Final Project
Using Churn Dataset

Course: CIND 119 - Introduction to Big Data
Ryerson Chang School of Continuing Studies F2019
Instructor: Dhanta Jothimani

Project Members
Alishay Akmal
Don Kim
Kyuhwan Kim
Richard Zhang

Index

Members

Alishay Akmal, alishay.akmal@ryerson.ca
Don Kim, don.kim@ryerson.ca
Kyuhwan Kim, kyuhwan.kim@ryerson.ca
Richard Zhang, ruifeng.zhang@ryerson.ca

1. Introduction

Our client is a company that is in an industry where there are numerous competitors and new entrants that are still coming into the already saturated market. Therefore, the company is developing a client retention strategy and have asked our data science consultants to help the company implement a strategy to minimize **churn**. Churn is a process whereby a customer would unsubscribe to the current company due to dissatisfaction and potentially move to another company.
The reason churn is investigated is to optimize revenue for this company since winning a new customer is more costly than retaining an existing customer. Since the market is very saturated, this would be a wise choice to look into customer retention rather than customer acquisition.
This paper will investigate the dataset provided by the company. What we set out to do is to investigate the differences between customers who churn and who do not churn. We will then visualize and present the results to assist in decision making process to potentially reduce churn for the company and to provide insights that alarm bells to provide opportunities to take precautionary measures.

<u>Tools</u>
The tools that will be used are WEKA (Waikato Environment for Knowledge Analysis) and R WEKA contains tools for data preprocessing, classification (J48 decision tree), clustering, and visualization; all the tools necessary for this project.

2. Workload Distribution

Member Names	**List of tasks performed**
Alishay Akmal	Presentation, attributes elimination, random forest classification
Don Kim	Data prep, Post Predictive Analysis
Kyuhwan Kim	Introduction, Data prep, attributes elimination, Conclusion
Richard Zhang	Predict Modeling, Classification (Decision tree, Naive Bayes)

3. Data Preparation

3.1 Information about Dataset :

The dataset provided came in 2 different formats (.csv and artf). ARTF stands for Atrribute Relation File Format. Both are suitable for analysis but artf format was used using a more robust WEKA tool. The dataset has 3333 instances (rows) and 21 attributes (columns) with the last attribute ‘churn’ as the classifier with TRUE or FALSE labels; TRUE = churned, FALSE = not churned. 2850 customers retained while 483 (483/2850) 16.9% churn rate. The company did an intricate job since the dataset was complete; none of the attributes were missing an entry. In terms of format, the datasets are combined with nominal and numerical types. The dataset is one set, one time interval time frame. For the future, a longitudinal study would be possible to conduct but we will presume that this dataset is an annual dataset.

	Attributes	Type	Min	Max	Mean	SD	Distinct	Missing Values
1	State	nominal					51	0
2	Account Length	numeric	1	243	101.065	39.82	212	0
3	Area Code	nominal					3	0
4	Phone Number	nominal					3333	0
5	Inter Plan	nominal					2	0
6	VoiceMail Plan	nominal					2	0
7	No of Vmail Mesgs	numeric	0	51	8.099	13.69	46	0
8	Total Day Min	numeric	0	350.8	179.775	54.47	1667	0
9	Total Day calls	numeric	0	165	100.436	20.07	119	0
10	Total Day Charge	numeric	0	59.64	30.562	9.259	1667	0
11	Total Evening Min	numeric	0	363.7	200.98	50.71	1611	0

12	Total Evening Calls	numeric	0	170	100.114	19.92	123	0
13	Total Evening Charge	numeric	0	30.91	17.084	4.311	1440	0
14	Total Night Minutes	numeric	23.2	395	200.872	50.57	1591	0
15	Total Night Calls	numeric	33	175	100.108	19.57	120	0
16	Total Night Charge	numeric	1.04	17.77	9.039	2.276	933	0
17	Total Int Min	numeric	0	20	10.237	2.792	162	0
18	Total Int Calls	numeric	0	20	4.479	2.461	21	0
19	Total Int Charge	numeric	0	5.4	2.765	0.754	162	0
20	No of Calls Customer Service	numeric	0	9	1.563	1.315	10	0
21	Churn	nominal					2	0

Figure 1 is a brief summary of attributes displaying its characteristics from artf file with WEKA:

3.2 Boxplots

All the numerical attributes can be plotted in boxplots and the boxplots will display the outliers. Figure 2 is boxplot plots of all the numerical attributes.

