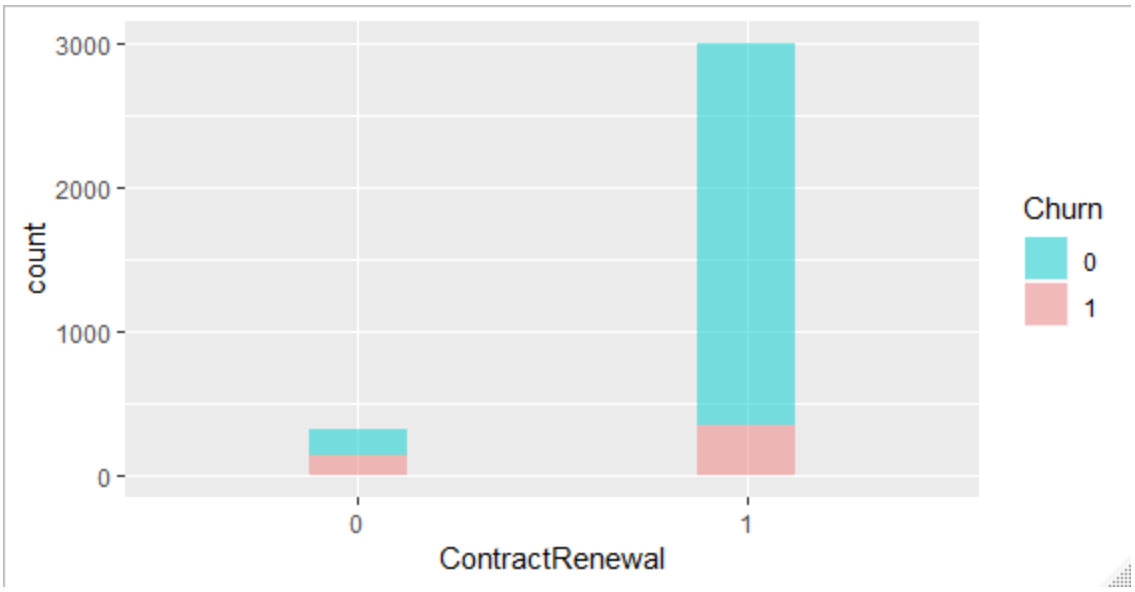
It is a dichotomous variable that specifies if the user or the customer has actually renewed the contract or not. Frequency in the dataset as stated are shown as follows:

0: 323

1: 3010

Clearly, the contract renewal is totally opposite to the churn as the churn value of 0 shows that the user not cancelled the service whereas the contract renewal of value 0 shows that user has not renewed the contract.

So, the first inference that we can make here is that both variables are negatively correlated.




```
> prop.table(table(ContractRenewal,Churn),1)*100
                Churn
ContractRenewal        0        1
              0 57.58514 42.41486
              1 88.50498 11.49502
```

