Assignment 2 Ireland Kuhn

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(class)  
library(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

data <- read.csv("C:\\Users\\Ireland\\Downloads\\UniversalBank.csv")  
library(ISLR)

bank\_data <- select(data,Age,Experience,Income,Personal.Loan,Family,CCAvg,Education,Mortgage,Securities.Account,CD.Account,Online,CreditCard)

set.seed(123)   
training\_Index <- createDataPartition(bank\_data$Personal.Loan, p = 0.6, list = FALSE)  
training\_data <- bank\_data[training\_Index, ]  
validation\_data <- bank\_data[-training\_Index, ]

k <- 1  
predictors <- select(training\_data, -Personal.Loan)  
target <- training\_data$Personal.Loan  
  
predicted\_class <- knn(predictors, select(validation\_data, -Personal.Loan), target, k = k)

library(gmodels)  
confusion\_matrix <- CrossTable(validation\_data$Personal.Loan, predicted\_class, prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 2000   
##   
##   
## | predicted\_class   
## validation\_data$Personal.Loan | 0 | 1 | Row Total |   
## ------------------------------|-----------|-----------|-----------|  
## 0 | 1714 | 84 | 1798 |   
## | 0.953 | 0.047 | 0.899 |   
## | 0.940 | 0.475 | |   
## | 0.857 | 0.042 | |   
## ------------------------------|-----------|-----------|-----------|  
## 1 | 109 | 93 | 202 |   
## | 0.540 | 0.460 | 0.101 |   
## | 0.060 | 0.525 | |   
## | 0.054 | 0.046 | |   
## ------------------------------|-----------|-----------|-----------|  
## Column Total | 1823 | 177 | 2000 |   
## | 0.911 | 0.088 | |   
## ------------------------------|-----------|-----------|-----------|  
##   
##

print(confusion\_matrix)

## $t  
## y  
## x 0 1  
## 0 1714 84  
## 1 109 93  
##   
## $prop.row  
## y  
## x 0 1  
## 0 0.95328142 0.04671858  
## 1 0.53960396 0.46039604  
##   
## $prop.col  
## y  
## x 0 1  
## 0 0.94020845 0.47457627  
## 1 0.05979155 0.52542373  
##   
## $prop.tbl  
## y  
## x 0 1  
## 0 0.8570 0.0420  
## 1 0.0545 0.0465

control <- trainControl(method = "cv", number = 5)  
k\_values <- seq(1, 20, by = 2)   
  
model <- train(Personal.Loan ~ ., data = training\_data, method = "knn", trControl = control, tuneGrid = data.frame(k = k\_values))

## Warning in train.default(x, y, weights = w, ...): You are trying to do  
## regression and your outcome only has two possible values Are you trying to do  
## classification? If so, use a 2 level factor as your outcome column.

best\_k <- model$bestTune$k  
print(paste("Best k:", best\_k))

## [1] "Best k: 17"

new\_customer <- data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,  
 Education = 2, Mortgage = 0, Securities.Account = 0, CD.Account = 0, Online = 1, CreditCard = 1)  
new\_customer\_pred <- knn(predictors, new\_customer, target, k = best\_k)  
print(paste("Classification for the new customer:", ifelse(new\_customer\_pred == 1, "Accepted", "Not Accepted")))

## [1] "Classification for the new customer: Not Accepted"

set.seed(123)   
training\_Index2 <- createDataPartition(bank\_data$Personal.Loan, p = 0.5, list = FALSE)  
training\_data2 <- bank\_data[training\_Index2, ]  
temp\_data2 <- bank\_data[-training\_Index2, ]  
validation\_Index2 <- createDataPartition(temp\_data2$Personal.Loan, p = 0.6, list = FALSE)  
validation\_data2 <- temp\_data2[validation\_Index2, ]  
test\_data2 <- temp\_data2[-validation\_Index2, ]

test\_predictors <- select(training\_data2, -Personal.Loan)  
test\_target <- training\_data2$Personal.Loan  
test\_pred <- knn(test\_predictors, select(test\_data2, -Personal.Loan), test\_target, k = best\_k)

confusion\_matrix\_test <- CrossTable(test\_data2$Personal.Loan, test\_pred, prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1000   
##   
##   
## | test\_pred   
## test\_data2$Personal.Loan | 0 | 1 | Row Total |   
## -------------------------|-----------|-----------|-----------|  
## 0 | 881 | 11 | 892 |   
## | 0.988 | 0.012 | 0.892 |   
## | 0.909 | 0.355 | |   
## | 0.881 | 0.011 | |   
## -------------------------|-----------|-----------|-----------|  
## 1 | 88 | 20 | 108 |   
## | 0.815 | 0.185 | 0.108 |   
## | 0.091 | 0.645 | |   
## | 0.088 | 0.020 | |   
## -------------------------|-----------|-----------|-----------|  
## Column Total | 969 | 31 | 1000 |   
## | 0.969 | 0.031 | |   
## -------------------------|-----------|-----------|-----------|  
##   
##

print("Confusion Matrix for Test Set:")

## [1] "Confusion Matrix for Test Set:"

print(confusion\_matrix\_test)

## $t  
## y  
## x 0 1  
## 0 881 11  
## 1 88 20  
##   
## $prop.row  
## y  
## x 0 1  
## 0 0.98766816 0.01233184  
## 1 0.81481481 0.18518519  
##   
## $prop.col  
## y  
## x 0 1  
## 0 0.90918473 0.35483871  
## 1 0.09081527 0.64516129  
##   
## $prop.tbl  
## y  
## x 0 1  
## 0 0.881 0.011  
## 1 0.088 0.020

# The accuracy of the validation set is 89.9% compared to 89.2% for the test set. The model performs similarly on both the validation and test sets, indicating consistent overall performance.  
# The model performs similarly on both the validation and test sets, indicating consistent overall performance.  
# Sensitivity is lower in the test set (35.5%) compared to the validation set (47.5%). This suggests that the model is less effective at identifying true positives in the test set. Specificity is higher in the test set (98.8%) compared to the validation set (95.3%). This indicates that the model is very good at correctly identifying true negatives in the test set. Precision is higher in the test set (64.5%) compared to the validation set (52.5%). When the model predicts positive, it is more likely to be correct in the test set.  
# The model seems to generalize well from the validation set to the test set, as the accuracy is consistent. The drop in sensitivity in the test set suggests that the model might not be as effective at identifying customers who accepted the loan. The increase in specificity in the test set suggests that the model is good at avoiding false positives (correctly identifying customers who did not accept the loan). The higher precision in the test set indicates that when the model predicts positive, it is more likely to be correct.