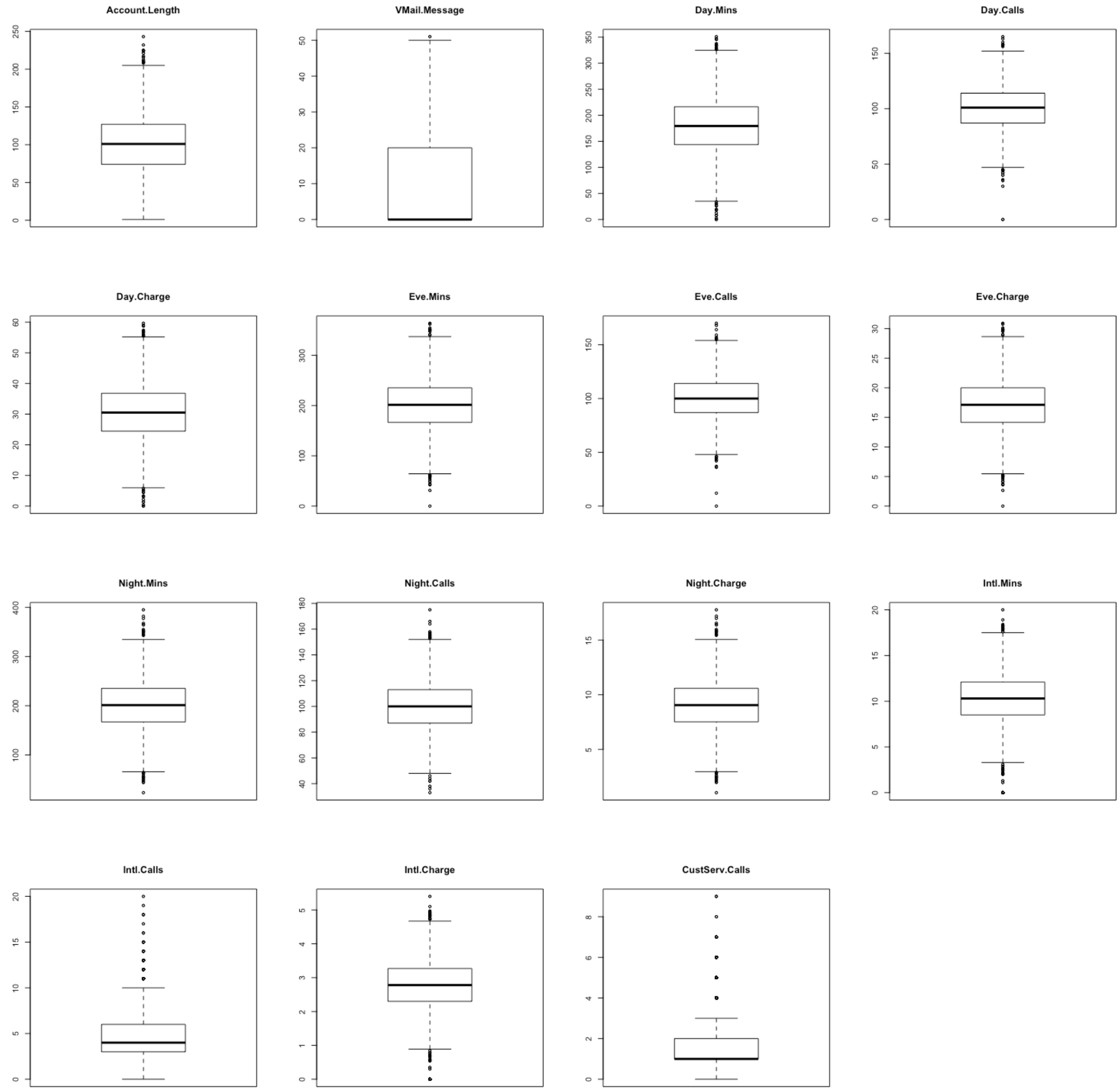

Figure 2. Boxplots for all numerical attributes

Below is the R code to have this output.

```
#Loading churn dataset
gc <- read.csv("R/churn.csv",header = TRUE, stringsAsFactors = FALSE,
               na.string=c("","NA"))

#draw multiple boxplots
plot_boxplots <- function(gcc) {
  x <- c(2,7:20) #all nominal attributes were not selected
  par(mfrow=c(4,4))
  for (i in x) {
    boxplot(gcc[i],main=names(gcc[i]))
  }
}
plot_boxplots(gc)
```

Dealing with Outliers
The dots that are outside of the 1.5 +/- IQR[Q3-Q1] are considered outliers.
Our decision was to keep the outlier since there were not many outliers and we wanted our data to be true from its source of record.

3.3 Visualization of data

3.3.1 Histogram

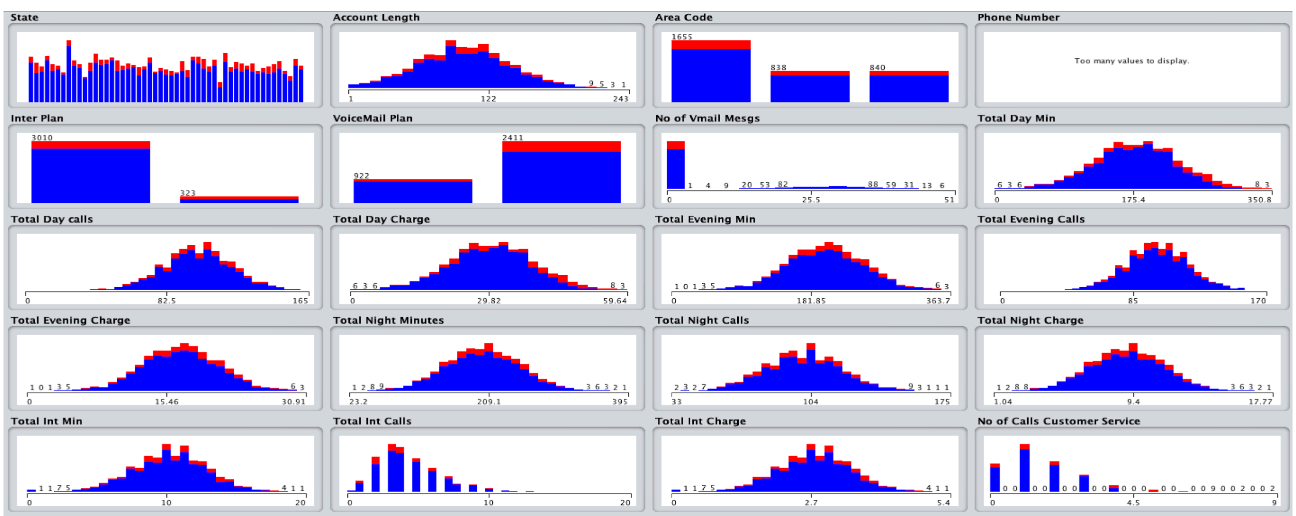

Figure 3. Output of Visualize All tab in Preprocess

Most of the plots had a normal distribution shape. Attribute 18 (International calls) and attribute 20 (No of Calls Customer Service) are skewed to the right. Checking the boxplots and histograms, these 2 attributes definitely show most skewness with most outliers.

3.3.2 Correlation Matrix

R Code for Correlation Matrix:
Here is the code for the correlation matrix in R:

```
#================================================
#open file and put into variable 'data'
data <- read.csv("churn.csv", header = TRUE)
#================================================
#this is to check the names of the attributes (header names)
head(data)
#=================================================
#convert the 3 columns with 'factor' data into numeric data
#check the outputs and R converted the bisectional (ex. T/F)
#data and grouped them into 1,2 and etc.
data$Churn. <- as.integer(data$Churn.)
data$Int.l.Plan <- as.integer(data$Int.l.Plan)
data$VMail.Plan <- as.integer(data$VMail.Plan)
#take that data and convert them to 0 and 1 (FALSE and TRUE)
data$Churn.[data$Churn.==1] <- 0
data$Churn.[data$Churn.==2] <- 1
data$Int.l.Plan[data$Int.l.Plan==1] <- 0
data$Int.l.Plan[data$Int.l.Plan==2] <- 1
data$VMail.Plan[data$VMail.Plan==1] <- 0
data$VMail.Plan[data$VMail.Plan==2] <- 1
#now all the columns are numeric
#=============================================
#  Drop uneeded variables
#=============================================
data$State <- NULL
data$Area.Code <- NULL
data$Phone <- NULL
#=============================================
#  Handling missing values
#=============================================
summary(data)
sapply(data,sd)
cormatrix <- round(cor(data), digits = 2)
```

