Assignment-2

Nikitha Chigurupati

10/5/2022

```
#Importing the required packages
library('caret')
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
library('ISLR')
```

```
## Warning: package 'ISLR' was built under R version 4.1.3
```

```
library('dplyr')
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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')
```

```
## Warning: package 'class' was built under R version 4.1.3
```

```
#Importing the data into R Studio
UniversalBank <- read.csv("C:/Users/Nikitha/Downloads/UniversalBank.csv")
```

#QUESTION-1

```
#Performing a K-NN classification with all attributes except ID and ZIP code.
UniversalBank$ID <- NULL
UniversalBank$ZIP.Code <- NULL
summary(UniversalBank)
```

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

```
UniversalBank$Personal.Loan =  as.factor(UniversalBank$Personal.Loan)
```

```
#Using the preProcess() from the caret package to normalize the data by dividing into training a
nd validation
Model_norm <- preProcess(UniversalBank[, -8],method = c("center", "scale"))
summary(UniversalBank)
```

```
##       Age          Experience       Income           Family
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.0   Median : 64.00   Median :2.000
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg          Education        Mortgage     Personal.Loan
##  Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4520
##  1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 480
##  Median : 1.500   Median :2.000   Median :  0.0
##  Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :10.000   Max.   :3.000   Max.   :635.0
##  Securities.Account   CD.Account         Online         CreditCard
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000     Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.1044     Mean   :0.0604   Mean   :0.5968   Mean   :0.294
##  3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

```
UniversalBank_norm <- predict(Model_norm,UniversalBank)
summary(UniversalBank_norm)
```

```
##       Age             Experience            Income            Family
##  Min.   :-1.94871   Min.   :-2.014710   Min.   :-1.4288   Min.   :-1.2167
##  1st Qu.:-0.90188   1st Qu.:-0.881116   1st Qu.:-0.7554   1st Qu.:-1.2167
##  Median :-0.02952   Median :-0.009121   Median :-0.2123   Median :-0.3454
##  Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84284   3rd Qu.: 0.862874   3rd Qu.: 0.5263   3rd Qu.: 0.5259
##  Max.   : 1.88967   Max.   : 1.996468   Max.   : 3.2634   Max.   : 1.3973
##      CCAvg           Education          Mortgage       Personal.Loan
##  Min.   :-1.1089   Min.   :-1.0490   Min.   :-0.5555   0:4520
##  1st Qu.:-0.7083   1st Qu.:-1.0490   1st Qu.:-0.5555   1: 480
##  Median :-0.2506   Median : 0.1417   Median :-0.5555
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3216   3rd Qu.: 1.3324   3rd Qu.: 0.4375
##  Max.   : 4.6131   Max.   : 1.3324   Max.   : 5.6875
##  Securities.Account   CD.Account          Online          CreditCard
##  Min.   :-0.3414    Min.   :-0.2535   Min.   :-1.2165   Min.   :-0.6452
##  1st Qu.:-0.3414    1st Qu.:-0.2535   1st Qu.:-1.2165   1st Qu.:-0.6452
##  Median :-0.3414    Median :-0.2535   Median : 0.8219   Median :-0.6452
##  Mean   : 0.0000    Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.3414    3rd Qu.:-0.2535   3rd Qu.: 0.8219   3rd Qu.: 1.5495
##  Max.   : 2.9286    Max.   : 3.9438   Max.   : 0.8219   Max.   : 1.5495
```

```
Index_Train <- createDataPartition(UniversalBank$Personal.Loan, p = 0.6, list = FALSE)
Train = UniversalBank_norm[Index_Train,]
validation = UniversalBank_norm[-Index_Train,]
```

```
#Prediction of data
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.1.3
```

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

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

```
Predict = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2,
                     CCAvg = 2, Education = 1, Mortgage = 0, Securities.Account =
                       0, CD.Account = 0, Online = 1, CreditCard = 1)
print(Predict)
```

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

```
Predict_Norm <- predict(Model_norm,Predict)
Prediction <- knn(train= as.data.frame(Train[,1:7,9:12]),
                  test = as.data.frame(Predict_Norm[,1:7,9:12]),
                  cl= Train$Personal.Loan,
                  k=1)
print(Prediction)
```

```
## [1] 0
## attr(,"nn.index")
##      [,1]
## [1,]  429
## attr(,"nn.dist")
##           [,1]
## [1,] 0.2986486
## Levels: 0
```

#QUESTION-2

```
set.seed(123)
UniversalBank <- trainControl(method= "repeatedcv", number = 3, repeats = 2)
searchGrid = expand.grid(k=1:10)
knn.model = train(Personal.Loan~., data = Train, method = 'knn', tuneGrid = searchGrid,trControl
= UniversalBank)
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  0.9490000  0.6711762
##    2  0.9435000  0.6356073
##    3  0.9531667  0.6840358
##    4  0.9521667  0.6731541
##    5  0.9506667  0.6553849
##    6  0.9491667  0.6403024
##    7  0.9473333  0.6169714
##    8  0.9463333  0.6087688
##    9  0.9463333  0.6075056
##   10  0.9451667  0.5947116
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

```
#The value of k is 3, which strikes a compromise between underfitTing and overfitting of the dat
a.
#Accuracy was used to select the optimal model using the largest value for the model was k = 3.
```

