Machine learning - Assignement 2: KNN- classifciation

10-02-2022

```
library(caret)
library(ISLR)
library(class)
```

```
getwd()
```

```
## [1] "/Users/thupiliabhinav/Desktop"
```

```
setwd("/Users/thupiliabhinav/Desktop")
bankdata<- read.csv("UniversalBank.csv")
str(bankdata)
```

```
## '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 ...
```

```
head(bankdata)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```
summary(bankdata)
```

```
##        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
```

```
test.na <- is.na.data.frame('bankdata')
test.na
```

```
##       [,1]
## [1,] FALSE
```

```
library(dplyr)
bankdata2<-bankdata %>%
  select(Age, Experience, Income, Family, CCAvg, Education, Mortgage, Personal.Loan, Securities.Account

head(bankdata2)
```

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

```
#converting numerical variables to characters and factors.
bankdata2$Education<-as.character(bankdata2$Education)
is.character(bankdata$Education)
```

```
## [1] FALSE
```

```
bankdata2$Personal.Loan <- as.factor(bankdata2$Personal.Loan)
is.factor(bankdata2$Personal.Loan)
```

```
## [1] TRUE
```

```
dummymodel <- dummyVars(~Education, data = bankdata2)
head(predict(dummymodel, bankdata2))
```

```
##   Education1 Education2 Education3
## 1          1          0          0
## 2          1          0          0
## 3          1          0          0
## 4          0          1          0
## 5          0          1          0
## 6          0          1          0
```

```
bankdata3 <- predict(dummymodel, bankdata2)
```

```
bankdata4 <- bankdata2[,-6]
bankdata5 <- cbind(bankdata4,bankdata3)
head(bankdata5)
```

```
##   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 Education1 Education2 Education3
## 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          1          0
## 5          0      0          1          0          1          0
## 6          0      1          0          0          1          0
```

```
set.seed(15)
Train_index = createDataPartition(bankdata5$Personal.Loan,p=0.60, list = FALSE)
Train_data = bankdata5[Train_index,]
Validation_data = bankdata5[-Train_index,]
```

```
#creating test data for testing the model.
Test_bankdata <- data.frame(Age = 40,Experience = 10,Income = 84,Family = 2,CCAvg = 2,Mortgage = 0,Secu
Test_bankdata
```

```
##   Age Experience Income Family CCAvg Mortgage Securities.Account CD.Account
## 1  40         10     84      2     2        0                  0          0
##   Online CreditCard Education_1 Education_2 Education_3
## 1      1          1           0           1           0
```

```
training_model <- preProcess(Train_data[,-c(7, 12:14)], method=c("center", "scale"))
model_train <- predict(training_model, Train_data)
model_validate <- predict(training_model, Validation_data)
model_test <- predict(training_model,Test_bankdata)
summary(model_train)
```

```
##       Age            Experience            Income            Family
##  Min.   :-1.9325   Min.   :-1.997167   Min.   :-1.4435   Min.   :-1.2237
##  1st Qu.:-0.8857   1st Qu.:-0.864443   1st Qu.:-0.7619   1st Qu.:-1.2237
##  Median :-0.0134   Median : 0.006883   Median :-0.2341   Median :-0.3482
##  Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8589   3rd Qu.: 0.878210   3rd Qu.: 0.5355   3rd Qu.: 0.5273
##  Max.   : 1.9057   Max.   : 2.010934   Max.   : 3.3061   Max.   : 1.4028
##      CCAvg            Mortgage       Personal.Loan Securities.Account
##  Min.   :-1.1014   Min.   :-0.5591   0:2712        Min.   :-0.3388
##  1st Qu.:-0.7024   1st Qu.:-0.5591   1: 288        1st Qu.:-0.3388
##  Median :-0.2465   Median :-0.5591                 Median :-0.3388
##  Mean   : 0.0000   Mean   : 0.0000                 Mean   : 0.0000
##  3rd Qu.: 0.3234   3rd Qu.: 0.4322                 3rd Qu.:-0.3388
##  Max.   : 4.5978   Max.   : 5.6581                 Max.   : 2.9506
##    CD.Account          Online          CreditCard       Education1
##  Min.   :-0.2404   Min.   :-1.1928   Min.   :-0.640   Min.   :0.0000
##  1st Qu.:-0.2404   1st Qu.:-1.1928   1st Qu.:-0.640   1st Qu.:0.0000
##  Median :-0.2404   Median : 0.8381   Median :-0.640   Median :0.0000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.000   Mean   :0.4163
##  3rd Qu.:-0.2404   3rd Qu.: 0.8381   3rd Qu.: 1.562   3rd Qu.:1.0000
##  Max.   : 4.1578   Max.   : 0.8381   Max.   : 1.562   Max.   :1.0000
##    Education2       Education3
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2873   Mean   :0.2963
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
#Predictors and Lables
Train_Bank_Predictors <- model_train[,-7]
Validate_Bank_Predictors <- model_validate[,-7]

Train_Bank_Label <- model_train[,7]
Validate_Bank_Label <- model_validate[,7]

K_NNmodel <- knn(Train_Bank_Predictors, model_test, cl= Train_Bank_Label, k=1)
K_NNmodel
```