- DataPlan –

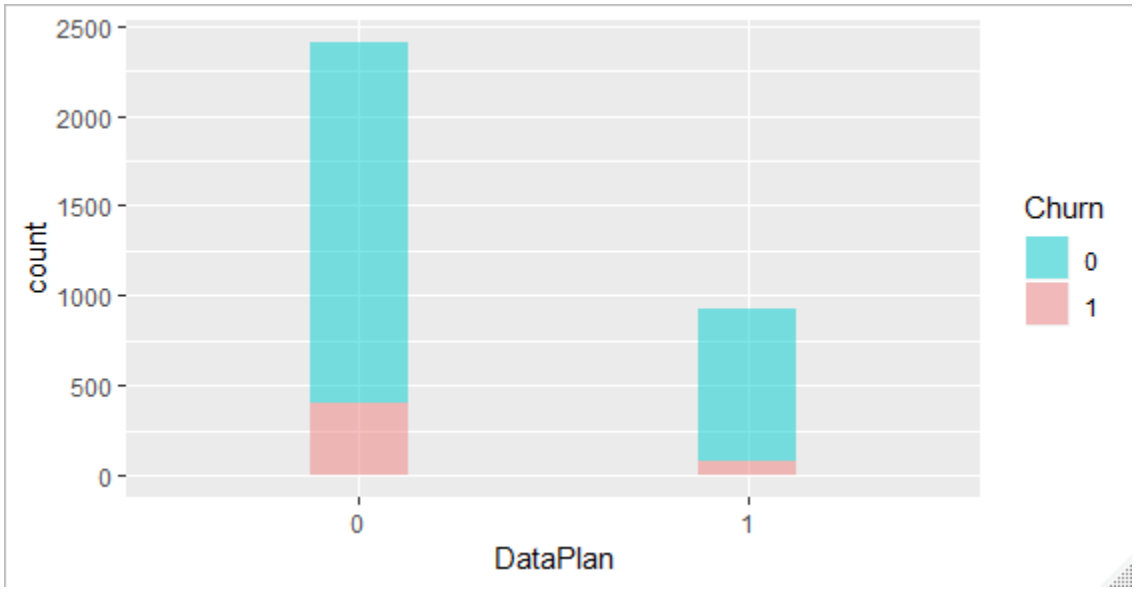
The data plan is the value of yes or no if the user has chosen the data plan or not. Frequency of 0's and 1's is as shown:

0 : 2411

1 : 922

Proportion table shows there isn't significant difference, implying this variable may not be very influencing in predicting churn.

```
> prop.table(table(DataPlan,Churn),1)*100
         Churn
DataPlan        0        1
       0 83.28494 16.71506
       1 91.32321  8.67679
```



3. Model building and tuning

3.1. Data Slicing

For classification problem, it is important to ensure that the train and test sets have approximately the same percentage of samples of each target class. Hence, we will use stratified sampling.

```
> prop.table(table(data$Churn))
        0         1
0.8550855 0.1449145

> prop.table(table(train.data$Churn))
        0         1
0.8547558 0.1452442

> prop.table(table(test.data$Churn))
        0         1
0.8558559 0.1441441
```

3.2. Logistic Regression

Let's build the initial Logistic Regression Model taking all independent variables into consideration.

3.2.1. Model Summary

```
> summary(logit_model1)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8613  -0.5257  -0.3650  -0.2255   2.9272

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.609070   0.642656  -8.728  < 2e-16 ***
AccountWeeks      0.001389   0.001654   0.840  0.40107
ContractRenewal1 -1.791007   0.169482 -10.568  < 2e-16 ***
DataPlan1        -0.734115   0.603669  -1.216  0.22395
DataUsage         0.042436   2.245106   0.019  0.98492
CustServCalls     0.456716   0.045536  10.030  < 2e-16 ***
DayMins           0.014172   0.037945   0.373  0.70879
DayCalls          0.001768   0.003212   0.550  0.58213
MonthlyCharge    -0.009112   0.222912  -0.041  0.96739
OverageFee        0.141130   0.380157   0.371  0.71046
RoamMins          0.066626   0.025510   2.612  0.00901 **
---
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: 1574.3  on 2323  degrees of freedom
AIC: 1596.3

Number of Fisher Scoring iterations: 5
```

3.2.2. Interpretation

The three significant variables highlighted in yellow are:

- Contract Renewal: Please note, this has a negative impact on Customer Churn.
- Customer Service Calls
- Roaming Minutes

Also, the AIC[#] Score is 1533.7. This will be observed in subsequent stages when we refine the model. The model having least AIC Score would be the most preferred and optimized one.

[#] Note on AIC score: The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters.

3.2.3. Evaluating model performance

1. Model significance test

Model significance is checked using log likelihood test[#].