	Account.Length	Int.l.Plan	VMail.Plan	VMail.Message	Day.Mins	Day.Calls	Day.Charge	Eve.Mins
Account.Length	1.00	0.02	0.00	0.00	0.01	0.04	0.01	-0.01
Int.l.Plan	0.02	1.00	0.01	0.01	0.05	0.00	0.05	0.02
VMail.Plan	0.00	0.01	1.00	0.96	0.00	-0.01	0.00	0.02
VMail.Message	0.00	0.01	0.96	1.00	0.00	-0.01	0.00	0.02
Day.Mins	0.01	0.05	0.00	0.00	1.00	0.01	1.00	0.01
Day.Calls	0.04	0.00	-0.01	-0.01	0.01	1.00	0.01	-0.02
Day.Charge	0.01	0.05	0.00	0.00	1.00	0.01	1.00	0.01
Eve.Mins	-0.01	0.02	0.02	0.02	0.01	-0.02	0.01	1.00
Eve.Calls	0.02	0.01	-0.01	-0.01	0.02	0.01	0.02	-0.01
Eve.Charge	-0.01	0.02	0.02	0.02	0.01	-0.02	0.01	1.00
Night.Mins	-0.01	-0.03	0.01	0.01	0.00	0.02	0.00	-0.01
Night.Calls	-0.01	0.01	0.02	0.01	0.02	-0.02	0.02	0.01
Night.Charge	-0.01	-0.03	0.01	0.01	0.00	0.02	0.00	-0.01
Intl.Mins	0.01	0.05	0.00	0.00	-0.01	0.02	-0.01	-0.01
Intl.Calls	0.02	0.02	0.01	0.01	0.01	0.00	0.01	0.00
Intl.Charge	0.01	0.05	0.00	0.00	-0.01	0.02	-0.01	-0.01
CustServ.Calls	0.00	-0.02	-0.02	-0.01	-0.01	-0.02	-0.01	-0.01
Churn.	0.02	0.26	-0.10	-0.09	0.21	0.02	0.21	0.09

	Eve.Calls	Eve.Charge	Night.Mins	Night.Calls	Night.Charge	Intl.Mins	Intl.Calls	Intl.Charge
Account.Length	0.02	-0.01	-0.01	-0.01	-0.01	0.01	0.02	0.01
Int.l.Plan	0.01	0.02	-0.03	0.01	-0.03	0.05	0.02	0.05
VMail.Plan	-0.01	0.02	0.01	0.02	0.01	0.00	0.01	0.00
VMail.Message	-0.01	0.02	0.01	0.01	0.01	0.00	0.01	0.00
Day.Mins	0.02	0.01	0.00	0.02	0.00	-0.01	0.01	-0.01
Day.Calls	0.01	-0.02	0.02	-0.02	0.02	0.02	0.00	0.02
Day.Charge	0.02	0.01	0.00	0.02	0.00	-0.01	0.01	-0.01

Figure 4. Output of Correlation Matrix using R code

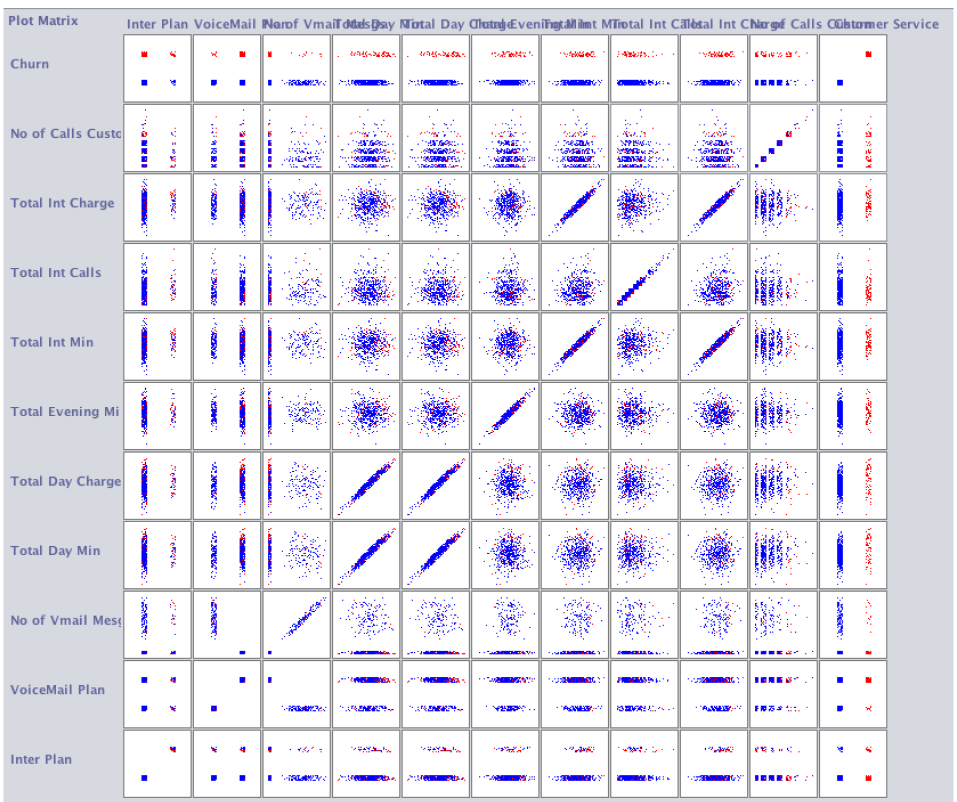

Figure 5. Plot matrix visual output from ‘Visualize’ tab

When there is 1.0 correlation on the matrix (code from R), this means that there is perfect correlation. This happens for instance when the attribute is compared to itself (ex. State vs State). Numerically, the output is 1.0 and visually you will see the graph representing y = x. An important things to note is that only numeric data was possible to have cor( ) so factors (state, phone #, area code) attributes were eliminated to function with the R code.

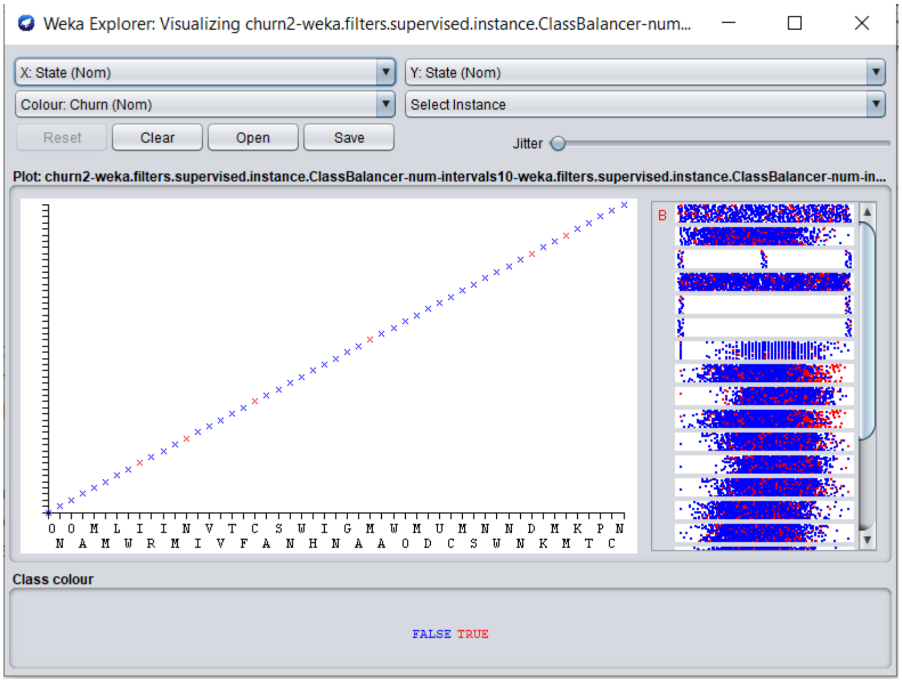

Figure 6. State vs state

Figure 6 is how an output should look like when an attribute is plotted to itself. The shape is a linear.

