Classification Decision Tree

# Part I: Research Question

Which customers are likely to churn?

The goal in answering this question is to decrease costs associated with portfolio maintenance. As the cost to on-board a new customer is already known to be 10 times that of retaining an existing customer, one way to cut costs is by raising retention rates among the existing customer base. Therefore, it would be a valuable undertaking for an organization to have the ability to identify existing customers likely to churn. Armed with this knowledge, new strategic initiatives could be developed as needed to target the retention of highly valued existing customers.

The research question will be addressed using a decision tree classification model.

# Part II: Method Justification

The classification tree method is chosen as it functions to determine classifications by using tree like decision flows which are transparent and relatively easy to follow. The data will flow through various levels of if/else decision points until arriving at a determination. Each branch indicates decision points, while the leaves of the tree indicate the determinations. In this case, the model will be trained using a set of historical observations. The model will use those historical observations to create decision points using the data attributes determining levels within the attributes at which the data can be split. The decision point splitting will continue until a determination can be reached. Then, the data set on which the predictions are to be made will be fed into the model thereby making classifications for each new observation. The assumption of the classification decision tree model is that the target variable will be categorical. Having the target variable as categorical means that it will be of an unordered data type which can be grouped. In this case, the target variable is categorical because the valid values are “Yes” or “No” for churn. If the target variable values were of an ordered and numerical data type (such as with measurements like height, width, age, etc..) then this would violate the classification decision tree model assumption and a different model would have to be used such as a regression decision tree model.  
R will be used for this analysis. R is open source software that was specifically made for statistical analysis. Using R, we can ingest the data set, and leveraging an extensive library of data manipulation and visualization packages, perform the necessary classification steps. More information can be found on the R project website (<https://www.r-project.org/>). The dplyr package will be used for data preparation and manipulation within R. More information for the dplyr package can be found on the tidyverse website (<https://dplyr.tidyverse.org/>). The rpart package will be used to construct the decision tree classification model. More information for the rpart package can be found on the Comprehensive R Archive Network (CRAN) website (<https://cran.r-project.org/>). Finally, the rpart.plot package will be used for visualizing the model. More information for the rpart.plot package can be found on the CRAN website (<https://cran.r-project.org/>).

# Part III: Data Preparation

To prepare the data, first a check is run for missing values in the data set using the sapply function looking for na’s in the data set. No action is needed as no missing values were detected. However, had missing values been detected more analysis would have been required for each instance to determine handling of the records containing nulls. Depending on the variable in view, the nulls could be filled with variable mean or median values, or even removed from the data set. Second, prior to creating the classification model, the data will be prepared by selecting only the variables which will be used in model construction to predict whether a customer will churn. Several variables which are assumed to be irrelevant to churn, such as latitude and longitude, will be excluded. First, it is necessary to investigate the data set to distinguish between discrete, continuous, and categorical variables. Discrete variables are those numerical values which can be counted. The discrete variables to be used are as follows: 1. Children 2. Age 3. Item 1 4. Item 2 5. Item 3 6. Item 4 7. Item 5 8. Item 6 9. Item 7 10. Item 8 Continuous variables are those numerical values which are not limited to whole numbers, likely to be measurements. The continuous variables to be used are as follows: 1. Income 2. Outage\_sec\_perweek 3. Tenure 4. MonthlyCharge 5. Bandwidth\_GB\_Year Categorical variables are those variables which are to be grouped and unordered. The categorical variables to be used from the original data set are as follows: 1. Churn (target variable) 2. Area Third, it is best to transform the any obscure variable names into a more readily understood naming convention. This is done with the variables “Item 1…. Item8” using the data dictionary provided. Finally, the variables to be used in the decision tree model are selected and the cleaned data set is provided in the Cleaned\_Data.csv attachment.

library(dplyr)

## Warning: replacing previous import 'vctrs::data\_frame' by 'tibble::data\_frame'  
## when loading 'dplyr'

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Load Data Set  
df<-read.csv("c:/users/shua/documents/Data Mining\_D209/churn\_clean.csv")  
  
