Riba-Assignment-2.R

ribakhan

2022-02-20

```
read.csv("~/Desktop/Fundamentals of Machine Learning/UniversalBank.csv")
```

```
##    ID Age Experience Income ZIP.Code Family
## 1   1  25          1     49    91107      4
## 2   2  45         19     34    90089      3
## 3   3  39         15     11    94720      1
## 4   4  35          9    100    94112      1
## 5   5  35          8     45    91330      4
## 6   6  37         13     29    92121      4
## 7   7  53         27     72    91711      2
## 8   8  50         24     22    93943      1
## 9   9  35         10     81    90089      3
## 10 10  34          9    180    93023      1
## 11 11  65         39    105    94710      4
## 12 12  29          5     45    90277      3
## 13 13  48         23    114    93106      2
## 14 14  59         32     40    94920      4
## 15 15  67         41    112    91741      1
## 16 16  60         30     22    95054      1
## 17 17  38         14    130    95010      4
## 18 18  42         18     81    94305      4
## 19 19  46         21    193    91604      2
## 20 20  55         28     21    94720      1
## 21 21  56         31     25    94015      4
## 22 22  57         27     63    90095      3
## 23 23  29          5     62    90277      1
## 24 24  44         18     43    91320      2
## 25 25  36         11    152    95521      2
## 26 26  43         19     29    94305      3
## 27 27  40         16     83    95064      4
## 28 28  46         20    158    90064      1
## 29 29  56         30     48    94539      1
## 30 30  38         13    119    94104      1
## 31 31  59         35     35    93106      1
## 32 32  40         16     29    94117      1
## 33 33  53         28     41    94801      2
## 34 34  30          6     18    91330      3
## 35 35  31          5     50    94035      4
## 36 36  48         24     81    92647      3
## 37 37  59         35    121    94720      1
## 38 38  51         25     71    95814      1
```

```
## 39 39  42         18    141    94114      3
## 40 40  38         13     80    94115      4
## 41 41  57         32     84    92672      3
## 42 42  34          9     60    94122      3
## 43 43  32          7    132    90019      4
## 44 44  39         15     45    95616      1
## 45 45  46         20    104    94065      1
## 46 46  57         31     52    94720      4
## 47 47  39         14     43    95014      3
## 48 48  37         12    194    91380      4
## 49 49  56         26     81    95747      2
## 50 50  40         16     49    92373      1
## 51 51  32          8      8    92093      4
## 52 52  61         37    131    94720      1
## 53 53  30          6     72    94005      1
## 54 54  50         26    190    90245      3
## 55 55  29          5     44    95819      1
## 56 56  41         17    139    94022      2
## 57 57  55         30     29    94005      3
## 58 58  56         31    131    95616      2
## 59 59  28          2     93    94065      2
## 60 60  31          5    188    91320      2
## 61 61  49         24     39    90404      3
## 62 62  47         21    125    93407      1
## 63 63  42         18     22    90089      1
## 64 64  42         17     32    94523      4
## 65 65  47         23    105    90024      2
## 66 66  59         35    131    91360      1
## 67 67  62         36    105    95670      2
## 68 68  53         23     45    95123      4
## 69 69  47         21     60    93407      3
## 70 70  53         29     20    90045      4
## 71 71  42         18    115    91335      1
##    CCAvg Education Mortgage Personal.Loan
## 1    1.6         1        0             0
## 2    1.5         1        0             0
## 3    1.0         1        0             0
## 4    2.7         2        0             0
## 5    1.0         2        0             0
## 6    0.4         2      155             0
## 7    1.5         2        0             0
## 8    0.3         3        0             0
## 9    0.6         2      104             0
## 10   8.9         3        0             1
## 11   2.4         3        0             0
## 12   0.1         2        0             0
## 13   3.8         3        0             0
## 14   2.5         2        0             0
## 15   2.0         1        0             0
## 16   1.5         3        0             0
## 17   4.7         3      134             1
## 18   2.4         1        0             0
## 19   8.1         3        0             1
## 20   0.5         2        0             0
```