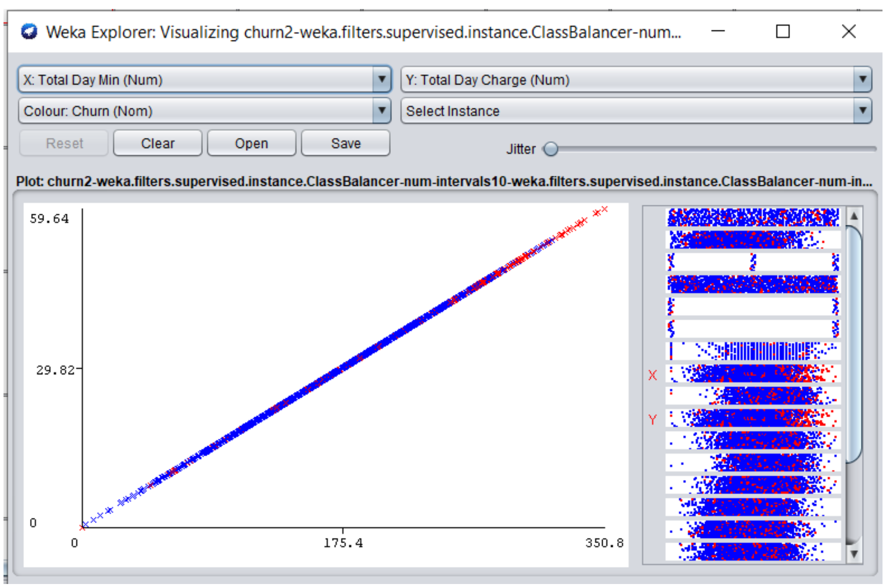

Figure 7. Total mins vs Total day charge

The output in figure 7 has a correlation of one. These attributes are required to remove. In figure 4, this correlation is actually 1.0

The cases where we see y=x should theoretically only be observed when an attribute is compared to itself. However, using WEKA, more of such representations were occurring and this realization has assisted in attribute elimination.

We were able to find in figure 5 that there were total of 4 charges in our dataset: Total Day Charge, Total Evening Charge, Total Night Charge, Total Int Charge. We compared these ‘charges’ towards their respective ‘minutes’ used. And we see a y=x relationship between the charges and the minutes used. This means that all the charges are laid out as fixed cost linear relationship. With our tests, we were able to verify that ‘charges’ and ‘minutes used’ had a correlation of 1.0.

3.4 Transforming attributes

Normalization is not required since none of the attributes necessarily require to be transformed into categorical ranking. Discretizing numeric attributes to categorical attributes are not required since it is already done. There was no missing values on attributes.

Class Label
To perform classification with WEKA, the last attribute is taken as a class label and it should be nominal. In churn dataset, the default was last attribute with nominal data type. Therefore, no transformation with class label was required in this case.

The question to investigate is to discover if we need to deal with Imbalanced Class Distribution. This is a case where observations belonging to one class is significantly lower than those belonging to other classes. This can be an issue since predictive models using machine learning algorithms could be biased and inaccurate due to overfitting of data. In our dataset, There are 2850 (83.1%) for FALSE and 483 (16.9%) TRUE.

To rebalance the class label, the method we did was: Preprocess -> Filter -> Choose -> Supervised (section related to class attribute) -> Instance (regarding rows) -> Class Balancer.

This is the description from ClassBalancer. “Reweights the instances in the data so that each class has the same total weight.”

Once this filter is applied, Figure 8 is the result:

Figure 8. Balance churn

So the counts of instances are the same, but the major change was in the weight. They have been equalized to have the same value of 1666.5

We also have noticed that when this filter was used, all the other attributes were balanced by adding surrogate data to other attributes more towards the churn = True data. This synthetically allowed balancing of data and added a lot more data towards churn = True values towards all of the attributes.

*A word of caution is that this process is to be done after attribute selection since many attribute tests were for some reason not available when conducting after this filter was used.

3.5 Elimination of Attributes

Based on observations of previous section, it appears that some attributes can be removed due to redundancy of the data and other various factors: raw data is often not suitable for modeling. The process of selecting which attributes for machine learning is called: feature selection.

Primarily, the feature selection can be done using the Weka Explorer at the "Select attributes" tab. From here, we can select the attribute evaluator and search method to allow Weka select which attributes would be most suitable for machine learning modelling.

Attribute evaluator is the technique by which each attribute (column) is evaluated in the context of the class. Search method is the technique to try different combinations of attributes in order to arrive at the short list of the desired results.

With experimentation, it was found that certain attribute evaluators require certain search methods; as the program would prompt you to select the specific search method. CorrelationAttributeEval can only be used by Ranker search method. Also, we have found out that each evaluator had a different output and it was difficult for our team to determine which attributes to keep and which attributes to discard. Here is a list of some attribute evaluator techniques that we have used: cfssubsetEval, CorrelationAttributesEval, InfrogainEval.

3.5.1 CfsSubsetEval

This is the default test when first opening WEKA. We have decided to run this test with the baseline dataset.

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 151
        Merit of best subset found:    0.152

Attribute Subset Evaluator (supervised, Class (nominal): 21 Churn):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 4,5,8,20 : 4
                     Phone Number
                     Inter Plan
                     Total Day Min
                     No of Calls Customer Service
```

Figure 8. Result from CfsSubsetEval using default settings and baseline model (all 21 attributes)

The 4 attributes that this algorithm has determined that were important in determining churn were: Phone Number, Inter Plan, Toal Day Min, No of Calls Customer Service. However, this we felt was not conclusive enough in determining the final dataset to be used for machine learning. Although Phone Number in actuality was not significant, the other attribute results: Inter Plan, Total Day Min, No of Calls Customer Service were very important. An interesting observation we have found was that when Phone Number was removed, it provided us with an entirely different result; regardless, though the first output was not perfect, with discernment, we would find that the algorithm was indeed useful since the result was significant in making our conclusion.

