Sharon Kasturiarachi

MIS 64060 Fundamentals of Machine Learning

Dr. Rouzbeh Razavi

Assignment 2- Answers

Kasturiarachi-Assignment 2

```
> summary(UBank_data)
      Age          Experience       Income           Family          CCAvg
 Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000   Min.   : 0.000
 1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000   1st Qu.: 0.700
 Median :45.00   Median :20.0   Median : 64.00   Median :2.000   Median : 1.500
 Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396   Mean   : 1.938
 3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000   3rd Qu.: 2.500
 Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000   Max.   :10.000
    Mortgage     Personal Loan Securities Account   CD Account         Online
 Min.   :  0.0   0:4520        Min.   :0.0000     Min.   :0.0000   Min.   :0.0000
 1st Qu.:  0.0   1: 480        1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000
 Median :  0.0                 Median :0.0000     Median :0.0000   Median :1.0000
 Mean   : 56.5                 Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
 3rd Qu.:101.0                 3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
 Max.   :635.0                 Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
   CreditCard    Education_Education_1 Education_Education_2 Education_Education_3
 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 :0.0000
 Mean   :0.294   Mean   :0.4192        Mean   :0.2806        Mean   :0.3002
 3rd Qu.:1.000   3rd Qu.:1.0000        3rd Qu.:1.0000        3rd Qu.:1.0000
 Max.   :1.000   Max.   :1.0000        Max.   :1.0000        Max.   :1.0000
>
```

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       5000 obs. of  14 variables:
 $ Age                   : num  25 45 39 35 35 37 53 50 35 34 ...
 $ Experience            : num  1 19 15 9 8 13 27 24 10 9 ...
 $ Income                : num  49 34 11 100 45 29 72 22 81 180 ...
 $ Family                : num  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 ...
 $ Mortgage              : num  0 0 0 0 0 155 0 0 104 0 ...
 $ Personal Loan         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ Securities Account    : num  1 1 0 0 0 0 0 0 0 0 ...
 $ CD Account            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Online                : num  0 0 0 0 0 1 1 0 1 0 ...
 $ CreditCard            : num  0 0 0 0 1 0 0 1 0 0 ...
 $ Education_Education_1: int  1 1 1 0 0 0 0 0 0 0 ...
 $ Education_Education_2: int  0 0 0 1 1 1 1 0 1 0 ...
 $ Education_Education_3: int  0 0 0 0 0 0 0 1 0 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
> 
```

```
> colnames(Bank)
 [1] "Age"                 "Experience"          "Income"              "Family"
 [5] "CCAvg"               "Education"           "Mortgage"            "Personal Loan"
 [9] "Securities Account"  "CD Account"          "Online"              "CreditCard"
[13] "Education_Education_1" "Education_Education_2" "Education_Education_3"
> colnames(Bank_without_education)
 [1] "Age"                 "Experience"          "Income"              "Family"
 [5] "CCAvg"               "Mortgage"            "Personal Loan"       "Securities Account"
 [9] "CD Account"          "Online"              "CreditCard"          "Education_Education_1"
[13] "Education_Education_2" "Education_Education_3"
> View(UBank_data)
> UBank_data <- Bank_without_education
> colnames(UBank_data)
 [1] "Age"                 "Experience"          "Income"              "Family"
 [5] "CCAvg"               "Mortgage"            "Personal Loan"       "Securities Account"
 [9] "CD Account"          "Online"              "CreditCard"          "Education_Education_1"
[13] "Education_Education_2" "Education_Education_3"
> View(UniversalBank)
> View(UBank_data)
> Train_Index   =createDataPartition(UBank_data$Age, p= 0.6, list =FALSE)
> Train_Data    =UBank_data[Train_Index,]
> Validation_Data  =UBank_data[-Train_Index,]
> Test_Data <- data.frame(Age=40 , Experience=10, Income = 84, Family = 2, CCAvg = 2, Education_Education_1 = 0, Education_Education_2 = 1, Education_E
ducation_3 = 0, Mortgage = 0, Securities.Account = 0, CD.Account = 0, Online = 1,  CreditCard = 1, stringsAsFactors = FALSE)
> View(UBank_data)
> ####Data Normalization
> train.norm.df    <- Train_Data
> valid.norm.df   <-Validation_Data
> test.norm.df    <- Test_Data
> maindata.norm.df <-UBank_data
> head(maindata.norm.df)
# A tibble: 6 x 14
    Age Experience Income Family CCAvg Mortgage `Personal Loan` `Securities Account` `CD Account` Online CreditCard
  <db1>      <db1>  <db1>  <db1> <db1>    <db1>           <db1>                <db1>        <db1>  <db1>      <db1>
1    25          1     49      4   1.6        0               0                    1            0      0          0
2    45         19     34      3   1.5        0               0                    1            0      0          0
3    39         15     11      1   1          0               0                    0            0      0          0
4    35          9    100      1   2.7        0               0                    0            0      0          0
5    35          8     45      4   1          0               0                    0            0      0          1
6    37         13     29      4   0.4      155               0                    0            0      1          0
# ... with 3 more variables: Education_Education_1 <int>, Education_Education_2 <int>, Education_Education_3 <int>
```

