Churn Reduction

Manjunath Nagendra

11 December 2018

Contents

Chapter 1

Introduction

1.1 Problem Statement

Churn (loss of customers to competition) is a problem for companies because it is more expensive to acquire a new customer than to keep your existing one from leaving. Customer retention or moving are based on the customer usage pattern/behavior. The objective of this project is to develop an algorithm to predict customer behavior whether customer will churn out or not depending on the customer usage pattern.

1.2 Data

Our task is to build classification model which will classify whether the customer will move out or not depending on his/her usage patterns. Given below is a sample of the data set that has customer usage pattern and if the customer has moved out or not.

Table 1.1: Churn Sample Data (Columns: 1-10)

state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge
KS	128	415	382-4657	no	yes	25	265.1	110	45.07
OH	107	415	371-7191	no	yes	26	161.6	123	27.47
NJ	137	415	358-1921	no	no	0	243.4	114	41.38
OH	84	408	375-9999	yes	no	0	299.4	71	50.9
OK	75	415	330-6626	yes	no	0	166.7	113	28.34
AL	118	510	391-8027	yes	no	0	223.4	98	37.98
MA	121	510	355-9993	no	yes	24	218.2	88	37.09
MO	147	415	329-9001	yes	no	0	157	79	26.69
LA	117	408	335-4719	no	no	0	184.5	97	31.37
WV	141	415	330-8173	yes	yes	37	258.6	84	43.96
IN	65	415	329-6603	no	no	0	129.1	137	21.95
RI	74	415	344-9403	no	no	0	187.7	127	31.91
IA	168	408	363-1107	no	no	0	128.8	96	21.9

Table 1.2: Churn Sample Data (Columns: 11-21)

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	Churn
197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	False.
195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	False.
121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	False.
61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	False.
148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	False.
220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	False.
348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	False.
103.1	94	8.76	211.8	96	9.53	7.1	6	1.92	0	False.
351.6	80	29.89	215.8	90	9.71	8.7	4	2.35	1	False.
222	111	18.87	326.4	97	14.69	11.2	5	3.02	0	False.
228.5	83	19.42	208.8	111	9.4	12.7	6	3.43	4	True.
163.4	148	13.89	196	94	8.82	9.1	5	2.46	0	False.
104.9	71	8.92	141.1	128	6.35	11.2	2	3.02	1	False.

The predictors provided are as follows:

Predictor
account length
international plan
voicemail plan
number of voicemail messages
total day minutes used
day calls made
total day charge
total evening minutes
total evening calls
total evening charge
total night minutes
total night calls
total night charge
total international minutes used
total international calls made
total international charge
number of customer service calls made

Target Variable: Churn: If the customer has moved (1=True; 0 = False)

Chapter 2

Methodology

2.1 Pre Processing

Before we even process the data or build a model above it, we need to have a good understanding about our dataset. Initial exploration of data set is the key step to building a good model. Initial data exploration involves cleaning of data like removing or replacing missing values, discovering few patterns to maximize insight, understanding the dimensions and visualization through plots and graphs. This whole process is termed as Exploratory Data Analysis.

To start with let's visualize the distribution of target variable and probability distribution of all other independent variables. In Figure 2.1 we have plotted the count distribution of the target variable 'Churn'. From the plot we can make out there are more loyal customers in the data set than customers who moved out. Figure illustrates the imbalance in target variable distribution. In Figure 2.2 We have plotted distribution plot with kernel density function. With a glance we can make out most of the numerical predictors are normally distributed.

Initial inspection with statistical technique illustrates there are no missing values in the dataset and data types of all the predictors (Refer Table 2.1). Initially we have 16 continuous independent variables.