```
2,3,5,8,14,15,17 : 7
Inter Plan
VoiceMail Plan
Total Day Min
Total Evening Min
Total Int Min
Total Int Calls
No of Calls Customer Service
```

Figure 8.1 Modified CfsSubsetEval with 3 nominal attributes removed (State, Area Code, Phone Number)

We have noticed that the order is quite different from the other one. Ex. No of Calls Customer Service is in a much lower ranking.

We decided to do more tests to validate our reasoning and to come to a more definite finding.

3.5.2 CorrelationAttributeEval

This technique requires ‘Ranker’ search method.
More formally known as Pearson’s correlation coefficient, we were able such values using the cor () function in R; looking at the documentation, we find that the default method is Pearson’s. So what the algorithm would do is calculate the correlation between each attribute and select the moderate to high values and drop the values which are closer to zero.

Figure 9. Comparing the R code and WEKA, we were able to know how CorrelationAttributeEval function works. First, obtain correlation matrix and rank correlation to obtain an order of attributes. (eg. Top scoring attribute ‘Inter Plan’ is both 0.25985 for both R and Weka.)

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 21 Churn):
        Correlation Ranking Filter
Ranked attributes:
 0.25985    5 Inter Plan
 0.20875   20 No of Calls Customer Service
 0.20515    8 Total Day Min
 0.20515   10 Total Day Charge
 0.10215    6 VoiceMail Plan
 0.0928    11 Total Evening Min
 0.09279   13 Total Evening Charge
 0.08973    7 No of Vmail Mesgs
 0.06826   19 Total Int Charge
 0.06824   17 Total Int Min
 0.05284   18 Total Int Calls
 0.0355    16 Total Night Charge
 0.03549   14 Total Night Minutes
 0.01874    1 State
 0.01846    9 Total Day calls
 0.01654    2 Account Length
 0.0122     4 Phone Number
 0.00923   12 Total Evening Calls
 0.00614   15 Total Night Calls
 0.00514    3 Area Code

Selected attributes: 5,20,8,10,6,11,13,7,19,17,18,16,14,1,9,2,4,12,15,3 : 20
```

```
                   Churn.
Account.Length   0.01654
Int.l.Plan       0.25985
VMail.Plan      -0.10215
VMail.Message   -0.08973
Day.Mins         0.20515
Day.Calls        0.01846
Day.Charge       0.20515
Eve.Mins         0.09280
Eve.Calls        0.00923
Eve.Charge       0.09279
Night.Mins       0.03549
Night.Calls      0.00614
Night.Charge     0.03550
Intl.Mins        0.06824
Intl.Calls      -0.05284
Intl.Charge      0.06826
CustServ.Calls   0.20875
Churn.           1.00000
```

Figure 9. Results from R and Weka were similar when using CorrelationAttributionEval

We also discovered that some relations had 1.0 relation (although not compared to itself). This was significant in determining to remove in conjunction with infogain.

3.5.3 InfoGainAttributeEval

This technique is used to calculate information gain (entropy) for each attribute for the class variable. Entry values ranges from 0 to 1: zero being no information and 1 being maximum information. Therefore, attributes that contribute more will have a higher value and scores that are too low will be removed. Like the Correlation Attribute value, Ranker search method must be used in conjunction.

```
=== Attribute Selection on all input data ===

Search Method:
	Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 21 Churn):
	Information Gain Ranking Filter

Ranked attributes:
 0.5969661    4 Phone Number
 0.0773975   10 Total Day Charge
 0.0773975    8 Total Day Min
 0.0500934   20 No of Calls Customer Service
 0.0368789    5 Inter Plan
 0.0180031    1 State
 0.0082165    6 VoiceMail Plan
 0.0082165    7 No of Vmail Mesgs
 0.0072895   18 Total Int Calls
 0.0067401   19 Total Int Charge
 0.0067401   17 Total Int Min
 0.0054209   13 Total Evening Charge
 0.0054209   11 Total Evening Min
 0.0000383    3 Area Code
 0           14 Total Night Minutes
 0            2 Account Length
 0            9 Total Day calls
 0           15 Total Night Calls
 0           12 Total Evening Calls
 0           16 Total Night Charge

Selected attributes: 4,10,8,20,5,1,6,7,18,19,17,13,11,3,14,2,9,15,12,16 : 20
```

Figure 10. This was the result of initial InfoGainAttributeEval using default dataset without removing any attributes

3.6 Decisions to remove/retain attributes

The first ranked attribute using this method was ‘Phone Number’. Logically, this does not make much sense since this can imply that the phone number a customer has can determine the churn outcome. This in our opinion happened by chance and decided that we would take out this attribute before running this test again.

Knowing this, before we ran this test again, with general consensus of other data scientists in our team, we have decided to remove attributes which have or should not have an impact when it comes to churn. These were the nominal data: State, Area Code, Phone Number. State in our opinion was a safe attribute to remove since each state appears to have a similar ratio of about 16.9% churn rate for each state (red part of the bar graph). We believe the 3 above mentioned attributes will not be helpful towards our machine learning algorithm since where and what phone service is provided will not be very insightful; it would be better to take these out to have a more reasonable machine learning result.

If we can make a suggestion to the company, the data selected for which state a customer is selected should be more equalized. For instance, we have noticed that CA (California) is the lowest number of instances yet CA is the most populous State in the United States. To do this study, all the states should randomly sampled equal number of instances for each state to have a general picture of the US. To make suggestions for each individual state office (we noticed that this telecom company had customers in all of 51 states), an idiosyncratic study for each study should be conducted to offer the best advice for each state office.

These were one of the cases where human intervention was required to make a conscious decision on how to go about filtering data than relying purely on machine learning algorithms.

Account length is also removed because infogain was = 0 on numerous attempts.

This was the result after cleaning the ‘unnecessary’ nominal data:

From Figure 10, we have noticed that usage and charges have the same info gain (ex. Total Day Min is equal to Total Day Charge at 0.0774) and also correlation between them were 1.0. Therefore, we have discovered more attributes that we can filter out to simply our model. We have decided to take out ‘charge’ to take out the redundancy (although taking out ‘minutes’ is another viable alternative).

It appears that there were many duplicates: charges, number of voicemail compared to voicemail (yes/no). So far, these are the candidates for attribute removal.

Although we were using infogaineval and correlationatrributeeval to mostly remove attributes, when deciding upon whether or not to remove/retain minutes called vs number of calls, we have decided to retain that relationship since there were a difference in infogain values to consider retaining those values.

3.7 Summary of data preparation

Following steps are used for data preparation.

1) Manually removed the attributes that are not relevant to churn decision: State, phone number and area code
2) Manually removed the attributes that are duplicated in terms of correlation result: Total Day Charge, Total Evening Charge and Total Night Charge and No. of Vmail Msgs
3) CfsSubsetEval, InfoGainAttributeEval and CorrelationAttributeEval are used to select best attributes to use for classification
4) Balance the dataset by using ClassBalancer

We have found out that we had to make a conscious ‘human’ decision in deciding which attributes to retain/remove as relying purely on the machine on this task did not have optimal outcomes.

12 attributes are selected: Inter Plan, VoiceMail Plan, Total Day Min, Total day charge, total evening min, total int min, total int calls, total int charge, no of calls customer service.