```
## 21   0.9         2      111             0
## 22   2.0         3        0             0
## 23   1.2         1      260             0
## 24   0.7         1      163             0
## 25   3.9         1      159             0
## 26   0.5         1       97             0
## 27   0.2         3        0             0
## 28   2.4         1        0             0
## 29   2.2         3        0             0
## 30   3.3         2        0             1
## 31   1.2         3      122             0
## 32   2.0         2        0             0
## 33   0.6         3      193             0
## 34   0.9         3        0             0
## 35   1.8         3        0             0
## 36   0.7         1        0             0
## 37   2.9         1        0             0
## 38   1.4         3      198             0
## 39   5.0         3        0             1
## 40   0.7         3      285             0
## 41   1.6         3        0             0
## 42   2.3         1        0             0
## 43   1.1         2      412             1
## 44   0.7         1        0             0
## 45   5.7         1        0             0
## 46   2.5         1        0             0
## 47   0.7         2      153             0
## 48   0.2         3      211             1
## 49   4.5         3        0             0
## 50   1.8         1        0             0
## 51   0.7         2        0             0
## 52   2.9         1        0             0
## 53   0.1         1      207             0
## 54   2.1         3      240             1
## 55   0.2         3        0             0
## 56   8.0         1        0             0
## 57   0.1         2        0             0
## 58   1.2         3        0             1
## 59   0.2         1        0             0
## 60   4.5         1      455             0
## 61   1.7         2        0             0
## 62   5.7         1      112             0
## 63   1.0         1        0             0
## 64   0.0         2        0             0
## 65   3.3         1        0             0
## 66   3.8         1        0             0
## 67   2.8         1      336             0
## 68   2.0         3      132             0
## 69   2.1         1        0             0
## 70   0.2         1        0             0
## 71   3.5         1        0             0
##    Securities.Account CD.Account Online
## 1                   1          0      0
## 2                   1          0      0
```

```
## 3                0          0       0
## 4                0          0       0
## 5                0          0       0
## 6                0          0       1
## 7                0          0       1
## 8                0          0       0
## 9                0          0       1
## 10               0          0       0
## 11               0          0       0
## 12               0          0       1
## 13               1          0       0
## 14               0          0       1
## 15               1          0       0
## 16               0          0       1
## 17               0          0       0
## 18               0          0       0
## 19               0          0       0
## 20               1          0       0
## 21               0          0       1
## 22               0          0       1
## 23               0          0       1
## 24               1          0       0
## 25               0          0       0
## 26               0          0       1
## 27               0          0       0
## 28               0          0       1
## 29               0          0       1
## 30               0          1       1
## 31               0          0       1
## 32               0          0       1
## 33               0          0       0
## 34               0          0       0
## 35               0          0       1
## 36               0          0       0
## 37               0          0       0
## 38               0          0       0
## 39               1          1       1
## 40               0          0       1
## 41               1          0       0
## 42               0          0       0
## 43               0          0       1
## 44               0          0       1
## 45               0          0       1
## 46               0          0       0
## 47               0          0       1
## 48               1          1       1
## 49               0          0       0
## 50               0          0       0
## 51               1          0       1
## 52               0          0       1
## 53               0          0       0
## 54               0          0       1
## 55               0          0       1
## 56               0          0       1
```

