Fundamentals of Machine Learning - Assignment 2

2022-09-27

bank.df <- read.csv("UniversalBank.csv", header = TRUE)  
dim(bank.df)

## [1] 5000 14

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

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

library(ggplot2)  
library(FNN)  
library(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(ISLR)  
library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(class)

##   
## Attaching package: 'class'

## The following objects are masked from 'package:FNN':  
##   
## knn, knn.cv

library(tidyr)  
library(datasets)

summary(bank.df)

## ID Age Experience Income ZIP.Code   
## Min. : 1 Min. :23.00 Min. :-3.0 Min. : 8.00 Min. : 9307   
## 1st Qu.:1251 1st Qu.:35.00 1st Qu.:10.0 1st Qu.: 39.00 1st Qu.:91911   
## Median :2500 Median :45.00 Median :20.0 Median : 64.00 Median :93437   
## Mean :2500 Mean :45.34 Mean :20.1 Mean : 73.77 Mean :93152   
## 3rd Qu.:3750 3rd Qu.:55.00 3rd Qu.:30.0 3rd Qu.: 98.00 3rd Qu.:94608   
## Max. :5000 Max. :67.00 Max. :43.0 Max. :224.00 Max. :96651   
## Family CCAvg Education Mortgage   
## Min. :1.000 Min. : 0.000 Min. :1.000 Min. : 0.0   
## 1st Qu.:1.000 1st Qu.: 0.700 1st Qu.:1.000 1st Qu.: 0.0   
## Median :2.000 Median : 1.500 Median :2.000 Median : 0.0   
## Mean :2.396 Mean : 1.938 Mean :1.881 Mean : 56.5   
## 3rd Qu.:3.000 3rd Qu.: 2.500 3rd Qu.:3.000 3rd Qu.:101.0   
## Max. :4.000 Max. :10.000 Max. :3.000 Max. :635.0   
## Personal.Loan Securities.Account CD.Account Online   
## Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.000 Median :0.0000 Median :0.0000 Median :1.0000   
## Mean :0.096 Mean :0.1044 Mean :0.0604 Mean :0.5968   
## 3rd Qu.:0.000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.0000   
## Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## CreditCard   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.294   
## 3rd Qu.:1.000   
## Max. :1.000

str(bank.df)

## 'data.frame': 5000 obs. of 14 variables:  
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ ZIP.Code : int 91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : int 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Securities.Account: int 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD.Account : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : int 0 0 0 0 0 1 1 0 1 0 ...  
## $ CreditCard : int 0 0 0 0 1 0 0 1 0 0 ...

names(bank.df)

## [1] "ID" "Age" "Experience"   
## [4] "Income" "ZIP.Code" "Family"   
## [7] "CCAvg" "Education" "Mortgage"   
## [10] "Personal.Loan" "Securities.Account" "CD.Account"   
## [13] "Online" "CreditCard"

bank.df$Education <- factor(bank.df$Education, levels = c(1, 2, 3), labels = c("Undergrad", "Graduate", "Advanced"))

df=subset(bank.df, select=-c(1,5))  
  
dumedu <- as.data.frame(dummy.code(bank.df$Education))  
  
df\_without\_education <- subset(df, select=-c(Education))  
bank\_data <- cbind(df\_without\_education, dumedu)  
head(bank\_data)

## Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account  
## 1 25 1 49 4 1.6 0 0 1  
## 2 45 19 34 3 1.5 0 0 1  
## 3 39 15 11 1 1.0 0 0 0  
## 4 35 9 100 1 2.7 0 0 0  
## 5 35 8 45 4 1.0 0 0 0  
## 6 37 13 29 4 0.4 155 0 0  
## CD.Account Online CreditCard Undergrad Advanced Graduate  
## 1 0 0 0 1 0 0  
## 2 0 0 0 1 0 0  
## 3 0 0 0 1 0 0  
## 4 0 0 0 0 0 1  
## 5 0 0 1 0 0 1  
## 6 0 1 0 0 0 1

bank\_data$Personal.Loan = as.factor(bank\_data$Personal.Loan)  
bank\_data$CCAvg = as.integer(bank\_data$CCAvg)  
  
set.seed(1)  
TrainRows <- sample(rownames(bank\_data), dim(bank.df)[1]\*0.6)  
TrainData = bank\_data[TrainRows, ]  
ValidRows <- setdiff(rownames(bank\_data), TrainRows)  
ValidData <- bank\_data[ValidRows, ]  
  
new.df <- data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education1 = 0, Education2 = 1, Education3 = 0, Mortgage = 0, Securities.Account = 0, CD.Account = 0, Online = 1, Credit.Card = 1)  
new.df