Table 2.1: Churn dataset description

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
state                           3333 non-null object
account length                  3333 non-null int64
area code                       3333 non-null int64
phone number                    3333 non-null object
international plan              3333 non-null object
voice mail plan                 3333 non-null object
number vmail messages           3333 non-null int64
total day minutes               3333 non-null float64
total day calls                 3333 non-null int64
total day charge                3333 non-null float64
total eve minutes               3333 non-null float64
total eve calls                 3333 non-null int64
total eve charge                3333 non-null float64
total night minutes             3333 non-null float64
total night calls               3333 non-null int64
total night charge              3333 non-null float64
total intl minutes              3333 non-null float64
total intl calls                3333 non-null int64
total intl charge               3333 non-null float64
number customer service calls   3333 non-null int64
Churn                           3333 non-null object
dtypes: float64(8), int64(8), object(5)
```

From Table 2.2, we discovered that *phone number* column has all unique values which won't add any value to the target variable. Hence dropped the respective column during the initial exploration.

Table 2.2: Churn Dataset Object Variable Statistics

	state	phone number	international plan	voice mail plan	
count	3333	3333	3333	3333	3333
unique	51	3333	2	2	2
top	WV	393-4949	no	no	False.
freq	106	1	3010	2411	2850

2.1.1 Outlier Analysis

From Figure 2.2 we can observe that most of the numerical predictors or independent variables are normally distributed except few are skewed like *area code, number customer service calls, number vmail messages, total intl calls*. Figure 2.3 illustrates that the variables like *area code, number customer service calls* falls under small number of categories. So we will categorize them into respective buckets.

