Assignment2.R

Jason

2021-02-21

```
#Assignment 2 Fundamentals of Machine Learning
#Data comes From UniversalBank.csv

library(utils)
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(class)
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(FNN)
```

```
##
## Attaching package: 'FNN'

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

```
library(e1071)

WD<-setwd("C:/Users/Jason/Documents/MSBA/Fundamentals for Machine Learning/Assignment2")
Bank<-read.csv("UniversalBank.csv", header = TRUE)
summary(Bank)
```

```
##        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   :93153
##  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
```

```
#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, and
Bank$Education_1 <- ifelse(Bank$Education == 1, 1, 0)
Bank$Education_2 <- ifelse(Bank$Education == 2, 1, 0)
Bank$Education_3 <- ifelse(Bank$Education == 3, 1, 0)


Bank<-Bank[,-1] #remove ID
Bank<-Bank[, -4] #remove zipcode
Bank<-Bank[, -6] #remove old Education column

Bank2 <- Bank
Bank2 <- Bank2[, c(7, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14)] #reorder

zNorm <- function(x){(x-mean(x))/sd(x)}
Bank_Norm <- as.data.frame(lapply(Bank[,2:14], zNorm))
Bank2[,2:14] <- Bank_Norm[,1:13]

summary(Bank2)
```

```
##  Personal.Loan        Age               Experience          Income
##  Min.   :0.000   Min.   :-2.014710   Min.   :-1.4288   Min.   :-1.2167
##  1st Qu.:0.000   1st Qu.:-0.881116   1st Qu.:-0.7554   1st Qu.:-1.2167
##  Median :0.000   Median :-0.009121   Median :-0.2123   Median :-0.3454
##  Mean   :0.096   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:0.000   3rd Qu.: 0.862874   3rd Qu.: 0.5263   3rd Qu.: 0.5259
##  Max.   :1.000   Max.   : 1.996468   Max.   : 3.2634   Max.   : 1.3973
##      Family            CCAvg            Mortgage       Securities.Account
##  Min.   :-1.1089  Min.   :-0.5555   Min.   :-0.3258   Min.   :-0.3414
##  1st Qu.:-0.7083  1st Qu.:-0.5555   1st Qu.:-0.3258   1st Qu.:-0.3414
##  Median :-0.2506  Median :-0.5555   Median :-0.3258   Median :-0.3414
##  Mean   : 0.0000  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3216  3rd Qu.: 0.4375   3rd Qu.:-0.3258   3rd Qu.:-0.3414
##  Max.   : 4.6131  Max.   : 5.6875   Max.   : 3.0684   Max.   : 2.9286
##    CD.Account          Online          CreditCard       Education_1
##  Min.   :-0.2535  Min.   :-1.2165   Min.   :-0.6452   Min.   :-0.8495
##  1st Qu.:-0.2535  1st Qu.:-1.2165   1st Qu.:-0.6452   1st Qu.:-0.8495
##  Median :-0.2535  Median : 0.8219   Median :-0.6452   Median :-0.8495
##  Mean   : 0.0000  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.2535  3rd Qu.: 0.8219   3rd Qu.: 1.5495   3rd Qu.: 1.1770
##  Max.   : 3.9438  Max.   : 0.8219   Max.   : 1.5495   Max.   : 1.1770
##   Education_2       Education_3
##  Min.   :-0.6245  Min.   :-0.6549
##  1st Qu.:-0.6245  1st Qu.:-0.6549
##  Median :-0.6245  Median :-0.6549
##  Mean   : 0.0000  Mean   : 0.0000
##  3rd Qu.: 1.6010  3rd Qu.: 1.5266
##  Max.   : 1.6010  Max.   : 1.5266
```

```r
set.seed(10)
Train_Index = createDataPartition(Bank2$Age, p=0.6, list = FALSE)
Train_Data = Bank2[Train_Index,]
Test_Data = Bank2[-Train_Index,]



KNN_Test<- knn(train = Train_Data[, 2:14], test = Test_Data[,2:14],
               cl = Train_Data[,"Personal.Loan"], k = 10, prob=TRUE)


table(KNN_Test, Test_Data[,1])
```

```
##
## KNN_Test    0    1
##        0 1820    1
##        1    0  177
```

```r
PL_Test<-as.factor(Test_Data[,1])


#Testing for best value of K
accuracy.df <- data.frame(k = seq(1, 25, 1), accuracy = rep(0, 25))
for(i in 1:25) {
  knn.pred <- knn(Train_Data[, 2:14], Test_Data[, 2:14],
                  cl = Train_Data[, 1], k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.pred, PL_Test)$overall[1]
}

print(accuracy.df)
```

```
##     k  accuracy
## 1   1 1.0000000
## 2   2 1.0000000
## 3   3 1.0000000
## 4   4 1.0000000
## 5   5 1.0000000
## 6   6 1.0000000
## 7   7 1.0000000
## 8   8 0.9994995
## 9   9 0.9994995
## 10 10 0.9994995
## 11 11 1.0000000
## 12 12 1.0000000
## 13 13 1.0000000
## 14 14 1.0000000
## 15 15 1.0000000
## 16 16 1.0000000
## 17 17 1.0000000
## 18 18 1.0000000
## 19 19 1.0000000
## 20 20 1.0000000
## 21 21 1.0000000
## 22 22 1.0000000
## 23 23 1.0000000
## 24 24 1.0000000
## 25 25 1.0000000
```