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

```
#For K=1 The customer is not accepting loan since the value is 0.
```

```
set.seed(123)
searchgrid <- expand.grid(k=c(1:40))
trtcontrol =
model <- train(Personal.Loan~.,data=model_train,tuneGrid = searchgrid, method="knn", trControl = trainC
model
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2700, 2700, 2700, 2701, 2700, 2700, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9573388  0.7296140
##    2  0.9463399  0.6638783
##    3  0.9536755  0.6846372
##    4  0.9543421  0.6934557
##    5  0.9523410  0.6672497
##    6  0.9503443  0.6424353
##    7  0.9483377  0.6243728
##    8  0.9470054  0.6106669
##    9  0.9466710  0.6006755
##   10  0.9453365  0.5862069
##   11  0.9436721  0.5724294
##   12  0.9420021  0.5534162
##   13  0.9403376  0.5375017
##   14  0.9396676  0.5356510
##   15  0.9396665  0.5294763
##   16  0.9386676  0.5207480
##   17  0.9396654  0.5252523
##   18  0.9393343  0.5189330
##   19  0.9390010  0.5192287
##   20  0.9379999  0.5095905
##   21  0.9369999  0.4945730
##   22  0.9370021  0.4964004
##   23  0.9353332  0.4776383
##   24  0.9336688  0.4611509
##   25  0.9343343  0.4647335
##   26  0.9340021  0.4619404
##   27  0.9336687  0.4597632
##   28  0.9326688  0.4490780
##   29  0.9336687  0.4555131
##   30  0.9336665  0.4549865
##   31  0.9316687  0.4329774
##   32  0.9313343  0.4322780
##   33  0.9310010  0.4283798
##   34  0.9306665  0.4233367
##   35  0.9300010  0.4150341
##   36  0.9293332  0.4054952
##   37  0.9280021  0.3973478
##   38  0.9279987  0.3973784
##   39  0.9279999  0.3940351
##   40  0.9269987  0.3837465
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
best_k <- model$bestTune[[1]]
```

```
model_v <- knn(Train_Bank_Predictors,Validate_Bank_Predictors,cl=Train_Bank_Label, k=best_k)

confusionMatrix(model_v,Validate_Bank_Label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1767   69
##          1   41  123
##
##                Accuracy : 0.945
##                  95% CI : (0.9341, 0.9546)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 1.359e-11
##
##                   Kappa : 0.661
##
##  Mcnemar's Test P-Value : 0.01004
##
##             Sensitivity : 0.9773
##             Specificity : 0.6406
##          Pos Pred Value : 0.9624
##          Neg Pred Value : 0.7500
##              Prevalence : 0.9040
##          Detection Rate : 0.8835
##    Detection Prevalence : 0.9180
##       Balanced Accuracy : 0.8090
##
##        'Positive' Class : 0
##
```

```
set.seed(123)
banktraindata <- createDataPartition(bankdata5$Personal.Loan, p=0.5, list = FALSE)
m_train_bankdata <- bankdata5[banktraindata,]
m_test_bankdata <- bankdata5[-banktraindata,]

bankdata7 <- createDataPartition(m_test_bankdata$Personal.Loan, p=0.6, list = FALSE)
m_validate_bankdata <- m_test_bankdata[bankdata7,]
m_test1_bankdata <- m_test_bankdata[-bankdata7,]
```

```
norm_bankdata <- preProcess(m_train_bankdata[,-c(7,12:14)], method = c("center", "scale"))

bankdata_train <- predict(norm_bankdata, m_train_bankdata)
bankdata_validate <- predict(norm_bankdata, m_validate_bankdata)
bankdata_test <- predict(norm_bankdata, m_test1_bankdata)
```

```
#defining predictors and labels

m_train_predictor <- bankdata_train[,-7]
m_validate_predictor <- bankdata_validate[,-7]
m_test_predictor <- bankdata_test[,-7]
```

```
m_train_label <- bankdata_train[,7]
m_validate_label<- bankdata_validate[,7]
m_test_label <- bankdata_test[,7]
```

```
m_bankmodel <- knn(m_train_predictor, m_train_predictor, cl=m_train_label, k=best_k)
head(m_bankmodel)
```