Next step is to check for outliers. We will use boxplot to visualize outliers for variables as illustrated in Figure 2.4. From the Figure 2.4 we can make out every continuous variable has outliers. Once outliers are detected, those values are replace by null/NaN. These Missing values are filled with Knn-imputation.

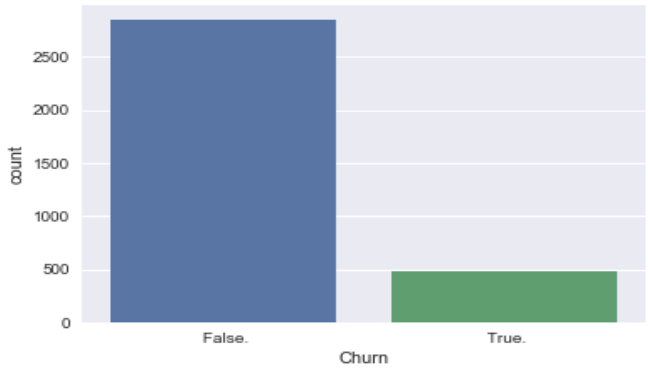
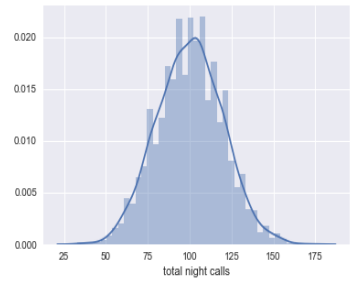
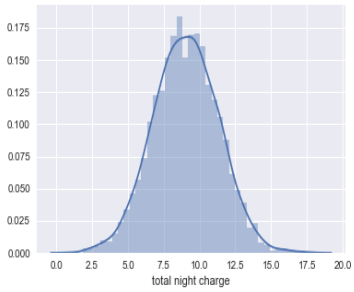
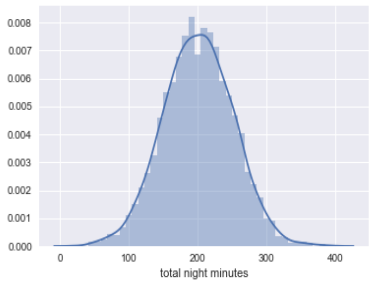
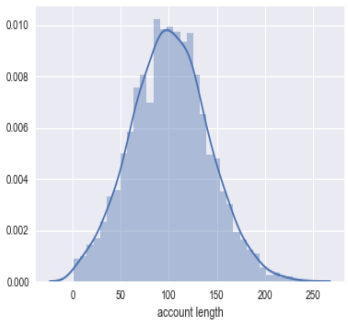


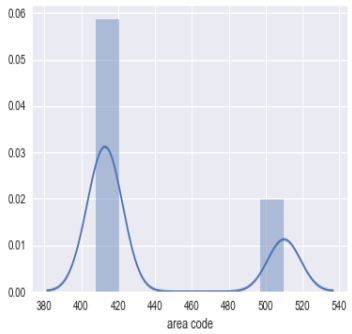
Figure 2.1: Frequency Distribution of target variable Churn


total night calls

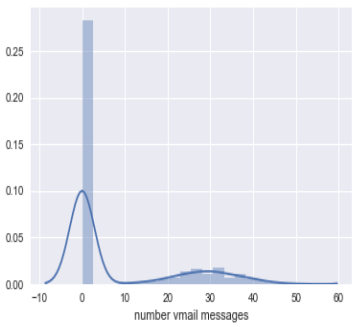

total night charge

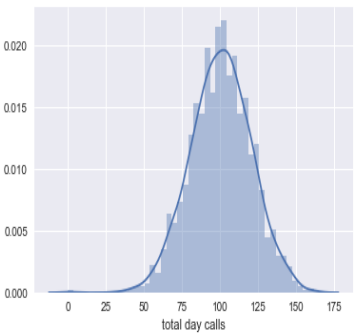

total night minutes

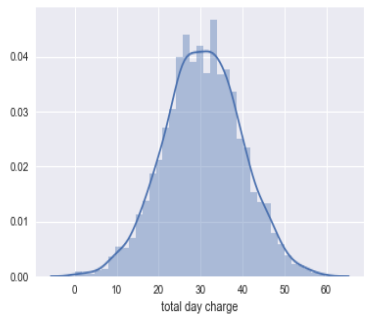

account length

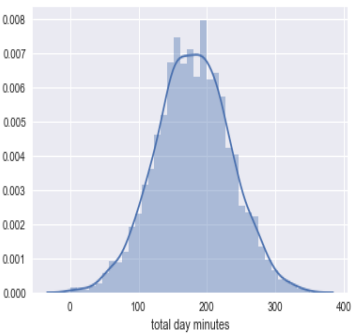

area code

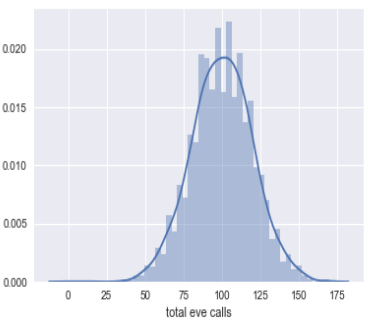

number customer service calls

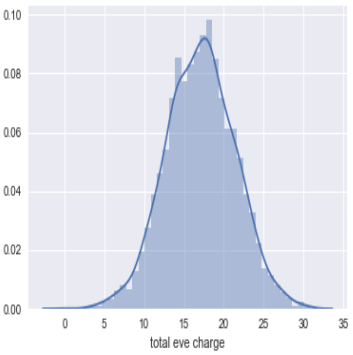

number vmail messages


total day calls


total day charge


total day minutes


total eve calls


total eve charge

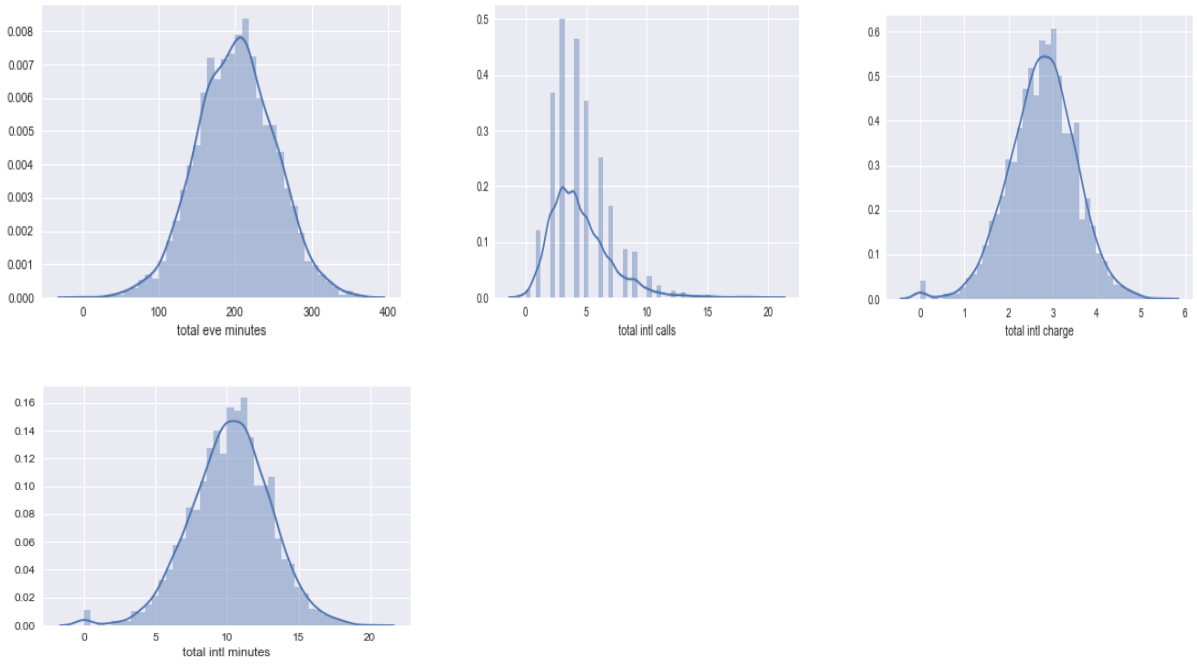


Figure 2.2: Disptribution plot with kernel desity of Churn Reduction Dataset

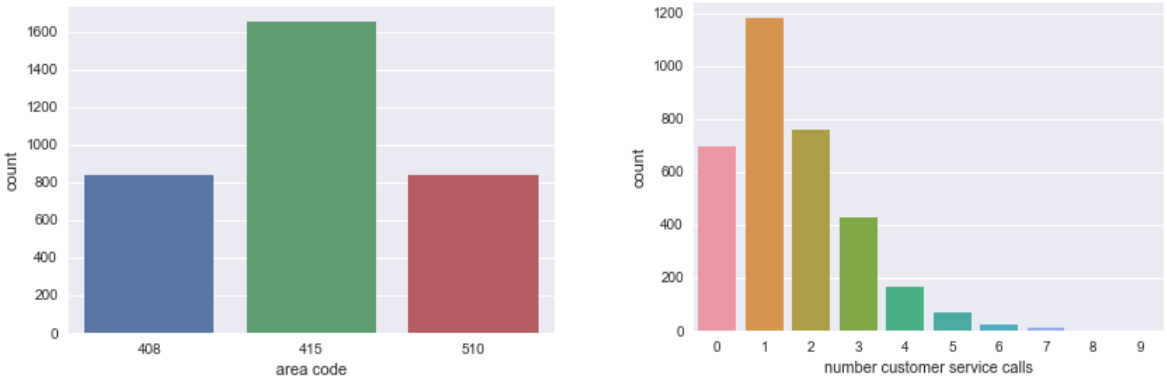
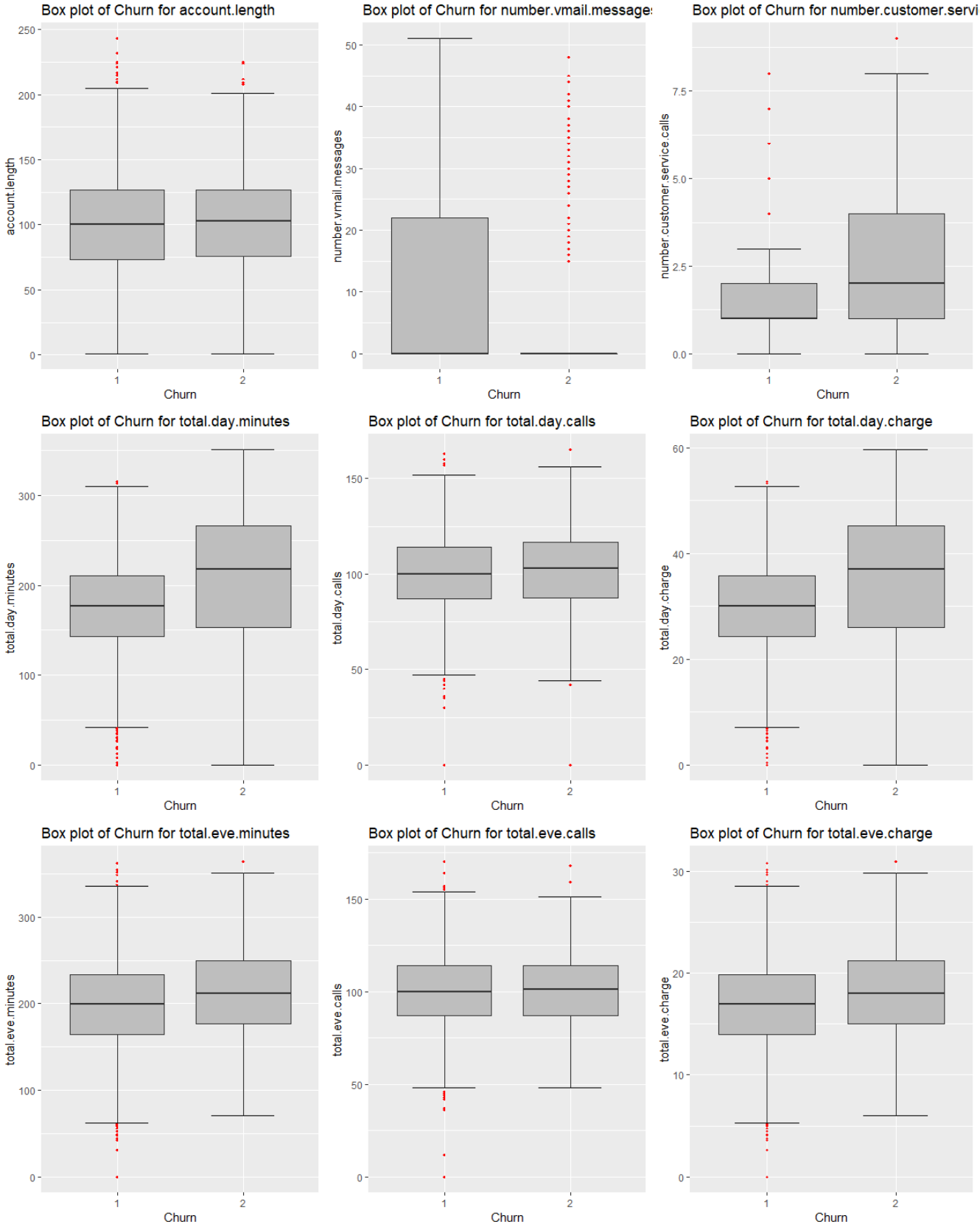


Figure 2.3: Count Distribution of *area code* and *number of customer service calls*


Box plot of Churn for account.length
account.length
Churn
Box plot of Churn for number.vmail.messages
number.vmail.messages
Churn
Box plot of Churn for number.customer.servi
number.customer.service.calls
Churn
Box plot of Churn for total.day.minutes
total.day.minutes
Churn
Box plot of Churn for total.day.calls
