Churn Reduction Project Report

Harshal Rajendra Kothawade

August 1, 2018

Contents

Chapter 1

Introduction

1.1 Project Description

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. This problem statement is targeted at enabling churn reduction using analytics concepts.

1.2 Problem statement

The objective of this Case is to predict customer behaviour. We are providing you a public dataset that has customer usage pattern and if the customer has moved or not. We expect you to develop an algorithm to predict the churn score based on usage pattern.

Target Variable : Churn: if the customer has moved (1=yes; 0 = no)

1.3 Data

Our task is to build classification models which will classify whether customer will move out or not depending on various factors given in the data. Given below is set of predictor variables given to classify the customer churn and the data for churn Reduction Classification.

Table 1.1: Predictor Variables

Sr, No.	Predictor
1	state
2	account length
3	area code
4	phone number
5	international plan
6	voice mail plan
7	number vmail messages
8	total day minutes
9	total day calls
10	total day charge
11	total eve minutes
12	total eve calls
13	total eve charge
14	total night minutes
15	total night calls
16	total night charge
17	total intl minutes
18	total intl calls
19	total intl charge
20	number customer service calls

Table 1.2: Churn Reduction sapmle Data (Columns 1-5)

state	account length	area code	phone number	international plan
HI	101	510	354-8815	no
MT	137	510	381-7211	no
OH	103	408	411-9481	no
NM	99	415	418-9100	no
SC	108	415	413-3643	no
IA	117	415	375-6180	no
ND	63	415	348-8073	no

Table 1.3: Churn Reduction sapmle Data (Columns 6-10)

voice mail plan	number vmail messages	total day minutes	total day calls	total day charge
no	0	70.9	123	12.05
no	0	223.6	86	38.01
yes	29	294.7	95	50.1
no	0	216.8	123	36.86
no	0	197.4	78	33.56
no	0	226.5	85	38.51
yes	32	218.9	124	37.21

Table 1.4: Churn Reduction sapmle Data (Columns 11-16)

total eve minutes	total eve calls	total eve charge	total night minutes	total night calls	total night charge
211.9	73	18.01	236	73	10.62
244.8	139	20.81	94.2	81	4.24
237.3	105	20.17	300.3	127	13.51
126.4	88	10.74	220.6	82	9.93
124	101	10.54	204.5	107	9.2
141.6	68	12.04	223	90	10.04
214.3	125	18.22	260.3	120	11.71

Table 1.5: Churn Reduction sapmle Data (Columns 17-21)

total intl minutes	total intl calls	total intl charge number	customer service calls	Churn
10.6	3	2.86	3	False.
9.5	7	2.57	0	False.
13.7	6	3.7	1	False.
15.7	2	4.24	1	False.
7.7	4	2.08	2	False.
6.9	5	1.86	1	False.
12.9	3	3.48	1	False.

Chapter 2

Methodology

2.1 Pre Processing

Any model requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. We can start with looking at the data types and unique values present in the variables and can make changes in them as per requirement. As we can see that there are only 3 unique values in "area code" variable and 10 unique values in "number customer service calls" variable, we can convert them into the categorical variable from numeric type variable for model simplification. Moreover, to save the memory used we can convert the actual values of categorical variables into the levels like 0, 1, 2... as per number of values present in the variable.

After converting the data as per required types. We can look for distribution of data in particular variable with the help of histogram. We can see from histogram distribution Figure 2.2 that most of the data is not normally distributed. This is because of the outliers present in the data.

2.1.1 Outlier Analysis

We can clearly observe from these probability distributions that most of the variables are skewed, for example, number customer service calls, number vmail messages, total day calls, total intl calls. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data.

One of the other steps of pre-processing apart from checking for normality is the presence of outliers. In this case we use a classic approach of outlier imputation, KNN imputation method. We visualize the outliers using boxplots. In figure 2.3 we have plotted the boxplots of the 14 continuous predictor variables with respect to each churn value. A lot of useful inferences can be made from these plots. As we can see there are lot of outliers and extreme values in each of the data set. First, we will replacing them with NA using boxplot method and then with KNN imputation method, we will impute missing values which will remove most of the outliers from the data set. In Figure 2.6 we have plotted the boxplots for the same 14 variables after outlier removals. As it can be seen clearly the number of of outliers are less and extreme values are removed from the dataset. And now data is normally distributed.

