Customer Churn Out Prediction

Venkata Subramanian R

03 January 2019

**Table of Contents:**

1. **Introduction**
   1. **Problem Statement-------------------------------------------------------3**
   2. **Data----------------------------------------------------------------------------3**
2. **Methodology**
   1. **Pre-Processing-------------------------------------------------------------5**
      1. **Missing Value Analysis----------------------------------------------5**
      2. **Outlier Analysis--------------------------------------------------------5**
      3. **Correlation Plot--------------------------------------------------------8**
      4. **Chi-Square Test--------------------------------------------------------9**
      5. **Feature Selection----------------------------------------------------10**
      6. **Modeling----------------------------------------------------------------12**
3. **Conclusion**
   1. **Model Evaluation--------------------------------------------------------16**
   2. **Model Selection----------------------------------------------------------17**
4. **Appendix---------------------------------------------------------------------------18**

**Chapter 1**

**Introduction**

**1.1 Problem Statement**

The objective of this project is to predict the Customer Churn Out Rate and enabling churn reduction. Analyze the data and develop a model to predict the churn out rate and develop a churn score.

**1.2 Data**

Our task is to build a classification model which will find the churn out of the customers based on the customer usage pattern. Below are some sample data which will predict the Customer Churn Out.

We are provided with Train and Test data separately, in which we develop alogorithm on training data and predict the value of test data using training data.

Table 1.1: Customer Sample Data (Columns 1-8)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **state** | **account length** | **area code** | **phone number** | **international plan** | **voice mail plan** | **number vmail messages** | **total day minutes** |
| KS | 128 | 415 | 382-4657 | no | yes | 25 | 265.1 |
| OH | 107 | 415 | 371-7191 | no | yes | 26 | 161.6 |
| NJ | 137 | 415 | 358-1921 | no | no | 0 | 243.4 |
| OH | 84 | 408 | 375-9999 | yes | no | 0 | 299.4 |
| OK | 75 | 415 | 330-6626 | yes | no | 0 | 166.7 |
| AL | 118 | 510 | 391-8027 | yes | no | 0 | 223.4 |
| MA | 121 | 510 | 355-9993 | no | yes | 24 | 218.2 |
| MO | 147 | 415 | 329-9001 | yes | no | 0 | 157 |
| LA | 117 | 408 | 335-4719 | no | no | 0 | 184.5 |
| WV | 141 | 415 | 330-8173 | yes | yes | 37 | 258.6 |
| IN | 65 | 415 | 329-6603 | no | no | 0 | 129.1 |

Table 1.2 Customer Sample Data (Columns 9-17)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **total day calls** | **total day charge** | **total eve minutes** | **total eve calls** | **total eve charge** | **total night minutes** | **total night calls** | **total night charge** | **total intl minutes** |
| 110 | 45.07 | 197.4 | 99 | 16.78 | 244.7 | 91 | 11.01 | 10 |
| 123 | 27.47 | 195.5 | 103 | 16.62 | 254.4 | 103 | 11.45 | 13.7 |
| 114 | 41.38 | 121.2 | 110 | 10.3 | 162.6 | 104 | 7.32 | 12.2 |
| 71 | 50.9 | 61.9 | 88 | 5.26 | 196.9 | 89 | 8.86 | 6.6 |
| 113 | 28.34 | 148.3 | 122 | 12.61 | 186.9 | 121 | 8.41 | 10.1 |
| 98 | 37.98 | 220.6 | 101 | 18.75 | 203.9 | 118 | 9.18 | 6.3 |
| 88 | 37.09 | 348.5 | 108 | 29.62 | 212.6 | 118 | 9.57 | 7.5 |
| 79 | 26.69 | 103.1 | 94 | 8.76 | 211.8 | 96 | 9.53 | 7.1 |
| 97 | 31.37 | 351.6 | 80 | 29.89 | 215.8 | 90 | 9.71 | 8.7 |
| 84 | 43.96 | 222 | 111 | 18.87 | 326.4 | 97 | 14.69 | 11.2 |
| 137 | 21.95 | 228.5 | 83 | 19.42 | 208.8 | 111 | 9.4 | 12.7 |