No.		Name
1		Inter Plan
2		VoiceMail Plan
3		Total Day Min
4		Total Day calls
5		Total Evening Min
6		Total Evening Calls
7		Total Night Minutes
8		Total Night Calls
9		Total Int Min
10		Total Int Calls
11		No of Calls Customer Service
12		Churn

Fig 11. Final attribute selections

4. Predictive Modeling

4.1 Data Split Strategy

The Data Split Strategy we used is 10-Fold Cross Validation. There are basically two common options: 3-way data splitting and K-Fold Cross Validation. The 3-way data splitting will hold 40 percent of data set for validation and testing. It can reduce the risk of overfitting, but at the same time, it might lose some important patterns. However, for K Fold Cross Validation, the data is split by K subsets and one of the subsets is used as the validation set and other K-1 subsets are put together to form a training set. The method is repeated K times with different validation set and the average of K trails is used to get total effectiveness of the Model. The advantage of this method is that it can reduce the overfitting and underfitting risk at the same time, since every data point gets to be in training set, it also gets to be in a validation set exactly once. Considering K value, 10-Fold split is most common used especially for sample size over 1000. In addition, less fold split might increase the risk of underfitting for each trail and eventually affects overall results and more fold split will take a longer time in the result. Therefore, 10-Fold Cross Validation is the best approach.

4.2 Applying Classification Algorithm

Three models were chosen to perform the classification algorithm: J48 Decision Tree, Naïve Bayes, and Random Forest

4.2.1 Decision Tree

First, we run the baseline model for J48 decision tree with original attributes and using training set. The accuracy result we got is 95.56%. However, the original dataset is unbalanced, churn is considered significantly rare event compared to not churn. Though the result is very high, the TP rate for churn is not high. This baseline model is overfitting. Then we rerun it with our selected attributes (balanced data) using default settings. We use ReducedErrorPruning in order to optimize the size of tree (reduced the risks of overfitting). The accuracy is improved from 83.09% to 86.42%. Thus, we kept ReducedErrorPruning as true and keep optimizing the tree through changing the minNumObj. Finally, we set it to 9, and this is not the best accuracy result we got. However, this gives a better tree to visualized with relatively simple shape and less risk of overfitting. Below is the performance results comparison table with mentioned setting.

	C 0.25 M 2 Default Setting Original Dataset	C 0.25 M 2 Default Setting	C 0.25 M 2 Reduced Error Pruning	C 0.25 M 9 Reduced Error Pruning
Accuracy	95.56%	83.09%	86.42%	86.42%
TP Rate	0.743	0.743	0.812	0.816
FP Rate	0.008	0.081	0.083	0.087
Precision	0.937	0.901	0.907	0.903
Recall	0.743	0.743	0.812	0.816

Figure 12. Performance Results of Decision Tree For True Class

Below is J48 Decision Tree with confidence factor equals to 0.25, minimum number of object equals to 9 and reduced error pruning. As we can see, when the number of calls to customer service (root node) is more than 3 times, the customer likely to churn. Also, Total Day Min, International plan and voiceMail plan are the effective attributes for churn, most of these attributes appears many times. People who have international plan likely to churn as well.

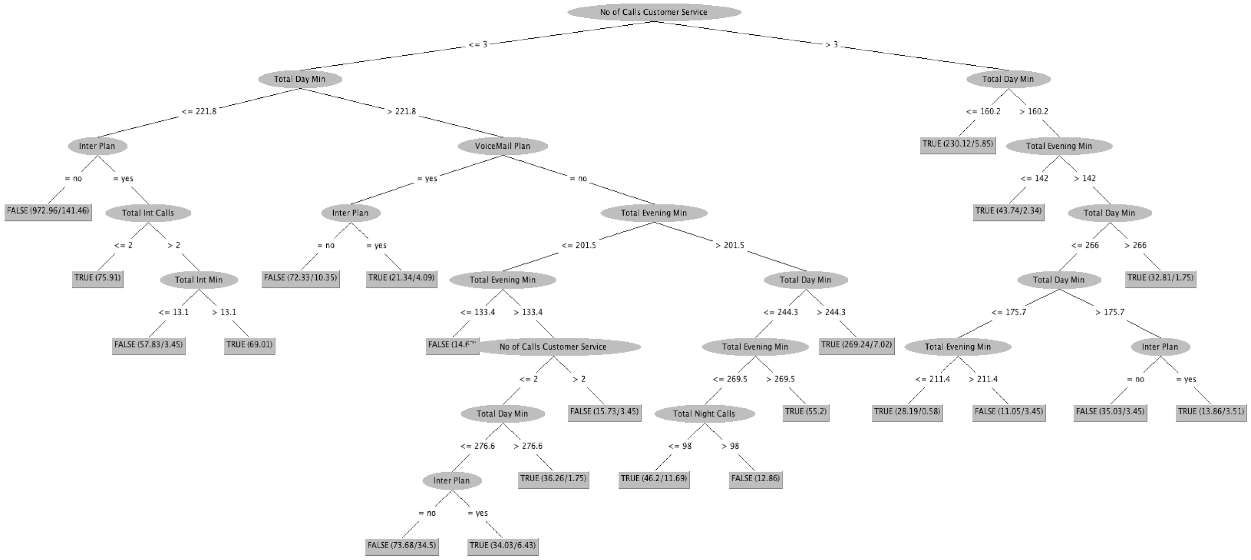

Figure 13. Decision Tree

4.2.2 Naive Bayes

Naive Bayes calculates the posterior probability for each class and makes a prediction for the class with the highest probability. We first ran Naive Bayes with default settings. Then we use supervised discretization to convert numeric attributes to nominal ones, since Naive Bayes performed better with categorical attributes. As we can see, the result didn't change much, and accuracy result is significantly lower than results from decision tree. Additionally, most of the attributes we chose are numeric. Decision Tree performed better with numeric attributes over Naive Bayes. In this case, we not gonna use Naive Bayes

	Naive Bayes Default Setting	Naive Bayes Use Supervised Discretization	Decision Tree C 0.25 M 9 Reduced Error Pruning
Accuracy	81.90%	81.67%	86.42%
TP Rate	0.819	0.817	0.816
FP Rate	0.181	0.180	0.087

Precision	0.819	0.819	0.903
Recall	0.819	0.814	0.816

Figure 14. Performance Results of Naive Bayes vs Decision Tree for True Class

4.2.3 Random Forest

Random forests is generally considered a better model if the goal is for prediction. In other words, we'd want to reduce the variance of the model. For example, the built-in OOB validation error rate is handy and can be efficiently implemented. Random forest is a bagged decision tree model that split on a subset of features on each split. First we break this down by first looking at a single decision tree which we have done in 4.2.1, then discussing bagged decision trees and finally introduce splitting on a random subset of features. The final predicted value is the average value of all our *X* decision trees. One single decision tree has high variance (tends to overfit), so by bagging or combining many weak learners into strong learners, we are averaging away the variance. One of the great advantages of Decision Tree-based models is their interpretability we can understand the reasoning behind each prediction. Using such a white-box model also gives us the ability to reason about which features were most important. Random forest handles outliers by essentially accumulating them. It is also indifferent to non-linear features, while protecting from individual errors, data set is in the form of binary features as random forest requires very little pre-processing and data does not need to be rescaled or transformed therefore this method was chosen.