```
## 57                  1          1       1
## 58                  0          0       0
## 59                  0          0       0
## 60                  0          0       0
## 61                  1          0       1
## 62                  1          0       0
## 63                  0          0       0
## 64                  0          0       1
## 65                  0          0       0
## 66                  0          0       1
## 67                  0          0       0
## 68                  1          0       0
## 69                  0          0       1
## 70                  0          0       1
## 71                  0          0       0
##    CreditCard
## 1           0
## 2           0
## 3           0
## 4           0
## 5           1
## 6           0
## 7           0
## 8           1
## 9           0
## 10          0
## 11          0
## 12          0
## 13          0
## 14          0
## 15          0
## 16          1
## 17          0
## 18          0
## 19          0
## 20          1
## 21          0
## 22          0
## 23          0
## 24          0
## 25          1
## 26          0
## 27          0
## 28          1
## 29          1
## 30          1
## 31          0
## 32          0
## 33          0
## 34          0
## 35          0
## 36          0
## 37          1
## 38          0
```

```
## 39          0
## 40          0
## 41          0
## 42          0
## 43          0
## 44          0
## 45          1
## 46          1
## 47          0
## 48          1
## 49          1
## 50          1
## 51          0
## 52          0
## 53          0
## 54          0
## 55          0
## 56          0
## 57          0
## 58          0
## 59          0
## 60          0
## 61          0
## 62          0
## 63          0
## 64          0
## 65          0
## 66          1
## 67          0
## 68          0
## 69          1
## 70          0
## 71          1
##  [ reached 'max' / getOption("max.print") -- omitted 4929 rows ]
```

```
summary(UniversalBank)
```

```
##       Age          Experience       Income
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00
##  Median :45.00   Median :20.0   Median : 64.00
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00
##  Max.   :67.00   Max.   :43.0   Max.   :224.00
##      Family          CCAvg        Education
##  Min.   :1.000   Min.   : 0.000   1:2096
##  1st Qu.:1.000   1st Qu.: 0.700   2:1403
##  Median :2.000   Median : 1.500   3:1501
##  Mean   :2.396   Mean   : 1.938
##  3rd Qu.:3.000   3rd Qu.: 2.500
##  Max.   :4.000   Max.   :10.000
##     Mortgage     Personal.Loan
##  Min.   :  0.0   0:4520
##  1st Qu.:  0.0   1: 480
```

```
##  Median :  0.0
##  Mean   : 56.5
##  3rd Qu.:101.0
##  Max.   :635.0
##  Securities.Account   CD.Account
##  Min.   :0.0000     Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000
##  Median :0.0000     Median :0.0000
##  Mean   :0.1044     Mean   :0.0604
##  3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   :1.0000     Max.   :1.0000
##      Online         CreditCard
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :1.0000   Median :0.000
##  Mean   :0.5968   Mean   :0.294
##  3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.000
```

```
#lets get rid of the two variables we don't need
UniversalBank$ID<-NULL
UniversalBank$ZIP.Code<-NULL
#Task 1
UniversalBank$Personal.Loan=as.factor(UniversalBank$Personal.Loan)
summary(UniversalBank)
```

```
##       Age           Experience       Income
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00
##  Median :45.00   Median :20.0   Median : 64.00
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00
##  Max.   :67.00   Max.   :43.0   Max.   :224.00
##      Family          CCAvg        Education
##  Min.   :1.000   Min.   : 0.000   1:2096
##  1st Qu.:1.000   1st Qu.: 0.700   2:1403
##  Median :2.000   Median : 1.500   3:1501
##  Mean   :2.396   Mean   : 1.938
##  3rd Qu.:3.000   3rd Qu.: 2.500
##  Max.   :4.000   Max.   :10.000
##     Mortgage     Personal.Loan
##  Min.   :  0.0   0:4520
##  1st Qu.:  0.0   1: 480
##  Median :  0.0
##  Mean   : 56.5
##  3rd Qu.:101.0
##  Max.   :635.0
##  Securities.Account   CD.Account
##  Min.   :0.0000     Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000
##  Median :0.0000     Median :0.0000
##  Mean   :0.1044     Mean   :0.0604
##  3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   :1.0000     Max.   :1.0000
```