[#] Note on Log Likelihood Test: In statistics, a likelihood ratio test is a statistical test used for comparing the goodness of fit of two models, one of which (the null model) is a special case of the other (the alternative model). The test is based on the likelihood ratio, which expresses how many times more likely the data are under one model than the other. This likelihood ratio, or equivalently its logarithm, can then be used to compute a p-value, or compared to a critical value to decide whether to reject the null model and hence accept the alternative model.

```
> lrtest(logit_model1)
Likelihood ratio test

Model 1: Churn ~ AccountWeeks + ContractRenewal + DataPlan + DataUsage
+ CustServCalls + DayMins + DayCalls + MonthlyCharge + OverageFee +
RoamMins

Model 2: Churn ~ 1
```

```
#Df LogLik Df Chisq Pr(>Chisq)
1 11 -755.87
2 1 -978.18 -10 444.61 < 2.2e-16 ***

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

H0: All betas are zero

H1: At least 1 beta is nonzero

From the log likelihood, we can see that, intercept only model -978.18 variance was unknown to us. When we take the full model, -755.87 variance was unknown to us.

So we can say that, 1-(-755.87 /-978.18)= 22.72% of the uncertainty inherent in the intercept only model is calibrated by the full model.

Chisq likelihood ratio is significant. Also the p value suggests that we can accept the Alternate Hypothesis that at least one of the beta is not zero.

So Model is significant.

2. Model robustness check

Now since we concluded that the model built is significant, let’s find out how robust it is with the help of McFadden pseudo-R Squared Test[#].

[#]Note on McFadden pseudo R-square: McFadden pseudo-R Squared: Logistic regression models are fitted using the method of maximum likelihood - i.e. the parameter estimates are those values which maximize the likelihood of the data which have been observed. McFadden's R squared measure is defined as (1-LogLc/LogLnull) where Lc denotes the (maximized) likelihood value from the current fitted model, and Lnull denotes the corresponding value but for the null model - the model with only an intercept and no covariates.

```
> pR2(logit_model1)
llh               llhNull            G2     McFadden     r2ML         r2CU
-755.8740291  -978.1767837    444.6055090 0.2272623    0.1739880    0.3059099
```

The McFadden’s pseudo-R Squared test suggests that at least 22.72% variance of the data is captured by our Model, which suggests it’s a robust model.

3. Odds explanatory power

```
> exp(coef(logit_model1)) # Odds Ratio
(Intercept)       AccountWeeks      ContractRenewal1  DataPlan1   DataUsage
0.001896793       1.001970708       0.126772710       0.289804648 7.671441061
CustServCalls     DayMins     DayCalls          MonthlyCharge     OverageFee
1.636667072       1.048031090 1.001901319       0.822894107       1.642410330
RoamMins
1.090383062

> exp(coef(logit_model1))/(1+exp(coef(logit_model1))) # Probability
(Intercept)       AccountWeeks      ContractRenewal1  DataPlan1   DataUsage
0.001893202       0.500492192       0.112509567       0.224688792 0.884678914
CustServCalls     DayMins     DayCalls          MonthlyCharge     OverageFee
0.620733307       0.511726162 0.500474878       0.451421782       0.621557640
RoamMins
0.521618780
```

4. In-sample Classification matrix

```
          Reference
Prediction    0    1
         0 1951  285
         1   44   54

               Accuracy : 0.859
            Sensitivity : 0.15929
            Specificity : 0.97794
```

5. Out-of-sample classification matrix

```
          Reference
Prediction   0   1
         0 843 123
         1  12  21

               Accuracy : 0.8649
            Sensitivity : 0.14583
            Specificity : 0.98596
```

6. Heteroscedasticity check

Solution of regression problem becomes unstable in presence of 2 or more correlated predictors. Multicollinearity can be measured by computing variance inflation factor (VIF) which gauges – how much the variance of regression coefficient is inflated due to multicolinearity.