# Check for missing values  
sapply(df, function(x) sum(is.na(x)))

## CaseOrder Customer\_id Interaction   
## 0 0 0   
## UID City State   
## 0 0 0   
## County Zip Lat   
## 0 0 0   
## Lng Population Area   
## 0 0 0   
## TimeZone Job Children   
## 0 0 0   
## Age Income Marital   
## 0 0 0   
## Gender Churn Outage\_sec\_perweek   
## 0 0 0   
## Email Contacts Yearly\_equip\_failure   
## 0 0 0   
## Techie Contract Port\_modem   
## 0 0 0   
## Tablet InternetService Phone   
## 0 0 0   
## Multiple OnlineSecurity OnlineBackup   
## 0 0 0   
## DeviceProtection TechSupport StreamingTV   
## 0 0 0   
## StreamingMovies PaperlessBilling PaymentMethod   
## 0 0 0   
## Tenure MonthlyCharge Bandwidth\_GB\_Year   
## 0 0 0   
## Item1 Item2 Item3   
## 0 0 0   
## Item4 Item5 Item6   
## 0 0 0   
## Item7 Item8   
## 0 0

# Identify numeric columns for use  
str(df)

## 'data.frame': 10000 obs. of 50 variables:  
## $ CaseOrder : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Customer\_id : chr "K409198" "S120509" "K191035" "D90850" ...  
## $ Interaction : chr "aa90260b-4141-4a24-8e36-b04ce1f4f77b" "fb76459f-c047-4a9d-8af9-e0f7d4ac2524" "344d114c-3736-4be5-98f7-c72c281e2d35" "abfa2b40-2d43-4994-b15a-989b8c79e311" ...  
## $ UID : chr "e885b299883d4f9fb18e39c75155d990" "f2de8bef964785f41a2959829830fb8a" "f1784cfa9f6d92ae816197eb175d3c71" "dc8a365077241bb5cd5ccd305136b05e" ...  
## $ City : chr "Point Baker" "West Branch" "Yamhill" "Del Mar" ...  
## $ State : chr "AK" "MI" "OR" "CA" ...  
## $ County : chr "Prince of Wales-Hyder" "Ogemaw" "Yamhill" "San Diego" ...  
## $ Zip : int 99927 48661 97148 92014 77461 31030 37847 73109 34771 45237 ...  
## $ Lat : num 56.3 44.3 45.4 33 29.4 ...  
## $ Lng : num -133.4 -84.2 -123.2 -117.2 -95.8 ...  
## $ Population : int 38 10446 3735 13863 11352 17701 2535 23144 17351 20193 ...  
## $ Area : chr "Urban" "Urban" "Urban" "Suburban" ...  
## $ TimeZone : chr "America/Sitka" "America/Detroit" "America/Los\_Angeles" "America/Los\_Angeles" ...  
## $ Job : chr "Environmental health practitioner" "Programmer, multimedia" "Chief Financial Officer" "Solicitor" ...  
## $ Children : int 0 1 4 1 0 3 0 2 2 1 ...  
## $ Age : int 68 27 50 48 83 83 79 30 49 86 ...  
## $ Income : num 28562 21705 9610 18925 40074 ...  
## $ Marital : chr "Widowed" "Married" "Widowed" "Married" ...  
## $ Gender : chr "Male" "Female" "Female" "Male" ...  
## $ Churn : chr "No" "Yes" "No" "No" ...  
## $ Outage\_sec\_perweek : num 7.98 11.7 10.75 14.91 8.15 ...  
## $ Email : int 10 12 9 15 16 15 10 16 20 18 ...  
## $ Contacts : int 0 0 0 2 2 3 0 0 2 1 ...  
## $ Yearly\_equip\_failure: int 1 1 1 0 1 1 1 0 3 0 ...  
## $ Techie : chr "No" "Yes" "Yes" "Yes" ...  
## $ Contract : chr "One year" "Month-to-month" "Two Year" "Two Year" ...  
## $ Port\_modem : chr "Yes" "No" "Yes" "No" ...  
## $ Tablet : chr "Yes" "Yes" "No" "No" ...  
## $ InternetService : chr "Fiber Optic" "Fiber Optic" "DSL" "DSL" ...  
## $ Phone : chr "Yes" "Yes" "Yes" "Yes" ...  
## $ Multiple : chr "No" "Yes" "Yes" "No" ...  
## $ OnlineSecurity : chr "Yes" "Yes" "No" "Yes" ...  
## $ OnlineBackup : chr "Yes" "No" "No" "No" ...  
## $ DeviceProtection : chr "No" "No" "No" "No" ...  
## $ TechSupport : chr "No" "No" "No" "No" ...  
## $ StreamingTV : chr "No" "Yes" "No" "Yes" ...  
## $ StreamingMovies : chr "Yes" "Yes" "Yes" "No" ...  
## $ PaperlessBilling : chr "Yes" "Yes" "Yes" "Yes" ...  
## $ PaymentMethod : chr "Credit Card (automatic)" "Bank Transfer(automatic)" "Credit Card (automatic)" "Mailed Check" ...  
## $ Tenure : num 6.8 1.16 15.75 17.09 1.67 ...  
## $ MonthlyCharge : num 172 243 160 120 150 ...  
## $ Bandwidth\_GB\_Year : num 905 801 2055 2165 271 ...  
## $ Item1 : int 5 3 4 4 4 3 6 2 5 2 ...  
## $ Item2 : int 5 4 4 4 4 3 5 2 4 2 ...  
## $ Item3 : int 5 3 2 4 4 3 6 2 4 2 ...  
## $ Item4 : int 3 3 4 2 3 2 4 5 3 2 ...  
## $ Item5 : int 4 4 4 5 4 4 1 2 4 5 ...  
## $ Item6 : int 4 3 3 4 4 3 5 3 3 2 ...  
## $ Item7 : int 3 4 3 3 4 3 5 4 4 3 ...  
## $ Item8 : int 4 4 3 3 5 3 5 5 4 3 ...