Table 1.3 Customer Sample Data (Columns 18-21)

|  |  |  |  |
| --- | --- | --- | --- |
| total intl calls | total intl charge | number customer service calls | Churn |
| 3 | 2.7 | 1 | False. |
| 3 | 3.7 | 1 | False. |
| 5 | 3.29 | 0 | False. |
| 7 | 1.78 | 2 | False. |
| 3 | 2.73 | 3 | False. |
| 6 | 1.7 | 0 | False. |
| 7 | 2.03 | 3 | False. |
| 6 | 1.92 | 0 | False. |
| 4 | 2.35 | 1 | False. |
| 5 | 3.02 | 0 | False. |
| 6 | 3.43 | 4 | True. |

**Chapter 2**

**Methodology**

**2.1** **Pre Processing**

Any predictive modeling 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.**

Since we have two data sets- Train data and Test data we will apply Pre-Processing Techiniques on both the data sets and prepare the data sets to apply model on them.

To Start with, we will compute the Missing Values

**2.1.1 Missing Value Analysis**

In the given train and test data set, there are no missing values found. So no need to perform any imputation methods analysis.

**2.1.2 Outlier Analysis**

We have found that there are no missing values to impute. Now as the next step in Data Pre-Processing we need to detect the outliers that is present in the data set and remove it.

We can detect the outliers in the data set using Box Plot Method. In the below figure we have plotted the outliers in the Training and Test Data of the Customer Report.

