Predicting customer churning possibilities based on service provider data

Shafeeq Ahmed

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The objective of this project is to predict whether a particular customer would discontinue his/her subscription with the service provider or not based on the statistical data given by the network service provider. Prediction and prevention of customer churn brings a huge additional revenue and moreover, retaining existing customer is significantly economical compared to spending resources to acquire more customers. We use the telecom customer data set in this project to predict the possibility of customers churning out.

1.2 DATASET

A sample of the entire dataset which we will use to build our classification model is given below in two parts: dependent and independent variables

	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls
0	KS	128	415	382-4657	no	yes	25	265.1	110
1	OH	107	415	371-7191	no	yes	26	161.6	123
2	NJ	137	415	358-1921	no	no	0	243.4	114
3	OH	84	408	375-9999	yes	no	0	299.4	71
4	OK	75	415	330-6626	yes	no	0	166.7	113

	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night calls	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls
0	45.07	197.4	99	16.78	244.7	91	11.01	10.0	3	2.70	1
1	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.70	1
2	41.38	121.2	110	10.30	162.6	104	7.32	12.2	5	3.29	0
3	50.90	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2
4	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3

TABLE 1.1 PREDICTOR VARIABLES

The predictor variables given in the above table 1.1 provides the overview of the service usage of the customer like total number of calls made, number of voice mail messages sent by the customer and the expenditure incurred for them. We are also provided with the geographic information about the customers.

Churn
False.
False.
False.
False.
False.

TABLE 1.2 RESPONSE VARIABLE

*Our task is to predict the class of the response variable **Churn** (sample given in table 1.2) by using these predictor variables. The target classes each represent the positive and negative chances for the customer to churn out ('False' meaning the customer would continue his subscription and 'True' meaning that he would terminate it)*

CHAPTER 2

EXPLORATORY DATA ANALYSIS

2.1 SIGNIFICANCE

For any given problem statement, exploring the given dataset and acquiring a deeper knowledge of the relationship between the variables is vital for producing more precise end results. Exploratory data analysis comprises calculating statistical parameters like mean, median, correlation coefficient etc., and also more powerful method of visualisations. Visualization has the added advantage of projecting out hidden patterns in the dataset which can be intuitively understood by any person and this helps in procuring a better business knowledge.

2.2 IMPORTANT VISUALISATIONS

Here we highlight some of the predictor variables and the positive/negative influence they inflict upon the customer decision to churn/stay. The corresponding python and R codes for the figures are given in appendix.

First of all, we start with the geographic location of the customers and we are interested in knowing how their location plays a role in satisfying their network needs. Figure 2.1 shows the relative percentage of customers that churned out in all the 50 states and in the national capital of USA.

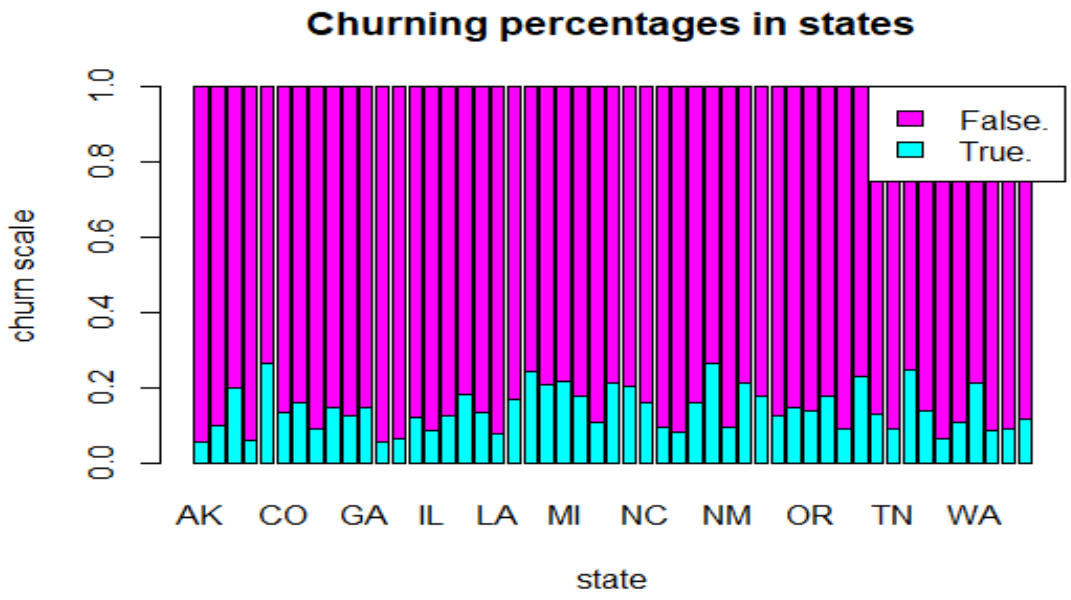


FIGURE 2.1

Since our target variable has a class imbalance problem, the data plotted as it is would be so clear. Hence the churning out of customers is expressed as a relative percentage of the total number of customers from the respective state. As it is evident from the above bar chart, both the states of California and New Jersey have the highest percentage (26% approx.) of customers churning out. In layman's terms, it can be said that if a particular customer is from California or New Jersey, there is a pretty good chance that he will not be happy with the service provided. This can be because of various reasons like poor network coverage, inadequate customer service centres in customer locality etc.,

Simple barplots depicting the satisfactory level of customers who have opted for international plan and Voice mail plan are given below:

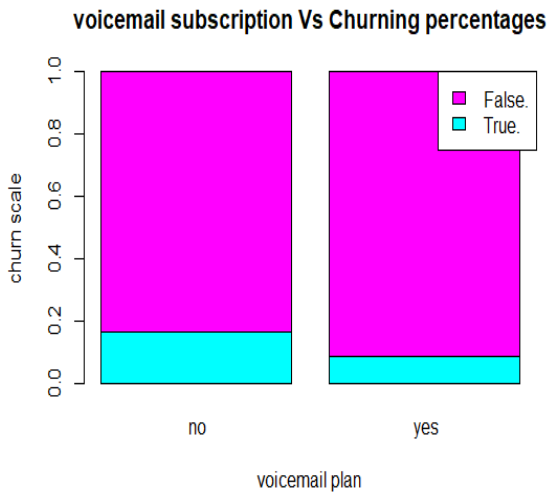


FIGURE 2.2

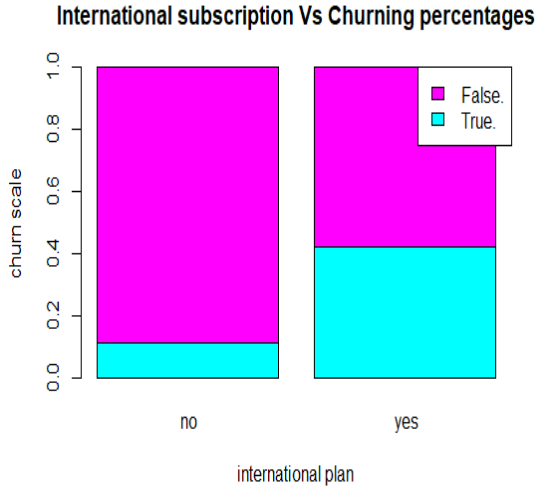


FIGURE 2.3

Again, the bars are scaled with relative percentages. We can infer from Figure 2.2 that percentage of the customers who opted for voice mail plan and churned out is relatively lower that the customers who got out without experiencing voice mail service. On the contrary, Figure 2.3 illustrates that significant fraction of customers who opted for international plan churned out of the network (more than 40%). Simply speaking, if a customer is found to be on the verge of churning out, the service provider can consider providing voice mail service to him/her for a trail period free of cost. Also, the company should review the facilities provided as part of international plan as it turns out to be infamous.

