DeSimone_Assignment 2

Heather DeSimone

2/19/2022

##First I have loaded in my data frame and called a summary of the information.

```
DF=read.csv("C:/Users/hdesi/Desktop/MBA/Machine Learning/UniversalBank.csv")
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

DF <- DF %>% relocate(Personal.Loan, .after = CreditCard)
summary(DF)

##        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
##  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
##  Personal.Loan
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.096
##  3rd Qu.:0.000
##  Max.   :1.000
```

##Next I will remove the 2 variables that will not be used in my classification/prediction: ID and Zip Code. ##I have also converted a few attributes over to factors - these attributes classify a yes (1) or no (0) response.I have called a summary to check my work.

```
DF$ID<-NULL
DF$ZIP.Code<-NULL
DF$Personal.Loan=as.factor(DF$Personal.Loan)
DF$Securities.Account=as.factor(DF$Securities.Account)
DF$CD.Account=as.factor(DF$CD.Account)
DF$Online=as.factor(DF$Online)
DF$CreditCard=as.factor(DF$CreditCard)
summary(DF)
```

```
##       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     Securities.Account
CD.Account
##  Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4478
0:4698
##  1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 522             1:
302
##  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
##  Online   CreditCard Personal.Loan
##  0:2016   0:3530     0:4520
##  1:2984   1:1470     1: 480
##
##
##
##
```

##I will now load the caret and class libraries.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

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

```
library(class)
```

##Next I have created a new data set for the normalization process - I have removed the target variable: Personal Loan as we cannot normalize it. ##I have also removed the attributes that were factored

```
Normalization_DF <- data.frame(DF)
Normalization_DF$Personal.Loan<-NULL
Normalization_DF$Securities.Account<-NULL
Normalization_DF$CD.Account<-NULL
Normalization_DF$Online<-NULL
Normalization_DF$CreditCard<-NULL
summary(Normalization_DF)
```

```
##       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
##  Min.   : 0.000   Min.   :1.000   Min.   :  0.0
##  1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0
##  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
```

##I will now normalize the data.

```
Norm_model <- preProcess(Normalization_DF,
                         method = c("center", "scale"))
loan_norm=predict(Norm_model,Normalization_DF)
summary(loan_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
##  Min.   :-1.1089   Min.   :-1.0490   Min.   :-0.5555
##  1st Qu.:-0.7083   1st Qu.:-1.0490   1st Qu.:-0.5555
##  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
```

##I will now add the attributes back in that I removed for normalization.

```
loan_norm$Personal.Loan=DF$Personal.Loan
loan_norm$Securities.Account=DF$Securities.Account
loan_norm$CD.Account=DF$CD.Account
loan_norm$Online=DF$Online
loan_norm$CreditCard=DF$CreditCard
summary(loan_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
##  0:4478             0:4698     0:2016   0:3530
##  1: 522             1: 302     1:2984   1:1470
##
##
##
##
```

##I will now separate my data into training and validating sets - training = 60% and validation = 40%.

```
Train_Index = createDataPartition(DF$Personal.Loan,p=0.6, list=FALSE)
Train.df=loan_norm[Train_Index,]
Validation.df=loan_norm[-Train_Index,]
```

##Question #1 ##I will now input the attributes of the 1st customer for prediction.

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

##I will remove the attributes that were factored.

```
To_Predict_norm=To_Predict
To_Predict_norm$Personal.Loan<-NULL
To_Predict_norm$Securities.Account<-NULL
To_Predict_norm$CD.Account<-NULL
To_Predict_norm$Online<-NULL
To_Predict_norm$CreditCard<-NULL
```

##I will now normalize the data.

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

##I will now add the attributes back in that I removed for normalization.

```
To_Predict_norm$Personal.Loan<-To_Predict$Personal.Loan
To_Predict_norm$Securities.Account<-To_Predict$Securities.Account
To_Predict_norm$CD.Account<-To_Predict$CD.Account
To_Predict_norm$Online<-To_Predict$Online
To_Predict_norm$CreditCard<-To_Predict$CreditCard
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education
Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 0.1416887 -
0.5554684
##   Securities.Account CD.Account Online CreditCard
## 1                  0          0      1          1
```

##I will now use the knn function to make my prediction.

```
Train.df <- Train.df %>% relocate(Personal.Loan, .after = CreditCard)
Prediction <-knn(train=Train.df[1:11],
                 test=To_Predict_norm[1:11],
                 cl=Train.df$Personal.Loan,
```

```
                  k=1)
print(Prediction)
```

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

##This customer is predicted NOT to accept the personal loan

##Question #2 ##I will now build the knn model that will give the best value of k that balances between overfitting and underfitting.

```
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  0.9625000  0.7603680
##    2  0.9561667  0.7152440
##    3  0.9613333  0.7351054
##    4  0.9588333  0.7141330
##    5  0.9583333  0.7049693
##    6  0.9580000  0.7029532
##    7  0.9555000  0.6796047
##    8  0.9556667  0.6812888
##    9  0.9560000  0.6830743
##   10  0.9533333  0.6609398
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

##The best k value to use is 3. ##Question #3 ##First I will use the predict function of the caret package.

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

#Now I will compute the confusion matrix using the caret package.

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1787   60
##          1   21  132
##
##                Accuracy : 0.9595
##                  95% CI : (0.9499, 0.9677)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7434
##
##  Mcnemar's Test P-Value : 2.419e-05
##
##             Sensitivity : 0.9884
##             Specificity : 0.6875
##          Pos Pred Value : 0.9675
##          Neg Pred Value : 0.8627
##              Prevalence : 0.9040
##          Detection Rate : 0.8935
##    Detection Prevalence : 0.9235
##       Balanced Accuracy : 0.8379
##
##        'Positive' Class : 0
##
```