Fig 2.1

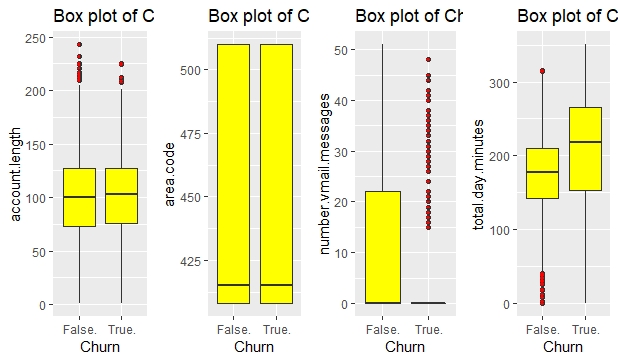


Fig 2.2

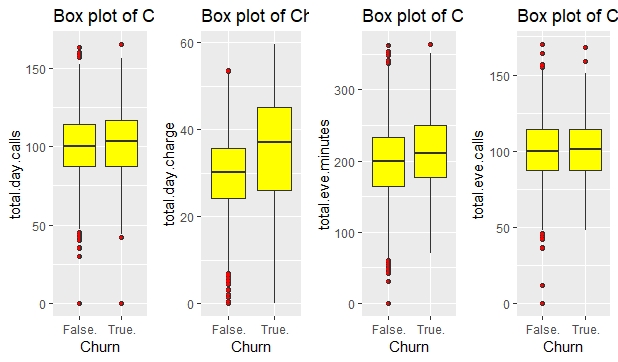


Fig 2.3

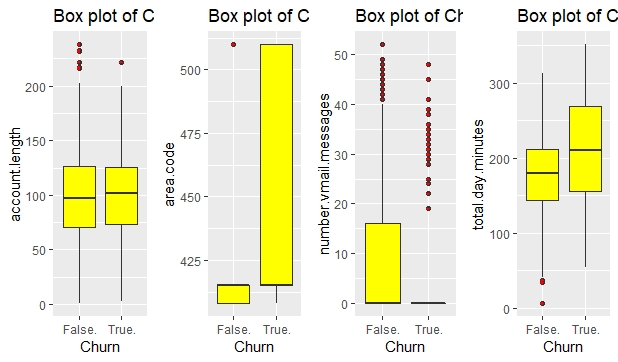
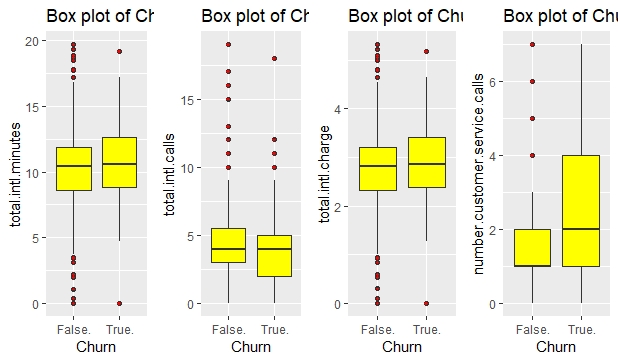


Fig 2.4



From the Figure 2.1,2.2,2.3 and 2.4 we can infer that we have some outliers in Training and Test Data sets. We can remove the values of these variables and replace them with NA.

Once replaced with NA, we can impute the missing value using KNN Imputation Method

Now with this Pre-Processing Technique, we have removed the outliers from the data set.

After computing the missing value using KNN Imputation, we can see in Fig 2.5 and 2.6 that almost 90% of the outliers values are removed.

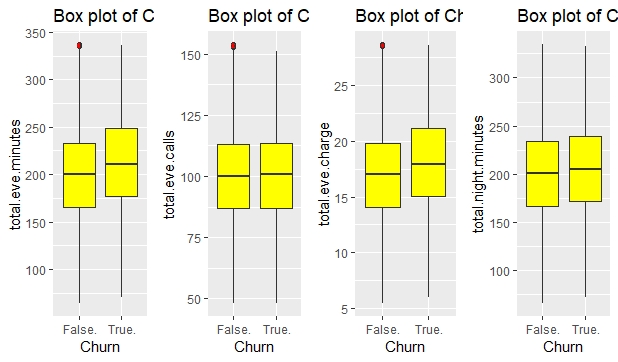


Fig 2.5

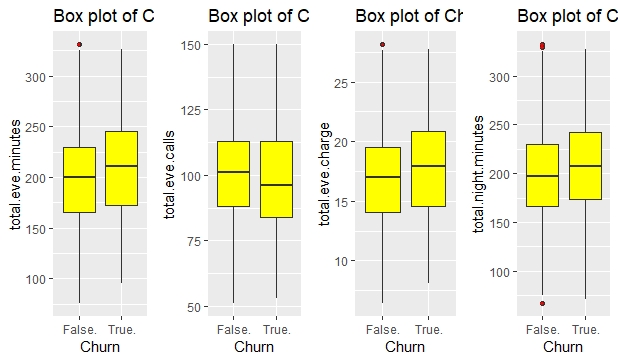
****

Fig2.6

**2.1.3 Correlation Plot**

As the next step in Pre-Processing we need to find the multi- collinearity between the variables. If two variables contain the same information, then we can discard one variable while passing it to the model. So, with the help of Correlation Plot we can find the multi collinearity between the continuous variables.

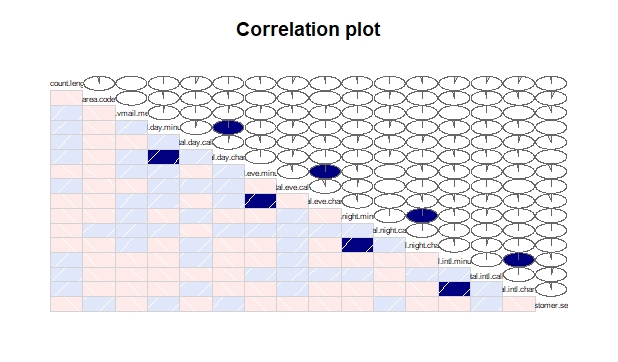


Fig 2.7

From Figure 2.7, we see that we high dependency between the following variables

* total.day.minutes and total.day.charge
* total.eve.minutes and total.eve.charge
* total.night.minutes and total.night.charge
* total.intl.minutes and total.intl.charge

So out of these 8 variables we can drop 4 variables and keep only 4 variables to build the model.