# Identify column order for renaming  
colnames(df)

## [1] "CaseOrder" "Customer\_id" "Interaction"   
## [4] "UID" "City" "State"   
## [7] "County" "Zip" "Lat"   
## [10] "Lng" "Population" "Area"   
## [13] "TimeZone" "Job" "Children"   
## [16] "Age" "Income" "Marital"   
## [19] "Gender" "Churn" "Outage\_sec\_perweek"   
## [22] "Email" "Contacts" "Yearly\_equip\_failure"  
## [25] "Techie" "Contract" "Port\_modem"   
## [28] "Tablet" "InternetService" "Phone"   
## [31] "Multiple" "OnlineSecurity" "OnlineBackup"   
## [34] "DeviceProtection" "TechSupport" "StreamingTV"   
## [37] "StreamingMovies" "PaperlessBilling" "PaymentMethod"   
## [40] "Tenure" "MonthlyCharge" "Bandwidth\_GB\_Year"   
## [43] "Item1" "Item2" "Item3"   
## [46] "Item4" "Item5" "Item6"   
## [49] "Item7" "Item8"

colnames(df)[43:50]

## [1] "Item1" "Item2" "Item3" "Item4" "Item5" "Item6" "Item7" "Item8"

colnames(df)[43:50]<-list("Timely.Response", "Timely.Fixes", "Timely.Replacements", "Reliability", "Options", "Respectful.Response", "Courteous.Exchange", "Active.Listening")  
colnames(df)

## [1] "CaseOrder" "Customer\_id" "Interaction"   
## [4] "UID" "City" "State"   
## [7] "County" "Zip" "Lat"   
## [10] "Lng" "Population" "Area"   
## [13] "TimeZone" "Job" "Children"   
## [16] "Age" "Income" "Marital"   
## [19] "Gender" "Churn" "Outage\_sec\_perweek"   
## [22] "Email" "Contacts" "Yearly\_equip\_failure"  
## [25] "Techie" "Contract" "Port\_modem"   
## [28] "Tablet" "InternetService" "Phone"   
## [31] "Multiple" "OnlineSecurity" "OnlineBackup"   
## [34] "DeviceProtection" "TechSupport" "StreamingTV"   
## [37] "StreamingMovies" "PaperlessBilling" "PaymentMethod"   
## [40] "Tenure" "MonthlyCharge" "Bandwidth\_GB\_Year"   
## [43] "Timely.Response" "Timely.Fixes" "Timely.Replacements"   
## [46] "Reliability" "Options" "Respectful.Response"   
## [49] "Courteous.Exchange" "Active.Listening"