(a) With Outlier

(b) Without Outlier

Figure 2.1: Effect of Outliers

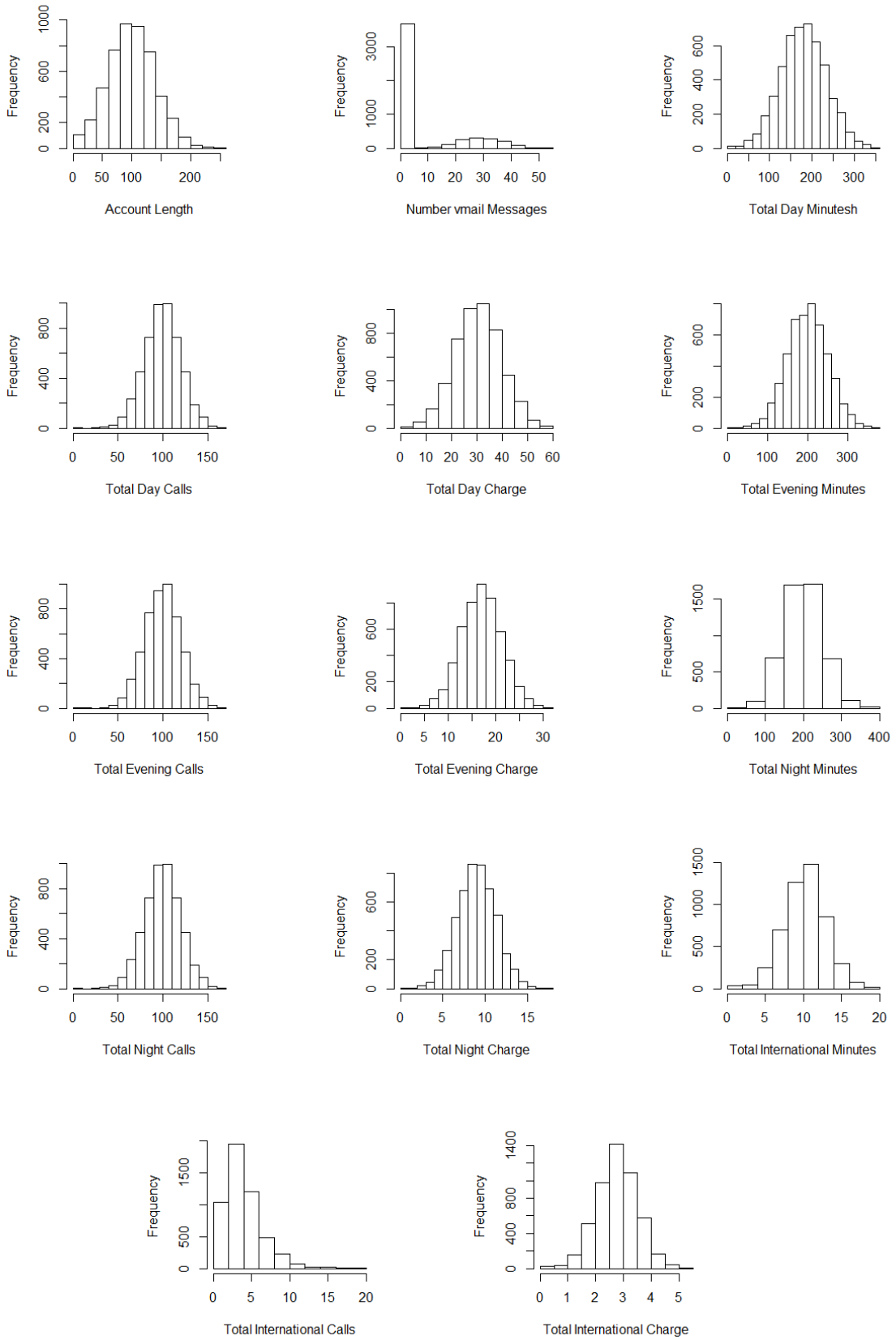


Figure 2.2: Histogram of Variables with Outliers

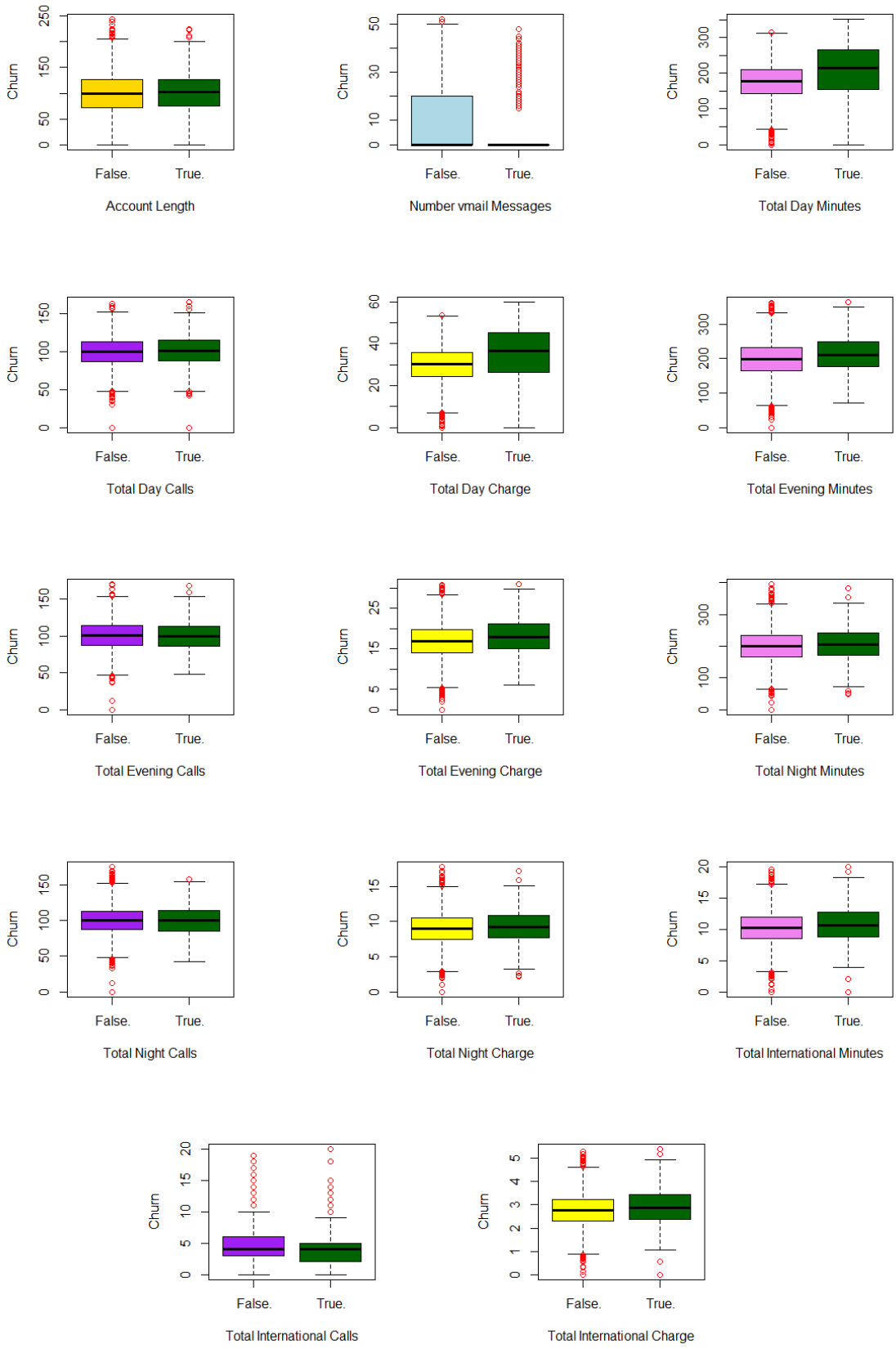


Figure 2.3: BoxPlot of Variables with Outliers

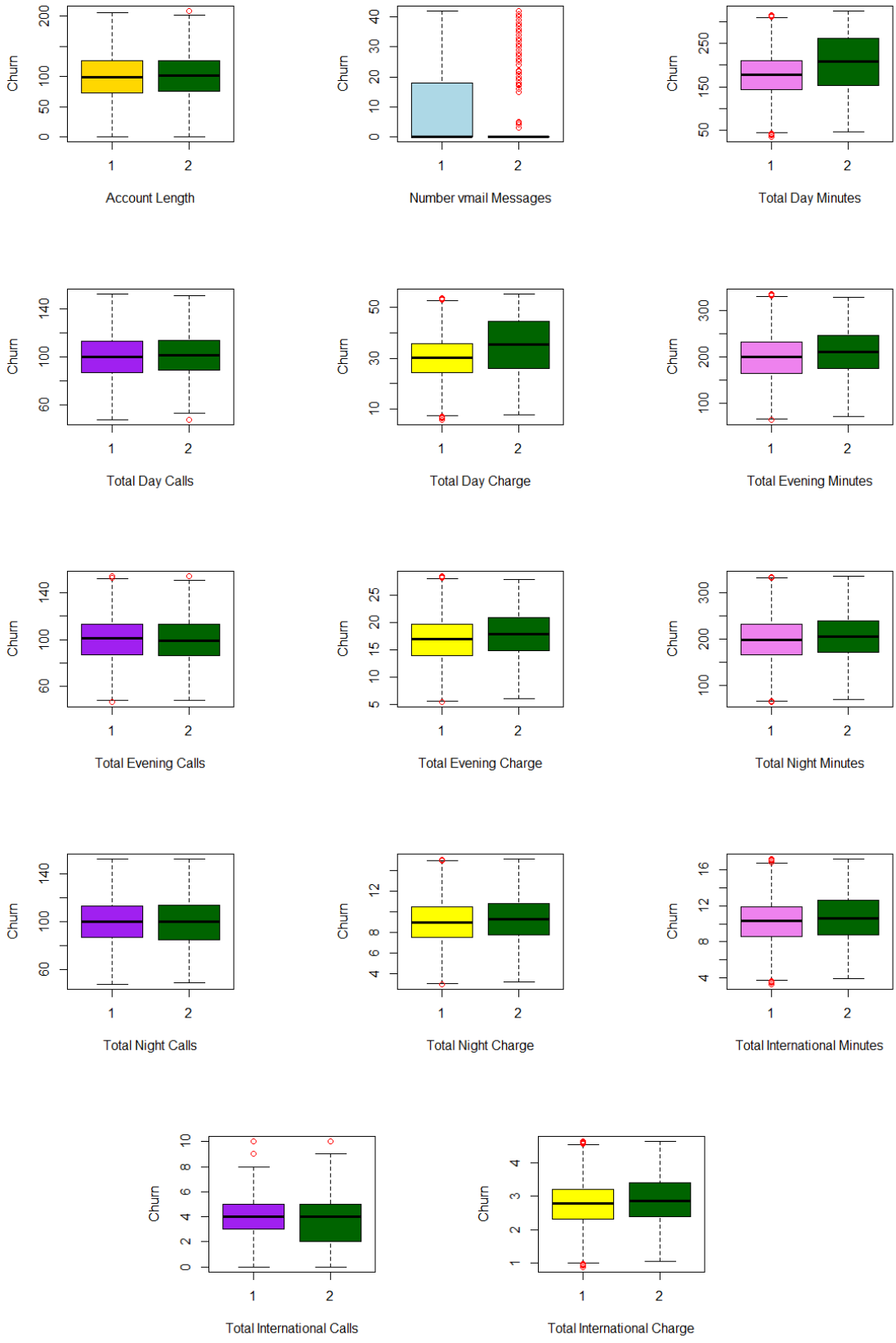
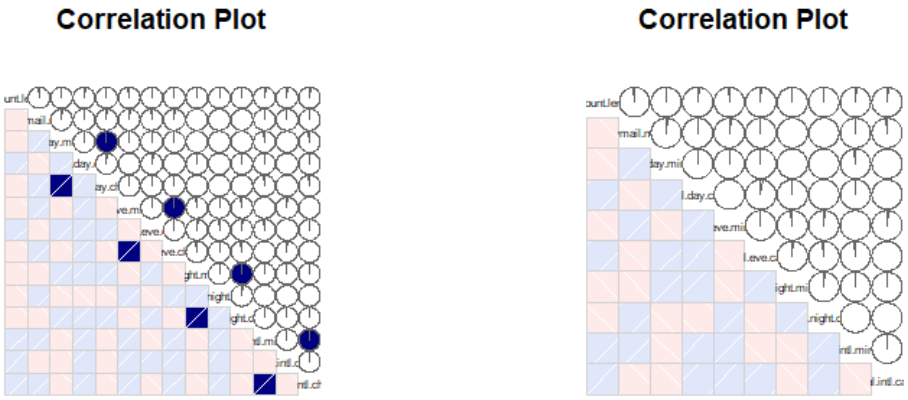


Figure 2.4: Histogram of Variables without Outliers

2.1.2 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of classification. There are several methods of doing that. We have used the correlation analysis for continuous variables and chi-square test for the categorical variables.



(a) Before feature selection

(b) After feature selection

Figure 2.5: Correlation Plot

The Figure 2.5a is correlation plot for Churn Reduction Data, it shows the correlation between two variables. The blue shade colour implies that two variables are positively correlated and orange shade implies negative correlation. As it can be clearly seen from Figure /reffig:bfs, total time variables is highly correlated with total charge as it should be. So, one of this variables can be dropped as they provide similar information and removing one of them wont effect model much.

