Machine Learning Assignment 2

Contents

---

k-NN

Install packages if necessary. Uncomment before running.

```
#install.packages("caret")
library(caret)

# install.packages("ISLR") # only install if needed
library(ISLR)

library(FNN)

#install.packages("dummies")
library(dummies)
library(gmodels)
```

Loading the data file and reviewing data structures

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

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```
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
```

Excluding ID and Zip Code and selecting the data set with the mentioned criteria

```
UniversalBank2<-UniversalBank[,c(-1,-5)]
UniversalBank1<-UniversalBank2[which(UniversalBank2$Age ==40 | UniversalBank2$Experience==10 | Universal
str(UniversalBank1)
```

```
## 'data.frame':    4999 obs. of  12 variables:
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

Creating Dummy Variables

```
levels(UniversalBank1$Education)
```

```
## NULL
```

```
dummy_model <- dummyVars(~Education,data=UniversalBank1)
head(predict(dummy_model,UniversalBank1))
```

```
##   Education
## 1         1
## 2         1
```

```
## 3         1
## 4         2
## 5         2
## 6         2
```

Creating New Data Frame with the Data variables and doing the Data Normalization

```
UBank<-dummy.data.frame(UniversalBank1, names = c("Education"), sep=".")
str(UBank)
```

```
## 'data.frame':    4999 obs. of  14 variables:
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education.1       : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ Education.2       : int  0 0 0 1 1 1 1 0 1 0 ...
##  $ Education.3       : int  0 0 0 0 0 0 0 1 0 1 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
##  - attr(*, "dummies")=List of 1
##   ..$ Education: int [1:3] 6 7 8
```

```
norm_model<-preProcess(UBank, method = c('range'))
UBank_normalized<-predict(norm_model,UBank)
UBank_Predictors<-UBank_normalized[,-10]
UBank_labels<-UBank_normalized[,10]
```

Doing the data partition of 60% Training and 40% Validation

```
set.seed(15)
inTrain = createDataPartition(UBank_normalized$Personal.Loan,p=0.6, list=FALSE)
Train_Data = UBank_normalized[inTrain,]
Val_Data = UBank_normalized[-inTrain,]
dim(Train_Data)
```

```
## [1] 3000   14
```

```
summary(Train_Data)
```

```
##       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   :0.5098   Mean   :0.5043   Mean   :0.3064   Mean   :0.4591
##  3rd Qu.:0.7273   3rd Qu.:0.7174   3rd Qu.:0.4213   3rd Qu.:0.6667
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.9722   Max.   :1.0000
##      CCAvg         Education.1      Education.2      Education.3
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0700   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.1500   Median :0.0000   Median :0.0000   Median :0.000
##  Mean   :0.1923   Mean   :0.4233   Mean   :0.2707   Mean   :0.306
##  3rd Qu.:0.2600   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000
```

```
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
##     Mortgage       Personal.Loan   Securities.Account   CD.Account
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000     Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.00000
##  Median :0.00000   Median :0.000   Median :0.0000     Median :0.00000
##  Mean   :0.08978   Mean   :0.102   Mean   :0.1053     Mean   :0.06167
##  3rd Qu.:0.16063   3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.00000
##  Max.   :0.97165   Max.   :1.000   Max.   :1.0000     Max.   :1.00000
##      Online         CreditCard
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000
##  Mean   :0.6027   Mean   :0.2877
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
summary(Val_Data)
```

```
##       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 :0.2546   Median :0.3333
##  Mean   :0.5044   Mean   :0.4991   Mean   :0.3014   Mean   :0.4749
##  3rd Qu.:0.7273   3rd Qu.:0.7174   3rd Qu.:0.4097   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##      CCAvg          Education.1      Education.2      Education.3
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0700   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.1600   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1958   Mean   :0.4132   Mean   :0.2951   Mean   :0.2916
##  3rd Qu.:0.2500   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     Mortgage       Personal.Loan     Securities.Account   CD.Account
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000     Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.0000     Median :0.00000
##  Mean   :0.08743   Mean   :0.08654   Mean   :0.1026     Mean   :0.05803
##  3rd Qu.:0.15591   3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000     Max.   :1.00000
##      Online         CreditCard
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000
##  Mean   :0.5883   Mean   :0.3037
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
Train_Predictors<-Train_Data[,-10]
Val_Predictors<-Val_Data[,-10]
Train_labels <-Train_Data[,10]
Val_labels <-Val_Data[,10]
Train_labels=as.factor(Train_labels)
Val_labels=as.factor(Val_labels)
UBank_labels<-as.factor(UBank_labels)
```

```
knn.pred <- knn(Train_Predictors,Val_Predictors,cl=Train_labels,k=1,prob = TRUE)
Q1 <- data.frame(40, 10, 84, 2, 2, 0, 1, 0, 0, 0, 0, 1, 1)
knn.pred1 <- knn(Train_Predictors, Q1, cl=Train_labels, k=1, prob = 0.5)
knn.pred1
```

```
## [1] 1
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1]
## [1,] 1802
## attr(,"nn.dist")
##         [,1]
## [1,] 92.3635
## Levels: 1
```

```
library(caret)
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
for(i in 1:14) {
knn <- knn(Train_Predictors, Val_Predictors, cl = Train_labels, k = i)
accuracy.df[i, 2] <- confusionMatrix(knn, Val_labels)$overall[1]
}
accuracy.df
```

```
##     k  accuracy
## 1   1 0.9629815
## 2   2 0.9534767
## 3   3 0.9624812
## 4   4 0.9504752
## 5   5 0.9559780
## 6   6 0.9479740
## 7   7 0.9499750
## 8   8 0.9439720
## 9   9 0.9464732
## 10 10 0.9434717
## 11 11 0.9449725
## 12 12 0.9424712
## 13 13 0.9419710
## 14 14 0.9399700
```

```
which.max( (accuracy.df$accuracy) )
```

```
## [1] 1
```