```
> head(maindata.norm.df)
# A tibble: 6 x 14
      Age Experience Income Family  CCAvg Mortgage `Personal Loan` `Securities Account` `CD Account` Online CreditCard
    <db1>      <db1>  <db1>  <db1>  <db1>    <db1> <fct>                          <db1>        <db1>  <db1>      <db1>
1 -1.77      -1.66   -0.541  1.39  -0.204   -0.561 0                              2.94        -0.251 -1.21      -0.647
2 -0.0286    -0.0951 -0.867  0.522 -0.261   -0.561 0                              2.94        -0.251 -1.21      -0.647
3 -0.552     -0.443  -1.37   -1.22 -0.547   -0.561 0                             -0.341       -0.251 -1.21      -0.647
4 -0.901     -0.966   0.567 -1.22   0.425   -0.561 0                             -0.341       -0.251 -1.21      -0.647
5 -0.901     -1.05   -0.628  1.39  -0.547   -0.561 0                             -0.341       -0.251 -1.21       1.54
6 -0.727     -0.617  -0.976  1.39  -0.890    0.938 0                             -0.341       -0.251  0.824     -0.647
# ... with 3 more variables: Education_Education_1 <dbl>, Education_Education_2 <dbl>, Education_Education_3 <dbl>
> source('D:/Fundamentals of Machine Learning Spring 2022/Kasturiarachi-Assignment 2/Kasturiarachi-Assignment 2.R')
Error in source("D:/Fundamentals of Machine Learning Spring 2022/Kasturiarachi-Assignment 2/Kasturiarachi-Assignment 2.R") :
```

Perfoming k-NN classification, using k = 1

```
> set.seed(2019)
> prediction <- knn(train = train.norm.df[,-7], test = valid.norm.df[,-7],
+                   cl = train.norm.df[,7], k = 1, prob=TRUE)
> actual= valid.norm.df$`Personal Loan`
> prediction_prob = attr(prediction,"prob")
> table(prediction,actual)
          actual
prediction    0    1
         0 1792   58
         1   23  126
> mean(prediction==actual)
[1] 0.9594797
>
```

```
> NROW(train.norm.df)
[1] 3001
> sqrt(3001)
[1] 54.78138
>
```

2: The value of k we choose is 3 as it provides the best result [i.e the choice of k that balances between overfitting and ignoring the predictor information]

```
> set.seed(123)
> ####Generating loop to find best k
> set.seed(2019)
> accuracy.df <- data.frame(k = seq(1,60,1), accuracy = rep(0,60))
> fitControl <- trainControl(method = "repeatedcv", number = 3, repeats = 2)
> searchGrid=expand.grid(k = 1:10)
> knn.model=train(`Personal Loan`~.,data = Train_Data, method='knn',tuneGrid=searchGrid, trControl = fitControl,)
> knn.model
k-Nearest Neighbors

3001 samples
  13 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (3 fold, repeated 2 times)
Summary of sample sizes: 2002, 2000, 2000, 2001, 2000, 2001, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.8987016  0.4084636
   2  0.8920363  0.3826519
   3  0.8998691  0.3743003
   4  0.8973679  0.3681927
   5  0.9003664  0.3398133
   6  0.8992009  0.3255275
   7  0.8988689  0.3078236
   8  0.8973683  0.3062081
   9  0.9021988  0.3322957
  10  0.8998679  0.3300653

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

The value of k we choose is 3 as it provides the best result [i.e the choice of k that balances between overfitting and ignoring the predictor information]

