Undersampling Models

bank <- read.csv("~/Desktop/MSA\_6440/customer\_churn/data/processed/BankChurners\_filtered.csv")  
source('/Users/JonathanVoth/Desktop/R Codes/myfunctions.R')  
  
bank <- bank[,c(-1,-2,-10)]  
  
RNGkind (sample.kind = "Rounding")

## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

set.seed(0)  
p2 <- partition.2(bank, 0.7) # 70:30 partition  
training.data <- p2$data.train  
test.data <- p2$data.test  
  
# Undersampling  
set.seed(0)  
all\_exist <- training.data[which(training.data$Attrition\_Flag == 'Existing Customer'),]  
all\_attrit <- training.data[which(training.data$Attrition\_Flag == 'Attrited Customer'),]  
  
random <- sample(1:nrow(all\_exist), nrow(all\_attrit), replace = FALSE)  
train.data <- all\_exist[random,]  
  
training.under <- rbind(all\_attrit, train.data)  
table(training.under$Attrition\_Flag)

##   
## Attrited Customer Existing Customer   
## 776 776

## CV Logistic Model

print(step\_cv)

## Generalized Linear Model with Stepwise Feature Selection   
##   
## 1552 samples  
## 18 predictor  
## 2 classes: 'Attrited Customer', 'Existing Customer'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 1397, 1396, 1397, 1397, 1397, 1397, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8324428 0.6649061

step\_cv$finalModel

##   
## Call: NULL  
##   
## Coefficients:  
## (Intercept) Customer\_Age   
## -7.9562225 0.0200569   
## GenderM Education\_LevelDoctorate   
## 0.5911277 -0.6088031   
## Marital\_StatusMarried `Income\_Category$40K - $60K`   
## 0.4545138 0.3879305   
## Total\_Relationship\_Count Months\_Inactive\_12\_mon   
## 0.3651510 -0.6075599   
## Contacts\_Count\_12\_mon Total\_Revolving\_Bal   
## -0.4868176 0.0006943   
## Total\_Amt\_Chng\_Q4\_Q1 Total\_Trans\_Amt   
## 0.6912743 -0.0005504   
## Total\_Trans\_Ct Total\_Ct\_Chng\_Q4\_Q1   
## 0.1289470 2.0349156   
## Avg\_Utilization\_Ratio   
## 1.3507459   
##   
## Degrees of Freedom: 1551 Total (i.e. Null); 1537 Residual  
## Null Deviance: 2152   
## Residual Deviance: 1079 AIC: 1109

# Confusion matrix for test data  
prob.test <- predict(step\_cv, newdata = test.data, type = "prob")  
y.test <- ifelse(prob.test[,2] > 0.6, "Existing Customer", "Attrited Customer")  
confusionMatrix(as.factor(y.test), as.factor(test.data$Attrition\_Flag), positive = "Existing Customer")

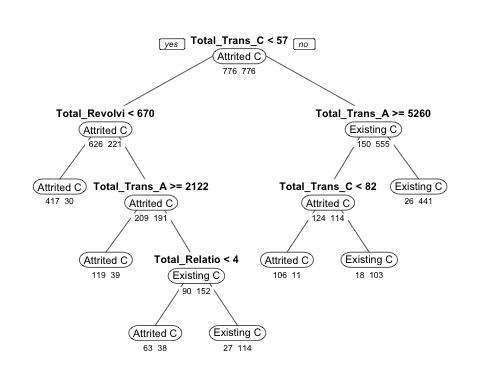
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Attrited Customer Existing Customer  
## Attrited Customer 297 382  
## Existing Customer 40 1405  
##   
## Accuracy : 0.8013   
## 95% CI : (0.7837, 0.8181)   
## No Information Rate : 0.8413   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4729   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.7862   
## Specificity : 0.8813   
## Pos Pred Value : 0.9723   
## Neg Pred Value : 0.4374   
## Prevalence : 0.8413   
## Detection Rate : 0.6615   
## Detection Prevalence : 0.6803   
## Balanced Accuracy : 0.8338   
##   
## 'Positive' Class : Existing Customer  
##

## Decision Tree

library(rpart)  
library(rpart.plot)  
library(caret)  
  