Having analysed the categorical variables, we shall now turn our attention towards the tariff details which take up the form of continuous predictor variables. We have plotted the respective boxplots for the tariff paid by the customers for the calls they made during different timings of a day (daytime, evening and night) and they are presented below:

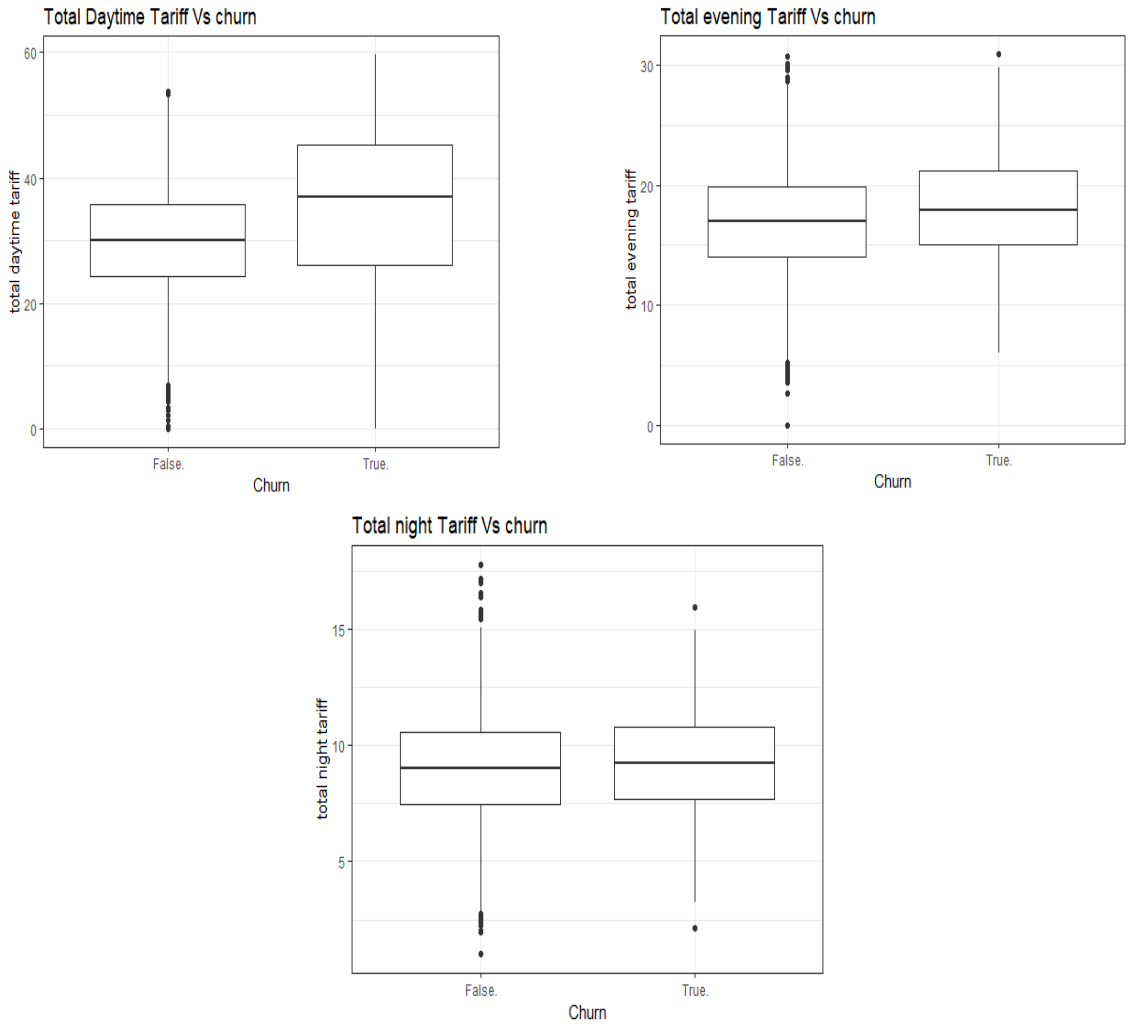


FIGURE 2.4

We could see that the quartile ranges are significantly lifted above in case of day tariff although the difference is barely visible for tariffs corresponding to evening and night timings. As will be demonstrated in correlation analysis, the duration in mins spent during a call is directly proportional to the tariff (charged on a minutely basis) and so people who spend a lot of time over the phone are more likely to churn out because of the heavy tariff imposed on them.

The variation of international call tariff with respect to international plan also reveals some information about the churning trend. Referring to Figure 2.5, we find that the boxplot distributions of international call tariff for customers who didn't opt for international plan is almost the same for churning/non-churning cases but once the international plan comes in as a factor, we find that the customers churned out as the tariff went up as shown below:

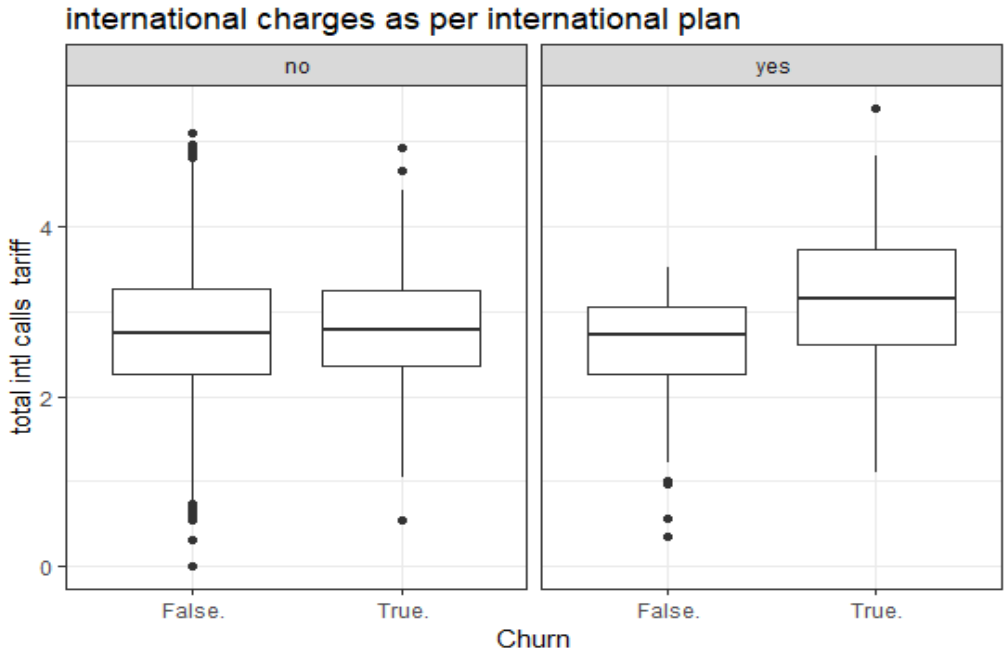


FIGURE 2.5

This again highlights the point that whatever offers that are being provided under the international plan is not very efficient in keeping the customers satisfied.

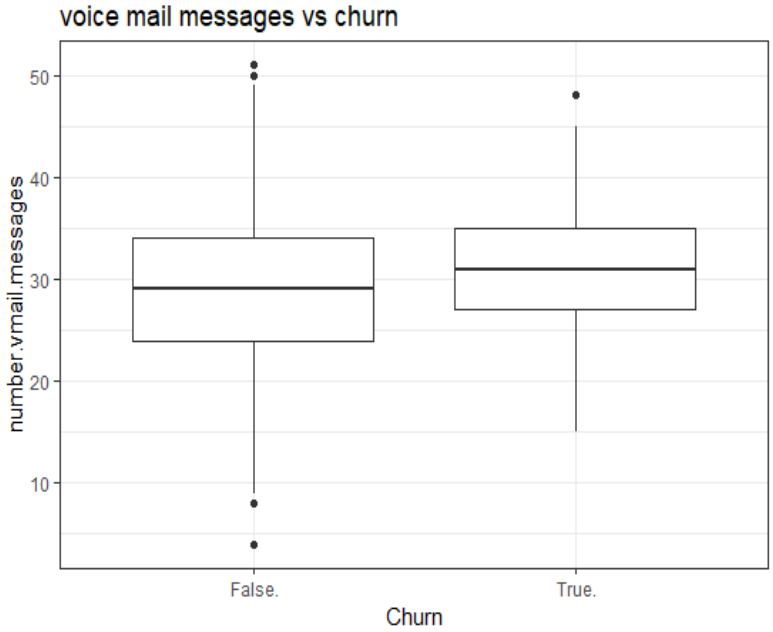


FIGURE 2.6

The above figure illustrates the distribution of voice mail messages sent by customers who have subscribed for voice mail plan. These boxplots emphasises the simple fact that those who have sent more number of voice mail messages would be charged more and hence it indirectly influences the churning decision of the customer.