Hence in both training and test data, we can drop the variables- **total.day.charge, total.eve.charge, total.night.charge, total.intl.charge**

From Correlation Plot Analysis we have dropped 4 variables from both training and test data set.

**2.1.4 Chi-Square Testing**

We have done Collinearity check between Continuous variables. Now we need to find the dependency between Categorical Variables. We can use Chi Square Testing to determine the dependency between these variables.

When executed the Chi SquareTesting, we got the following observations.

[1] "state"

Pearson's Chi-squared test

data: table(train\_data\_val$Churn, train\_data\_val[, i])

X-squared = 83.044, df = 50, p-value = 0.002296

[1] "phone.number"

Pearson's Chi-squared test

data: table(train\_data\_val$Churn, train\_data\_val[, i])

X-squared = 3333, df = 3332, p-value = 0.4919

[1] "international.plan"

Pearson's Chi-squared test with Yates' continuity correction

data: table(train\_data\_val$Churn, train\_data\_val[, i])

X-squared = 222.57, df = 1, p-value < 2.2e-16

[1] "voice.mail.plan"

Pearson's Chi-squared test with Yates' continuity correction

data: table(train\_data\_val$Churn, train\_data\_val[, i])

X-squared = 34.132, df = 1, p-value = 5.151e-09

[1] "Churn"

Pearson's Chi-squared test with Yates' continuity correction

data: table(train\_data\_val$Churn, train\_data\_val[, i])

X-squared = 3324.9, df = 1, p-value < 2.2e-16

Based on the above observations, we need to remove the variables whose p-value is >0.05 since our significance value is 0.05.

On Interpreting the results, we can remove the variables **phone.number** from our training as well as test data set.

**2.1.5 Feature Selection**

Before Performing any type of modeling, we need to access the importance of each predictor variables in our data sets. There is every chance that all variables in our analysis set may not be as important as all in the problem of class prediction. Below we have used Random Forest method to Perform Feature Selection and got the following observations.

MeanDecreaseAccuracy

state 0.97113009

account.length -1.42178408

area.code 1.31562771

international.plan 54.49055554

voice.mail.plan 16.63832364

number.vmail.messages 16.88587626

total.day.minutes 31.57546901

total.day.calls 1.28040034

total.day.charge 34.28028727

total.eve.minutes 19.31004991

total.eve.calls -1.14564077

total.eve.charge 18.78572953

total.night.minutes 15.61315968

total.night.calls 0.02147584

total.night.charge 14.93386267

total.intl.minutes 19.38092936

total.intl.calls 26.76708633

total.intl.charge 19.61917274

number.customer.service.calls 19.23869423

From the above information we can infer that **international.plan, total.day.charge, total.day.minutes** variables are some of the most important predictors in our data set.

The variables **area.code,account.length,total.day.calls,total.eve.calls,total.night.calls,state** contain much less information for deriving target variable and hence can be dropped from the data set. So we can do dimension reduction and drop these variables.

Since our Prediction is there should be less dependency between the independent variables in a dataset.

So, we need to find the correlation between the independent variables. We can have the threshold value as 0.9.

vifcor(train\_cor\_data,th=0.9)

No variable from the 7 input variables has collinearity problem.

