**Introduction**

For this case study, I chose to work with the term “customer retention”. The search resulted in 2 datasets, both of which were datasets in the field of telecommunications, and both datasets were in relation to customer churn analysis. One dataset had more than double the amount of observations than the other had.

The dataset I chose is called “Telco Customer Churn”, and I chose this dataset because I knew it would be a challenge for me. Customer retention is something I hadn’t gotten the chance to experiment with in my schooling so I felt this dataset was perfect for testing myself. Another reason I chose the dataset is because I thought it would be interesting to compare my own habits to my findings.

There is an interesting problem to address here, and it’s pretty straightforward: We have 21 variables making up our dataset that profile each customer, which variables are the strongest identifiers of churning, and how can we use the variables to identify a pattern resulting in customer churning? Answering these questions helps to focus in on areas of interest to better understand ways of retaining customers.

I plan to solve the problem using Random Forests and Classification Trees. I will use a 75-25 training and testing data split. The response variable I’ll be working with is “Churn” which is a binary variable with responses of “No” if the customer did not churn, and “Yes” if the customer did churn.

**Analysis**

dat=read.csv("TelcoCustomerChurn.csv", header=TRUE)  
library(VIM)

library(randomForest)

library(tree)

Set seed and create training and testing sets, check for missing values.

set.seed(2019)  
summary(aggr(dat, plot=FALSE))

##   
## Missings per variable:   
## Variable Count  
## customerID 0  
## gender 0  
## SeniorCitizen 0  
## Partner 0  
## Dependents 0  
## tenure 0  
## PhoneService 0  
## MultipleLines 0  
## InternetService 0  
## OnlineSecurity 0  
## OnlineBackup 0  
## DeviceProtection 0  
## TechSupport 0  
## StreamingTV 0  
## StreamingMovies 0  
## Contract 0  
## PaperlessBilling 0  
## PaymentMethod 0  
## MonthlyCharges 0  
## TotalCharges 11  
## Churn 0

dat1=na.omit(dat)  
train = sample(1:nrow(dat1), 0.75\*nrow(dat1))  
dat.train = dat1[train,]  
dat.test = dat1[-train,]

After checking for missing values we see that we have 11 observations with values missing for “TotalCharges”. It is safe to assume that these values are simply missing at random, so we’ll just remove these observations considering they only account for 0.16% of the data.

On to the Random Forest.

dat.rf=randomForest(Churn~.-customerID, mtry=6, importance=TRUE, dat.train)

dat.rf

##   
## Call:  
## randomForest(formula = Churn ~ . - customerID, data = dat.train, mtry = 6, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 6  
##   
## OOB estimate of error rate: 21.05%  
## Confusion matrix:  
## No Yes class.error  
## No 3453 408 0.1056721  
## Yes 702 711 0.4968153

We see that this model has an error rate of 21.05%. Next we’ll view the model’s ability to predict outcomes on the training set, followed by the testing set.

train.rf.pred=predict(dat.rf, dat.train, type = "class")  
table(train.rf.pred, dat.train$Churn)

##   
## train.rf.pred No Yes  
## No 3858 23  
## Yes 3 1390

test.rf.pred=predict(dat.rf, dat.test, type = "class")  
table(test.rf.pred, dat.test$Churn)

##   
## test.rf.pred No Yes  
## No 1184 224  
## Yes 118 232

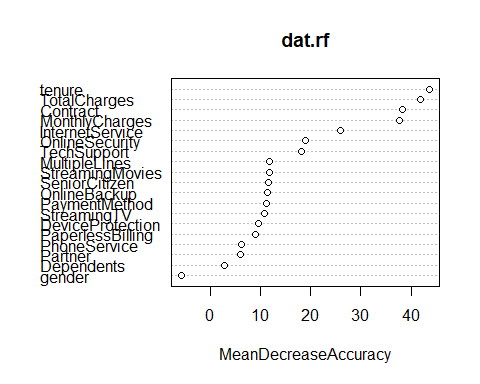
We see that the training data is predicting extremely well with only 26 misclassifications out of 5274. However, this model is not doing a great job with the testing set when predicting that customers will churn. The model has an error rate of 15.9% when predicting a customer will not churn, and a 19.45% error rate overall.