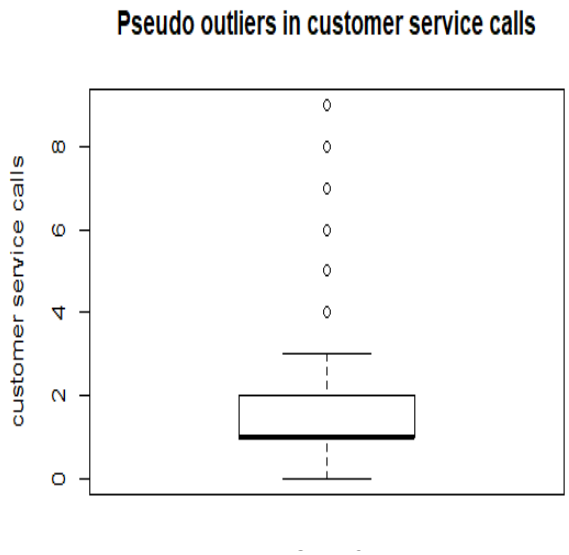


FIGURE 2.7

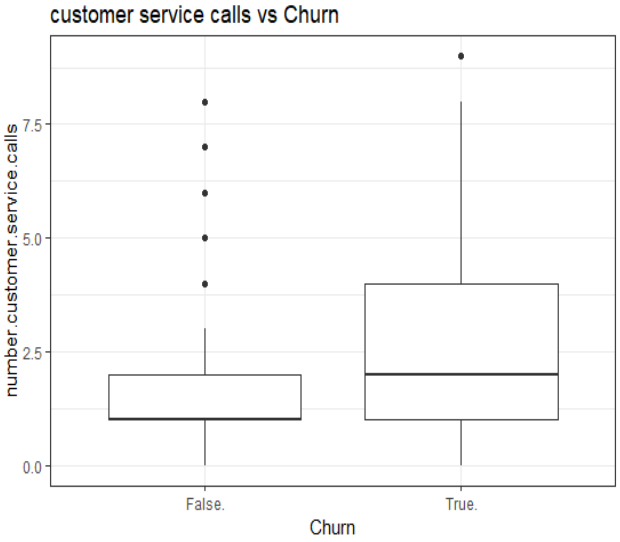


FIGURE 2.8

Before proceeding to the final analysis, it is necessary to point out a natural intuition that we get while looking at the relation between customer care calls and churning. If a customer calls the customer care numerous times, then it is safe to assume that he/she has grievances that needs to be addressed. Hence a high number (or abnormal number) of customer care calls indicate that the customer is reporting a lot of issues and hence becoming increasingly uncomfortable with the service.

Having said this, if we look at the figures above, a simple boxplot of the total number of customer care calls made by the customer suggests that there are lot of outliers in the distribution. But the moment, we facet the plot using churn, the outliers are significantly reduced. This indicates that the outliers are indeed not outliers but valid data points providing information about the churning decision. Taking this into account along with the results of the ANOVA test (described in later chapters), ***outlier removal is not done for this particular predictor variable.***

CHAPTER 3

DATA PRE-PROCESSING

3.1 NEED FOR PRE-PROCESSING

The raw data that we are considering for our problem statement may not be in the ideal shape that meet the standards of training our model. To name a few, it may contain irregularities like missing values, redundant variables, outliers etc., so it is a preliminary requirement for building our model that we clean our data. The subsequent topics elaborately describe the different steps involved in data pre-processing.

3.2 OUTLIER DETECTION AND REMOVAL

Outliers are data points that differ significantly from the overall spread of the data. The presence of outliers in the variables degrade the quality of learning of the model as it distorts the weightage that is assigned to the predictor variables. Removal of outliers from the dataset is mandatory before using it for further analysis. We have already seen individual boxplots for predictor variables in the previous chapters which also indicates the presence of outliers

Observations that contain outliers can either be removed or imputed with suitable values. Our Dataset has a ***peculiar target class imbalance*** *which means that the number of cases available in our dataset where the customer actually churned out is extremely lower than the cases where the customer did not churn. Therefore, we cannot afford to lose more data which describes the minority class by removing outliers. Hence, we have opted for a combination of both.*

Using the below function, we have removed the observations which contains outliers but the customer didn't churn out and imputed the outlier cases where the customer did quit so that the minority class remains unaffected.

```
outlier_removal=function(df,num,to_variable){
  for (i in num){
    outliers=df[,i][df[,i] %in% boxplot.stats(df[,i])$out]
    out.index=row.names(df[which(df[,i] %in% outliers & df$Churn==1),])
    in.indices=!row.names(df) %in% out.index
    df=df[in.indices,]
    df[which(df[,i] %in% outliers),i]=NaN
    assign(to_variable,df,envir = .GlobalEnv)
  }
}
```

3.3 Missing value Imputation:

Our Dataset now contains missing values which were induced as part of outlier analysis. Missing values impede the performance of the classification model that we are going to build on our dataset. Moreover, the various feature selection tests that follow data pre-processing cannot function properly if there are missing values in the dataset. Hence the missing values must imputed and there are several ways to do that. We can use the mean, median values of the predictor variables to fill up NAN values or we can use a more dynamic distance approach.

For our dataset, we have used KNN (k Nearest Neighbours) imputation method which is a distance based method to find k – number of closest observations to the particular observation which contains missing values and uses the average of those k-values to impute the missing values.

We have also tested the accuracy of all the three methods to confirm that KNN gives best results. An artificial NAN is induced at the 1000th value of 'total data charge' variable and the imputed values are compared with the real value. As indicated below, KNN produces a value that is closest to the real value

```
imputation method       Actual value: 28.27 (churn['total day charge'][1000])
mean                    30.636593192868744
median                  30.5
KNN                     29.88962811568968
```

3.4 Feature Selection

The data acquired for resolving a particular problem statement may not always be fully relevant to the case in point. Since there may be multiple sources from where data is extracted, there is a good chance for irrelevant features to find their way in to the dataset. If a predictor variable that has no useful information to predict the outcome of a response variable is included in training set of our model, it can cause severe performance degradation of the model.

Moreover, if two or more predictor variables included in the dataset contains the same information or in technical terms, highly correlated, then the redundant information can also impact the performance of our model. Hence it is a usual practice to subject the features to statistical tests like correlation analysis, chi-square and ANOVA tests to determine the level of contribution they make to the prediction of the response variable. We have carried out the following tests to determine the validity and relevance of the predictor variables.

3.3.1 Correlation Analysis:

The following correlation plot illustrates the pairwise correlation between numerical variables of the dataset. The lower triangle and diagonal of the correlation matrix fed in to this heatmap are masked to avoid redundancy.