#QUESTION-3

```
prediction_of_bank <- predict(knn.model,validation)
confusionMatrix(prediction_of_bank,validation$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1797   72
##          1   11  120
##
##                Accuracy : 0.9585
##                  95% CI : (0.9488, 0.9668)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7213
##
##  Mcnemar's Test P-Value : 4.523e-11
##
##             Sensitivity : 0.9939
##             Specificity : 0.6250
##          Pos Pred Value : 0.9615
##          Neg Pred Value : 0.9160
##              Prevalence : 0.9040
##          Detection Rate : 0.8985
##    Detection Prevalence : 0.9345
##       Balanced Accuracy : 0.8095
##
##        'Positive' Class : 0
##
```

```
#This matrix has a 95.9% accuracy.
```

#QUESTION-4

```
For_Predict_Norm = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2,
                              CCAvg = 2, Education = 1, Mortgage = 0,
                              Securities.Account =0, CD.Account = 0, Online = 1,
                              CreditCard = 1)
For_Predict_Norm = predict(Model_norm, For_Predict_Norm)
predict(knn.model, For_Predict_Norm)
```

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

#QUESTION-5

```
#Creating Training, Test, and validation sets from the data collection.
Train_size = 0.5 #training(50%)
Index_Train = createDataPartition(UniversalBank_norm$Personal.Loan, p = 0.5, list = FALSE)
Train = UniversalBank_norm[Index_Train,]

valid_size = 0.3 #validation(30%)
Index_Validation = createDataPartition(UniversalBank_norm$Personal.Loan, p = 0.3, list = FALSE)
validation = UniversalBank_norm[Index_Validation,]

Test_size = 0.2 #Test Data(20%)
Index_Test = createDataPartition(UniversalBank_norm$Personal.Loan, p = 0.2, list = FALSE)
Test = UniversalBank_norm[Index_Test,]


Trainingknn <- knn(train = Train[,-8], test = Train[,-8], cl = Train[,8], k =3)
Validknn <- knn(train = Train[,-8], test = validation[,-8], cl = Train[,8], k =3)
Testingknn <- knn(train = Train[,-8], test = Test[,-8], cl = Train[,8], k =3)

Train_Predictors<-Train[,9:12]
Test_Predictors<-Test[,9:12]

Train_labels <-Train[,8]
Test_labels  <-Test[,8]

Predicted_Test_labels <-knn(Train_Predictors,
                            Test_Predictors,
                            cl=Train_labels,
                            k=3 )

library("gmodels")
```

```
## Warning: package 'gmodels' was built under R version 4.1.3
```

```
CrossTable(x=Test_labels,y=Predicted_Test_labels, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | Predicted_Test_labels
##  Test_labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       893 |        11 |       904 |
##              |     0.988 |     0.012 |     0.904 |
##              |     0.923 |     0.333 |           |
##              |     0.893 |     0.011 |           |
## -------------|-----------|-----------|-----------|
##            1 |        74 |        22 |        96 |
##              |     0.771 |     0.229 |     0.096 |
##              |     0.077 |     0.667 |           |
##              |     0.074 |     0.022 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       967 |        33 |      1000 |
##              |     0.967 |     0.033 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
confusionMatrix(Trainingknn, Train[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2255   59
##          1    5  181
##
##                Accuracy : 0.9744
##                  95% CI : (0.9674, 0.9802)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.836
##
##  Mcnemar's Test P-Value : 3.472e-11
##
##             Sensitivity : 0.9978
##             Specificity : 0.7542
##          Pos Pred Value : 0.9745
##          Neg Pred Value : 0.9731
##              Prevalence : 0.9040
##          Detection Rate : 0.9020
##    Detection Prevalence : 0.9256
##       Balanced Accuracy : 0.8760
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(Validknn, validation[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1353   45
##          1    3   99
##
##                Accuracy : 0.968
##                  95% CI : (0.9578, 0.9763)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.788
##
##  Mcnemar's Test P-Value : 3.262e-09
##
##             Sensitivity : 0.9978
##             Specificity : 0.6875
##          Pos Pred Value : 0.9678
##          Neg Pred Value : 0.9706
##              Prevalence : 0.9040
##          Detection Rate : 0.9020
##    Detection Prevalence : 0.9320
##       Balanced Accuracy : 0.8426
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(Testingknn, Test[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 900  23
##          1   4  73
##
##                Accuracy : 0.973
##                  95% CI : (0.961, 0.9821)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8293
##
##  Mcnemar's Test P-Value : 0.000532
##
##             Sensitivity : 0.9956
##             Specificity : 0.7604
##          Pos Pred Value : 0.9751
##          Neg Pred Value : 0.9481
##              Prevalence : 0.9040
##          Detection Rate : 0.9000
##    Detection Prevalence : 0.9230
##       Balanced Accuracy : 0.8780
##
##        'Positive' Class : 0
##
```