The linear correlation coefficients ranges between:

min correlation ( number.customer.service.calls ~ total.day.minutes ): 0.000116526

max correlation ( number.customer.service.calls ~ total.eve.minutes ): -0.03216692

---------- VIFs of the remained variables --------

Variables VIF

1 number.vmail.messages 1.000857

2 total.day.minutes 1.000453

3 total.eve.minutes 1.002374

4 total.night.minutes 1.000794

5 total.intl.minutes 1.000979

6 total.intl.calls 1.001181

7 number.customer.service.calls 1.002456

From the above observation we can find that no variable has collinearity problem.

**2.6 Modeling**

Now we have done Pre-Processing Techniques on the data set and our data is ready to develop a model.

**2.2.1 Model Selection**

We need to Predict the Count Variable which is a continuous variable. For dependent variable we have 4 categories:

* Nominal
* Ordinal
* Interval
* Ratio

Since our Variable is a nominal, we will go for classification model.So, we will start from the basic model and try to predict the Target variable.

We will first use Logistic Regression Model to Predict the value of Target Variable

**Logistic Regression**

Summary of the Logistic Regression Model is as below.

Call:

glm(formula = Churn ~ ., family = "binomial", data = customer\_train\_data)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.5855 -0.5517 -0.4194 -0.2720 3.0523

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.975322 0.481060 -12.421 < 2e-16 \*\*\*

international.plan yes 1.826220 0.136327 13.396 < 2e-16 \*\*\*

voice.mail.plan yes -1.906626 0.534709 -3.566 0.000363 \*\*\*

number.vmail.messages 0.033744 0.016826 2.005 0.044916 \*

total.day.minutes 0.010862 0.001045 10.397 < 2e-16 \*\*\*

total.eve.minutes 0.005550 0.001101 5.042 4.62e-07 \*\*\*

total.night.minutes 0.003544 0.001095 3.238 0.001204 \*\*

total.intl.minutes 0.065331 0.020783 3.144 0.001669 \*\*

total.intl.calls -0.113328 0.026677 -4.248 2.16e-05 \*\*\*

number.customer.service.calls -0.012391 0.055719 -0.222 0.824018

---

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: 2367.0 on 3323 degrees of freedom

AIC: 2387

Number of Fisher Scoring iterations: 5

From the above observations, we can find that the difference between Null Deviance and Residual Deviance is 391.3. As per our assumption the difference between these two should be very high. So, 391.3 is a good difference and we can see the error metrics of this model, compared with other models and less error model can be freezed.

**Decision Tree Classification**

We have built Logistic Regression for our data previously and now we will build a decision tree using CART Model and we will check for accuracy and few other error metrics.

Since Decision Tree is a rule based algorithm we extract rules from the decision tree.

Here are the few sample rules which we got on building the decision tree algorithm.

Class specified by attribute `outcome'

Read 3333 cases (10 attributes) from undefined.data

Rules:

Rule 1: (160/4, lift 1.1)

international.plan = no

voice.mail.plan = yes

total.day.minutes > 223.2

-> class False. [0.969]

Rule 2: (499/15, lift 1.1)

international.plan = no

total.day.minutes <= 223.2

number.customer.service.calls <= 0.43716

-> class False. [0.968]

Rule 3: (207/6, lift 1.1)

international.plan = no

total.day.minutes > 176

total.day.minutes <= 223.2

total.eve.minutes > 135.4

total.eve.minutes <= 212.1

number.customer.service.calls > 1

-> class False. [0.967]

Rule 4: (876/29, lift 1.1)

international.plan = no

total.day.minutes <= 223.2

number.customer.service.calls > 0.9878923

number.customer.service.calls <= 1

-> class False. [0.966]

Rule 5: (875/31, lift 1.1)

international.plan = no

total.day.minutes <= 223.2

number.customer.service.calls > 1.972828

-> class False. [0.964]

Rule 18: (42/2, lift 6.4)

international.plan = no

voice.mail.plan = no

total.day.minutes > 246.6

total.eve.minutes > 241

-> class True. [0.932]

Out of the total rules given we need to find the rule which is statistically significant based on three parameters:

* **Support**
* **Confidence**
* **Lift**

From the above rules, let’s take a sample rule and evaluate these parameters.