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

```
m_bankdatamodel <- knn(m_train_predictor, m_validate_predictor, cl=m_train_label, k=best_k)
head(m_bankdatamodel)
```

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

```
m_bankmodel2 <- knn(m_train_predictor, m_test_predictor, cl=m_train_label, k=best_k)
head(m_bankmodel2)
```

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

```
confusionMatrix(m_bankmodel, m_train_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2260    0
##          1    0  240
##
##                Accuracy : 1
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.000
##             Specificity : 1.000
##          Pos Pred Value : 1.000
##          Neg Pred Value : 1.000
##              Prevalence : 0.904
##          Detection Rate : 0.904
##    Detection Prevalence : 0.904
##       Balanced Accuracy : 1.000
##
##        'Positive' Class : 0
##
```

```
#Number of miscalculations = 0. Accuracy is 100% for training model.
```

```
confusionMatrix(m_bankdatamodel, m_validate_label)
```

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

```
##
##           Reference
## Prediction    0    1
##          0 1335   47
##          1   21   97
##
##                Accuracy : 0.9547
##                  95% CI : (0.9429, 0.9646)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 1.551e-13
##
##                   Kappa : 0.7159
##
##  Mcnemar's Test P-Value : 0.002432
##
##             Sensitivity : 0.9845
##             Specificity : 0.6736
##          Pos Pred Value : 0.9660
##          Neg Pred Value : 0.8220
##              Prevalence : 0.9040
##          Detection Rate : 0.8900
##    Detection Prevalence : 0.9213
##       Balanced Accuracy : 0.8291
##
##        'Positive' Class : 0
##
```

```
#Number of miscalculations = 68. Accuracy is 95% for validation model.
```

```
confusionMatrix(m_bankmodel2, m_test_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 899  31
##          1   5  65
##
##                Accuracy : 0.964
##                  95% CI : (0.9505, 0.9747)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 2.787e-13
##
##                   Kappa : 0.764
##
##  Mcnemar's Test P-Value : 3.091e-05
##
##             Sensitivity : 0.9945
##             Specificity : 0.6771
##          Pos Pred Value : 0.9667
##          Neg Pred Value : 0.9286
##              Prevalence : 0.9040
##          Detection Rate : 0.8990
##    Detection Prevalence : 0.9300
##       Balanced Accuracy : 0.8358
```

```
##
##        'Positive' Class : 0
##
```

```
#Number of miscalculations = 36. Accuracy is 96% for Test Model.
```