total.day.calls
Churn
Box plot of Churn for total.day.charge
total.day.charge
Churn
Box plot of Churn for total.eve.minutes
total.eve.minutes
Churn
Box plot of Churn for total.eve.calls
total.eve.calls
Churn
Box plot of Churn for total.eve.charge
total.eve.charge
Churn

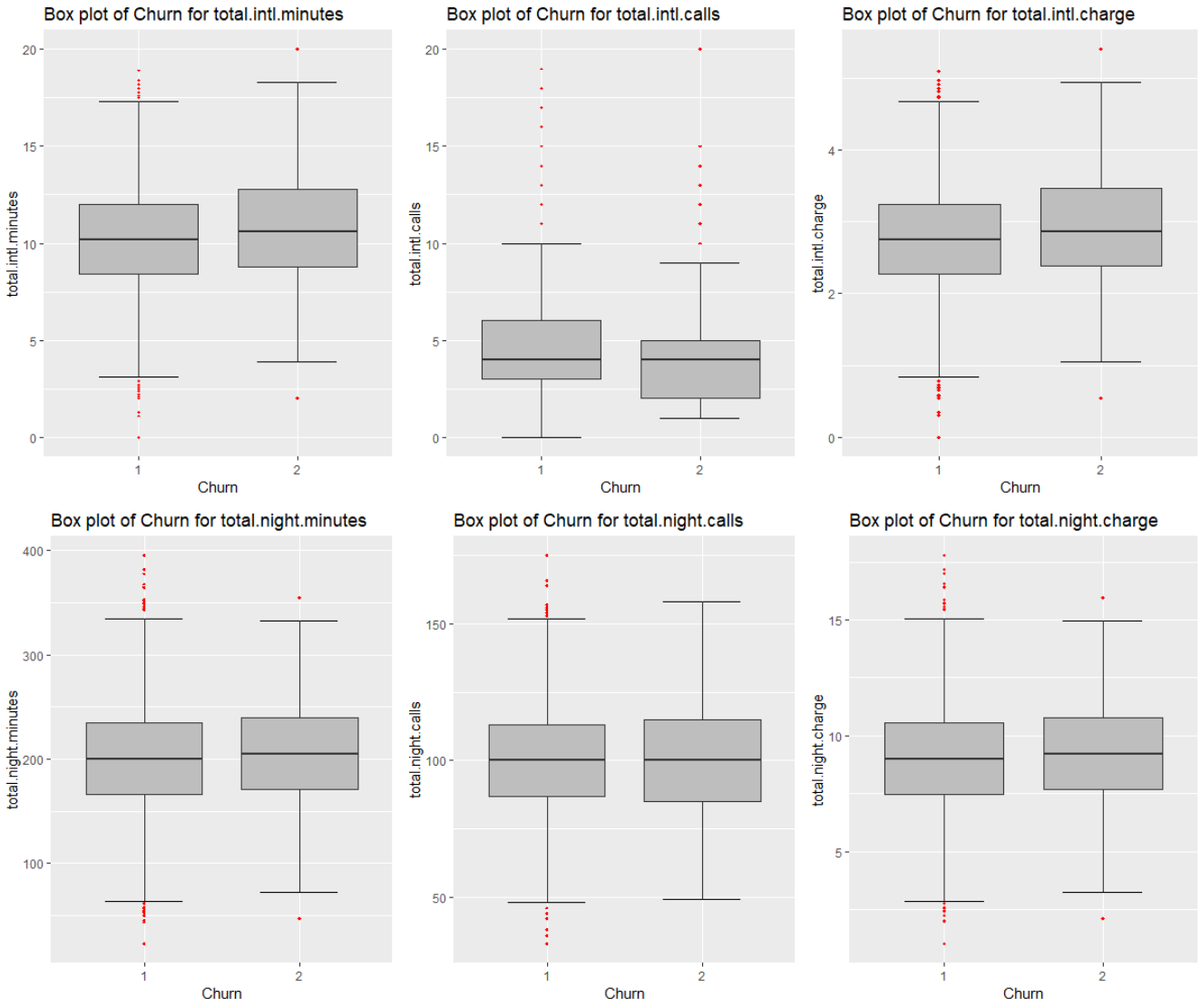


Figure 2.4: Boxplot for each predictor

2.1.2 Feature Selection

Before we get into model building, we need to assess the importance of each predictor or independent variables. Then select only relevant variables needed for model construction by eliminating other variables which does not add any value for our classification problem.

The methods used here to do Feature Selection are classified as:

a. **Correlation Analysis**. For all the continuous variable we will build a head map to inspect which variables are highly correlated to each other. We will use corr() method to calculate the correlation between each continuous variables and pass those values to head map to visualize them as in Figure 2.5.
b. **Chi-Square test of independence**: For all the categorical variables we will use chi-square test to test relationship between 2 categorical variables. From chi-square test, we make 2 hypothesis one is *Null Hypothesis*: 2 variables (one of them is target variable) are independent variables and other is *Alternate hypothesis:* 2 variables (one of them is target variable) are not independent variables. Variables which has p-value above 0.05 will be dropped for model construction further.

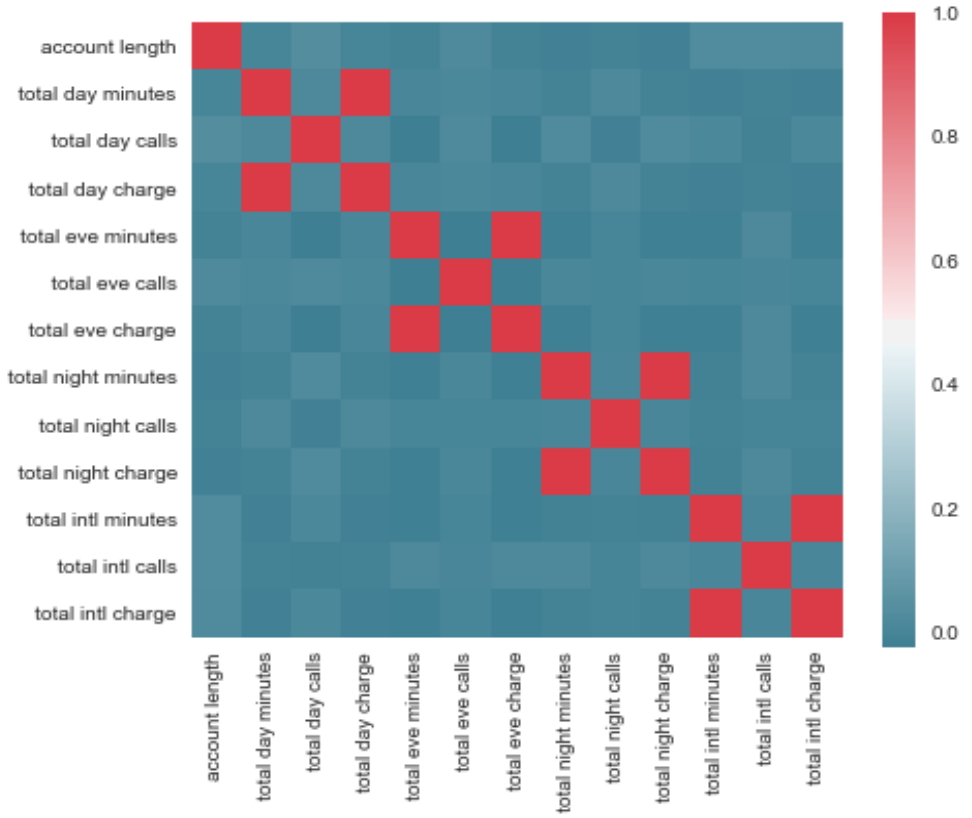


Figure 2.5: Heat Map to show Correlation Matrix of Continuous Variables

From the Figure 2.5, we can see that there are 4 variables which are highly correlated like *total night charge, total day charge, total eve charge, total intl charge* where they are directly calculated from *total night minutes, total day minutes, total eve minutes, total intl minutes* respectively. So we can drop any one of the correlated pair for further model building. We will consider variables [*'total day charge', 'total eve charge', 'total night charge', 'total intl charge'*] to be dropped as they won't add any added value.

