KNN Model

bank <- read.csv("~/Desktop/MSA\_6440/customer\_churn/data/processed/BankChurners\_filtered.csv")  
source('/Users/JonathanVoth/Desktop/R Codes/myfunctions.R')  
  
# Creating dummy variables for Education  
bank$High\_School <- ifelse(bank$Education\_Level == 'High School', 1, 0)  
bank$College <- ifelse(bank$Education\_Level == 'College', 1, 0)  
bank$Graduate <- ifelse(bank$Education\_Level == 'Graduate', 1, 0)  
bank$Uneducated <- ifelse(bank$Education\_Level == 'Uneducated', 1, 0)  
bank$Post\_Graduate <- ifelse(bank$Education\_Level == 'Post-Graduate', 1, 0)  
bank$Doctorate <- ifelse(bank$Education\_Level == 'Doctorate', 1, 0)  
  
# Creating dummy variables for Marital Status  
bank$Married <- ifelse(bank$Marital\_Status == 'Married', 1, 0)  
bank$Single <- ifelse(bank$Marital\_Status == 'Single', 1, 0)  
bank$Divorced <- ifelse(bank$Marital\_Status == 'Divorced', 1, 0)  
  
# Creating dummy variables for Gender  
bank$Male <- ifelse(bank$Gender == 'M', 1, 0)  
bank$Female <- ifelse(bank$Gender == 'F', 1, 0)  
  
# Creating dummy variables for Income  
bank$Income\_5 <- ifelse(bank$Income\_Category == '$120K +', 1, 0)  
bank$Income\_4 <- ifelse(bank$Income\_Category == '$80K - $120K', 1, 0)  
bank$Income\_3 <- ifelse(bank$Income\_Category == '$60K - $80K', 1, 0)  
bank$Income\_2 <- ifelse(bank$Income\_Category == '$40K - $60K', 1, 0)  
bank$Income\_1 <- ifelse(bank$Income\_Category == 'Less than $40K', 1, 0)  
  
# Creating dummy variables for Card category  
bank$Blue\_Card <- ifelse(bank$Card\_Category == 'Blue', 1, 0)  
bank$Gold\_Card <- ifelse(bank$Card\_Category == 'Gold', 1, 0)  
bank$Plat\_Card <- ifelse(bank$Card\_Category == 'Platinum', 1, 0)  
bank$Silver\_Card <- ifelse(bank$Card\_Category == 'Silver', 1, 0)  
  
bank <- bank[,c(-1,-2,-5,-7:-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

## KNN Model

# Scaling the data  
training.scaled <- scale(training.data[,-1], center = TRUE, scale = TRUE)  
attrib <- attributes(training.scaled)  
test.scaled <- scale(test.data[,-1], center = attrib$`scaled:center`, scale = attrib$`scaled:scale`)  
  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(FNN)  
  
# Fitting 10-fold CV model  
set.seed(0)  
train\_control <- trainControl(method = "cv", number = 10)   
knn\_cv <- train(Attrition\_Flag ~ ., data = training.data, method = "knn", trControl = train\_control, preProcess = c("center","scale"), tuneGrid = data.frame(k = seq(1,7,1)), metric = "Kappa")  
  
# Final model  
print(knn\_cv)

## k-Nearest Neighbors   
##   
## 4957 samples  
## 34 predictor  
## 2 classes: 'Attrited Customer', 'Existing Customer'   
##   
## Pre-processing: centered (34), scaled (34)   
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 4461, 4461, 4461, 4461, 4461, 4462, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.8448607 0.3575738  
## 2 0.8327525 0.3154456  
## 3 0.8664488 0.3640722  
## 4 0.8601955 0.3340201  
## 5 0.8656431 0.3253668  
## 6 0.8636270 0.2972761  
## 7 0.8678625 0.2909634  
##   
## Kappa was used to select the optimal model using the largest value.  
## The final value used for the model was k = 3.

knn\_cv$finalModel

## 3-nearest neighbor model  
## Training set outcome distribution:  
##   
## Attrited Customer Existing Customer   
## 776 4181

# Fit model on test data with k = 3  
knn.test <- knn(train = training.scaled, test = test.scaled, cl = training.data[,1], k = 3)  
confusionMatrix(as.factor(knn.test), as.factor(test.data[,1]), positive = "Existing Customer")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Attrited Customer Existing Customer  
## Attrited Customer 129 56  
## Existing Customer 208 1731  
##   
## Accuracy : 0.8757   
## 95% CI : (0.8609, 0.8894)   
## No Information Rate : 0.8413   
## P-Value [Acc > NIR] : 4.547e-06   
##   
## Kappa : 0.4302   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9687   
## Specificity : 0.3828   
## Pos Pred Value : 0.8927   
## Neg Pred Value : 0.6973   
## Prevalence : 0.8413   
## Detection Rate : 0.8150   
## Detection Prevalence : 0.9129   
## Balanced Accuracy : 0.6757   
##   
## 'Positive' Class : Existing Customer  
##