Optimal value is of K=1

#Confision Matrix

```
knn.pred3 <- knn(Train_Predictors,Val_Predictors,cl=Train_labels,k=3,prob = TRUE)
confusionMatrix(knn.pred3,Val_labels)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1815   64
##          1   11  109
```

```
##
##                Accuracy : 0.9625
##                  95% CI : (0.9532, 0.9704)
##     No Information Rate : 0.9135
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7245
##
##  Mcnemar's Test P-Value : 1.92e-09
##
##             Sensitivity : 0.9940
##             Specificity : 0.6301
##          Pos Pred Value : 0.9659
##          Neg Pred Value : 0.9083
##              Prevalence : 0.9135
##          Detection Rate : 0.9080
##    Detection Prevalence : 0.9400
##       Balanced Accuracy : 0.8120
##
##        'Positive' Class : 0
##
```

Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%). Apply the k-NN method with the k chosen above.

```
set.seed(15)
Bank_Partition = createDataPartition(UBank_normalized$Personal,p=0.5, list=FALSE)
TrainingData = UBank_normalized[Bank_Partition,]
TestValidData = UBank_normalized[-Bank_Partition,]
Test_Index = createDataPartition(TestValidData$Personal.Loan, p=0.6, list=FALSE)
ValidationData = TestValidData[Test_Index,]
Test_Data = TestValidData[-Test_Index,]
```

```
Training_Predictors<-TrainingData[,-10]
Test_Predictors<-Test_Data[,-10]
Validation_Predictors<-ValidationData[,-10]
Training_labels <-TrainingData[,10]
Test_labels <-Test_Data[,10]
Validation_labels <-ValidationData[,10]
Training_labels=as.factor(Training_labels)
Test_labels<-as.factor(Test_labels)
Validation_labels=as.factor(Validation_labels)
```

Confusion Matrix on Training data

```
knn.pred5 <- knn(Training_Predictors, Test_Predictors , cl=Training_labels, k=1, prob = TRUE)
confusionMatrix(knn.pred5,Test_labels)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 910  30
##          1   7  52
##
##                Accuracy : 0.963
```

```
##                  95% CI : (0.9493, 0.9738)
##     No Information Rate : 0.9179
##     P-Value [Acc > NIR] : 6.824e-09
##
##                   Kappa : 0.7182
##
##  Mcnemar's Test P-Value : 0.0002983
##
##             Sensitivity : 0.9924
##             Specificity : 0.6341
##          Pos Pred Value : 0.9681
##          Neg Pred Value : 0.8814
##              Prevalence : 0.9179
##          Detection Rate : 0.9109
##    Detection Prevalence : 0.9409
##       Balanced Accuracy : 0.8133
##
##        'Positive' Class : 0
##
```

Confusion Matrix on Validation data

```
knn.pred6 <- knn(Validation_Predictors, Test_Predictors, cl=Validation_labels, k=1, prob = TRUE)
confusionMatrix(knn.pred6,Test_labels)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 898  32
##          1  19  50
##
##                Accuracy : 0.9489
##                  95% CI : (0.9334, 0.9618)
##     No Information Rate : 0.9179
##     P-Value [Acc > NIR] : 9.323e-05
##
##                   Kappa : 0.6349
##
##  Mcnemar's Test P-Value : 0.09289
##
##             Sensitivity : 0.9793
##             Specificity : 0.6098
##          Pos Pred Value : 0.9656
##          Neg Pred Value : 0.7246
##              Prevalence : 0.9179
##          Detection Rate : 0.8989
##    Detection Prevalence : 0.9309
##       Balanced Accuracy : 0.7945
##
##        'Positive' Class : 0
##
```