From Chi-Square test we will drop *area code* as it has p value = 0.915055 > 0.05 by rejecting the alternate Hypothesis mentioned above indicating that the 2 variables are dependent on each other.

So, overall variables considered to be dropped from the Churn Reduction dataset are

[*'total day charge', 'total eve charge', 'total night charge', 'total intl charge', area code*]

2.1.3 Feature Scaling

Another aspect under Pre-Processing is to normalize the data in the dataset before moving to model building phase so that the data are in proportion with one another. Feature Scaling can be done only on continuous variables. So before we deal with continuous variables, let us deal with categorical variables which could be normalized. If we observe from Table 2.2, we observe that *state* has 51 unique values. So we will map these states to a specific number from a range. As we discussed earlier, categorical variables which could be categorized to small number of buckets like *number customer service calls, number vmail messages* could be scaled between a small numbers range. WE have binomial variables like *'international plan', 'voice mail plan', 'Churn'* which could be directly converted into 0 and 1. With this technique we are converting categorical variables to numbers which we can feed to model development stage. We can do Feature Scaling on continuous variables by 2 methods:

a. Normalization - convert all categorical variable values between 0 and 1
b. Standardization (Z-score) – Standardization can be applied on continuous variables only when they are normally distributed.

From Figure 2.2, though we observe most of the continuous variables are normally distributed, here we will use normalization technique for Feature Scaling.

After Data Pre-Processing, Dataset will be

a. Free of missing values.
b. Outliers will be replaced with values within variable/feature range.
c. Only those features/variables are selected which add value for predicting target variable.
d. All variables values are normalized/standardized.

After this stage, dataset is ready to build Machine Learning/Statistical model on them to predict our target variable.

2.2 Modeling

2.2.1 Model Selection

Since our target variable here is binomial in nature, we are dealing with classification problem here. So we can use below algorithms to build a classification model on our pre-processed dataset and predict our target variable whether customer *churned* out or not.

a. Decision Tree Classifier
b. Random Forest Classifier
c. Logistic Regression
d. KNN
e. Naïve Bayes

Let's go one by one and check which algorithm performs better on this dataset to predict our target variable. Performance of above mentioned algorithms will be calculated with help of *Error Metrics* which we will discuss later.

2.2.1.1 Decision Tree Classifier

As Decision Tree is a predictive model based on series of Boolean test, can be used for classification which uses concept of *Information Gain* i.e variables with highest *Information gain* is used to spilt at root node first.

2.2.1.2 Random Forest Classifier

Random Forest is an ensemble technique which uses n-Decision Trees to predict target variable which works on the concept of bagging. It uses *gini-index* to select parent node and split from randomly selected small set of variables (sqrt (total independent variable)).Here I have used 500 trees. From the n-trees generated, new test case will traverse tree according to the condition it carries and displays relevant output.

2.2.1.3 Logistic Regression

