Assignment 2

lava kumar

2/17/2022

```
#First load the all the required packages using library for smooth execution.
library('caret')
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library('ISLR')
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(readr)
library(gmodels)
library(FNN)
```

```
##
## Attaching package: ’FNN’

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

```
#To import the UniversalBank.csv file
Universal_Bank <- read.csv("C:/Users/lavak/Documents/R/Assignment2/UniversalBank.csv")
colnames(Universal_Bank)    #for displaying the column names
```

```
##  [1] "ID"                 "Age"                "Experience"
##  [4] "Income"             "ZIP.Code"           "Family"
##  [7] "CCAvg"              "Education"          "Mortgage"
## [10] "Personal.Loan"      "Securities.Account" "CD.Account"
## [13] "Online"             "CreditCard"
```

```
summary(Universal_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
```

```
#Removing columns ID and ZIP.Code  as per the Question instruction by assigning them to NULL
Universal_Bank$ID <- NULL
Universal_Bank$ZIP.Code <- NULL
summary(Universal_Bank)
```

```
##       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
```

```
#Making the Personal Loan column as factor value

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

```
#Applying normalization to Universal_Bank dataset
Normal_model <- preProcess(Universal_Bank,method = "range")
Universal_Bank_Norm <- predict(Normal_model,Universal_Bank)
summary(Universal_Bank_Norm)
```

```
##       Age           Experience         Income           Family      
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.2727   1st Qu.:0.2826   1st Qu.:0.1435   1st Qu.:0.0000  
##  Median :0.5000   Median :0.5000   Median :0.2593   Median :0.3333  
##  Mean   :0.5077   Mean   :0.5023   Mean   :0.3045   Mean   :0.4655  
##  3rd Qu.:0.7273   3rd Qu.:0.7174   3rd Qu.:0.4167   3rd Qu.:0.6667  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
##      CCAvg          Education         Mortgage       Personal.Loan
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   0:4520       
##  1st Qu.:0.0700   1st Qu.:0.0000   1st Qu.:0.00000   1: 480       
##  Median :0.1500   Median :0.5000   Median :0.00000                
##  Mean   :0.1938   Mean   :0.4405   Mean   :0.08897                
##  3rd Qu.:0.2500   3rd Qu.:1.0000   3rd Qu.:0.15906                
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000                
##  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
```

```
#As per the question instructions partitioning  the data into 60% for training and 40% for testing

Train_index <- createDataPartition(Universal_Bank$Personal.Loan, p = 0.6, list = FALSE)
train.df = Universal_Bank_Norm[Train_index,]
validation.df = Universal_Bank_Norm[-Train_index,]
```

```
#Task 1 classifying the customer as per the date provided in the question 1
To_Predict = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education = 1, M
print(To_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
```

```
Prediction <- knn(train = train.df[,1:7],
                  test = To_Predict[,1:7], cl = train.df$Personal.Loan, k = 1)
print(Prediction)
```

```
## [1] 1
## attr(,"nn.index")
##      [,1]
## [1,] 1804
## attr(,"nn.dist")
##          [,1]
## [1,] 92.34726
## Levels: 1
```

```
#Customer is classified as 1.
```

```
#Task2

set.seed(123)
Universal_Bank_control <- 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 = U

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  0.9546667  0.7070340
##    2  0.9460000  0.6543675
##    3  0.9533333  0.6759682
##    4  0.9490000  0.6382196
##    5  0.9458333  0.6016805
##    6  0.9428333  0.5783428
##    7  0.9396667  0.5421195
##    8  0.9385000  0.5310286
##    9  0.9375000  0.5165528
##   10  0.9358333  0.4984294
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
#The choice of K that balances between overfitting and ignoring predictor information appears as K=3
```

```
#Question 3
#confusion matrix for the validation data that results from using the best k.

predictions <- predict(knn.model,validation.df)

confusionMatrix(predictions,validation.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction     0    1
##          0 1789   54
##          1   19  138
##
##                Accuracy : 0.9635
##                  95% CI : (0.9543, 0.9713)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7711
##
##  Mcnemar's Test P-Value : 6.909e-05
##
##             Sensitivity : 0.9895
##             Specificity : 0.7188
##          Pos Pred Value : 0.9707
##          Neg Pred Value : 0.8790
##              Prevalence : 0.9040
##          Detection Rate : 0.8945
##    Detection Prevalence : 0.9215
##       Balanced Accuracy : 0.8541
##
##        'Positive' Class : 0
##
```

```
#Question 4
#Classify the customer using the best k
To_Predict_Normaliz = 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)
To_Predict_Normaliz = predict(Normal_model, To_Predict)
predict(knn.model, To_Predict_Normaliz)
```