# Select Columns for Model  
df2<-df%>%select(Churn, Area, Children, Age, Income, Outage\_sec\_perweek, Tenure, MonthlyCharge, Bandwidth\_GB\_Year, Timely.Response, Timely.Fixes, Timely.Replacements, Reliability, Options, Respectful.Response, Courteous.Exchange, Active.Listening)  
  
# Write cleaned data set  
write.csv(df, "c:/users/shua/documents/Data Mining\_D209/Cleaned\_Data.csv")

# Part IV: Analysis

To begin, the analysis the data is split into training and test data. In this case, 75% of the data set provided will be used as a training data set. This fits the model with historical observations which the model will use to make future predictions. The remaining 25% of the data set will be the test data on which the predictions of churn will be made. The split data sets are stored in the “trainDF” and “testDF” data frames respectively. An analysis of the train and test data frames using R’s summary function confirms the data has been appropriately normalized and split. Now having the data and split into training and test data sets, the classification model is built. First the model is built using only the predictor variables of Tenure and Area to determine the churn outcome. The resulting accuracy of this approach is 74.36%. Next, the model is rerun using all of the predictor variables in the cleaned data set. The resulting accuracy shows some improvement at 81.8%. A visualization of the model at this point reveals over-plotting which reduces the transparency benefit of the decision tree model. In order to reduce the over-plotting, pruning techniques will be used. Using the rpart.plot function the appropriate complexity parameter (cp) can be visualized. In this case, the plot indicates that a cp value of 0.0026 may produce a more accurate model. Additionally, to help resole the over-plotting issues a maxdepth of 6 can also be used. Finally, the model is rerun changing the cp and maxdepth attributes to the aforementioned values. The resulting accuracy nudges even higher to 84.32%.

# Split training and testing  
sampNumbers<-sort(sample(nrow(df2), nrow(df2)\*.75))  
trainDF<-df2[sampNumbers,]  
testDF<-df2[-sampNumbers,]  
summary(trainDF)

## Churn Area Children Age   
## Length:7500 Length:7500 Min. : 0.000 Min. :18.0   
## Class :character Class :character 1st Qu.: 0.000 1st Qu.:35.0   
## Mode :character Mode :character Median : 1.000 Median :53.0   
## Mean : 2.089 Mean :53.2   
## 3rd Qu.: 3.000 3rd Qu.:71.0   
## Max. :10.000 Max. :89.0   
## Income Outage\_sec\_perweek Tenure MonthlyCharge   
## Min. : 348.7 Min. : 0.09975 Min. : 1.005 Min. : 79.98   
## 1st Qu.: 19255.9 1st Qu.: 8.08585 1st Qu.: 7.848 1st Qu.:139.98   
## Median : 33289.0 Median :10.04798 Median :25.378 Median :167.46   
## Mean : 39951.1 Mean :10.03798 Mean :34.127 Mean :172.64   
## 3rd Qu.: 53517.2 3rd Qu.:11.99041 3rd Qu.:61.278 3rd Qu.:200.17   
## Max. :258900.7 Max. :21.20723 Max. :71.999 Max. :290.16   
## Bandwidth\_GB\_Year Timely.Response Timely.Fixes Timely.Replacements  
## Min. : 155.5 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:1224.8 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Median :2766.1 Median :3.000 Median :4.000 Median :3.000   
## Mean :3359.9 Mean :3.496 Mean :3.509 Mean :3.485   
## 3rd Qu.:5562.2 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :7159.0 Max. :7.000 Max. :7.000 Max. :8.000   
## Reliability Options Respectful.Response Courteous.Exchange  
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Median :3.000 Median :3.000 Median :3.000 Median :4.000   
## Mean :3.485 Mean :3.498 Mean :3.494 Mean :3.511   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :7.000 Max. :7.000 Max. :8.000 Max. :7.000   
## Active.Listening  
## Min. :1.000   
## 1st Qu.:3.000   
## Median :3.000   
## Mean :3.493   
## 3rd Qu.:4.000   
## Max. :8.000

summary(testDF)