Logistic Regression is used only for classification purpose. Here regression coefficient for independent categorical variable is calculated for each category.

2.2.1.4 KNN

Knn works on nearest neighbor concept using distance method. It calculates distance between each test case with all training cases. Here depending on value of *n (number of neighbors)* results might differ, so it is important to use the right value of *n* for Knn. I created a plot Figure 2.6 to choose the right value of *n* according to which I use n to be 13.

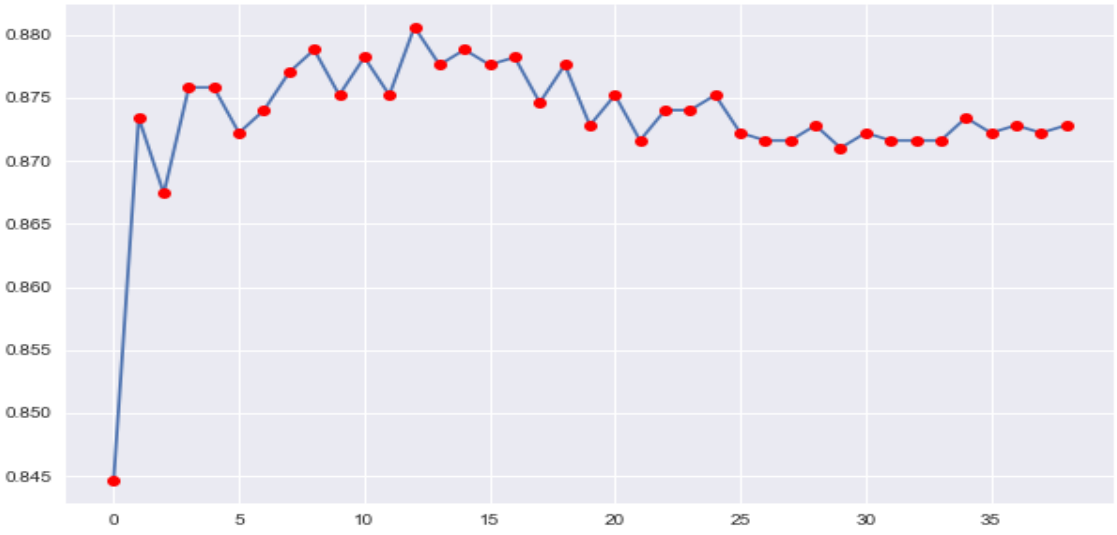


Figure 2.6: Plot to select right value for n in KNN implementation

2.2.1.5 Naïve Bayes

Naïve Bayes uses concept of Probabilistic Classification.

Chapter 3

Conclusion

3.1 Model Evaluation

Since we have few models now to predict our target variable, we need to decide which one to choose. There are several criteria exist for evaluating and comparing model performance, since our models are based on classification let's use classification metrics. We will compare our model using below 2 criteria:

a. Accuracy
b. False Negative Rate

3.1.1 **Accuracy**: This metric is important to check the overall performance of the model i.e. how well the model predicts the correct decision.

Accuracy = TP + TN / Total observations

3.1.2 **False Negative Rate**: With Accuracy, FNR is important in this project because here FNR refers to customers who are actually to churn out but we predicted as they won't, by this we are actually missing out those customers. So this metric becomes important in this project.

FNR = FN/ FN+TP

So let's see how our models performed with these two metrics considered

Note: I have completed building model for this project both in R and Python. But I am publishing scores here in this report of only the best though difference between both are very minimal along with statistics from R.

Decision Tree Classifier