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

```
#Question 5
#As per the question instructions Repartition the data into 50% for training ,30%  for validation, 20%
train_size = 0.5
```

```
Train_index = createDataPartition(Universal_Bank$Personal.Loan, p = 0.5, list = FALSE)
train.df = Universal_Bank_Norm[Train_index,]

test_size = 0.2
Test_index = createDataPartition(Universal_Bank$Personal.Loan, p = 0.2, list = FALSE)
Test.df = Universal_Bank_Norm[Test_index,]

valid_size = 0.3
Validation_index = createDataPartition(Universal_Bank$Personal.Loan, p = 0.3, list = FALSE)
validation.df = Universal_Bank_Norm[Validation_index,]

Testingknn <- knn(train = train.df[,-8], test = Test.df[,-8], cl = train.df[,8], k =3)
Validationknn <- knn(train = train.df[,-8], test = validation.df[,-8], cl = train.df[,8], k =3)
Trainingknn <- knn(train = train.df[,-8], test = train.df[,-8], cl = train.df[,8], k =3)

#Comparing the confusion matrix of the test set with that of the training and validation sets.

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 901  36
##          1   3  60
##
##                Accuracy : 0.961
##                  95% CI : (0.9471, 0.9721)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 5.695e-12
##
##                   Kappa : 0.7345
##
##  Mcnemar’s Test P-Value : 2.990e-07
##
##             Sensitivity : 0.9967
##             Specificity : 0.6250
##          Pos Pred Value : 0.9616
##          Neg Pred Value : 0.9524
##              Prevalence : 0.9040
##          Detection Rate : 0.9010
##    Detection Prevalence : 0.9370
##       Balanced Accuracy : 0.8108
##
##        ’Positive’ Class : 0
##
```

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2254   52
##          1    6  188
##
##                Accuracy : 0.9768
##                  95% CI : (0.9701, 0.9823)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8538
##
##  Mcnemar’s Test P-Value : 3.446e-09
##
##             Sensitivity : 0.9973
##             Specificity : 0.7833
##          Pos Pred Value : 0.9775
##          Neg Pred Value : 0.9691
##              Prevalence : 0.9040
##          Detection Rate : 0.9016
##    Detection Prevalence : 0.9224
##       Balanced Accuracy : 0.8903
##
##        ’Positive’ Class : 0
##
```

```
confusionMatrix(Validationknn, validation.df[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1347   45
##          1    9   99
##
##                Accuracy : 0.964
##                  95% CI : (0.9533, 0.9728)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7665
##
##  Mcnemar’s Test P-Value : 1.908e-06
##
##             Sensitivity : 0.9934
##             Specificity : 0.6875
##          Pos Pred Value : 0.9677
##          Neg Pred Value : 0.9167
##              Prevalence : 0.9040
##          Detection Rate : 0.8980
##    Detection Prevalence : 0.9280
##       Balanced Accuracy : 0.8404
##
##        ’Positive’ Class : 0
##
```

```
#After comparing the date obtained from both confusion matrices. We can observe it training accuracy is

#We can also  determine that Training accuracy is slightly higher than the test and validation sets whi
```