## Churn Area Children Age   
## Length:2500 Length:2500 Min. : 0.000 Min. :18.00   
## Class :character Class :character 1st Qu.: 0.000 1st Qu.:34.75   
## Mode :character Mode :character Median : 2.000 Median :53.00   
## Mean : 2.085 Mean :52.72   
## 3rd Qu.: 3.000 3rd Qu.:70.00   
## Max. :10.000 Max. :89.00   
## Income Outage\_sec\_perweek Tenure MonthlyCharge   
## Min. : 368.5 Min. : 0.1201 Min. : 1.00 Min. : 79.98   
## 1st Qu.: 19092.0 1st Qu.: 7.8543 1st Qu.: 8.35 1st Qu.:139.97   
## Median : 32963.3 Median : 9.9267 Median :42.53 Median :169.94   
## Mean : 39374.4 Mean : 9.8935 Mean :35.72 Mean :172.57   
## 3rd Qu.: 52515.0 3rd Qu.:11.9033 3rd Qu.:62.09 3rd Qu.:202.44   
## Max. :169580.7 Max. :20.6250 Max. :71.98 Max. :287.64   
## Bandwidth\_GB\_Year Timely.Response Timely.Fixes Timely.Replacements  
## Min. : 243.3 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:1259.8 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Median :3999.7 Median :3.000 Median :3.000 Median :3.500   
## Mean :3489.7 Mean :3.476 Mean :3.493 Mean :3.494   
## 3rd Qu.:5661.7 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :7138.3 Max. :7.000 Max. :7.000 Max. :7.000   
## Reliability Options Respectful.Response Courteous.Exchange  
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Median :4.000 Median :3.000 Median :3.000 Median :3.000   
## Mean :3.535 Mean :3.478 Mean :3.506 Mean :3.506   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :7.000 Max. :6.000 Max. :7.000 Max. :7.000   
## Active.Listening  
## Min. :1.000   
## 1st Qu.:3.000   
## Median :3.000   
## Mean :3.504   
## 3rd Qu.:4.000   
## Max. :7.000

#Build classification model  
library(rpart)  
churn\_model1<- rpart(Churn ~ Tenure + Area, data = trainDF, method="class", control = rpart.control(cp=0, maxdepth = 6))  
  
# Prediction on Test Data Set  
testDF$pred<-predict(churn\_model1, testDF, type="class")  
  
#Confusion Matrix  
table(testDF$Churn, testDF$pred)

##   
## No Yes  
## No 1577 297  
## Yes 300 326

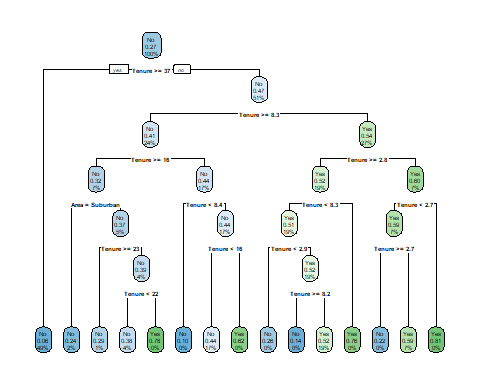
# Model Accuracy  
mean(testDF$Churn == testDF$pred, na.rm=TRUE)

## [1] 0.7612

# Examine Model  
#churn\_model1  
  
# Plot model 1  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

rpart.plot(churn\_model1)



#Remodel with additional variables  
churn\_model2<- rpart(Churn ~ ., data = trainDF, method="class", control = rpart.control(cp=0))  
  
# Prediction on Test Data Set  
testDF$pred<-predict(churn\_model2, testDF, type="class")  
  
#Confusion Matrix  
table(testDF$Churn, testDF$pred)