For categorical variables, we used chi-square test of independence to check the dependence of variable with each other. If p value in chi-square test is less than 0.05, then we accept the variable for model otherwise reject it. The p-value implies dependence of variable on the target variable.

Table 2.1: Chi-Square Test p-values

Varaible	p-value
state	7.851e-05
area code	0.7547
phone number	0.4934
international plan	2.2e-16
voice mail plan	7.165e-15
number customer service calls	2.2e-16

Therefore, following continuous variables can be removed after correlation analysis :

1. Numeric Variables:
   - total day charge
   - total eve charge
   - total night charge
   - total intl charge
2. Categorical Variables:
   - area code
   - phone number

2.1.3 Feature Scaling

From figure 2.2, it is clear that the range of all the variables is not same. Some variables range from 0 to 10 while other range from 0 to 1000. We need to normalise this, so that model should not be more prone towards the high value variables. We can do this either by standardization or normalization. Standardization is more suited for the data which is normally distributed. As for Churn Reduction data is not normally distributed. So we can't use standardization technique, Normalization is more suitable for such data set. After normalization all numeric data values will be between 0 and 1.

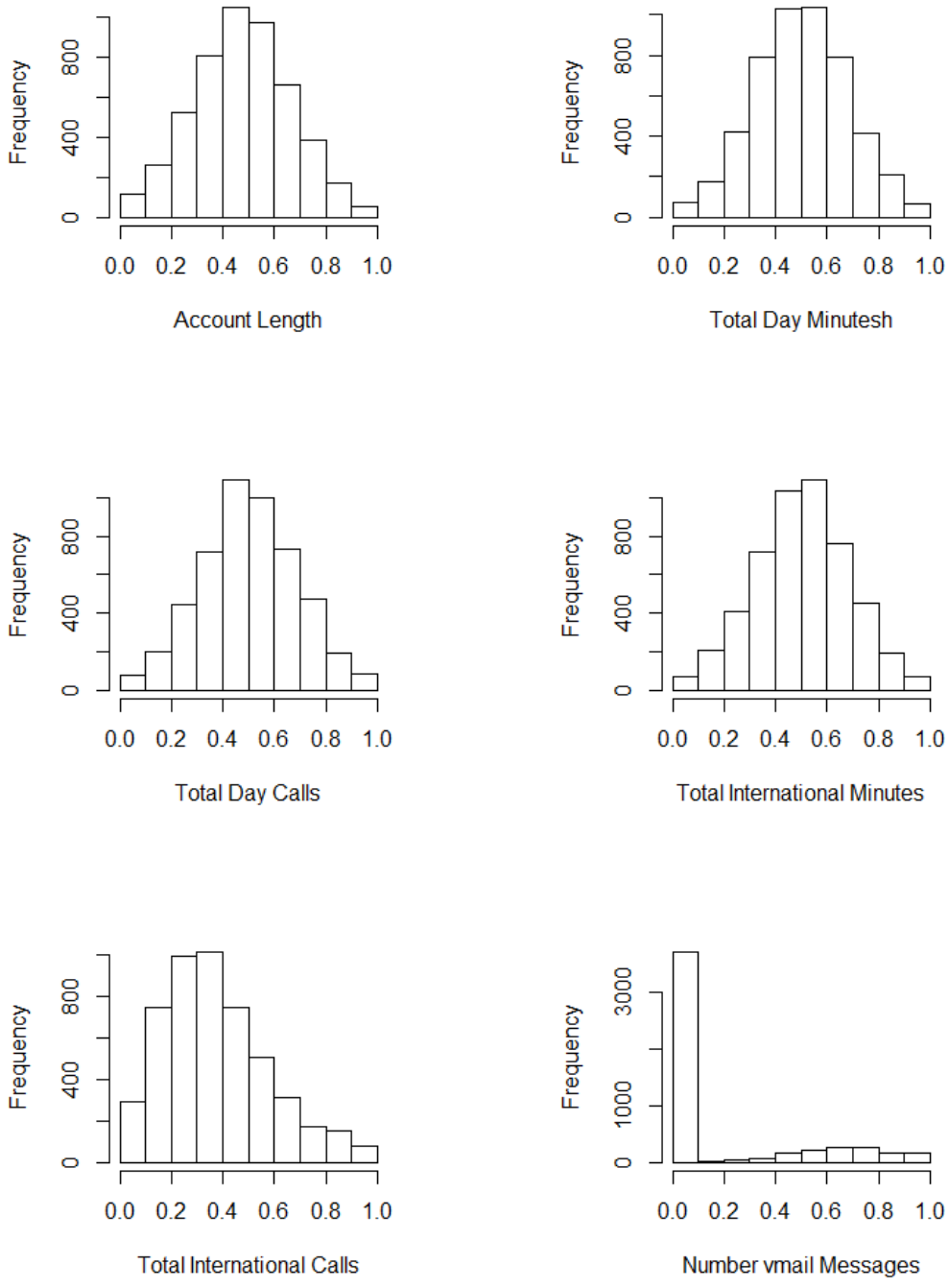


Figure 2.6: Histogram of Normalized Variables

2.2 Modeling

2.2.1 Model Selection

After pre-processing of data, we must proceed with model development. In this case, we have to predict whether customer will churn out or not. So this can be considered as binary classification problem, like if the customer has moved (1=yes; 0 = no). If the target variable is numeric or interval, then we must go for regression model. We can implement following models for data classification

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

2.2.2 Decision Tree

Decision Tree is rule. Each rule connects nodes with 'and' and multiple branches connected with 'or'. Following is the example of rules in Decision Tree with atribute usage values.

```
-----  Trial 99:  -----

Rules:

Rule 99/1: (207.9, lift 1.5)