```
#k=8 is the best choice, other K's over fitted


#breaking data into Train, Test, and validation
set.seed(123)
Train_Index2 = createDataPartition(Bank2$Age, p=0.5, list = FALSE) #50%
Train_Data2 = Bank2[Train_Index2,]
Valid_Index = createDataPartition(-Train_Index2, p=0.6, list = FALSE) #30%
Validation_Data = Bank2[Valid_Index,]
Test_Index2 = createDataPartition(-Train_Index2, p=0.4, list = FALSE) #20%
Test_Data2 = Bank2[Test_Index2,]

#Train vs. Test
KNN_Test2<- knn(train = Train_Data2[, 2:14], test = Test_Data2[,2:14],
               cl = Train_Data2[,"Personal.Loan"], k = 8, prob=TRUE)

#Valid vs. Test
KNN_Valid<-knn(train = Validation_Data[, 2:14], test = Test_Data2[,2:14],
                         cl = Validation_Data[,"Personal.Loan"], k = 8, prob=TRUE)

PL_Test2<-as.factor(Test_Data2[,1])


Train_vs_Test.df <- data.frame(k = seq(1, 25, 1), accuracy = rep(0, 25))
for(i in 1:25) {
  TrainTest.pred <- knn(Train_Data2[, 2:14], Test_Data2[, 2:14],
                   cl = Train_Data2[, 1], k = i)
  Train_vs_Test.df[i, 2] <- confusionMatrix(TrainTest.pred, PL_Test2)$overall[1]
}
print(Train_vs_Test.df)
```

```
##     k accuracy
## 1   1 1.000000
## 2   2 1.000000
## 3   3 1.000000
## 4   4 1.000000
## 5   5 1.000000
## 6   6 1.000000
## 7   7 1.000000
## 8   8 1.000000
## 9   9 1.000000
## 10 10 1.000000
## 11 11 1.000000
## 12 12 1.000000
## 13 13 1.000000
## 14 14 1.000000
## 15 15 1.000000
## 16 16 1.000000
## 17 17 1.000000
## 18 18 1.000000
## 19 19 1.000000
## 20 20 0.999002
## 21 21 0.999002
## 22 22 0.999002
## 23 23 0.999002
## 24 24 0.999002
## 25 25 0.999002
```

```
#Best K = 20, other K over fitted


Valid_vs_Test.df <- data.frame(k = seq(1, 50, 1), accuracy = rep(0, 50))
for(i in 1:50) {
  ValidTest.pred <- knn(Validation_Data[, 2:14], Test_Data2[, 2:14],
                        cl = Validation_Data[, 1], k = i)
  Valid_vs_Test.df[i, 2] <- confusionMatrix(ValidTest.pred, PL_Test2)$overall[1]
}
print(Valid_vs_Test.df)
```

```
##     k accuracy
## 1   1 1.000000
## 2   2 1.000000
## 3   3 1.000000
## 4   4 1.000000
## 5   5 1.000000
## 6   6 1.000000
## 7   7 1.000000
## 8   8 1.000000
## 9   9 1.000000
## 10 10 1.000000
## 11 11 1.000000
## 12 12 0.999002
## 13 13 0.999002
## 14 14 0.999002
## 15 15 0.999002
## 16 16 0.999002
## 17 17 0.999002
## 18 18 0.999002
## 19 19 0.999002
## 20 20 0.999002
## 21 21 0.999002
## 22 22 0.999002
## 23 23 0.999002
## 24 24 0.999002
## 25 25 0.999002
## 26 26 0.999002
## 27 27 0.999002
## 28 28 0.999002
## 29 29 0.999002
## 30 30 0.997006
## 31 31 0.998004
## 32 32 0.997006
## 33 33 0.997006
## 34 34 0.996008
## 35 35 0.996008
## 36 36 0.996008
## 37 37 0.996008
## 38 38 0.994012
## 39 39 0.994012
## 40 40 0.994012
## 41 41 0.994012
## 42 42 0.993014
## 43 43 0.993014
## 44 44 0.989022
## 45 45 0.992016
## 46 46 0.989022
## 47 47 0.989022
## 48 48 0.988024
## 49 49 0.988024
## 50 50 0.988024
```

```
#Best K = 12, other K over fitted




#Below is random code for my reference to learn R syntax



# DS1<-filter(Bank, Age==40, Experience==10, Income==84, Family==2, CCAvg==2, Education ==1, Mortgage==
#         Securities.Account==0, CD.Account==0, Online==1, CreditCard==1)
# DS1<-Bank[c(Bank[Bank$Age == 40,],Bank[Bank$Experience == 10,],Bank[Bank$Income == 84,],
#          Bank[Bank$CCAvg == 2,],Bank[Bank$Education == 1,],
#          Bank[Bank$Securities.Account == 0,],
#          Bank[Bank$Online == 1,],Bank[Bank$CreditCard == 1,])]

# AGE<-Bank$Age[Bank$Age == 40]
# EXPER<-Bank$Experience[Bank$Experience == 10]
# Inco<-Bank$Income[Bank$Income == 84]
# CCAv<-Bank$CCAvg[Bank$CCAvg == 2]
# Ed<- ifelse(Bank$Education[Bank$Education == 2],1,0)
# Mort<-Bank$Mortgage[Bank$Mortgage == 0]
# SecAct<-Bank$Securities.Account[Bank$Securities.Account == 0]
# CDAct<-Bank$CD.Account[Bank$CD.Account == 0]
# Onl<-Bank$Online[Bank$Online == 1]
# CredC<-Bank$CreditCard[Bank$CreditCard == 1]

# DataSet1 <- data.frame("AGE" = AGE, "EXP" = EXPER, "Income"= Inco, "CCAvg"=CCAv, "Education"=Ed, "Mor
#                        "Securities_Acct"=SecAct, "CD_Acct"=CDAct, "Online"=Onl, "CreditCard"=CredC)
```