KNN Algorithm

Arbuda Sivani

2/20/2022

```
#Import
UniversalBank <- read.csv("~/ML/Assignment/Assignement_2/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
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
```

```
#Installing packages
library(caret)
```

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

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

```
library(class)
```

```
#Normalization
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
```

```
UnivBank_norm<- UniversalBank

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

```
#Data Partition
set.seed(100)
Train_Index = createDataPartition(UniversalBank$Personal.Loan,p=0.6,list=FALSE) # 60% reserved for Trai
Train.df=UnivBank_norm[Train_Index,]
Validation.df=UnivBank_norm[-Train_Index,]
```

```
#Task1
To_Predict=data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Education=0,Mortgage=0,Securities

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
```

#Given the conditions mentioned the customer will not be taking a loan and hence it is classified as 0.

```
#Task2
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  0.9506667  0.6886329
##    2  0.9435000  0.6427522
##    3  0.9515000  0.6692239
##    4  0.9473333  0.6325288
##    5  0.9483333  0.6291648
##    6  0.9460000  0.6091451
##    7  0.9445000  0.5858059
##    8  0.9445000  0.5854667
##    9  0.9418333  0.5595318
##   10  0.9398333  0.5387812
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

#The best choice of k is k=3

```
#Task3
predictions<-predict(Knn.model,Validation.df)

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1797   70
##          1   11  122
##
##                Accuracy : 0.9595
##                  95% CI : (0.9499, 0.9677)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7295
##
##  Mcnemar's Test P-Value : 1.16e-10
##
##             Sensitivity : 0.9939
##             Specificity : 0.6354
##          Pos Pred Value : 0.9625
##          Neg Pred Value : 0.9173
##              Prevalence : 0.9040
##          Detection Rate : 0.8985
##    Detection Prevalence : 0.9335
##       Balanced Accuracy : 0.8147
##
##        'Positive' Class : 0
##
```

```
#Task4
To_Predict=data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Education=1,Mortgage=0,Securities

To_Predict_norm=predict(Norm_model,To_Predict)

predict(Knn.model,To_Predict_norm)
```

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

#Here we considered Education = 1

```
#Task5
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

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

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

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

confusionMatrix(Testknn, test.data[,8])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 874  52
##          1  29  45
##
##                Accuracy : 0.919
##                  95% CI : (0.9003, 0.9352)
##     No Information Rate : 0.903
##     P-Value [Acc > NIR] : 0.04613
##
##                   Kappa : 0.4829
##
##  Mcnemar's Test P-Value : 0.01451
##
##             Sensitivity : 0.9679
##             Specificity : 0.4639
##          Pos Pred Value : 0.9438
##          Neg Pred Value : 0.6081
##              Prevalence : 0.9030
##          Detection Rate : 0.8740
##    Detection Prevalence : 0.9260
##       Balanced Accuracy : 0.7159
##
##        'Positive' Class : 0
##
```

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

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

```
## Prediction    0    1
##          0 2211   89
##          1   39  161
##
##                Accuracy : 0.9488
##                  95% CI : (0.9394, 0.9571)
##     No Information Rate : 0.9
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6878
##
##  Mcnemar's Test P-Value : 1.484e-05
##
##             Sensitivity : 0.9827
##             Specificity : 0.6440
##          Pos Pred Value : 0.9613
##          Neg Pred Value : 0.8050
##              Prevalence : 0.9000
##          Detection Rate : 0.8844
##    Detection Prevalence : 0.9200
##       Balanced Accuracy : 0.8133
##
##        'Positive' Class : 0
##
```

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1327   92
##          1   40   41
##
##                Accuracy : 0.912
##                  95% CI : (0.8965, 0.9259)
##     No Information Rate : 0.9113
##     P-Value [Acc > NIR] : 0.4869
##
##                   Kappa : 0.3388
##
##  Mcnemar's Test P-Value : 9.039e-06
##
##             Sensitivity : 0.9707
##             Specificity : 0.3083
##          Pos Pred Value : 0.9352
##          Neg Pred Value : 0.5062
##              Prevalence : 0.9113
##          Detection Rate : 0.8847
##    Detection Prevalence : 0.9460
##       Balanced Accuracy : 0.6395
##
##        'Positive' Class : 0
##
```

#Comments: #Accuracy = TP+TN/TP+TN+FP+FN #It can be seen that the accuracy for the testing,training and validation is approximately different.The differences in the accuracy is due the confusion matrix and the confusion matrix clearly shows the reason behind it, the classification has been pretty decent considering how relatively large the number of true positives is.