Assignment_2

Manasa Chelukala

2/20/2022

#Importing the Dataset

```
UniversalBank <- read.csv('C:/Users/HP/Desktop/Machine learning slides/M-3/UniversalBank.csv')

summary(UniversalBank)
```

```
##        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 ID & ZIP.Code Variables

```
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
```

#Calling Libraries

```
library(caret)
```

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

```
## Warning in register(): Can't find generic 'scale_type' in package ggplot2 to
## register S3 method.
```

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

```
library(class)
```

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

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 <- UniversalBank
```

#Normalizing the data

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

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
```

#Dividing the data

```
Train_Index = createDataPartition(UniversalBank$Personal.Loan,p=0.6, list=FALSE) # 60% reserved for Tra
Train.df=UniversalBank_norm[Train_Index,]
Validation.df=UniversalBank_norm[-Train_Index,]
```

#TASK-1

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

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

```
To_Predict_norm=predict(Norm_model,To_Predict)
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -2.239635 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

```
Prediction <-knn(train=Train.df[,1:7,9:12],
                 test=To_Predict_norm[,1:7,9:12],
                 cl=Train.df$Personal.Loan,
                 k=1)
print(Prediction)
```

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

->Here the customer is classified as '0'.So,the customer doesn't accepts the loan offer.

#TASK-2

```
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  0.9500000  0.6786152
##    2  0.9430000  0.6410927
##    3  0.9516667  0.6686382
##    4  0.9496667  0.6498429
##    5  0.9478333  0.6227188
##    6  0.9480000  0.6284429
##    7  0.9488333  0.6296237
##    8  0.9470000  0.6127640
##    9  0.9453333  0.5944008
##   10  0.9435000  0.5749530
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

-> 'k = 3' is the optimum choice for k that also prevents the model from overfitting.

#Task-3

```
predictions<-predict(Knn.model,Validation.df)


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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1799   65
##          1    9  127
##
##                Accuracy : 0.963
##                  95% CI : (0.9538, 0.9708)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7549
##
##  Mcnemar's Test P-Value : 1.62e-10
##
##             Sensitivity : 0.9950
##             Specificity : 0.6615
##          Pos Pred Value : 0.9651
##          Neg Pred Value : 0.9338
##              Prevalence : 0.9040
##          Detection Rate : 0.8995
##    Detection Prevalence : 0.9320
##       Balanced Accuracy : 0.8282
```

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

-> The Confusion matrix for the above k-value.

#Task-4

```
To_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(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
```

```
To_Predict_norm=predict(Norm_model,To_Predict)
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -1.048973 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

```
Prediction <-knn(train=Train.df[,1:7,9:12],
                 test=To_Predict_norm[,1:7,9:12],
                 cl=Train.df$Personal.Loan,
                 k=3)
print(Prediction)
```

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

-> Here the customer is classified as '0'.So,the customer doesn't accepts the loan offer.

#Task-5

```
set.seed(123)

train.rows <- sample(rownames(UniversalBank), dim(UniversalBank)[1] * .50)

validation.rows <- sample(setdiff(rownames(UniversalBank), train.rows), dim(UniversalBank)[1]*0.30)

test.rows <- setdiff(rownames(UniversalBank), union(train.rows, validation.rows))
```

```
train.data <- UniversalBank[train.rows,]
rownames(train.data) <- NULL

validation.data <- UniversalBank[validation.rows,]
rownames(validation.data) <- NULL

test.data <- UniversalBank[test.rows,]
rownames(validation.data) <- NULL
```

```r
Test_knn<-knn(train=train.data[,-8],test
              =test.data[,-8],cl= train.data[,8], k=3)

Validation_knn<-knn(train = train.data[,-8],test = validation.data[,-8],cl = train.data[,8], k=3)

Train_knn<-knn(train = train.data[,-8],test = train.data[,-8],cl = train.data[,8], k=3)
```

```r
confusionMatrix(Test_knn, test.data[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 845  67
##          1  47  41
##
##                Accuracy : 0.886
##                  95% CI : (0.8647, 0.905)
##     No Information Rate : 0.892
##     P-Value [Acc > NIR] : 0.74848
##
##                   Kappa : 0.3559
##
##  Mcnemar's Test P-Value : 0.07516
##
##             Sensitivity : 0.9473
##             Specificity : 0.3796
##          Pos Pred Value : 0.9265
##          Neg Pred Value : 0.4659
##              Prevalence : 0.8920
##          Detection Rate : 0.8450
##    Detection Prevalence : 0.9120
##       Balanced Accuracy : 0.6635
##
##        'Positive' Class : 0
##
```

->Test Accuracy = 0.886

```r
confusionMatrix(Validation_knn, validation.data[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1296   89
##          1   61   54
##
##                Accuracy : 0.9
##                  95% CI : (0.8837, 0.9147)
##     No Information Rate : 0.9047
##     P-Value [Acc > NIR] : 0.74733
```

```
##
##                   Kappa : 0.3646
##
##  Mcnemar's Test P-Value : 0.02749
##
##             Sensitivity : 0.9550
##             Specificity : 0.3776
##          Pos Pred Value : 0.9357
##          Neg Pred Value : 0.4696
##              Prevalence : 0.9047
##          Detection Rate : 0.8640
##    Detection Prevalence : 0.9233
##       Balanced Accuracy : 0.6663
##
##        'Positive' Class : 0
##
```

->Validation Accuracy = 0.9

```
confusionMatrix(Train_knn, train.data[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2232   89
##          1   39  140
##
##                Accuracy : 0.9488
##                  95% CI : (0.9394, 0.9571)
##     No Information Rate : 0.9084
##     P-Value [Acc > NIR] : 2.497e-14
##
##                   Kappa : 0.6589
##
##  Mcnemar's Test P-Value : 1.484e-05
##
##             Sensitivity : 0.9828
##             Specificity : 0.6114
##          Pos Pred Value : 0.9617
##          Neg Pred Value : 0.7821
##              Prevalence : 0.9084
##          Detection Rate : 0.8928
##    Detection Prevalence : 0.9284
##       Balanced Accuracy : 0.7971
##
##        'Positive' Class : 0
##
```

-> Train Accuracy = 0.9488

-> Here,the classifications should be most accurate on the training data set and least accurate on the test data sets, given that the model is fitted on the training data.