When we try to change the setting, the change are all minimal, as a result states in Figure 15, we choose the one with best accuracy result. Kappa statistic shows that 0.78 accuracy between the classified attributes showing they are positively correlated to one another with each other. People who churn are based on these following attributes this finding was confirmed through the low root mean squared error.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       2978.6749          89.3692 %
Incorrectly Classified Instances      354.3251          10.6308 %
Kappa statistic                          0.7874
Mean absolute error                      0.1976
Root mean squared error                  0.3161
Relative absolute error                 39.5223 %
Root relative squared error             63.2144 %
Total Number of Instances             3333
```

	Random Forest Default Setting	Random Forest
Accuracy	89.3692%	89.6973%
TP Rate	0.976	0.818
FP Rate	0.188	0.024
Precision	0.838	0.972
Recall	0.976	0.818

```
=== Classifier model (full training set) ===

RandomForest

Bagging with 200 iterations and base learner

weka.classifiers.trees.RandomTree -K 6 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Attribute importance based on average impurity decrease (and number of nodes using that attribute)

      0.41 (   879)  Inter Plan
      0.4  (  7190)  Total Day Min
      0.4  (  1297)  VoiceMail Plan
      0.39 (  4238)  Total Day calls
      0.33 (  3227)  Total Evening Calls
      0.32 (  5733)  Total Evening Min
      0.29 (  3225)  Total Int Min
      0.28 (  4129)  Total Night Minutes
      0.28 (  1753)  Total Int Calls
      0.28 (  2915)  Total Night Calls
      0.25 (  1212)  No of Calls Customer Service
```

Figure 15. Performance Results of Random Forest Tree for True Class & Importance output (setting we chose)

4.3 Summary of predictive modeling

	Decision Tree	Naive Bayes	Random Forest
Accuracy	86.42%	81.67%	89.70 %
TP Rate	0.816	0.817	0.818
FP Rate	0.087	0.180	0.024
Precision	0.903	0.819	0.972
Recall	0.816	0.814	0.818

Figure 16. Best Performance Results of Different Classifiers

Overall, Random forest performs the best and it gives the best predictive result for decision making compared to other models as we can see from the figure 4.3.

5. Post Predictive Analysis

5.1 K-means Algorithm

SimpleKMeans is used for post predictive analysis. The original dataset and the selected data from the original dataset are used for k-mean. Only churn true dataset is retrieved. Seed is set to 25 and algorithm ran 9 times with cluster size from 2 to 10. Using Elbow method, within cluster sum of squared errors are recorded to find the best cluster size in figure 17. The best cluster size is found to be 4. However, it is hard to find elbow location on graph from original data. We decided to use the same cluster size as one from the first graph.

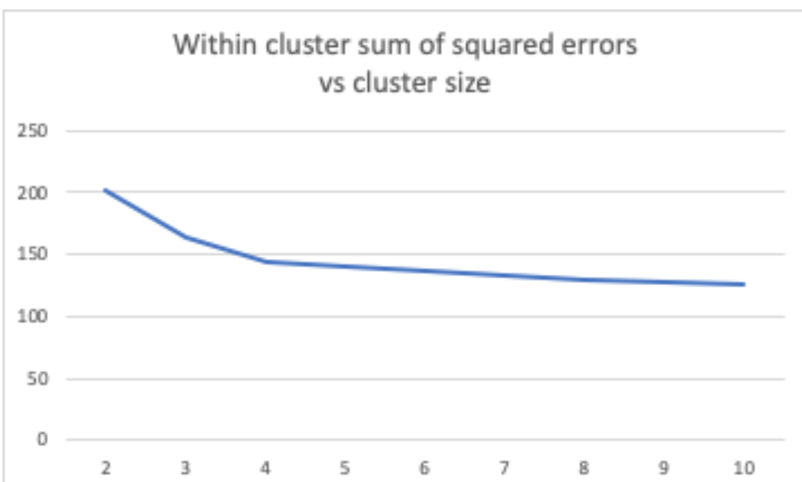

Figure 17 Within cluster sum of squared errors vs cluster size from selected dataset

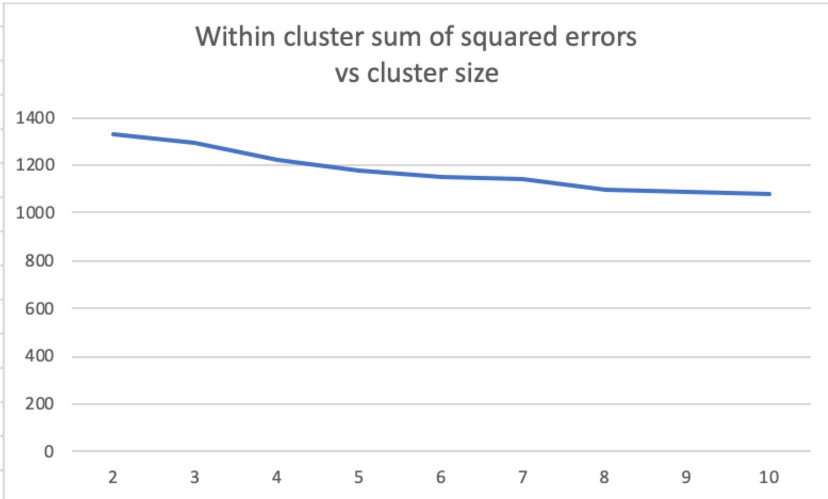

Figure 18 Within cluster sum of squared errors vs cluster size from original dataset

```
Final cluster centroids:
                                Cluster#
Attribute            Full Data          0          1          2          3
                       (483.0)     (51.0)    (211.0)    (101.0)    (120.0)
==========================================================================
Int.l.Plan                  no        yes         no        yes         no
VMail.Plan                  no        yes         no         no         no
Day.Mins              206.9141   185.0255   257.8521   193.9149   137.5917
Day.Calls             101.3354   100.2549   103.5545    97.7921    100.875
Eve.Mins              212.4101   210.0667    234.046   208.4129   178.7275
Eve.Calls             100.5611   102.5294   101.0806    99.0891     100.05
Night.Mins            205.2317   189.2882   217.3474   197.2139   197.4525
Night.Calls           100.3996   103.6275   100.9194   100.9307    97.6667
Intl.Mins                 10.7    11.9451    10.3412     11.599     10.045
Intl.Calls              4.1636     4.9216     4.3839     3.5743       3.95
CustServ.Calls          2.2298     1.6863     1.4171     1.5644       4.45
Churn.                  True.      True.      True.      True.      True.