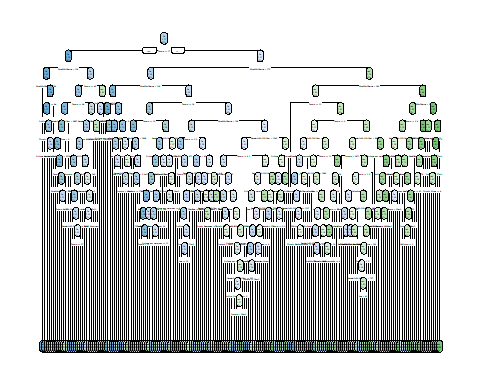
##   
## No Yes  
## No 1654 220  
## Yes 249 377

# Model Accuracy  
mean(testDF$Churn == testDF$pred, na.rm=TRUE)

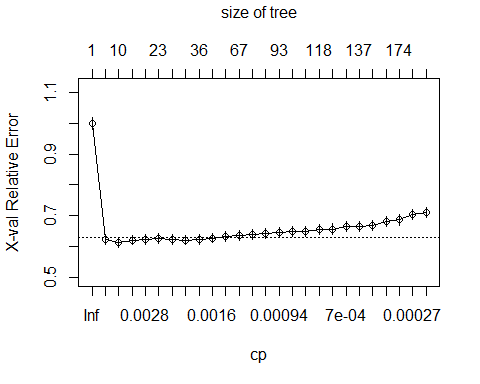
## [1] 0.8124

# Plot model 2  
rpart.plot(churn\_model2)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



# Visualize Post Pruning  
plotcp(churn\_model2)



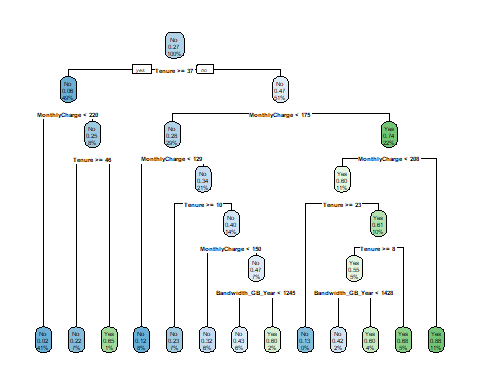
# Post Prune adjusting cp value and maxdepth  
churn\_model3<-rpart(Churn ~ ., data = trainDF, method="class", control = rpart.control(cp=0.0026, maxdepth = 6))  
  
# Prediction on Test Data Set  
testDF$pred<-predict(churn\_model3, testDF, type="class")  
  
#Confusion Matrix  
table(testDF$Churn, testDF$pred)

##   
## No Yes  
## No 1723 151  
## Yes 248 378

# Model Accuracy  
mean(testDF$Churn == testDF$pred, na.rm=TRUE)

## [1] 0.8404

rpart.plot(churn\_model3)



# Part V: Data Summary and Implications

Having found the overall accuracy to be 84.32%, this means that the model is correctly predicting the churn outcome (whether positive or negative churn) 84.32% of the time. The final model can be plotted to reveal the over-plotting issues have been resolved. Further, accuracy is the performance metric for the overall performance of the model, and as in this instance it is more important to predict customer who will churn than customers who will not churn, the sensitivity should also be calculated. For this reason, a confusion matrix is constructed. A confusion matrix shows the number of times that the model predicted all of the relevant outcomes. Using the numbers from the confusion matrix, the sensitivity (the rate at which the model correctly predicted the actual positive churn cases) is shown to be 70.89% (419 actual “Yes” results correctly predicted, 172 actual “Yes” results not correctly predicted) With an accuracy of 84.32% and a sensitivity of 70.89% is performing well but will have some limitations. This analysis did not incorporate all of the variables in the original data set such as latitude and longitude. Also, this analysis only attempted a maxdepth attribute of 6 and did not adjust the minsplit attribute which would direct the model on how many instances for a given grouping before a decision split would be made. Other model attribute values may be tried to yield optimal results. The recommended course of action to the organization is to implement this model for regular use to identify customer likely to churn. These customer can then become part of targeted service retention initiatives, thereby decreasing the organization’s customer on-boarding costs.