# Decision tree model with CV  
set.seed(0)  
train\_control <- trainControl(method = "cv", number = 10)   
tree\_cv <- train(Attrition\_Flag ~ ., data = training.under, method = "rpart", trControl = train\_control, tuneLength = 10, metric = "Kappa", control = rpart.control(minsplit = 150, minbucket = 100))  
  
print(tree\_cv)

## CART   
##   
## 1552 samples  
## 18 predictor  
## 2 classes: 'Attrited Customer', 'Existing Customer'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 1397, 1396, 1397, 1397, 1397, 1397, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.003865979 0.8428028 0.6855599  
## 0.004510309 0.8447259 0.6894060  
## 0.006443299 0.8447259 0.6894060  
## 0.007731959 0.8447259 0.6894060  
## 0.016752577 0.8485803 0.6970879  
## 0.020618557 0.8466448 0.6933188  
## 0.021907216 0.8453628 0.6907547  
## 0.032216495 0.8453628 0.6907547  
## 0.060567010 0.7874636 0.5750404  
## 0.521907216 0.5933871 0.1887352  
##   
## Kappa was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.01675258.

# Decision tree rules plot  
prp(tree\_cv$finalModel, type = 1, extra = 1, under = TRUE, split.font = 2, varlen = -10)

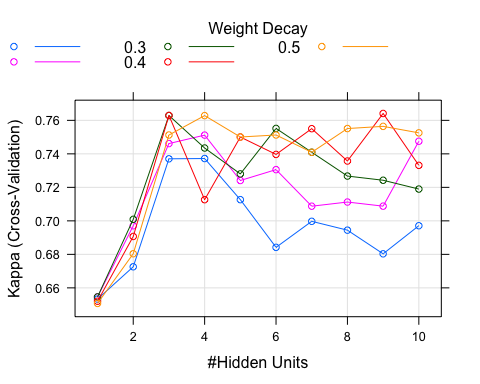


# Evaluation on the test data  
prob.test <- predict(tree\_cv, test.data, type = 'prob')  
y.test <- ifelse(prob.test[,2] > 0.6, "Existing Customer", "Attrited Customer")  
confusionMatrix(as.factor(y.test), as.factor(test.data$Attrition\_Flag), positive = 'Existing Customer')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Attrited Customer Existing Customer  
## Attrited Customer 306 291  
## Existing Customer 31 1496  
##   
## Accuracy : 0.8484   
## 95% CI : (0.8324, 0.8634)   
## No Information Rate : 0.8413   
## P-Value [Acc > NIR] : 0.1951   
##   
## Kappa : 0.5675   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8372   
## Specificity : 0.9080   
## Pos Pred Value : 0.9797   
## Neg Pred Value : 0.5126   
## Prevalence : 0.8413   
## Detection Rate : 0.7043   
## Detection Prevalence : 0.7189   
## Balanced Accuracy : 0.8726   
##   
## 'Positive' Class : Existing Customer  
##

## Neural Network

# Plot  
plot(cv.nn)



# Best parameters  
cv.nn$bestTune

## size decay  
## 44 9 0.4

# Evaluation on the test data  
prob.test <- predict(cv.nn, test.data, type = 'prob')  
y.test <- ifelse(prob.test[,2] > 0.65, "Existing Customer", "Attrited Customer")  
confusionMatrix(as.factor(y.test), as.factor(test.data$Attrition\_Flag), positive = 'Existing Customer')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Attrited Customer Existing Customer  
## Attrited Customer 302 347  
## Existing Customer 35 1440  
##   
## Accuracy : 0.8202   
## 95% CI : (0.8031, 0.8363)   
## No Information Rate : 0.8413   
## P-Value [Acc > NIR] : 0.9961   
##   
## Kappa : 0.5103   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8058   
## Specificity : 0.8961   
## Pos Pred Value : 0.9763   
## Neg Pred Value : 0.4653   
## Prevalence : 0.8413   
## Detection Rate : 0.6780   
## Detection Prevalence : 0.6944   
## Balanced Accuracy : 0.8510   
##   
## 'Positive' Class : Existing Customer  
##