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       51 ( 11%)
1      211 ( 44%)
2      101 ( 21%)
3      120 ( 25%)
```

Figure 19 clusters characters from selected dataset

		Cluster#			
Attribute	Full Data	0	1	2	3
	(483.0)	(116.0)	(79.0)	(112.0)	(176.0)
State	TX	MI	NJ	TX	MD
Account.Length	102.6646	107.6121	104.2785	97.4554	101.9943
Area.Code	437.8178	412.6466	440.7089	500.4821	413.233
Phone	329-6603	329-6603	383-6029	374-8042	351-7269
Int.l.Plan	no	no	no	no	no
VMail.Plan	no	no	yes	no	no
VMail.Message	5.1159	0	30.8861	0.2768	0
Day.Mins	206.9141	147.0647	176.2443	207.3509	259.8489
Day.Calls	101.3354	99.4828	101.481	102.0625	102.0284
Day.Charge	35.1759	25.0012	29.9624	35.2501	44.1749
Eve.Mins	212.4101	182.2233	202.662	214.042	235.6432
Eve.Calls	100.5611	101.4052	101.3291	100.4911	99.7045
Eve.Charge	18.055	15.4891	17.2267	18.1939	20.0295
Night.Mins	205.2317	196.6276	192.9684	208.6759	214.2153
Night.Calls	100.3996	101.0517	101.8228	101.75	98.4716
Night.Charge	9.2355	8.849	8.6832	9.3903	9.6398
Intl.Mins	10.7	10.9724	11.1165	10.525	10.4449
Intl.Calls	4.1636	3.931	4.6203	3.9821	4.2273
Intl.Charge	2.8895	2.9629	3.0019	2.8421	2.8209
CustServ.Calls	2.2298	3.2845	2.6076	2.0357	1.4886
Churn.	True.	True.	True.	True.	True.

```
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      116 ( 24%)
1       79 ( 16%)
2      112 ( 23%)
3      176 ( 36%)
```

Figure 20 clusters characters from original dataset

Figure 19 and 20 give same similar pattern. So we decide to use figure 19 to simplify our finding. Figure 19 shows the characteristic of 4 clusters and cluster 0 and 2 are similar characteristics in terms of plan attributes and mins used. We categorize two clusters as same and explain them as one.

Following chart is a further analysis from kmean result. Cluster 4 has low day mins per call usage but average night mins per call. Also it has above average no. of customer calls. Cluster 2 have high mins per call usage on day and night and average no of customer calls. Cluster 1 has average mins per call usage but have voice mail and international plan

	Cluster 1	Cluster 2	Cluster 4
Day mins per call	1.85 mins	2.50 mins	1.37 mins
Day min vs mean	185 vs 179	257 vs 179	137 vs 179
Eve mins per call	2.05	2.31	1.78
Eve min vs mean	210 vs 209	234 vs 209	178 vs 209
Night mins per call	1.83	2.17	2.03
Night mins vs mean	189 vs 200	217 vs 200	197 vs 200
Customer call vs mean	1.68 vs 1.31	1.42 vs 1.31	4.45 vs 1.31

Figure 20. Comparison Chart of clusters

We can conclude each cluster in the following:
First and third cluster characteristic: Customer who used phone as average consumption but have voice mail and international plan will churn. 32% of customers belongs to this cluster.
Second cluster characteristic: Churn True: Customer who used phone as high consumption and doesn't have a voicemail and international plan will churn. 44% of customers belongs to this cluster.
Fourth cluster characteristic: Churn True: Customer who used phone as low consumption and doesn't have a voicemail and international plan will churn. 25% of customers belongs to this cluster.

5.2 Summary of Post Predictive Analysis

The apriori algorithm cannot be applied to this dataset since all the numerical attributes could not be able to convert to nominal attributes. Even running apriori with nominal attributes available, it won't give good results since dataset doesn't include major attributes that will contribute to a decision. Kmean gives good validated results that solution to the problem the company faces. We found 3 clusters that f

6. Conclusion and Recommendations

Here are the conclusions based on the tests conducted and the following steps were taken:

First step which was used was to clean the data where we concluded that only 12 attributes and one classifier (churn) were used. These attributes helped us get further precision in predicting future churning by removing noise as per explanation in section 1.

Second step is once we have cleaned the data we focused on running different algorithms to find the best algorithm to predict the model. Random tests that we have conducted to investigate churn are: Decision Tree, Naive Bayes, Random Forest, and K-means Cluster Algorithms. Naive baye was least accurate as it works best with nominal data and we only have two nominal data in our dataset. Therefore, it did not help in our predictions. Decision tree was used to predict the model and we have a better accuracy rate than using Naive bayes but not less accuracy rate than using Random Forest.

Random forest is the best one to use as mentioned above as it averages 100 decision tree and provides four top attributes which impact the churning is int plan, total day min, voicemail plan, total day calls affect the overall compared to the benchmark which chose phone number and gave 100% accuracy which means it's over fitting.

K-mean Clustering algorithm is used to predict the model. It gave 3 clusters that describe the customer who would be likely to churn in the future. First cluster describes as average mins usage and have both plans; Second cluster describes as high mins usage and no plans; Third cluster describes as low mins usage but high frequency of call number, no plans and high volume of calling customer service.

From our analysis, we were able to conclude that the most important attributes to consider to make a decision were: Inter Plan, Daytime Charges, Voicemail Plan and No of Customer Calls.

What we have noticed was that customers who churn have made several calls to the customer centre. The critical number seems to be 3; as customers who go beyond this point has lost patience. Therefore, we would like to state that customers do give an opportunity with the company. We would like to state that customer service is of utmost importance and this would be the best way to find out for sure why customers are churning. Our data team were able to verify other possible reasons for churn but the best method in our opinion would be to hear why customers churn directly from the source.

We have found that people who have International Plans and Voicemail Plan are less likely to churn. It seems that they will be less likely to churn once the customer has more services with us. Thus an advice to the company is to have the customers to sign up with more features since the likelihood people will switch will be less.

Lastly, we would like to mention that charges might need some revision. We have noticed that number of minutes and charges have a perfect correlation. We would like the company to introduce phone plans to satisfy customers who would churn early. They used the least amount of minutes and made most amount of phone calls and the evidence is from K-means tests.

In conclusion, we would advise the company to understand why customers call to customer service and to attempt to resolve the issues with the customer on average within the customers' third call towards the centre. Since customers provide several chances, there seems to be less issue with branding of the company but rather the servicing provided. We would also advise to let customers have many features as possible since customers with more features are less likely to churn. Finally, we would advise to have different tiers for phone plans since this company seems to run in a pay as you go model. We would advise to have plans that would satisfy most amount of customers possible. We recommend these advice that will help to churn the customers.