Next we’ll check importance.

imp.df=as.data.frame(round(importance(dat.rf), digits=3))  
imp.df[order(imp.df$MeanDecreaseAccuracy, decreasing = TRUE),]

## No Yes MeanDecreaseAccuracy MeanDecreaseGini  
## tenure 28.549 21.425 43.731 306.494  
## TotalCharges 30.019 16.331 41.844 381.357  
## Contract 2.472 32.623 38.256 184.422  
## MonthlyCharges 23.218 20.505 37.738 361.533  
## InternetService 14.416 23.909 25.842 77.764  
## OnlineSecurity 10.622 25.451 18.917 84.926  
## TechSupport 8.478 26.446 18.176 78.036  
## MultipleLines -0.163 17.287 11.826 43.361  
## StreamingMovies 12.580 -0.523 11.802 33.294  
## SeniorCitizen 10.472 4.168 11.523 42.514  
## OnlineBackup 8.182 9.495 11.432 43.891  
## PaymentMethod 4.487 11.346 11.157 113.287  
## StreamingTV 10.843 0.618 10.824 31.707  
## DeviceProtection 11.040 -3.812 9.514 36.981  
## PaperlessBilling -7.661 20.133 9.014 44.327  
## PhoneService -0.293 7.818 6.147 7.934  
## Partner 7.239 -0.237 5.977 40.528  
## Dependents -1.884 6.147 2.915 35.357  
## gender -5.680 -1.726 -5.684 48.394

varImpPlot(dat.rf, type=1)



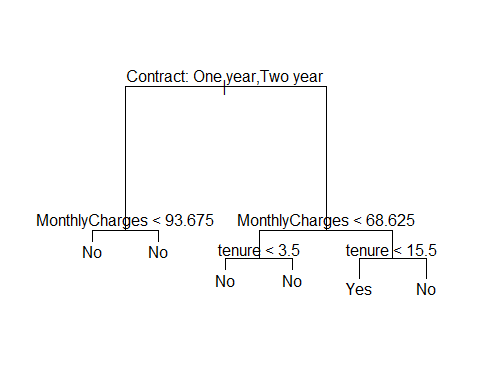
We see from the plot that the 4 most important variables stand out in how important they are. Those variables are tenure, TotalCharges, Contract, and MonthlyCharges. From the table of importance, we see that some of the variables have major differences in how they effect accuracy. For instance, with the variable “Contract”, customers that churn have a much greater impact on the accuracy of the model in comparison to customers that don’t churn.

Let’s use these four variables in a Classification Tree.

tree.dat=tree(Churn~tenure+TotalCharges+Contract+MonthlyCharges, dat.train)  
summary(tree.dat)

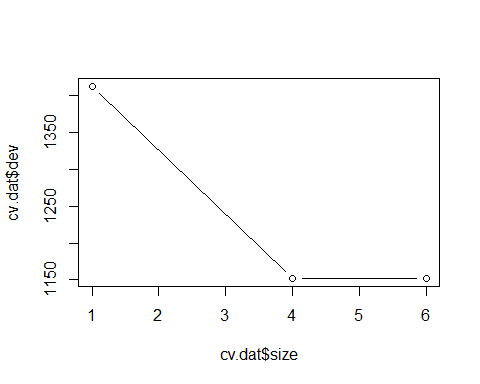
##   
## Classification tree:  
## tree(formula = Churn ~ tenure + TotalCharges + Contract + MonthlyCharges,   
## data = dat.train)  
## Variables actually used in tree construction:  
## [1] "Contract" "MonthlyCharges" "tenure"   
## Number of terminal nodes: 6   
## Residual mean deviance: 0.8962 = 4721 / 5268   
## Misclassification error rate: 0.2139 = 1128 / 5274

plot(tree.dat)  
text(tree.dat, pretty=0)