```
##      Online         CreditCard
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :1.0000   Median :0.000
##  Mean   :0.5968   Mean   :0.294
##  3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.000
```

```
#Call the libraries
library(lattice)
library(ggplot2)
library(caret)
library(class)

#Normalisation of the data
UniversalBank1_Norm<-UniversalBank

Norm_model<-preProcess(UniversalBank[,-8],method = c("center","scale"))
UniversalBank1_Norm[,-8]=predict(Norm_model, UniversalBank[,-8])
summary(UniversalBank1_Norm)
```

```
##       Age              Experience
##  Min.   :-1.94871   Min.   :-2.014710
##  1st Qu.:-0.90188   1st Qu.:-0.881116
##  Median :-0.02952   Median :-0.009121
##  Mean   : 0.00000   Mean   : 0.000000
##  3rd Qu.: 0.84284   3rd Qu.: 0.862874
##  Max.   : 1.88967   Max.   : 1.996468
##      Income            Family
##  Min.   :-1.4288   Min.   :-1.2167
##  1st Qu.:-0.7554   1st Qu.:-1.2167
##  Median :-0.2123   Median :-0.3454
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.5263   3rd Qu.: 0.5259
##  Max.   : 3.2634   Max.   : 1.3973
##      CCAvg         Education    Mortgage
##  Min.   :-1.1089   1:2096    Min.   :-0.5555
##  1st Qu.:-0.7083   2:1403    1st Qu.:-0.5555
##  Median :-0.2506   3:1501    Median :-0.5555
##  Mean   : 0.0000             Mean   : 0.0000
##  3rd Qu.: 0.3216             3rd Qu.: 0.4375
##  Max.   : 4.6131             Max.   : 5.6875
##  Personal.Loan Securities.Account
##  0:4520        Min.   :-0.3414
##  1: 480        1st Qu.:-0.3414
##                Median :-0.3414
##                Mean   : 0.0000
##                3rd Qu.:-0.3414
##                Max.   : 2.9286
##    CD.Account          Online
##  Min.   :-0.2535   Min.   :-1.2165
##  1st Qu.:-0.2535   1st Qu.:-1.2165
##  Median :-0.2535   Median : 0.8219
##  Mean   : 0.0000   Mean   : 0.0000
```

```
##  3rd Qu.:-0.2535   3rd Qu.: 0.8219
##  Max.   : 3.9438   Max.   : 0.8219
##    CreditCard
##  Min.   :-0.6452
##  1st Qu.:-0.6452
##  Median :-0.6452
##  Mean   : 0.0000
##  3rd Qu.: 1.5495
##  Max.   : 1.5495
```

```
UniversalBank1_Norm$Personal.Loan=UniversalBank$Personal.Loan


#Train
train.index=createDataPartition(UniversalBank$Personal.Loan,p=0.6, list = FALSE)
train.df=UniversalBank1_Norm[train.index,]
valid.df=UniversalBank1_Norm[-train.index,]


#To predict
To_Predict=data.frame(Age=40, Experience=10, Income=84, Family=2, CCAvg=2, Education_1=0,
                      Mortgage=0, Securities.Account=0, CD.Account=0, Online=1, CreditCard=1)
print(To_Predict)
```

```
##   Age Experience Income Family CCAvg Education_1
## 1  40         10     84      2     2           0
##   Mortgage Securities.Account CD.Account Online
## 1        0                  0          0      1
##   CreditCard
## 1          1
```

```
#Now, we need to apply the normalisation to this record. We must use the same model.
To_Predict_norm=predict(Norm_model,To_Predict)
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975
##       CCAvg Education_1   Mortgage
## 1 0.0355115           0 -0.5554684
##   Securities.Account CD.Account    Online
## 1         -0.3413892 -0.2535149 0.8218687
##   CreditCard
## 1   1.549477
```

```
#Now we will use the Knn function to make the prediction
Prediction <-knn(train = train.df[,1:7],
                 test = To_Predict_norm[,1:7],
                 cl=train.df$Personal.Loan,
                 k=1)
print(Prediction)
```

```
## [1] 0
## Levels: 0 1
```

```r
#Task 2
#The best choice of K= 3 which prevents the model form over fitting and ignoring the predictor informat
#setseed
#setting the seed of the random number generator will make sure that results are productive
set.seed(123)

fitControl <- trainControl(method= "repeatedcv",
                           number = 3,
                           repeats = 2)

searchGrid=expand.grid(k =1:10)

knn.model=train(Personal.Loan~.,
                data=train.df,
                method='knn',
                tuneGrid = searchGrid,
                trControl = fitControl,)

knn.model
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   11 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 2 times)