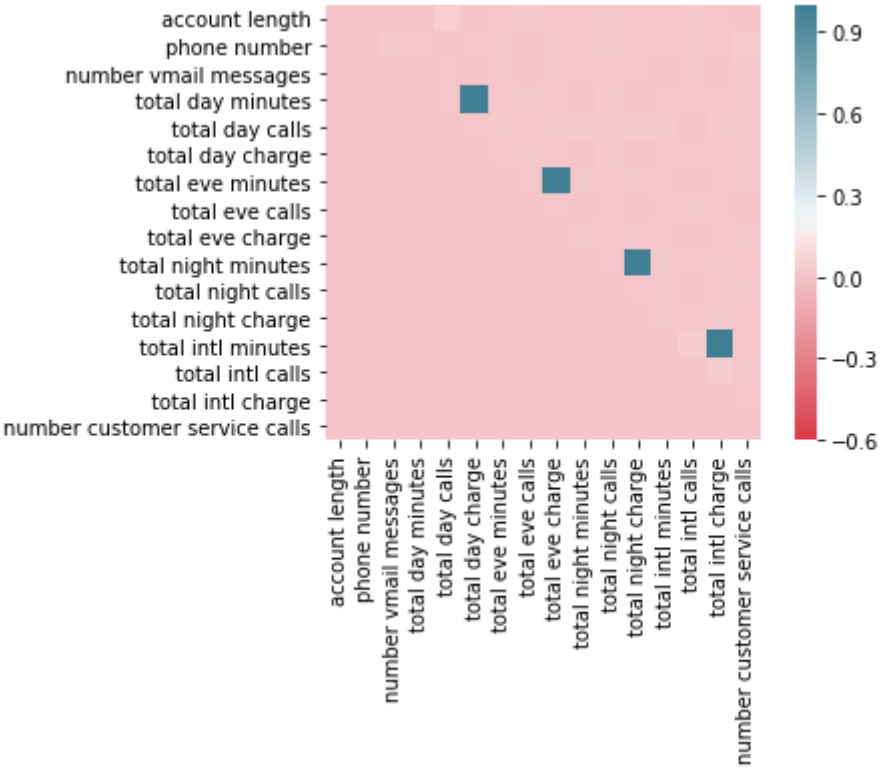


FIGURE 3.1

The value of correlation coefficient ranges between +1 to -1, +1 being highly positively correlated and -1 being highly negatively correlated. As we can see from the above heatmap, number of minutes spent on the phone is strongly correlated with the respective charges which is obvious because the customers are charged on a minutely basis. In fact, there is a constant amount per minute fixed by the network company and the total charge is an exact multiple of this base amount.

Therefore, as per the results of correlation test, we will ***remove the attributes representing total minutes spent on the phone during day, evening and night and on international calls*** *as charges incurred has more direct relation with customer churn.*

3.3.2 Chi-Square test:

Chi-square statistics examines the independence of two categorical vectors. By calculating the chi-squared statistic between a feature and the target vector, we obtain a measurement of the independence between the two. If the target is independent of the feature variable, then it is irrelevant for our purposes because it contains no useful information for our model. On the other hand, if the two variables are highly dependent, it is very likely that the predictor is useful for our model.

We have used the below block of code to create a function that will accept the dataset and factors and returns only the list of factor variables that are related to the target vector.

```
chi_sqre_test=function(df,fact,trgt){
  col_fact=c()
  for (i in fact){
    x=chisq.test(table(df[,trgt],df[,i]))
    if (x$p.value<0.05){
      col_fact=append(col_fact,i)
    }
  }
  return(col_fact)
}
```

*From the Chi-square test, we have found out the **area code of the customer is irrelevant for our model** as it contains no useful information and so it is removed.*

3.3.3 ANOVA test:

Analysis of Variance (ANOVA) test gives us the F-statistic value which is a measure of independence between a numerical attribute and the target vector. F-value scores examine if, when we group the numerical attribute by the classes of the target variable, the means for each group are significantly different. In other words, if there is high variance between those groups, then we can conclude that the feature has some pattern which the target classes follow.

We have created the below function to let us know only those numerical attributes which has useful information to drive the classes of the target variable.

```
anova_test=function(df,num,trgt){
  col_ano=c()
  for (i in num){
    ftest=aov(df[,i] ~ df[,trgt], data=df)
    qu=summary(ftest)
    if (qu[[1]][1,5]<0.05){
      col_ano=append(col_ano,i)
    }
  }
  return(col_ano)
}
```

The results of the ANOVA test were significantly useful in simplifying the dataset as it removes a bunch of unnecessary variables as below.

1. *Phone number of the customer*
2. *Number of calls made by the customer during various times of the day (since the only relevant information is the duration of the calls or the charge incurred)*
3. *Customer account length*

Important Note: *our assumption regarding the validity of the customer service calls attribute is further supported by ANOVA test. When the variable was subjected to outlier analysis and then fed to ANOVA, the test rejected the variable. But when the variable was fed as it is to the test, it considered the variable to contain useful information for the target.*

Hence, after applying all these tests, we have simplified the dataset to contain only the relevant variables. A sample of the same is given below.

	number customer service calls	number vmail messages	total day charge	total eve charge	total intl calls	total intl charge	total night charge	state	international plan	voice mail plan	Churn
0	1.0	25.0	45.07	16.78	3.0	2.70	11.01	16.0	0.0	1.0	0.0
1	1.0	26.0	27.47	16.62	3.0	3.70	11.45	35.0	0.0	1.0	0.0
2	0.0	0.0	41.38	10.30	5.0	3.29	7.32	31.0	0.0	0.0	0.0
3	3.0	0.0	28.34	12.61	3.0	2.73	8.41	36.0	1.0	0.0	0.0
4	0.0	0.0	37.98	18.75	6.0	1.70	9.18	1.0	1.0	0.0	0.0

3.5 Feature Scaling

Our dataset contains numerical values that are measured in different scales. This makes the range of each variable disproportionate to each other. For example, the call tariffs may be measured in terms of U.S dollars ranging from 0 to say 100 dollars whereas the customer service calls are just discrete numbers having a much lower range. Considering them with their different unit scales as they are, will result in significant imbalance while building our model. The model will be inclined towards the variable that has higher range of values.

Therefore it is necessary to apply corrective measures to the variable scales so that they fall under a universal scale. For our dataset, we have normalised the variables to convert their range so that their values fall within a common limit (between 0 and 1). A sample of the normalised dataset is given below.

number customer service calls	number vmail messages	total day charge	total eve charge	total intl calls	total intl charge	total night charge	state	international plan	voice mail plan	Churn
0.111111	0.50	0.792375	0.486370	0.222222	0.474667	0.665289	16.0	0.0	1.0	0.0
0.111111	0.52	0.431646	0.479446	0.222222	0.741333	0.701653	35.0	0.0	1.0	0.0
0.000000	0.00	0.716745	0.205971	0.444444	0.632000	0.360331	31.0	0.0	0.0	0.0
0.333333	0.00	0.449477	0.305928	0.222222	0.482667	0.450413	36.0	1.0	0.0	0.0
0.000000	0.00	0.647059	0.571614	0.555556	0.208000	0.514050	1.0	1.0	0.0	0.0

CHAPTER 4

MODEL PHASE

4.1 MODEL SELECTION

As we have already defined in our problem statement, the main goal of this project is to predict whether the customer would churn out or not and hence, our problem statement falls under the category "classification". We can choose a variety of machine learning algorithms for building a binary classification model. For this project, the following 4 algorithms are chosen and their performance is evaluated on the same train and test data to select the best one of them.

1. *Logistic Regression*
2. *Naïve Bayes*
3. *Decision Tree*
4. *Random Forest*

4.2 LOGISTIC REGRESSION

Logistic regression works in a similar manner to linear regression. In fact, it uses the same linear formula but the result of the linear equation is manipulated to indicate the probabilities of the target classes. The following formula is a simplified version that is used by logistic regression to calculate the probabilities of the target class for a particular observation.

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

The formula indicates that we are still interested in the linear equation but the outcome is interpreted as the respective probabilities of the target classes.