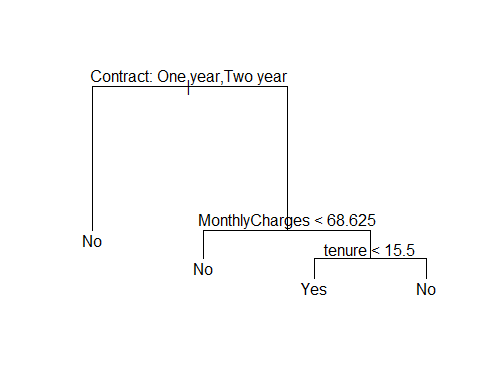
This tree plot is a bit confusing so in attempt to clean things up, we’ll try fine tuning with cross validation and pruning.

cv.dat=cv.tree(tree.dat, FUN=prune.misclass)  
plot(cv.dat$size, cv.dat$dev, type="b")



Based on the plot we’ll use a tree with 4 terminal nodes.

prune.dat=prune.misclass(tree.dat, best=4)  
plot(prune.dat)  
text(prune.dat, pretty=0)



This tree considers Contract to be the most important variable, and has chosen not to use TotalCharges. It suggests that One Year versus Two Year contracts will be most efficient in determining customer churns. Customers on a 1 year contract are most likely not to churn, while two year contract customers churn depending on some Monthly Charges and tenure. If a customer with a two year contract has Monthly Charges less than $68.63 they are more likely to stay than one with greater than $68.63 in Monthly Charges. The tree determines that, a customer with a two year contract, with more than $68.63 in Monthly Charges, is likely to leave sometime around 1 year and 3.5 months with the company.

Let’s see how well this tree can predict outcomes on the training set and testing set.

train.cv.pred=predict(prune.dat, dat.train, type="class")  
table(train.cv.pred, dat.train$Churn)

##   
## train.cv.pred No Yes  
## No 3599 866  
## Yes 262 547

test.cv.pred=predict(prune.dat, dat.test, type="class")  
table(test.cv.pred, dat.test$Churn)

##   
## test.cv.pred No Yes  
## No 1210 276  
## Yes 92 180

Based on these confusion matrices, we can see that the Random Forest Model is performing with slightly better accuracy. Our error rate on the testing set is 20.93% for this model versus 19.45% for the Random Forest model. Also, this model has an error rate of 18.57% when predicting a customer will not churn.

**Conclusion**

We’ve determined that our strongest identifiers for customer churning are tenure, TotalCharges, Contract, and MonthlyCharges. What we can do going forward is use these variables to focus on ways to get more customers to stay away from the tipping point of churning.

The Random Forest Model suggested that Contract was the 3rd most important variable, and the Decision Tree suggested that One Year Contracted customers are least likely to leave. How can we get more customers to sign One Year contracts?

Monthly Charges was considered an important variable suggesting that customers are more likely to leave when their charges exceed $68.63. Is there a way that is not harmful to business, to identify customers who’s Monthly Charges are exceeding this amount, and reduce the amount effectively, causing them to stay?

Random Forest considered TotalCharges as its second most important variable, suggesting that when a customer leaves, it is less likely to reduce the model’s ability to predict churning, than a customer who stays. I’d interpret this as customers are not as bothered by charges signed at the contract as they are by their monthly bill. Can we reduce monthly bill costs and increase fees when a customer first signs the contract?

Finally, tenure seems to play a bigger role with people who’ve signed a two-year contract, shortly after their second year begins, they want to buyout. What is causing this? Are there first year discounts that expire in the second year? Why is our customer happy for a full year, but shortly after is unhappy?

In conclusion, we now know where our focus is. We can begin to dive deeper and strategize around that focus. We’ve raised more questions, but that is important for getting to the best answers. We’ve reached our goal today, and further analysis is needed to reach the ultimate goal.