```
> vif(logit_model1)
   AccountWeeks ContractRenewal        DataPlan       DataUsage
       1.001517        1.046133       12.878313     1541.203760
```

```
CustServCalls   DayMins   DayCalls   MonthlyCharge   OverageFee   RoamMins
1.070081        942.8935   1.003864   2759.467445     207.198784   1.210579
```

As a thumb rule, VIF of more than 5 or 10 is considered to be significant. And such variables can be removed to improve stability of the regression model.

7. ROC plot

Finally, let's draw the Receiver Operating Characteristic (ROC) plot. It is a plot of the True Positive Rate against the False Positive Rate for the different possible cut-points of a diagnostic test.

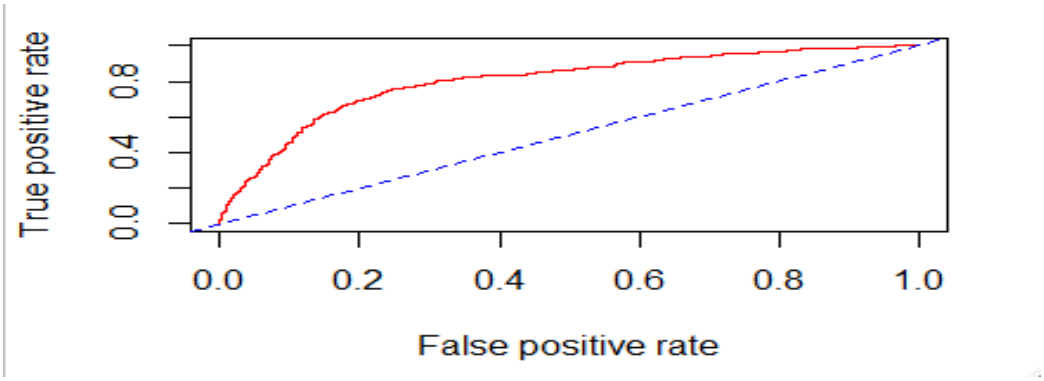
An ROC curve demonstrates several things:

- It shows the trade-off between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
- The slope of the tangent line at a cut-point gives the likelihood ratio (LR) for that value of the test.
- The area under the curve (AUC) is a measure of text accuracy.

Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test. A rough guide for classifying the accuracy of a model verification test is the traditional academic point system, as follows:

0.90-1 = excellent (A)
0.80-0.90 = good (B)
0.70-0.80 = fair (C)
0.60-0.70 = poor (D)
0.50-0.60 = fail (F)

AUC At 0.8028, the ROC Curve of our model demonstrates fairly good results.



8. Area under the curve

AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive data point higher than a randomly chosen negative data point. Higher the probability better is the classifier.

```
> train.area
[1] 0.8026955
```

9. Gini Coefficient

Gini coefficient is a ratio of two areas:

- the area between the ROC curve and the random model line
- top left triangle above the random model line – which is just 0.5

It can also be simplified as: (2 * AUC – 1)

```
> train.gini
[1] 0.6053911
```

10. Kolmogorov–Smirnov test

This performance measure is defined as maximum difference between TPR and FPR. Higher KS stat value indicates better model.

```
> train.ks
[1] 0.510493
```

3.3. Logistic Regression – significant variables

So far we have checked the Model's overall significance, and we are pretty happy with its results.

Hence shall we conclude and finalize the Model? Or we can do some more refining?

Let's revisit the predictors which are less significant and also see if there is any interaction between variables.

As seen in section 3.2.4, variable "MonthlyCharge" is insignificant (odds less than 1) and also as per section 2.1.1, it is correlated with the following three variables:

- Data Plan
- Data Usage
- DayMins

So we may decide to exclude it in our refinement step.

So far in our Logistic Regression journey, we have included all the explanatory variables in our model. However, selecting the one's which really matters for the model becomes really important.

There are two main approaches towards selecting variables: the all possible regression approach and automatic methods.

The all possible regressions approach considers all possible subsets of the pool of explanatory variables and finds the model that best fits the data according to some criteria (e.g. Adjusted R2, AIC and BIC). These criteria assign scores to each model and allow us to choose the model with the best score.

Automatic methods are useful when the number of explanatory variables is large and it is not feasible to fit all possible models. In this case, it is more efficient to use a search algorithm (e.g., Forward selection, Backward elimination and Stepwise regression) to find the best model.

Let's use the R function step() to perform the Variable Selection. The Model giving minimum value of the AIC would be the best one to choose from.