The summary statistics of the logistic model which indicates the coefficients of the predictors (beta values in the formula) along with their significance level is given below

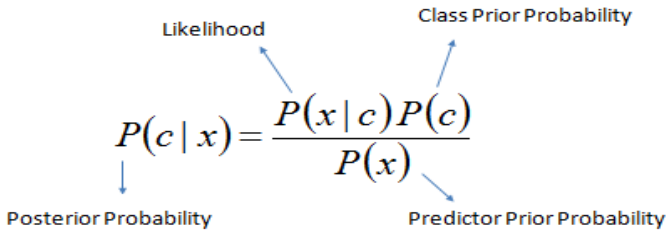
```
> summary(logistic_model)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9532  -0.5260  -0.3290  -0.1776   3.0275

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -7.30240    0.74963  -9.741  < 2e-16 ***
number.vmail.messages      1.84092    0.92989   1.980 0.047736 *
total.day.charge           3.47538    0.32411  10.723  < 2e-16 ***
```

```
total.eve.charge               1.95258    0.32970   5.922 3.18e-09 ***
total.night.charge             1.18713    0.32018   3.708 0.000209 ***
total.intl.calls              -1.19551    0.26048  -4.590 4.44e-06 ***
total.intl.charge              1.01116    0.31269   3.234 0.001222 **
number.customer.service.calls  4.77492    0.36859  12.954  < 2e-16 ***
state2                         0.38542    0.76127   0.506 0.612656
state3                         1.07658    0.75287   1.430 0.152727
state4                         0.18547    0.84347   0.220 0.825956
state5                         2.12912    0.79022   2.694 0.007053 **
state6                         0.76572    0.76087   1.006 0.314241
state7                         1.06808    0.72462   1.474 0.140485
state8                         0.71092    0.80687   0.881 0.378273
state9                         0.79269    0.75067   1.056 0.290981
state10                        0.70061    0.75738   0.925 0.354944
state11                        0.74798    0.77605   0.964 0.335130
state12                       -0.10014    0.89841  -0.111 0.911249
state13                        0.29949    0.90178   0.332 0.739804
state14                        0.98978    0.74602   1.327 0.184594
state15                       -0.09582    0.83158  -0.115 0.908266
state16                        0.58088    0.75006   0.774 0.438667
state17                        1.20299    0.73003   1.648 0.099380 .
state18                        0.80411    0.76373   1.053 0.292403
state19                        0.77962    0.83704   0.931 0.351643
state20                        1.31641    0.74310   1.771 0.076478 .
state21                        1.22800    0.71587   1.715 0.086270 .
state22                        1.44965    0.72658   1.995 0.046024 *
state23                        1.46843    0.71082   2.066 0.038846 *
state24                        1.18088    0.71432   1.653 0.098300 .
state25                        0.79399    0.77446   1.025 0.305261
state26                        1.40812    0.72449   1.944 0.051942 .
state27                        1.92529    0.71586   2.689 0.007156 **
state28                        0.69897    0.75011   0.932 0.351430
state29                        0.21230    0.79391   0.267 0.789158
state30                        0.43422    0.80402   0.540 0.589150
state31                        1.26283    0.76806   1.644 0.100137
state32                        1.62439    0.70729   2.297 0.021638 *
state33                        0.57301    0.78880   0.726 0.467577
state34                        1.29382    0.72482   1.785 0.074258 .
state35                        1.28992    0.71555   1.803 0.071438 .
state36                        1.00270    0.73985   1.355 0.175328
state37                        1.12821    0.75029   1.504 0.132660
state38                        0.81522    0.73638   1.107 0.268267
state39                        1.21974    0.78029   1.563 0.118006
state40                       -0.06490    0.82283  -0.079 0.937128
state41                        1.87877    0.73408   2.559 0.010487 *
state42                        0.92756    0.76067   1.219 0.222694
state43                        0.27099    0.81741   0.332 0.740252
state44                        1.86362    0.70649   2.638 0.008343 **
state45                        1.10506    0.74381   1.486 0.137365
state46                       -0.17106    0.82553  -0.207 0.835842
state47                        0.17108    0.77515   0.221 0.825326
state48                        1.47898    0.72209   2.048 0.040542 *
state49                        0.45229    0.77941   0.580 0.561715
state50                        0.62366    0.73034   0.854 0.393144
state51                        0.49781    0.75753   0.657 0.511086
international.plan2            2.20132    0.15410  14.285  < 2e-16 ***
voice.mail.plan2              -2.09421    0.59252  -3.534 0.000409 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 2681.8  on 3097  degrees of freedom
Residual deviance: 2041.0  on 3038  degrees of freedom
AIC: 2161

Number of Fisher Scoring iterations: 6
```

4.3 NAÏVE BAYES

Naïve Bayes classifier works on the basis of Bayes theorem which uses conditional probabilities to predict the target classes. The Bayes formula is as follows:



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Naïve Bayes classifier uses the conditional probability of the predictors when the target classes are known, to predict the target classes using the predictor variables. Since our dataset contains numerical attributes, naïve Bayes will assume that they are normally distributed and use the probability density function of normal distribution (given below) to calculate the conditional probabilities

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$$

Where μ is the mean and σ is the standard deviation of the numerical variable.

4.4 DECISION TREE

The decision tree that was developed to predict the churning possibility of the customers is given in figure 4.1. Decision trees work on the principle of information gain that is produced due to the introduction of predictor variables. The variable that reduces the entropy (or chaos) of the target variable to the maximum extent is chosen as the root node and subsequent decision nodes are selected to form a tree that finally arrives at a particular decision, which will be one of the classes of the target variable.

The figure illustrates that the variable "total day charge" is taken as the root node and the dataset is continuously divided by the remaining variables until a root node is reached. The prediction, as far as this project is concerned has improved significantly in terms of

performance by decision tree algorithm. The details regarding the performance statistics is illustrated in a detailed manner in the following chapter.

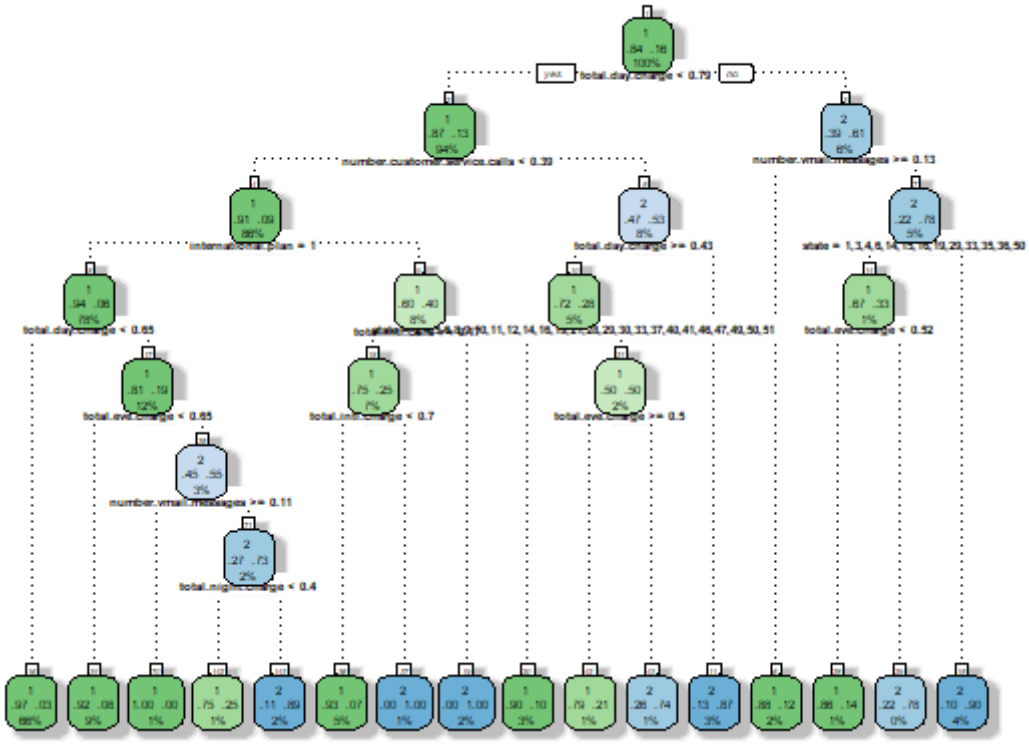


FIGURE 4.1

4.5 RANDOM FOREST

Random forest is an ensemble technique which uses multiple decision trees to further enhance the accuracy of prediction. A suitable number of decision trees is chosen and their result are iterated to produce a collective result.