```
4   Confusion Matrix and Statistics
5
6      c50_Predictions
7          1    2
8     1 1435    8
9     2   67  157
10
11                Accuracy : 0.955
12                  95% CI : (0.9439, 0.9645)
```

```
      No Information Rate : 0.901
      P-Value [Acc > NIR] : 2.288e-16

                    Kappa : 0.7824
   Mcnemar's Test P-Value : 2.124e-11

               Sensitivity : 0.9554
               Specificity : 0.9515
            Pos Pred Value : 0.9945
            Neg Pred Value : 0.7009
                Prevalence : 0.9010
            Detection Rate : 0.8608
      Detection Prevalence : 0.8656
         Balanced Accuracy : 0.9535

          'Positive' Class : 1
```

Accuracy: 95.5%

FNR: 29.91%

Random Forest Classifier

```
    Confusion Matrix and Statistics

       RF_Predictions
           1    2
      1 1438    5
      2   82  142

                   Accuracy : 0.9478
                     95% CI : (0.936, 0.958)
       No Information Rate : 0.9118
       P-Value [Acc > NIR] : 1.789e-08

                      Kappa : 0.7376
    Mcnemar's Test P-Value : 3.698e-16

                Sensitivity : 0.9461
                Specificity : 0.9660
             Pos Pred Value : 0.9965
             Neg Pred Value : 0.6339
                 Prevalence : 0.9118
             Detection Rate : 0.8626
      Detection Prevalence : 0.8656
         Balanced Accuracy : 0.9560

          'Positive' Class : 1
```

Accuracy: 94.78%

FNR: 36.6%

Logistic Regression

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1285  -0.5191  -0.3514  -0.2139   3.2603

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -6.45915    0.49725 -12.990  < 2e-16 ***
state                         0.06937    0.18721   0.371 0.710953
account.length                0.14270    0.29189   0.489 0.624931
international.plan2           2.05510    0.14379  14.292  < 2e-16 ***
voice.mail.plan2             -1.61465    0.40232  -4.013 5.99e-05 ***
number.vmail.messages         1.48501    0.80229   1.851 0.064176 .
total.day.minutes             3.35314    0.31496  10.646  < 2e-16 ***
total.day.calls               0.42141    0.29513   1.428 0.153335
total.eve.minutes             1.75849    0.31437   5.594 2.22e-08 ***
total.eve.calls              -0.01748    0.30482  -0.057 0.954278
total.night.minutes           1.14193    0.30914   3.694 0.000221 ***
total.night.calls             0.11416    0.30030   0.380 0.703836
total.intl.minutes            0.96610    0.30387   3.179 0.001476 **
total.intl.calls             -1.19902    0.27626  -4.340 1.42e-05 ***
number.customer.service.calls 4.55080    0.34941  13.024  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2758.3  on 3332  degrees of freedom
Residual deviance: 2200.4  on 3318  degrees of freedom
AIC: 2230.4

Number of Fisher Scoring iterations: 5
```

Accuracy: 87.52%

FNR: 80%

KNN

```
2    Confusion Matrix and Statistics
3
4
5    KNN_Predictions    1    2
6                   1 1430  179
7                   2   13   45
8
9                    Accuracy : 0.8848
10                     95% CI : (0.8685, 0.8998)
11       No Information Rate : 0.8656
12       P-Value [Acc > NIR] : 0.0107
```

```
13
14                    Kappa : 0.2793
15    Mcnemar's Test P-Value : <2e-16
16
17               Sensitivity : 0.9910
18               Specificity : 0.2009
19            Pos Pred Value : 0.8888
20            Neg Pred Value : 0.7759
21                Prevalence : 0.8656
22            Detection Rate : 0.8578
23      Detection Prevalence : 0.9652
24         Balanced Accuracy : 0.5959
25
26          'Positive' Class : 1
```

Accuracy: 88.48%

FNR: 22.41%

Naïve Bayes

```
Confusion Matrix and Statistics

          predicted
observed    1    2
       1 1406   37
       2  174   50

               Accuracy : 0.8734
                 95% CI : (0.8565, 0.889)
    No Information Rate : 0.9478
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2664
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8899
            Specificity : 0.5747
         Pos Pred Value : 0.9744
         Neg Pred Value : 0.2232
             Prevalence : 0.9478
         Detection Rate : 0.8434
   Detection Prevalence : 0.8656
      Balanced Accuracy : 0.7323

       'Positive' Class : 1
```

Accuracy: 87.34%

FNR: 77.67%

3.2 Model Selection

Considering both Accuracy and False Negative Rate, I am considering Decision Tree for this model.

Using Decision Tree Classifier we can predict if customers churned out or not at an accuracy of 95.5% and false negative rate of 29.91%.