```
> ### Model refining - Logistic Regression
> logit_model2  =  glm(Churn ~ . -DataUsage -MonthlyCharge,
+                      data = train.data,
+                      family = binomial(link="logit"))
>
> summary(logit_model2)

Call:
glm(formula = Churn ~ . - DataUsage - MonthlyCharge, family = binomial
(link = "logit"), data = train.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8580  -0.5271  -0.3667  -0.2249   2.9227

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -5.593738   0.627886  -8.909  < 2e-16 ***
AccountWeeks    0.001387   0.001654   0.839   0.4016
ContractRenewal -1.789193   0.169215 -10.573  < 2e-16 ***
DataPlan       -0.870395   0.169323  -5.140 2.74e-07 ***
CustServCalls   0.457208   0.045479  10.053  < 2e-16 ***
DayMins         0.012619   0.001255  10.056  < 2e-16 ***
DayCalls        0.001776   0.003211   0.553   0.5803
OverageFee      0.125623   0.026682   4.708 2.50e-06 ***
RoamMins        0.064212   0.023275   2.759   0.0058 **
---
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: 1574.4  on 2325  degrees of freedom
AIC: 1592.4

Number of Fisher Scoring iterations: 5
```

Recall earlier when we built our all-inclusive model wherein we had got the higher AIC value. Using the step() function, now we got the best AIC of 1528.4, with only 6 significant predictor variables.

We may proceed with this model, to check its overall significance now, by performing the below tests:

1. Log Likelihood Test (model significance)
2. McFadden's pseudo-R Squared Test (model robustness)
3. Odds Explanatory Power
4. In Sample classification matrix
5. Out of the sample classification matrix
6. Heteroscedasticity check (VIF test)
7. ROC Plot
8. Area Under the Curve (AUC)
9. Gini Coefficient
10. KS test

3.4. Logistic Regression – calibrating threshold

The threshold parameter creates a trade-off between the false positives and false negatives. But due to different cost of errors and/or based on business requirements, it could be tuned in to fulfill the needs of problem-in-hand.

In our case, telecom industry would like to predict correctly all the customers who have probability to churn. Cost of wrongly predicting customers as positive is low. Hence, to improve sensitivity, threshold may be decreased and decided based on in-sample results.

```
# Performance metrics (out-of-the-sample)

pred = predict(logit_model2, newdata=test.data, type="response")
y_pred_num = ifelse(pred>0.35,1,0)
y_pred = factor(y_pred_num, levels=c(0,1))
y_actual = test.data$Churn
confusionMatrix(y_pred,y_actual,positive="1")
```