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9496667  0.6740813
##    2  0.9390000  0.5992582
##    3  0.9448333  0.6055640
##    4  0.9420000  0.5753160
##    5  0.9396667  0.5490626
##    6  0.9395000  0.5491111
##    7  0.9381667  0.5244721
##    8  0.9385000  0.5310332
##    9  0.9391667  0.5291252
##   10  0.9370000  0.5078993
##
## Accuracy was used to select the optimal
##  model using the largest value.
## The final value used for the model was k = 1.
```

```r
Predictions<-predict(knn.model,valid.df)

#Task 3
#Confusion Matrix
confusionMatrix(Predictions, valid.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    0    1
##          0 1774   59
##          1   34  133
##
##                Accuracy : 0.9535
##                  95% CI : (0.9433, 0.9623)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7155
##
##  Mcnemar's Test P-Value : 0.01282
##
##             Sensitivity : 0.9812
##             Specificity : 0.6927
##          Pos Pred Value : 0.9678
##          Neg Pred Value : 0.7964
##              Prevalence : 0.9040
##          Detection Rate : 0.8870
##    Detection Prevalence : 0.9165
##       Balanced Accuracy : 0.8370
##
##        'Positive' Class : 0
##
```

```
#Task 4
library(class)
#prediction
customer.df= data.frame(Age=40, Experience=10, Income=84, Family=2, CCAvg=2, Education_1=0,
                        Mortgage=0, Securities.Account=0, CD.Account=0, Online=1, CreditCard=1)
knn.4 <- knn(train = train.df[,-8],test = customer.df, cl = train.df[,8], k=3, prob=TRUE)
knn.4
```

```
## [1] 1
## attr(,"prob")
## [1] 0.6666667
## Levels: 0 1
```

```
#Task 5
#Repartition the data this time into training, validation and test set.seed()
#50% data for training
#30% data for validation
#20% data for test sets
# K value used = 3

set.seed(1)
train.index <- sample(rownames(UniversalBank1_Norm), 0.5*dim(UniversalBank1_Norm)[1])
set.seed(1)
valid.index <- sample(setdiff(rownames(UniversalBank1_Norm),train.index), 0.3*dim(UniversalBank1_Norm)[
test.index = setdiff(rownames(UniversalBank1_Norm), union(train.index, valid.index))
```

```
train.df <- UniversalBank1_Norm[train.index,]
valid.df <- UniversalBank1_Norm[valid.index,]
test.df <- UniversalBank1_Norm[test.index,]


norm.values <- preProcess(train.df[, -c(8)], method=c("center", "scale"))
train.df[, -c(8)] <- predict(norm.values, train.df[, -c(8)])
valid.df[, -c(8)] <- predict(norm.values, valid.df[, -c(8)])
test.df[,-c(8)] <- predict(norm.values, test.df[,-c(8)])

testknn <- knn(train = train.df[,-c(8)],test = test.df[,-c(8)], cl = train.df[,8], k=3, prob=TRUE)
validknn <- knn(train = train.df[,-c(8)],test = valid.df[,-c(8)], cl = train.df[,8], k=3, prob=TRUE)
trainknn <- knn(train = train.df[,-c(8)],test = train.df[,-c(8)], cl = train.df[,8], k=3, prob=TRUE)

confusionMatrix(testknn, test.df[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 888  35
##          1   4  73
##
##                Accuracy : 0.961
##                  95% CI : (0.9471, 0.9721)
##     No Information Rate : 0.892
##     P-Value [Acc > NIR] : 1.389e-15
##
##                   Kappa : 0.7684
##
##  Mcnemar's Test P-Value : 1.556e-06
##
##             Sensitivity : 0.9955
##             Specificity : 0.6759
##          Pos Pred Value : 0.9621
##          Neg Pred Value : 0.9481
##              Prevalence : 0.8920
##          Detection Rate : 0.8880
##    Detection Prevalence : 0.9230
##       Balanced Accuracy : 0.8357
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(validknn, valid.df[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1351   51
##          1    9   89
```

```
##
##                Accuracy : 0.96
##                  95% CI : (0.9488, 0.9693)
##     No Information Rate : 0.9067
##     P-Value [Acc > NIR] : 1.882e-15
##
##                   Kappa : 0.7269
##
##  Mcnemar's Test P-Value : 1.203e-07
##
##             Sensitivity : 0.9934
##             Specificity : 0.6357
##          Pos Pred Value : 0.9636
##          Neg Pred Value : 0.9082
##              Prevalence : 0.9067
##          Detection Rate : 0.9007
##    Detection Prevalence : 0.9347
##       Balanced Accuracy : 0.8145
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(trainknn, train.df[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2259   63
##          1    9  169
##
##                Accuracy : 0.9712
##                  95% CI : (0.9639, 0.9774)
##     No Information Rate : 0.9072
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.809
##
##  Mcnemar's Test P-Value : 4.208e-10
##
##             Sensitivity : 0.9960
##             Specificity : 0.7284
##          Pos Pred Value : 0.9729
##          Neg Pred Value : 0.9494
##              Prevalence : 0.9072
##          Detection Rate : 0.9036
##    Detection Prevalence : 0.9288
##       Balanced Accuracy : 0.8622
##
##        'Positive' Class : 0
##
```

```
#Test Accuracy: 0.961
#Valid Accuracy: 0.96
#Training Accuracy: 0.9712
#As the model is being fit on the training data it would make sense to say that the classifications are
```