CHAPTER 5

MODEL EVALUATION & SELECTION

5.1 PERFORMANCE METRICS

We have developed four models so far to predict the possibility of a customer churning out. In order to select the best suitable model, we have to use some sort of performance metrics which can be used as a touchstone to compare the performance of different models. In case of classification problems, the performance level is generally determined in terms of combinations of any of the following measures

1. ***Accuracy*** *– percentage of the total observations correctly classified by the model*
2. ***True positive rate*** *– fraction of total positive cases correctly classified as positive by the model*
3. ***True Negative rate*** *– fraction of total negative cases correctly classified as negative by the model*
4. ***False Positive rate*** *- fraction of total negative cases misclassified as positive by the model*
5. ***False negative rate*** *- fraction of total positive cases misclassified as negative by the model*

For any classification models, a decent level of accuracy is expected and that is a direct measure of the model performance. But in our case, in addition to accuracy, we also have to make sure that no customer who is likely to churn out is missed by our model. Therefore, we have to try and reduce ***False Negative Rate*** *(FNR) as far as possible (customer churning out is the positive case of our model although in practice, this event is not positive). An alternative parameter to FNR is* ***sensitivity*** *and is a measure of how sensitive our model is, towards predicting the positive class. Sensitivity is calculated as follows:*

*Sensitivity (%) = (1 – FNR) * 100*

Hence, the primary focus of the model building stage is to build models with high accuracy and high sensitivity (or low false negative rate). We also have to keep in mind, the ***severe class imbalance problem*** *that our dataset is suffering from. The total number of observations available for the customers who did not quit is enormous compared to the ones where the customers did churn out. A simple table containing the respective proportions illustrate this fact.*

```
> im=table(churn$Churn)
> prop.table(im)

        1         2
0.844093 0.155907
```

This indicates that more than 84% of the observations relate to the customers not churning out (represented by numeric code 1) and a mere 15.6% of the data provides information about

unsatisfied customers who churned out. If we were to blindly classify all the customers as belonging to the non-churning category in the training dataset, without considering any information, we would still be right about 84.4% of the time due to the class imbalance. This is called the ***no information rate*** *and we will have kept this parameter as our baseline while evaluating accuracy of various models. The different ways to overcome class imbalance problem are also discussed later in this chapter.*

5.2 Accuracy and FNR comparison of different models

The Accuracy and FNR values of different models are given below. The training and testing datasets are separately provided and so the different models will be trained and tested on same datasets. The performance metrics along with the confusion matrix that depicts both the predicted and actual classes in the form of a contingency table are given below:

1. ***Logistic Regression:***

```
> logistic_conf_matrix
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1343  132
         2  100   92

               Accuracy : 0.8608
                 95% CI : (0.8433, 0.8771)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 0.73102

                  Kappa : 0.3633
 Mcnemar's Test P-Value : 0.04183

            Sensitivity : 0.41071
```

2. ***Naïve Bayes:***

```
> NB_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1332  118
         2  111  106

               Accuracy : 0.8626
                 95% CI : (0.8452, 0.8788)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 0.6563

                  Kappa : 0.4016
 Mcnemar's Test P-Value : 0.6917

            Sensitivity : 0.47321
```

3. ***Decision Tree:***

```
> DT_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1370   89
         2   73  135

               Accuracy : 0.9028
                 95% CI : (0.8876, 0.9166)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 2.087e-06

                  Kappa : 0.5693
 Mcnemar's Test P-Value : 0.2386

            Sensitivity : 0.60268
```

4. ***Random Forest:***

```
> RF_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1294   67
         2  149  157

               Accuracy : 0.8704
                 95% CI : (0.8534, 0.8862)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 0.2972

                  Kappa : 0.5176
 Mcnemar's Test P-Value : 3.561e-08

            Sensitivity : 0.70089
```

5.3 ADDRESSING THE TARGET CLASS IMBALANCE PROBLEM

Before proceeding with model selection, we have to resolve the severe class imbalance problem that is found in the training dataset. As mentioned before, the ratio between observations available for positive and negative classes is hugely disproportionate. This causes the model to be biased and it will be inclined towards to majority class of the target variable. Therefore it is necessary to rectify this imbalance and the methods that can be used for the achieving balance are described and their performance is evaluated below. Since Random Forest and Decision Tree outperformed the remaining two models, we will consider only these two for further enhancement.

1. ***Under-sampling the majority class:***

This method removes random samples from the majority class so that the count is reduced and made equal to the minority class. This method is prone to information loss as a lot of samples containing useful information is disregarded. This is shown in the code below where the majority class is under-sampled to level the minority class so that number of observations (given by N) is met. The performance of decision tree and random forest on the testing data when trained by under-sampled is given for comparison.

```
> under=ovun.sample(Churn~.,data = churn,method='under',N=966)$data
> table(under$Churn)

  1   2
483 483
```

```
> DT_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1052   23
         2  391  201

               Accuracy : 0.7516
                 95% CI : (0.7302, 0.7722)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3698
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8973
```

```
> RF_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1005   24
         2  438  200

               Accuracy : 0.7229
                 95% CI : (0.7007, 0.7442)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 1

                  Kappa : 0.331
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8929
```

2. ***Over-sampling the minority class:***

This method duplicates the observations belong to the minority class and inserts them randomly into the dataset so that the ratio of observations become balanced.

```
> over=ovun.sample(Churn~.,data = churn,method='over',N=5230)$data
> table(over$Churn)

   1    2
2615 2615
```

```
> DT_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1052   23
         2  391  201

               Accuracy : 0.7516
                 95% CI : (0.7302, 0.7722)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3698
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8973
```

```
> RF_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1273   51
         2  170  173

               Accuracy : 0.8674
                 95% CI : (0.8502, 0.8833)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 0.4321

                  Kappa : 0.5346
 Mcnemar's Test P-Value : 2.062e-15

            Sensitivity : 0.7723
```

3. ***Combination of under and over sampling:***

This method includes a combination of both i.e., random under-sampling of Majority classes and over-sampling of minority classes

```
> both=ovun.sample(Churn~.,data = churn,method='both')$data
> table(both$churn)

   1    2
1522 1576
```

```
> DT_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1294   59
         2  149  165

               Accuracy : 0.8752
                 95% CI : (0.8584, 0.8907)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 0.1323

                  Kappa : 0.5415
 Mcnemar's Test P-Value : 6.784e-10

            Sensitivity : 0.73661
```

```
> RF_conf
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1 1170   38
         2  273  186

               Accuracy : 0.8134
                 95% CI : (0.7939, 0.8319)
    No Information Rate : 0.8656
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4443
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8304
```

5.4 CONCLUSION

From the above detailed portrayal of the performance statistics of individual models, it is evident that ***a combination of both under and over sampling*** *of the training data significantly improves the sensitivity although there is a small decrease in accuracy and the overall trade-off is advantageous. Out of the two models,* ***Decision Tree*** *meets the mandatory criteria of achieving an accuracy level of 87.5% which is more than the no information rate of 86.56%. The sensitivity is also significantly improved from 60% to 73.6%. Hence Decision Tree can be selected for predicting the possibility of customer churning for future test cases and the same can be used for deployment.*

APPENDIX - A

R function for plotting categorical variables

```
bar_visual = function(cat_a,cat_b,xmark,ymark,heading){
  tab =table(cat_a,cat_b)
  c=tab[,1]+tab[,2]
  tab[,1]=tab[,1]/c
  tab[,2]=tab[,2]/c
  tab=tab[,c(2,1)]
  return(barplot(t(tab),main=heading,col = c(5,6),xlab=xmark,ylab=ymark))
}
```

R code for Fig 2.1 (churning percentages in states)

```
bar_visual(churn$state,churn$Churn,"state","churn scale","Churning percentages in states")
legend("topright",legend=unique(churn$Churn),fill=c(6,5))
```

R code for Fig 2.2 & Fig 2.3 (voice mail plan and international plan vs churn)

```
#international plan
bar_visual(churn$international.plan,churn$Churn,"international plan","churn scale",
           "International subscription Vs Churning percentages")
legend("topright",legend=unique(churn$Churn),fill=c(6,5))

#vmail plan
bar_visual(churn$voice.mail.plan,churn$Churn,"voicemail plan","churn scale",
           "voicemail subscription Vs Churning percentages")
legend("topright",legend=unique(churn$Churn),fill=c(6,5))
```