```
#                  Actual
#Prediction      0          1
#    0           811        94
#    1           44         50
# Accuracy       : 0.862
# Sensitivity    : 0.347
# Specificity    : 0.949
```

3.5. Logistic Regression – uncovering interactions

Having more than one input variable in a regression model brings up several issues that do not come up when there is only a single input.

- Ideally, all inputs should be linearly independent with respect to each other
- There may be interactions between inputs that should be explicitly included in the model
- Adding a new input changes the coefficient values for any inputs added previously

So, what exactly interaction is?

Even when two variables are completely independent, their effect on the target may not be. The attractiveness of an ice-cream cone may depend on both its price and the weather — especially how hot the day is. These variables may safely be assumed to be independent. (Certainly, the price of ice cream does not determine the temperature; temperature could conceivably affect the price of ice cream, but let us assume it does not.)

Despite the independence of these variables, the effect of price may still be affected by temperature. On a very hot day, people may buy ice cream at almost any price, whereas only a really good deal might tempt them when the weather is cold, damp, and drizzly.

In Model2, we considered 6 input variables, and there are chances of interactions between any of those.

When interactions are considered important, they are often included in the model by adding a new variable that is the product of the standardized values of the variables involved in the interaction.

A step() function and product of the input variables can be used to add interactions and build a further refined Model. Step() function will help us in coming up with the appropriate combination of input variables and the interaction terms.

Refining the Model by understanding interactions between variables is not in scope for this exercise; hence we are not going ahead with it. However, using interactions, we can further optimize the model and achieve accuracy beyond 90%, as well as sensitivity beyond 50%.

3.6. K-Nearest Neighbor's algorithm

KNN is supervised classifier, which uses neighbor data points' information to predict outcome variable. Neighbors are identified using distance measures such as Euclidean distance.

3.6.1. Normalizing continuous variables

Distance metric is highly influenced by the scale of the variable. Hence, it is important to standardize variables before utilizing them in model building. We will use min-max standardization method to bring all variables in same scale.

```
scale = preProcess(train.data, method = "range")
```

```
train.norm.data = predict(scale, train.data)
test.norm.data = predict(scale, test.data)
```

3.6.2. KNN model building

Caret package has train() method for training our data for various algorithms. We just need to pass different parameter values for desired algorithms.

We will first use trainControl() method to control the computational nuances of the train() method.

- "method" parameter refers to resampling method. Let's try to use CV i.e., cross-validation
- "number" parameter implies number of resampling iterations

It automatically iterates through different values of "k" and identifies the optimal value.

```
> knn_fit = train(Churn ~., data = train.norm.data, method = "knn",
+                 trControl = trainControl(method = "cv", number = 3),
+                 tuneLength = 10)
```

```
> knn_fit
k-Nearest Neighbors

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

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 1556, 1556, 1556
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.8946015  0.4654674
   7  0.8967438  0.4551969
   9  0.8954584  0.4348789
  11  0.8958869  0.4305237
  13  0.8946015  0.4184169
  15  0.8916024  0.3849744
  17  0.8886033  0.3624811
  19  0.8881748  0.3520873
  21  0.8856041  0.3228548
  23  0.8821765  0.2971357

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

3.6.3. Model Performance

1. In Sample confusion matrix

```
#                 Actual
#Prediction    0       1
#    0       1986     181
#    1          9     158
# Accuracy : 0.919
# Sensitivity : 0.466
# Specificity : 0.996
```

2. Out of the Sample confusion matrix

```
pred = predict(knn_fit, newdata = test.norm.data[-1], type = "raw")
confusionMatrix(pred,test.norm.data$Churn,positive="1")
```

```
#                  Actual
#Prediction    0       1
#    0        841      91
#    1         14      53
# Accuracy : 0.895
# Sensitivity : 0.368
# Specificity : 0.984
```

3.7. Naïve Bayes method

The e1071 package holds the naiveBayes function. It allows continuous and categorical features to be used in the naive bayes model. It is count-based classifier i.e. only thing it does is – count how often each variable's distinct values occur for each class.