        international.plan = 1
        voice.mail.plan = 2
        number.customer.service.calls in {1, 2, 3, 4, 8}
        ->  class 1  [0.995]

Rule 99/2: (760.4/7.7, lift 1.5)
        account.length <= 0.9613526
        international.plan = 1
        total.day.minutes <= 0.628453
        number.customer.service.calls in {1, 2, 3, 4, 8}
        ->  class 1  [0.989]

Rule 99/3: (502.2/9.2, lift 1.5)
        account.length <= 0.9613526
        international.plan = 1
        total.day.minutes <= 0.7883287
        total.eve.minutes <= 0.489464
        number.customer.service.calls in {1, 2, 3, 4, 8}
        ->  class 1  [0.980]


  (a)   (b)    <-classified as
 ----  ----
 2846     4    (a): class 1
   84   399    (b): class 2


Attribute usage:

100.00% state
100.00% international.plan
100.00% voice.mail.plan
100.00% total.day.minutes
100.00% total.day.calls
100.00% total.eve.minutes
100.00% total.eve.calls
100.00% total.night.minutes
100.00% total.intl.minutes
100.00% total.intl.calls
100.00% number.customer.service.calls
 99.91% number.vmail.messages
 99.73% total.night.calls
 99.52% account.length
```

2.2.3 Random Forest

```
                Length Class  Mode
call               5   -none- call
type               1   -none- character
predicted       3333   factor numeric
err.rate        1500   -none- numeric
confusion          6   -none- numeric
votes           6666   matrix numeric
oob.times       3333   -none- numeric
classes            2   -none- character
importance        56   -none- numeric
importanceSD      42   -none- numeric
localImportance    0   -none- NULL
proximity          0   -none- NULL
ntree              1   -none- numeric
mtry               1   -none- numeric
forest            14   -none- list
y               3333   factor numeric
test               0   -none- NULL
inbag              0   -none- NULL
terms              3   terms  call
```

2.2.4 KNN

The KNN is K nearest Neighbours algorithm, in this case we used KNN value as 7.

```
> summary(KNN_Predict)
   1    2
1611   56
```

2.2.5 Naïve Bayes

```
> summary(NB_model)
        Length Class  Mode
apriori  2     table  numeric
tables  14     -none- list
levels   2     -none- character
call     4     -none- call
```