R code for Fig 2.4 (Tariff vs churn)

```
#day
ggplot(churn,aes(x=Churn,y=total.day.charge))+theme_bw()+geom_boxplot()+
  labs(y="total daytime tariff",title="Total Daytime Tariff Vs churn")

#eve
ggplot(churn,aes(x=Churn,y=total.eve.charge))+theme_bw()+geom_boxplot()+
  labs(y="total evening tariff",title="Total evening Tariff Vs churn")

#night
ggplot(churn,aes(x=Churn,y=total.night.charge))+theme_bw()+geom_boxplot()+
  labs(y="total night tariff",title="Total night Tariff Vs churn")
```

R code for Fig 2.5 (International Tariff vs churn)

```
#international
ggplot(churn,aes(x=Churn,y=total.intl.charge))+theme_bw()+geom_boxplot()+
  facet_wrap(~international.plan)+
  labs(title="international charges as per international plan",
       y="total intl calls  tariff")
```

R code for Fig 2.6 (voicemail messages vs churn)

```
#vmail plan
bar_visual(churn$voice.mail.plan,churn$Churn,"voicemail plan","churn scale",
           "voicemail subscription Vs Churning percentages")
legend("topright",legend=unique(churn$Churn),fill=c(6,5))
```

R - Code for Fig 2.7 & Fig 2.8 (customer service calls vs churn)

```
#customer service calls
boxplot(churn$number.customer.service.calls,ylab="customer service calls")
title("Pseudo outliers in customer service calls")
ggplot(churn,aes(x=Churn,y=number.customer.service.calls))+
  theme_bw()+geom_boxplot()+labs(title="customer service calls vs Churn")
```

Python code for correlation plot (Fig 3.1)

```
df_corr=churn.loc[:,churn_numeric]
df_corr=df_corr.corr().abs()
df_corr=pd.DataFrame(np.triu(df_corr,k=1),columns=df_corr.columns,index=df_corr.index)
pl=sns.diverging_palette(10,220,as_cmap=True)
sns.heatmap(df_corr,cmap=pl,square=True,vmin=-0.6)
```

R code for figure 4.1 (decision tree for classification)

```
#DECISON TREE VISUALISATION
library(rpart)
library(rattle)
churn_tree=rpart(Churn~.,data=churn,method='class')
fancyRpartPlot(churn_tree,cex=0.4)
```

Python Code for Models:

```
#Logistic Regression
logit_churn=sm.Logit(ytrain_logit,xtrain_logit).fit()
logit_predictions=pd.DataFrame(logit_churn.predict(xtest_logit),columns=['probs'])
logit_predictions['value']=1
logit_predictions.loc[logit_predictions.probs<0.5,'value']=0
```

```
#Naive Bayes
NB_churn=GaussianNB().fit(xtrain,ytrain)
NB_predictions=NB_churn.predict(xtest)
```

```
#Decision Tree
DT_churn=tree.DecisionTreeClassifier(criterion='entropy').fit(xtrain,ytrain)
DT_predictions=DT_churn.predict(xtest)
```

```
#Random Forest
RF_churn=RandomForestClassifier().fit(xtrain,ytrain)
RF_predictions=RF_churn.predict(xtest)
```

Python Function written to calculate Accuracy and FNR:

```
# Performance metrics
def perf(true_val,predict_val):
    cm=(pd.crosstab(true_val,predict_val))
    cm1=np.array(cm)
    tn= cm1[0,0]
    fp=cm1[0,1]
    fn=cm1[1,0]
    tp=cm1[1,1]
    accuracy=(tn+tp)/(tn+fp+fn+tp)
    sensitivity=abs(1-(fn/(fn+tp)))
    print(cm)
    print("Accuracy:",round(accuracy*100,ndigits=2),"%")
    print("sensitivity:",round(sensitivity*100,ndigits=2),"%")
```

Accuracy and FNR of different Models:

```
#Logistic_regression
perf(ytest,logit_predictions['value'])
```

```
value     0   1
Churn
0      1392  51
1       167  57
Accuracy: 86.92 %
sensitivity: 25.45 %
```

```
#Naive Bayes
perf(ytest,NB_predictions)
```

```
col_0   0.0  1.0
Churn
0      1303  140
1       103  121
Accuracy: 85.42 %
sensitivity: 54.02 %
```

```
#Decision Tree
perf(ytest,DT_predictions)
```

```
col_0   0.0  1.0
Churn
0      1253  190
1        94  130
Accuracy: 82.96 %
sensitivity: 58.04 %
```

```
#Random Forest
perf(ytest,RF_predictions)
```

```
col_0   0.0  1.0
Churn
0      1345   98
1        95  129
Accuracy: 88.42 %
sensitivity: 57.59 %
```

In Python, Random Forest turned out to be the best model after rectifying class imbalance

```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
under=RandomUnderSampler(sampling_strategy=0.25, return_indices=False, random_state=None, replacement=False)
xunder,yunder=under.fit_resample(xtrain,ytrain)
over=RandomOverSampler(sampling_strategy='minority', return_indices=False, random_state=None, ratio=None)
xover,yover=over.fit_resample(xunder,yunder)
```

```
RF_churn_bl=RandomForestClassifier().fit(xover,yover)
RF_predictions_bl=RF_churn_bl.predict(xtest)
perf(ytest,RF_predictions_bl)
```

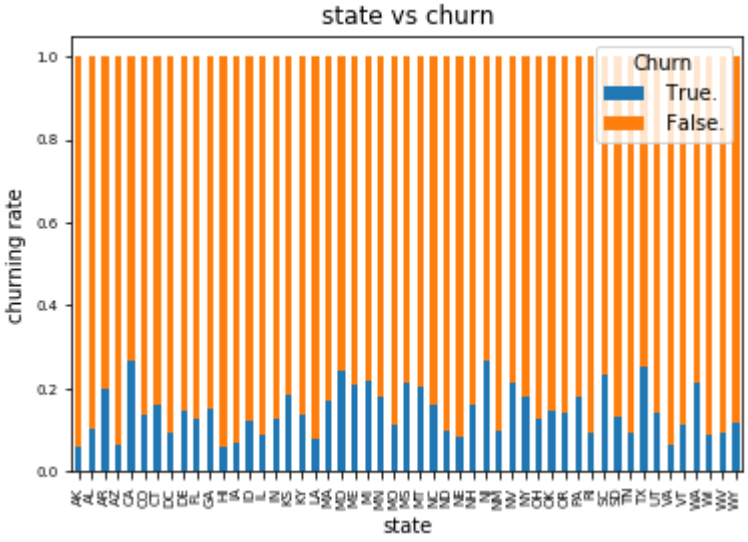
```
col_0   0.0  1.0
Churn
0      1301  142
1        67  157
Accuracy: 87.46 %
sensitivity: 70.09 %
```

APPENDIX – B

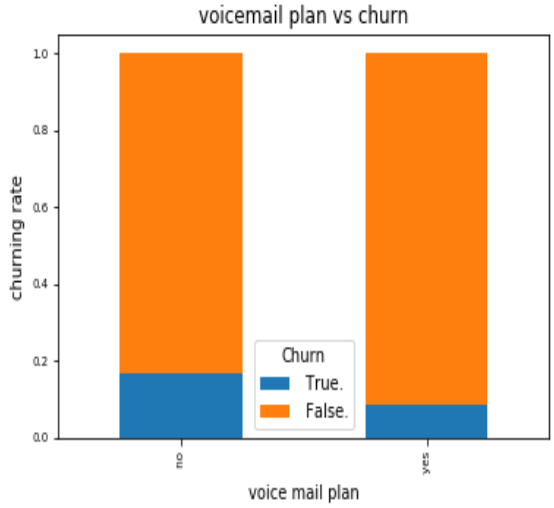
Exploratory Data Analysis –Python:

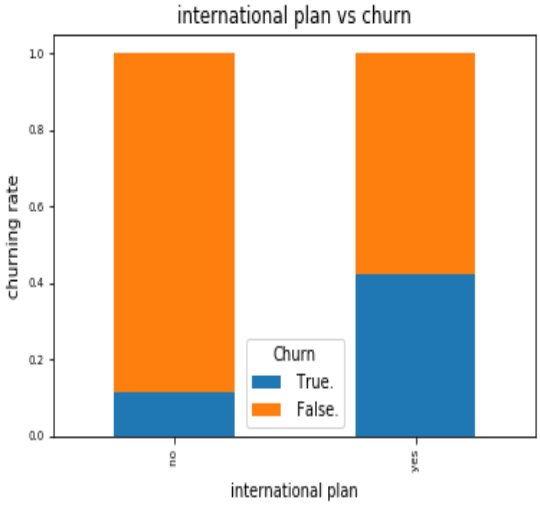
Function for plotting categorical variables:

```
def bar_visual(categ_a,categ_b,heading):
    tab=pd.crosstab(churn[categ_a],churn[categ_b])
    c=tab.iloc[:,0]+tab.iloc[:,1]
    tab.iloc[:,0]=tab.iloc[:,0]/c
    tab.iloc[:,1]=tab.iloc[:,1]/c
    tab=tab.iloc[:,[1,0]]
    t=tab.plot.bar(stacked=True,fontsize=7)
    t.set_ylabel('churning rate')
    t.set_title(heading)
    return t
```

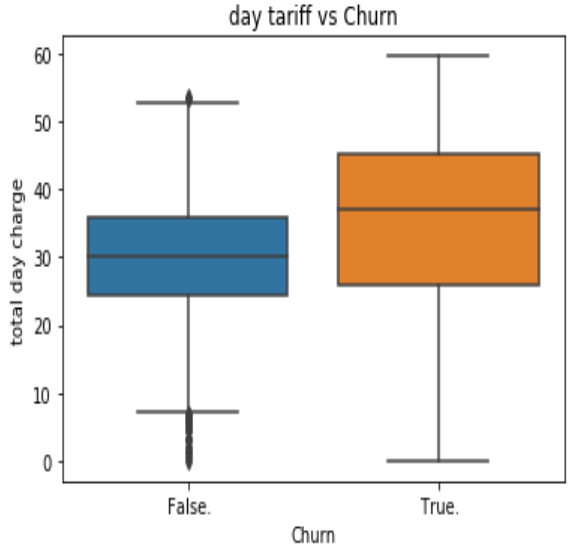


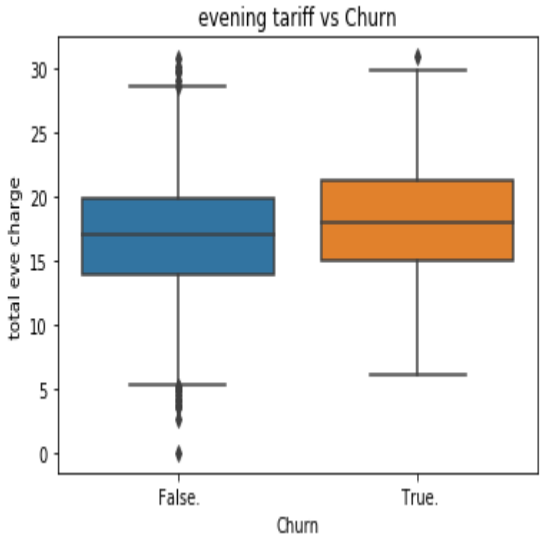

```
bar_visual('state','Churn','state vs churn')
```

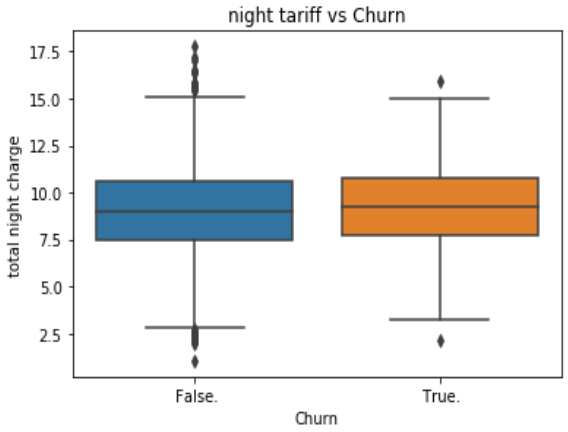




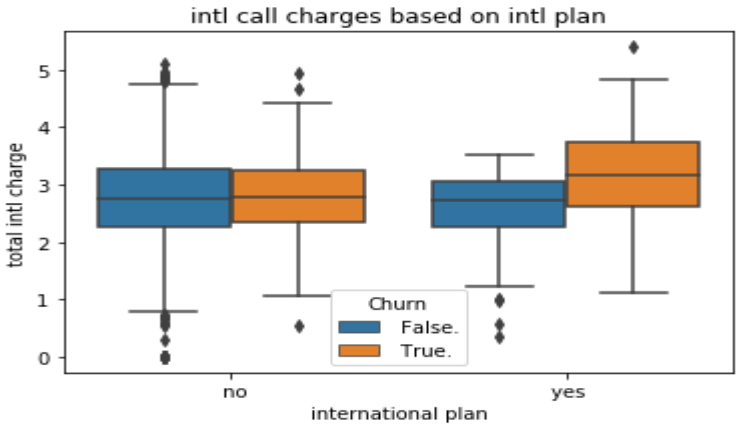
```
bar_visual('voice mail plan','Churn','voicemail plan vs churn')
bar_visual('international plan','Churn','international plan vs churn')
```



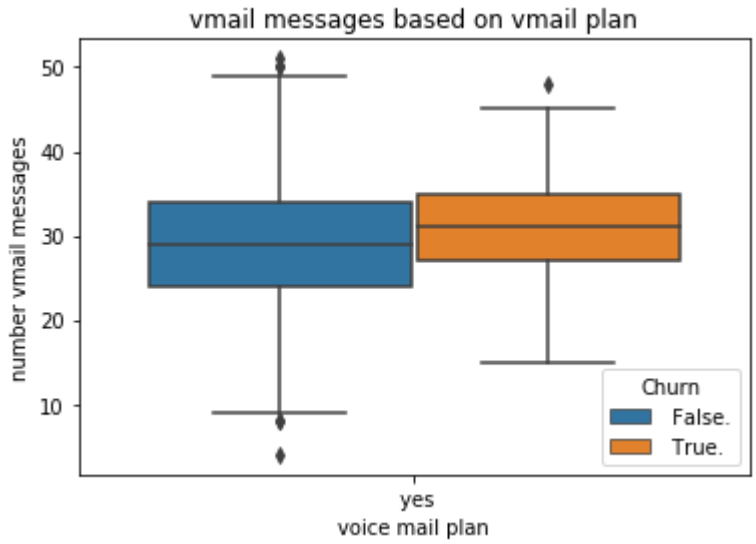




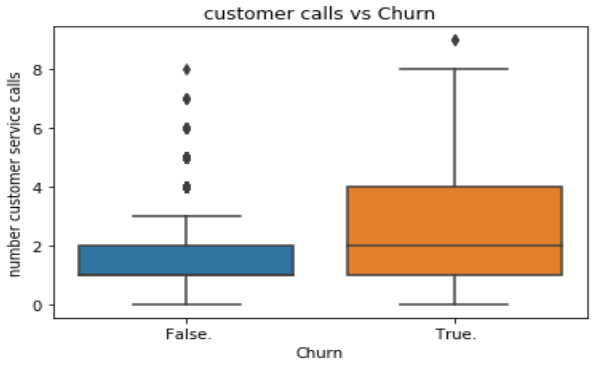

```
sns.boxplot(churn['Churn'],churn['total day charge'])
plt.title("day tariff vs Churn")
sns.boxplot(churn['Churn'],churn['total eve charge'])
plt.title("evening tariff vs Churn")
sns.boxplot(churn['Churn'],churn['total night charge'])
plt.title("night tariff vs Churn")
```



```
sns.boxplot('international plan','total intl charge',hue='Churn',data=churn)
plt.title("intl call charges based on intl plan")
```




```
sns.boxplot('voice mail plan','number vmail messages',hue='Churn',data=a)
plt.title("vmail messages based on vmail plan")
```




```
sns.boxplot(churn['Churn'],churn['number customer service calls'])
plt.title("customer calls vs Churn")
```