## Age Experience Income Family CCAvg Education1 Education2 Education3 Mortgage  
## 1 40 10 84 2 2 0 1 0 0  
## Securities.Account CD.Account Online Credit.Card  
## 1 0 0 1 1

train.norm.df <- TrainData  
valid.norm.df <- ValidData  
bank.norm.df <- bank\_data  
  
norm.values <- preProcess(TrainData[, -7], method=c("center", "scale"))  
train.norm.df[, -7] <-predict(norm.values, TrainData[, -7])  
valid.norm.df[, -7] <-predict(norm.values, ValidData[, -7])  
bank.norm.df[, -7] <-predict(norm.values, bank\_data[, -7])  
  
knn.predictor <- knn(train = train.norm.df[, -7], test = bank.norm.df[, -7], cl = train.norm.df[, 7], k=5, prob=TRUE)  
knn.attributes <- attributes(knn.predictor)  
knn.attributes[1]

## $levels  
## [1] "0" "1"

accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))  
  
for(i in 1:14) {  
 KNN2 <- knn(train = train.norm.df[ ,-7],test = valid.norm.df[ ,-7], cl = train.norm.df[ ,7], k=i, prob=TRUE)  
 accuracy.df[i, 2] <- confusionMatrix(KNN2, valid.norm.df[ ,7])$overall[1]  
}  
accuracy.df

## k accuracy  
## 1 1 0.9625  
## 2 2 0.9560  
## 3 3 0.9640  
## 4 4 0.9570  
## 5 5 0.9590  
## 6 6 0.9570  
## 7 7 0.9580  
## 8 8 0.9560  
## 9 9 0.9545  
## 10 10 0.9535  
## 11 11 0.9510  
## 12 12 0.9500  
## 13 13 0.9500  
## 14 14 0.9495

1. k=3 is the best choice becuase it is closest to 1

KNN3 <- knn(train = train.norm.df[ ,-7],test = valid.norm.df[ ,-7], cl = train.norm.df[ ,7], k=3, prob=TRUE)  
confusionMatrix(KNN3, valid.norm.df[ ,7])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1787 64  
## 1 8 141  
##   
## Accuracy : 0.964   
## 95% CI : (0.9549, 0.9717)  
## No Information Rate : 0.8975   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7774   
##   
## Mcnemar's Test P-Value : 9.063e-11   
##   
## Sensitivity : 0.9955   
## Specificity : 0.6878   
## Pos Pred Value : 0.9654   
## Neg Pred Value : 0.9463   
## Prevalence : 0.8975   
## Detection Rate : 0.8935   
## Detection Prevalence : 0.9255   
## Balanced Accuracy : 0.8417   
##   
## 'Positive' Class : 0   
##

newCustomer.df= data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education\_1 = 0, Education\_2 = 1, Education\_3 = 0, Mortgage = 0, Securities.Account = 0, CD.Account = 0, Online = 1, CreditCard = 1)  
knn.4 <- knn(train = train.norm.df[,-7],test = newCustomer.df, cl = train.norm.df[,7], k=3, prob=TRUE)  
knn.4

## [1] 1  
## attr(,"prob")  
## [1] 1  
## Levels: 0 1

Customer will be a 1 as classification.

df=subset(bank.df, select=-c(1,5))  
  
dumedu <- as.data.frame(dummy.code(bank.df$Education))  
  
df\_without\_education <- subset(df, select=-c(Education))  
bank\_data <- cbind(df\_without\_education, dumedu)  
head(bank\_data)

## Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account  
## 1 25 1 49 4 1.6 0 0 1  
## 2 45 19 34 3 1.5 0 0 1  
## 3 39 15 11 1 1.0 0 0 0  
## 4 35 9 100 1 2.7 0 0 0  
## 5 35 8 45 4 1.0 0 0 0  
## 6 37 13 29 4 0.4 155 0 0  
## CD.Account Online CreditCard Undergrad Advanced Graduate  
## 1 0 0 0 1 0 0  
## 2 0 0 0 1 0 0  
## 3 0 0 0 1 0 0  
## 4 0 0 0 0 0 1  
## 5 0 0 1 0 0 1  
## 6 0 1 0 0 0 1

bank\_data$Personal.Loan = as.factor(bank\_data$Personal.Loan)  
bank\_data$CCAvg = as.integer(bank\_data$CCAvg)  
  
set.seed(1)  
train.index <- sample(rownames(bank\_data), 0.5\*dim(bank.df)[1])  
valid.index <- sample(setdiff(rownames(bank\_data),train.index), 0.3\*dim(bank.df)[1])  
test.index = setdiff(rownames(bank\_data), union(train.index, valid.index))  
  
train.df <- bank\_data[train.index, ]  
valid.df <- bank\_data[valid.index, ]  
test.df <- bank\_data[test.index, ]  
  
norm.values <- preProcess(train.df[, -7], method=c("center", "scale"))  
train.df[, -7] <- predict(norm.values, train.df[, -7])  
valid.df[, -7] <- predict(norm.values, valid.df[, -7])  
test.df[,-7] <- predict(norm.values, test.df[,-7])  
  
TestKNN <- knn(train = train.df[,-7],test = test.df[,-7], cl = train.df[,7], k=3, prob=TRUE)  
ValidKNN <- knn(train = train.df[,-7],test = valid.df[,-7], cl = train.df[,7], k=3, prob=TRUE)  
TrainKNN <- knn(train = train.df[,-7],test = train.df[,-7], cl = train.df[,7], k=3, prob=TRUE)  
  
confusionMatrix(TestKNN, test.df[,7])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 885 35  
## 1 3 77  
##   
## Accuracy : 0.962   
## 95% CI : (0.9482, 0.973)  
## No Information Rate : 0.888   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7817   
##   
## Mcnemar's Test P-Value : 4.934e-07   
##   
## Sensitivity : 0.9966   
## Specificity : 0.6875   
## Pos Pred Value : 0.9620   
## Neg Pred Value : 0.9625   
## Prevalence : 0.8880   
## Detection Rate : 0.8850   
## Detection Prevalence : 0.9200   
## Balanced Accuracy : 0.8421   
##   
## 'Positive' Class : 0   
##

confusionMatrix(ValidKNN, valid.df[,7])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1360 44  
## 1 4 92  
##   
## Accuracy : 0.968   
## 95% CI : (0.9578, 0.9763)  
## No Information Rate : 0.9093   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7763   
##   
## Mcnemar's Test P-Value : 1.811e-08   
##   
## Sensitivity : 0.9971   
## Specificity : 0.6765   
## Pos Pred Value : 0.9687   
## Neg Pred Value : 0.9583   
## Prevalence : 0.9093   
## Detection Rate : 0.9067   
## Detection Prevalence : 0.9360   
## Balanced Accuracy : 0.8368   
##   
## 'Positive' Class : 0   
##

confusionMatrix(TrainKNN, train.df[,7])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2264 51  
## 1 4 181  
##   
## Accuracy : 0.978   
## 95% CI : (0.9715, 0.9834)  
## No Information Rate : 0.9072   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8563   
##   
## Mcnemar's Test P-Value : 5.552e-10   
##   
## Sensitivity : 0.9982   
## Specificity : 0.7802   
## Pos Pred Value : 0.9780   
## Neg Pred Value : 0.9784   
## Prevalence : 0.9072   
## Detection Rate : 0.9056   
## Detection Prevalence : 0.9260   
## Balanced Accuracy : 0.8892   
##   
## 'Positive' Class : 0   
##

Test Set - .962 Validation Set - .968 Training Set - .978 As Training Set is closest to 1, it will have the most accuracy. Testing Set is furthest from 1, so it will have lowest accuracy.