2.2.6 Logistic Regression

```
> summary(logit_model)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2005  -0.4627  -0.2973  -0.1642   3.1191

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -7.05830    0.82332  -8.573  < 2e-16 ***
state2                         0.04222    0.78229   0.054  0.95696
state3                         0.77271    0.77554   0.996  0.31908
state4                        -0.33123    0.86217  -0.384  0.70084
state5                         1.35312    0.80069   1.690  0.09104 .
state6                         0.43526    0.78592   0.554  0.57970
state7                         0.80480    0.73905   1.089  0.27617
state8                         0.41679    0.83213   0.501  0.61646
state9                         0.54013    0.76394   0.707  0.47955
state10                        0.56135    0.76667   0.732  0.46406
state11                        0.47793    0.80306   0.595  0.55175
state12                       -0.64070    0.92551  -0.692  0.48877
state13                       -0.19023    0.93524  -0.203  0.83882
state14                        0.71240    0.76552   0.931  0.35206
state15                       -0.47275    0.85180  -0.555  0.57890
state16                        0.28009    0.77896   0.360  0.71917
state17                        1.01302    0.73504   1.378  0.16815
state18                        0.63300    0.77562   0.816  0.41443
state19                        0.37364    0.85942   0.435  0.66374
state20                        0.97224    0.75348   1.290  0.19694
state21                        1.04219    0.73256   1.423  0.15483
state22                        1.29754    0.74122   1.751  0.08003 .
state23                        1.32669    0.72171   1.838  0.06603 .
state24                        0.98315    0.72594   1.354  0.17564
state25                        0.44965    0.79074   0.569  0.56959
state26                        1.38733    0.73609   1.885  0.05947 .
state27                        1.82073    0.72589   2.508  0.01213 *
state28                        0.51177    0.76518   0.669  0.50360
state29                        0.22139    0.80069   0.276  0.78217
state30                        0.26513    0.81985   0.323  0.74640
state31                        0.95898    0.77757   1.233  0.21746
state32                        1.55687    0.71919   2.165  0.03041 *
state33                        0.31008    0.79853   0.388  0.69778
state34                        1.16122    0.73445   1.581  0.11386
state35                        1.04624    0.72511   1.443  0.14906
state36                        0.59704    0.75492   0.791  0.42902
state37                        0.81998    0.77117   1.063  0.28765
state38                        0.68856    0.74444   0.925  0.35500
state39                        0.96852    0.80143   1.208  0.22686
state40                       -0.26868    0.84357  -0.319  0.75010
state41                        1.54966    0.74068   2.092  0.03642 *
state42                        0.92546    0.76685   1.207  0.22750
state43                        0.18880    0.83094   0.227  0.82026
state44                        1.51435    0.72130   2.099  0.03577 *
state45                        0.93036    0.75172   1.238  0.21585
state46                       -0.56107    0.83041  -0.676  0.49926
state47                       -0.24964    0.79730  -0.313  0.75420
state48                        1.23420    0.73852   1.671  0.09469 .
state49                        0.11040    0.79907   0.138  0.89011
state50                        0.34318    0.74660   0.460  0.64576
state51                       -0.04234    0.77079  -0.055  0.95619
account.length                 0.24783    0.31611   0.784  0.43303
international.plan2            2.24343    0.15579  14.401  < 2e-16 ***
voice.mail.plan2              -1.82951    0.59951  -3.052  0.00228 **
number.vmail.messages          1.00020    0.83592   1.197  0.23149
total.day.minutes              3.77748    0.34096  11.079  < 2e-16 ***
total.day.calls                0.50294    0.31731   1.585  0.11297
total.eve.minutes              2.02071    0.33826   5.974 2.32e-09 ***
total.eve.calls               -0.15802    0.33105  -0.477  0.63313
total.night.minutes            1.34195    0.33492   4.007 6.16e-05 ***
total.night.calls              0.16374    0.32098   0.510  0.60996
total.intl.minutes             0.98167    0.32565   3.014  0.00257 **
total.intl.calls              -1.17827    0.29612  -3.979 6.92e-05 ***
number.customer.service.calls2  -0.16836    0.16977  -0.992  0.32134
number.customer.service.calls3   0.02800    0.18337   0.153  0.87862
number.customer.service.calls4  -0.21086    0.21782  -0.968  0.33303
number.customer.service.calls5   2.28482    0.23014   9.928  < 2e-16 ***
number.customer.service.calls6   3.28290    0.32038  10.247  < 2e-16 ***
number.customer.service.calls7   3.73164    0.50872   7.335 2.21e-13 ***
number.customer.service.calls8   3.62428    0.75673   4.789 1.67e-06 ***
number.customer.service.calls9   2.61719    1.48577   1.762  0.07815 .
number.customer.service.calls10 13.91949  354.87698   0.039  0.96871
---
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: 1977.7  on 3261  degrees of freedom
AIC: 2121.7

Number of Fisher Scoring iterations: 12
```