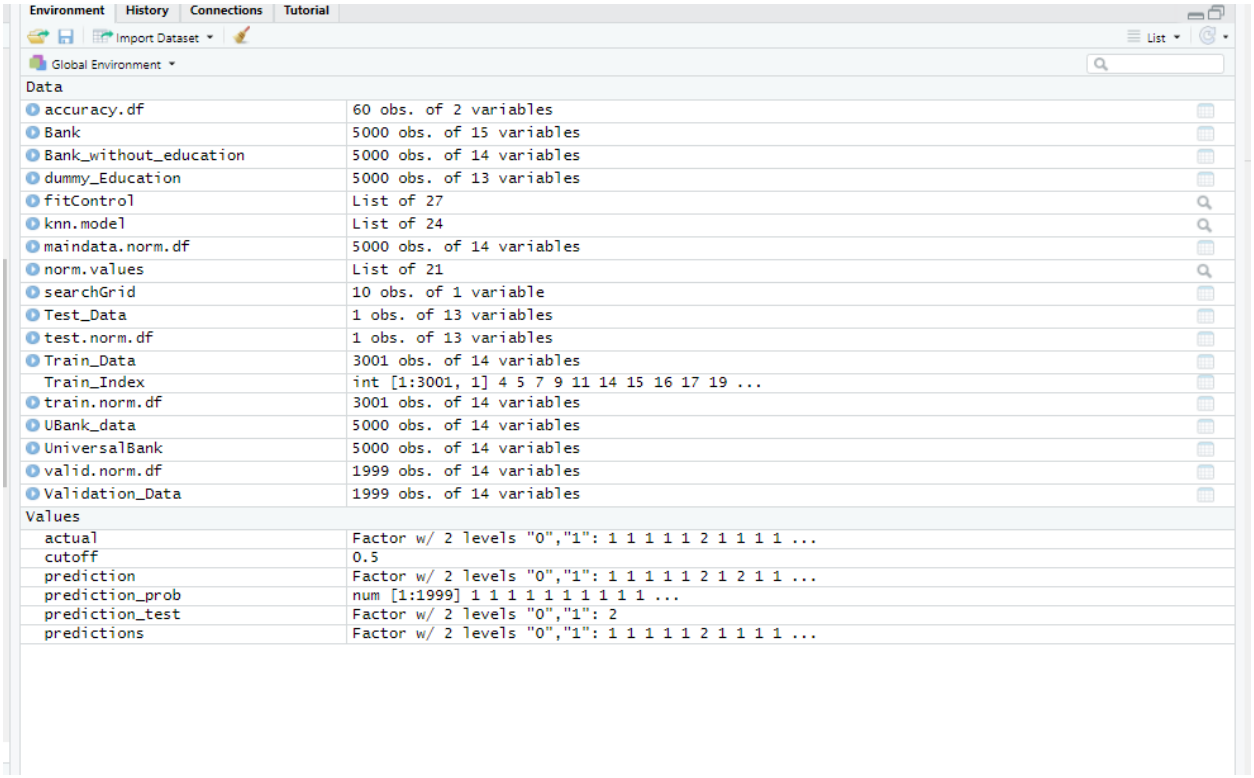
confusion matrix for the best k value =3

```
> confusionMatrix(predictions,valid.norm.df$`Personal Loan`)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1754  129
         1   61   55

               Accuracy : 0.905
                 95% CI : (0.8912, 0.9175)
    No Information Rate : 0.908
    P-Value [Acc > NIR] : 0.6952

                  Kappa : 0.3181

 Mcnemar's Test P-Value : 1.17e-06

            Sensitivity : 0.9664
            Specificity : 0.2989
         Pos Pred Value : 0.9315
         Neg Pred Value : 0.4741
             Prevalence : 0.9080
         Detection Rate : 0.8774
   Detection Prevalence : 0.9420
      Balanced Accuracy : 0.6327

       'Positive' Class : 0
```

4. Classifying the customer using the best k [performing k-NN classification on test data]

customer: 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 Credit Card = 1. Classify the customer using the best k

```
> Test_Data <- data.frame(Age=40 , Experience=10, Income = 84, Family = 2, CCAvg = 2, Education_Education_1 = 0, Education_Education_2 = 1, Education_Education_3 = 0, Mortgage =
 0, Securities.Account = 0, CD.Account = 0, Online = 1,  CreditCard = 1, stringsAsFactors = FALSE)
> maindata.norm.df <- as.data.frame(maindata.norm.df)
> head(prediction_test)
[1] 1
Levels: 0 1
>
```


Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Data
accuracy.df
60 obs. of 2 variables
Bank
5000 obs. of 15 variables
Bank_without_education
5000 obs. of 14 variables
dummy_Education
5000 obs. of 13 variables
fitControl
List of 27
knn.model
List of 24
maindata.norm.df
5000 obs. of 14 variables
norm.values
List of 21
searchGrid
10 obs. of 1 variable
Test_Data
1 obs. of 13 variables
test.norm.df
1 obs. of 13 variables
Train_Data
3001 obs. of 14 variables
Train_Index
int [1:3001, 1] 4 5 7 9 11 14 15 16 17 19 ...
train.norm.df
3001 obs. of 14 variables
UBank_data
5000 obs. of 14 variables
UniversalBank
5000 obs. of 14 variables
valid.norm.df
1999 obs. of 14 variables
Validation_Data
1999 obs. of 14 variables
Values
actual
Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
cutoff
0.5
prediction
Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 2 1 1 ...
prediction_prob
num [1:1999] 1 1 1 1 1 1 1 1 1 1 ...
prediction_test
Factor w/ 2 levels "0","1": 2
predictions
Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...