#title: “Assinment2FML” #author: “Mpurumandla” #date: “20/02/2022” #output: pdf_document

#Importing data and setting data as working directory

```
UniversalBank <- read.csv("universalbank.csv")

colnames<-c('ID','Age','Experience','Income','zIP.Code','Family','CCAvg','Education','Mortgage','Persona

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 some of attributes we do not use in our model and set them to NULL

```
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(class)
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(caret)
```

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

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

```
library(ggplot2)

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  
```

converting categorical variables(“Education”,“Personal.Loan”) to factors

```
UniversalBank$Personal.Loan=as.factor(UniversalBank$Personal.Loan)
UniversalBank$Income=as.factor(UniversalBank$Income)
Bank_norm<-UniversalBank
```

Normalize the data,removing target attribute before normalization

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

```
##       Age             Experience            Income          Family       
##  Min.   :-1.94871   Min.   :-2.014710   44     :  85   Min.   :-1.2167  
##  1st Qu.:-0.90188   1st Qu.:-0.881116   38     :  84   1st Qu.:-1.2167  
##  Median :-0.02952   Median :-0.009121   81     :  83   Median :-0.3454  
##  Mean   : 0.00000   Mean   : 0.000000   41     :  82   Mean   : 0.0000  
##  3rd Qu.: 0.84284   3rd Qu.: 0.862874   39     :  81   3rd Qu.: 0.5259  
##  Max.   : 1.88967   Max.   : 1.996468   40     :  78   Max.   : 1.3973  
##                                         (Other):4507                     
##      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
##
```

```
Bank_norm$personal.Loan=UniversalBank$Personal.Loan
```

#Dividing the data into train and validation.

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

#1 -> Modelling k-NN with K=1 and sample data

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

print(To_Predict)
```

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

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

```
##          Age Experience Income     Family     CCAvg   Mortgage
## 1 -0.4657003 -0.8811162     84 -0.3453975 0.0355115 -0.5554684
##   Securities.Account CD.Account    Online CreditCard Education
## 1         -0.3413892 -0.2535149 0.8218687   1.549477 -1.048973
```

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

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

#2- Finding the best value of K to avoid over fitting

```
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
##   12 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  0.9656667  0.7765121
##    2  0.9581667  0.7250803
##    3  0.9613333  0.7354740
##    4  0.9586667  0.7135486
##    5  0.9538333  0.6738772
##    6  0.9530000  0.6643577
##    7  0.9513333  0.6474319
##    8  0.9466667  0.6008159
##    9  0.9451667  0.5878615
##   10  0.9436667  0.5702564
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

#3 - Show the 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 1802   56
##          1    6  136
##
##                Accuracy : 0.969
##                  95% CI : (0.9604, 0.9762)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar's Test P-Value : 4.877e-10
```

```
##
##             Sensitivity : 0.9967
##             Specificity : 0.7083
##          Pos Pred Value : 0.9699
##          Neg Pred Value : 0.9577
##              Prevalence : 0.9040
##          Detection Rate : 0.9010
##    Detection Prevalence : 0.9290
##       Balanced Accuracy : 0.8525
##
##        'Positive' Class : 0
##
```

#4

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


print(To_Predict)
```

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

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

```
##          Age Experience Income     Family     CCAvg   Mortgage
## 1 -0.4657003 -0.8811162     84 -0.3453975 0.0355115 -0.5554684
##   Securities.Account CD.Account    Online CreditCard Education
## 1         -0.3413892 -0.2535149 0.8218687   1.549477 -1.048973
```

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

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

#5

```
splitSample <- sample(1:3, size=nrow(Bank_norm), prob=c(0.5,0.3,0.2), replace = TRUE)
train_Data <- Bank_norm[splitSample==1,]
valid_Data <- Bank_norm[splitSample==2,]
test_Data <- Bank_norm[splitSample==3,]

Predict=data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education= 1,Mortgage

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(Norm_model,Predict)

print(Predict_norm)
```

```
##          Age Experience Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162     84 -0.3453975 0.0355115 -1.048973 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

```
Prediction_newsplit <-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_newsplit)
```

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

```
fitControl2 <- trainControl(method = "repeatedcv",
                            number = 3,
                            repeats = 2)
searchGrid=expand.grid(k = 1:10)

Knn.model2 =train(Personal.Loan~.,
                  data=Train.df,
                  method='knn',
                  tuneGrid=searchGrid,
                  trControl = fitControl2,)
Knn.model2
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   12 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  0.9620000  0.7520622
##    2  0.9551667  0.6985799
##    3  0.9601667  0.7279976
##    4  0.9525000  0.6638537
##    5  0.9523333  0.6589676
##    6  0.9520000  0.6558529
##    7  0.9481667  0.6204255
##    8  0.9445000  0.5890596
##    9  0.9423333  0.5668552
##   10  0.9421667  0.5667225
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
predictions2<-predict(Knn.model2,Validation.df)
confusionMatrix(predictions2,Validation.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1802   56
##          1    6  136
##
##                Accuracy : 0.969
##                  95% CI : (0.9604, 0.9762)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar’s Test P-Value : 4.877e-10
##
##             Sensitivity : 0.9967
##             Specificity : 0.7083
##          Pos Pred Value : 0.9699
##          Neg Pred Value : 0.9577
##              Prevalence : 0.9040
##          Detection Rate : 0.9010
##    Detection Prevalence : 0.9290
##       Balanced Accuracy : 0.8525
##
##        ’Positive’ Class : 0
##
```