Chapter 3

Conclusion

3.1 Model Evaluation

3.1.1 Accuracy

```
> print(paste0("Decision Tree Accuracy: ", DT_valid_Matrix[1]))
[1] "Decision Tree Accuracy: 95.0809838032393"
> print(paste0("Random Forest Accuracy: ", RF_valid_Matrix[1]))
[1] "Random Forest Accuracy: 94.121175764847"
> print(paste0("Logistic Regression Accuracy: ", logit_valid_Matrix[1]))
[1] "Logistic Regression Accuracy: 86.9226154769046"
> print(paste0("KNN Accuracy: ", knn_valid_Matrix[1]))
[1] "KNN Accuracy: 86.8026394721056"
> print(paste0("Naive Bayes Accuracy: ", NB_valid_Matrix[1]))
[1] "Naive Bayes Accuracy: 87.9424115176965"
```

Figure 3.1: Model Accuracy

3.1.2 False Negative Rate

```
> print(paste0("Decision Tree False Negative rate: ", DT_valid_Matrix[2]))
[1] "Decision Tree False Negative rate: 33.4821428571429"
> print(paste0("Random Forest False Negative rate: ", RF_valid_Matrix[2]))
[1] "Random Forest False Negative rate: 32.1428571428571"
> print(paste0("Logistic Regression False Negative rate: ", logit_valid_Matrix[2]))
[1] "Logistic Regression False Negative rate: 69.6428571428571"
> print(paste0("KNN False Negative rate: ", knn_valid_Matrix[2]))
[1] "KNN False Negative rate: 46.4285714285714"
> print(paste0("Naive Bayes False Negative rate: ", NB_valid_Matrix[2]))
[1] "Naive Bayes False Negative rate: 72.3214285714286"
```

Figure 3.2: Model False Negative Rate

Model Name	%Accuracy	%FNR
Decision Tree	95.08	33.48
Random Forest	94.12	32.14
Naive Bayes	87.94	72.32
KNN	86.80	46.43
Logistic Regression	86.92	69.64

Table 3.1: Accuracy and False Negative Rate for Models

3.2 Model Selection

As it can be seen from Table 3.1, that Decision Tree has highest Accuracy as compared to any other model and Random Forest performs better in the False Negative Rate. But there is no drastic difference in their performance. So any one of them can be selected based on the customer requirement i.e. if the accuracy is more important for customer then we must select Decision Tree, but customer is interested in False Negative Rate then select Random Forest.

Appendix A

R Code

BoxPlot:
boxplot(variable Churn,data=totalData, main=" ", xlab="Account Length", ylab="Churn", col=(c("gold","darkgreen")), outcol="red")
Histogram:
hist(totalData$variable, xlab="Account Length", ylab="Frequency", main=" ")

Following is the snipper of R code:

```
# load Churn data
churn_train <- read.csv("Train_data.csv", header = TRUE)
churn_test <- read.csv("Test_data.csv", header = TRUE)
totalData <- rbind(churn_train, churn_test)

# Convert the numeric variables into categorical variables
totalData$area.code <- as.factor(totalData$area.code)
totalData$number.customer.service.calls <-
as.factor(totalData$number.customer.service.calls)

for(i in 1:ncol(totalData)){
  if(class(totalData[,i]) == 'factor'){
    totalData[,i] = factor(totalData[,i],
labels=(1:length(levels(factor(totalData[,i])))))
  }
}

#Decision Tree Model
DT_model = C5.0(Churn ~., trainData, trials = 100, rules = TRUE)
summary(DT_model)
#Lets predict for test cases
DT_Predict = predict(DT_model, testData[,-15], type = "class")

#Evaluate the performance of classification model
ConfMatrix_DT = table(testData$Churn, DT_Predict)

DT_valid_Matrix <- validationMatrix(ConfMatrix_DT)
print(paste0("Decision Tree Accuracy: ", DT_valid_Matrix[1]))
print(paste0("Decision Tree False Negative rate: ", DT_valid_Matrix[2]))
```

Appendix B

Python Code

Following is the snipper of Python code:

```
#Correlation plot
churn_corr = churn_data.corr()
fig, ax = plt.subplots(figsize=(10, 8))
ax.matshow(churn_corr)
plt.xticks(range(len(churn_data.columns)), churn_data.columns,
rotation='vertical')
plt.yticks(range(len(churn_data.columns)), churn_data.columns)

#Model Accuracy and FNR calculating function
def validationMatrix(confMatrix):
    #let us save TP, TN, FP, FN
    TN = confMatrix.iloc[0,0]
    FN = confMatrix.iloc[1,0]
    TP = confMatrix.iloc[1,1]
    FP = confMatrix.iloc[0,1]

    #check accuracy of model
    Accuracy = ((TP+TN)*100)/(TP+TN+FP+FN)

    #False Negative rate
    FNR = (FN*100)/(FN+TP)
    return(Accuracy,FNR)
```