3.7.1. Prior probabilities and conditional probabilities

```
> NB

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = train.norm.data[-1], y = train.norm.data$Churn)

A-priori probabilities:
train.norm.data$Churn
        0         1
0.8547558 0.1452442

Conditional probabilities:
                     AccountWeeks
train.norm.data$Churn      [,1]      [,2]
                   0 0.4107500 0.1625249
                   1 0.4233404 0.1640356


                     ContractRenewal
train.norm.data$Churn          0          1
                   0 0.06766917 0.93233083
                   1 0.27138643 0.72861357

                     DataPlan
train.norm.data$Churn         0         1
                   0 0.7082707 0.2917293
                   1 0.8348083 0.1651917
```

```
                      DataUsage
train.norm.data$Churn       [,1]      [,2]
                    0 0.1575364 0.2371174
                    1 0.0992243 0.2158593

                      CustServCalls
train.norm.data$Churn       [,1]      [,2]
                    0 0.1630744 0.1308800
                    1 0.2425434 0.2057204

                      DayMins
train.norm.data$Churn       [,1]      [,2]
                    0 0.4868149 0.1467107
                    1 0.5839357 0.2024648

                      DayCalls
train.norm.data$Churn       [,1]      [,2]
                    0 0.5223949 0.1459424
                    1 0.5266033 0.1567201

                      MonthlyCharge
train.norm.data$Churn       [,1]      [,2]
                    0 0.4241457 0.1729747
                    1 0.4623135 0.1732142

                      OverageFee
train.norm.data$Churn       [,1]      [,2]
                    0 0.5067509 0.1498000
                    1 0.5452757 0.1571581

                      RoamMins
train.norm.data$Churn       [,1]      [,2]
                    0 0.5091153 0.1409820
                    1 0.5322271 0.1459401
```

3.7.2. Model performance

1. In Sample confusion matrix

```
#                 Actual
#Prediction    0       1
#    0        1940    250
#    1          55     89
# Accuracy : 0.869
# Sensitivity : 0.263
# Specificity : 0.972
```

2. Out of the Sample confusion matrix

```
#                  Actual
#Prediction     0       1
#     0        840      109
#     1         15       35
# Accuracy : 0.876
# Sensitivity : 0.243
# Specificity : 0.982
```

4. Communicating results

4.1. Model Comparison

In this business problem, decision makers will be keen to identify positives very accurately. Hence, we will not just evaluate models based on accuracy on test data, we will also use sensitivity as metric to compare model performances.

Algorithm	Accuracy	Sensitivity	Specificity
Logistic Regression	86.2	34.7	94.9
K Nearest Neighbors classifier	91.9	46.6	99.6
Naïve Bayes classifier	87.6	24.3	98.2

Accuracy and Sensitivity is relatively high for KNN among the above methods. Yet, insights from logistic regression model can still be utilized to assist decision makers.

4.2. Insights sharing

An organization loses its customers to its competition for various reasons. Churn can affect the company's overall growth. The reputation of the company also goes down in the market if the percentage churn increases year on year. For a company to expand its clientele, its growth rate, as measured by the number of new customers, must exceed its churn rate.

For the data provided to our assignment, the Customer Churn is significantly affected by following variables:

- **Contract Renewal:** Our Model suggests that if a Customer renews contract, then there is 11% probability that the customer will not churn compared to the one who has not renewed his contract.

- **Data Plan:** Data plan also has negative impact on Customer Churn. The customer opting for Data Plan has 27% probability that the customer will not churn compared to the one who has not opted for Data Plan.

- **Customer Service Calls:** The odds of Customer churning out are 1.63 when he makes one unit of Service Calls, who is not making Service Calls. This translates to 62% probability of the customer churning out. The Telecom companies need to ensure that their customers are happy with their services, so that the Customer Churn is reduced.

- **Day Mins:** The odds of Customer churning out are 1.01 when there is an increase of one unit of average daytime minutes per month. This translates to 50% probability of that customer churning out. The Telecom companies need to keep an eye on this parameter.

- **Overage Fee:** The odds of Customer churning out are 1.01 when there is an increase of one unit in overage fee. This translates to 54% probability in the customer churn.

- **Roaming Minutes:** The odds of Customer churning out are 1.09 when there is an increase of one unit in Roaming Minutes of a Customer usage, compared to the one who is not availing roaming minutes. This translates to 52% probability of that customer churning out. The Telecom companies need to keep an eye on this parameter.

-- x --