Assignment 2

Harsh Patel

2025-09-22

# My k-NN Model on Universal Bank Dataset  
  
  
# First, I’ll load the packages I need  
  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(class)  
library(e1071)  
library(dplyr)

##   
## Attaching package: 'dplyr'

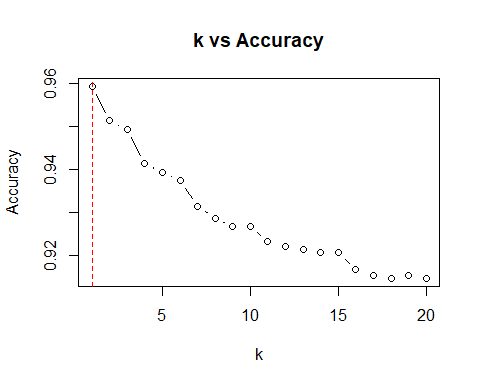
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

# k-NN model on Universal Bank data  
  
# load data (renamed file to avoid spaces)  
df <- read.csv("C:/Users/jbhsp/Downloads/UniversalBank - Copy.csv")  
df <- select(df, -c(ID, ZIP.Code))  
  
# convert categorical variables  
df$Personal.Loan <- as.factor(df$Personal.Loan)  
df$Family <- as.factor(df$Family)  
df$Education <- as.factor(df$Education)  
df$Securities.Account <- as.factor(df$Securities.Account)  
df$CD.Account <- as.factor(df$CD.Account)  
df$Online <- as.factor(df$Online)  
df$CreditCard <- as.factor(df$CreditCard)  
  
# split data  
set.seed(42)  
ind <- createDataPartition(df$Personal.Loan, p = 0.5, list = FALSE)  
train\_df <- df[ind, ]  
temp\_df <- df[-ind, ]  
ind\_val <- createDataPartition(temp\_df$Personal.Loan, p = 0.6, list = FALSE)  
validation\_df <- temp\_df[ind\_val, ]  
test\_df <- temp\_df[-ind\_val, ]  
  
# normalize numeric features  
numeric\_cols <- c("Age", "Experience", "Income", "CCAvg", "Mortgage")  
pre\_process <- preProcess(train\_df[, numeric\_cols], method = c("range"))  
train\_norm <- predict(pre\_process, train\_df[, numeric\_cols])  
validation\_norm <- predict(pre\_process, validation\_df[, numeric\_cols])  
test\_norm <- predict(pre\_process, test\_df[, numeric\_cols])  
  
train\_ready <- bind\_cols(train\_norm, train\_df %>% select(-all\_of(numeric\_cols)))  
validation\_ready <- bind\_cols(validation\_norm, validation\_df %>% select(-all\_of(numeric\_cols)))  
test\_ready <- bind\_cols(test\_norm, test\_df %>% select(-all\_of(numeric\_cols)))  
  
predictor\_cols <- names(train\_ready)[!names(train\_ready) %in% "Personal.Loan"]  
target\_col <- "Personal.Loan"  
  
# find best k  
k\_values <- 1:20  
accuracy\_df <- data.frame(k = k\_values, accuracy = rep(0, length(k\_values)))  
  
for (k\_val in k\_values) {  
 knn\_pred <- knn(train = train\_ready[, predictor\_cols],  
 test = validation\_ready[, predictor\_cols],  
 cl = train\_ready[, target\_col],  
 k = k\_val)  
 accuracy\_df[k\_val, "accuracy"] <- mean(knn\_pred == validation\_ready[, target\_col])  
}  
  
optimal\_k <- accuracy\_df[which.max(accuracy\_df$accuracy), "k"]  
cat("Best k:", optimal\_k, "\n")

## Best k: 1

plot(accuracy\_df$k, accuracy\_df$accuracy, type = "b",  
 main = "k vs Accuracy", xlab = "k", ylab = "Accuracy")  
abline(v = optimal\_k, col = "red", lty = 2)



# evaluate model  
knn\_pred\_train <- knn(train = train\_ready[, predictor\_cols],  
 test = train\_ready[, predictor\_cols],  
 cl = train\_ready[, target\_col],  
 k = optimal\_k)  
train\_cm <- confusionMatrix(knn\_pred\_train, train\_ready[, target\_col])  
cat("\nTraining Accuracy:", round(train\_cm$overall['Accuracy'], 4), "\n")

##   
## Training Accuracy: 1

knn\_pred\_val <- knn(train = train\_ready[, predictor\_cols],  
 test = validation\_ready[, predictor\_cols],  
 cl = train\_ready[, target\_col],  
 k = optimal\_k)  
validation\_cm <- confusionMatrix(knn\_pred\_val, validation\_ready[, target\_col])  
cat("Validation Accuracy:", round(validation\_cm$overall['Accuracy'], 4), "\n")

## Validation Accuracy: 0.9593

knn\_pred\_test <- knn(train = train\_ready[, predictor\_cols],  
 test = test\_ready[, predictor\_cols],  
 cl = train\_ready[, target\_col],  
 k = optimal\_k)  
test\_cm <- confusionMatrix(knn\_pred\_test, test\_ready[, target\_col])  
cat("Test Accuracy:", round(test\_cm$overall['Accuracy'], 4), "\n")

## Test Accuracy: 0.951