Rule 18: (42/2, lift 6.4)

international.plan = no

voice.mail.plan = no

total.day.minutes > 246.6

total.eve.minutes > 241

-> class True. [0.932]

Here our Confidence Value is **93.2**% and our lift is **6.4**. According to Market Standard, if Lift is greater than 1 and Confidence is greater than 20%, then that rule can be classified as good. So, the above rule is more powerful. Similarly we can calculate for other rules and find the rules which is statistically significant.

Based on the above sample rules. Decision tree will calculate the target variable value. From Decision tree algorithm, we got total of 24 rule sets based on which target value will be calculated.

Default class: False.

Evaluation on training data (3333 cases):

Rules

----------------

No Errors

24 172( 5.2%) <<

(a) (b) <-classified as

---- ----

2826 24 (a): class False.

148 335 (b): class True.

Attribute usage:

96.40% total.day.minutes

88.48% international.plan

83.02% total.eve.minutes

72.76% number.customer.service.calls

66.58% total.intl.calls

66.40% total.intl.minutes

30.48% total.night.minutes

14.40% voice.mail.plan

Decision Tree also gives us the important variables which have contributed more to find the target variable. Based on the above attribute usage, we can find that total day minutes variable contributes much in deriving the target variable.

We have now build the model and need to check for some error metrics and find whether we can go with this model.

**Random Forest Classification**

We have built Logistic Regression and Decision Tree Algorithm for our data set. Now we will go with Random Forest Approach.

In Random Forest Algorithm, initially we will build the model with 500 Decision Tree and check the performance of the model. Based on the error metrics, we can freeze the number of trees to be used.

Performance with 500 Decision Trees:

Accuracy Rate-94.24%

False Negative Rate-42.8%

Performance with 1000 Decision Trees:

Accuracy Rate- 94.1%

False Negative Rate-41.6%

Performance with 1500 Decision Trees:

Accuracy Rate-94.3%

False Negative Rate-41.2%

Performance with 2000 Decision Trees:

Accuracy Rate-93.8%

False Negative Rate-41.6%

Based on the above analysis, we can see that Error rate increases in 2000 trees compared to that using 1500 trees. So, we can freeze 1500 trees for building our model.

**Conclusion**

**3.1 Model Evaluation**

Now that we have developed a few models for predicting our target variable and we have to decide which one to choose. We will evaluate some error metrics and decide which one to choose.

We can evaluate the Predictive Performance of the model with real values of the target variables and calculating some average error measures.

**3.1.1 Accuracy:**

We need to find the accuracy of the models which we have developed and choose the best among the three models.

**Logistic Regression Model:**

For Logistic Regression Model we get Accuracy as **94.24%**

**Decision Tree Model:**

For Decision Tree Model we get Accuracy as **93.4%**

**Random Forest Algorithm:**

For Random Forest Model we get Accuracy as **94.3%**

Based on the above resultd we can see that the Accuracy for all the models are good.

**3.1.2 False Negative Rate(FNR):**

We need to find the False Negative Rate to find the efficiency of the model and how good the model predicts the target value. Lesser the FNR, the more suitable the model will be.

**Logistic Regression Model:**

For Logistic Regression Model we get FNR as **1.3%**

**Decision Tree Model:**

For Decision Tree Model we get FNR as **42.4%**

**Random Forest Algorithm:**

For Random Forest Model we get Accuracy as **41.9%**

**3.2 Model Selection**

Based on the Above Observations, we can see that Logistic Model is best suitable since it has very low FNR Value. But one problem with Logistic Model is that, they are sensitive to outliers. So, if outliers value increases in the above dataset, we cannot go Logistic model since the error rate will get increased.

So, based on the above calculations, we can go for Logistic Model or Random Forest Classifier to predict the target class value.

**Appendix A - Extra Figures**