##Question #4 ##I will now input the attributes of the 2nd customer for prediction. *Note that this customer information is the same as the 1st customer.

```
To_Predict2=data.frame(Age=40, Experience=10,
                       Income=84,Family=2,
                       CCAvg=2,Education=2,
                       Mortgage=0,
                       Securities.Account=0,
                       CD.Account=0,
                       Online=1,
                       CreditCard=1)
```

```
print(To_Predict2)
```

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

##I will remove the attributes that were factored.

```
To_Predict_norm2=To_Predict2
To_Predict_norm2$Personal.Loan<-NULL
To_Predict_norm2$Securities.Account<-NULL
To_Predict_norm2$CD.Account<-NULL
To_Predict_norm2$Online<-NULL
To_Predict_norm2$CreditCard<-NULL
```

##I will now normalize the data.

```
To_Predict_norm2=predict(Norm_model,To_Predict_norm2)
```

##I will now add the attributes back in that I removed for normalization.

```
To_Predict_norm2$Personal.Loan<-To_Predict2$Personal.Loan
To_Predict_norm2$Securities.Account<-To_Predict2$Securities.Account
To_Predict_norm2$CD.Account<-To_Predict2$CD.Account
To_Predict_norm2$Online<-To_Predict2$Online
To_Predict_norm2$CreditCard<-To_Predict2$CreditCard
print(To_Predict_norm2)
```

```
##          Age Experience    Income     Family     CCAvg Education
Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 0.1416887 -
0.5554684
##   Securities.Account CD.Account Online CreditCard
## 1                  0          0      1          1
```

##I will now use the knn function to make my prediction.I am using k=3 as it is the best k value.

```
Train.df <- Train.df %>% relocate(Personal.Loan, .after = CreditCard)
Prediction <-knn(train=Train.df[1:11],
                 test=To_Predict_norm2[1:11],
                 cl=Train.df$Personal.Loan,
                 k=3)
print(Prediction)
```

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

##This customer is predicted NOT to take out the personal loan ##Question 5 ##I will now repartition my data into training (50%), Validation (30%), and test (20%).

```
Train_Index2 = createDataPartition(DF$Personal.Loan,p=0.5, list=FALSE)
Train.df2=loan_norm[Train_Index2,]
Validation.df2=loan_norm[-Train_Index2,]
```

##I will now input the attributes of the 1st customer for prediction.

```
To_Predict3=data.frame(Age=40, Experience=10,
                       Income=84,Family=2,
                       CCAvg=2,Education=2,
                       Mortgage=0,
                       Securities.Account=0,
                       CD.Account=0,
                       Online=1,
                       CreditCard=1)

print(To_Predict3)
```

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

##I will remove the attributes that were factored.

```
To_Predict_norm3=To_Predict3
To_Predict_norm3$Personal.Loan<-NULL
To_Predict_norm3$Securities.Account<-NULL
To_Predict_norm3$CD.Account<-NULL
To_Predict_norm3$Online<-NULL
To_Predict_norm3$CreditCard<-NULL
```

##I will now normalize the data.

```
To_Predict_norm3=predict(Norm_model,To_Predict_norm3)
```

##I will now add the attributes back in that I removed for normalization.

```
To_Predict_norm3$Personal.Loan<-To_Predict3$Personal.Loan
To_Predict_norm3$Securities.Account<-To_Predict3$Securities.Account
To_Predict_norm3$CD.Account<-To_Predict3$CD.Account
To_Predict_norm3$Online<-To_Predict3$Online
To_Predict_norm3$CreditCard<-To_Predict3$CreditCard
print(To_Predict_norm3)
```

```
##          Age Experience    Income     Family     CCAvg Education
Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 0.1416887 -
0.5554684
##   Securities.Account CD.Account Online CreditCard
## 1                  0          0      1          1
```

##I will now use the knn function to make my prediction.

```
Train.df2 <- Train.df2 %>% relocate(Personal.Loan, .after = CreditCard)
Prediction <-knn(train=Train.df2[1:11],
                 test=To_Predict_norm2[1:11],
                 cl=Train.df2$Personal.Loan,
                 k=3)
print(Prediction)
```

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

##Now I will create my confusion matrix ##First I will use the predict function of the caret package.

```
predictions2<-predict(Knn.model,Validation.df2)
```

#Now I will compute the confusion matrix using the caret package.

```
confusionMatrix(predictions2,Validation.df2$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2247   36
##          1   13  204
##
##                Accuracy : 0.9804
##                  95% CI : (0.9742, 0.9855)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.882
##
##  Mcnemar's Test P-Value : 0.001673
##
##             Sensitivity : 0.9942
##             Specificity : 0.8500
##          Pos Pred Value : 0.9842
##          Neg Pred Value : 0.9401
##              Prevalence : 0.9040
##          Detection Rate : 0.8988
##    Detection Prevalence : 0.9132
##       Balanced Accuracy : 0.9221
##
##        'Positive' Class : 0
##
```

##The prediction stands that the customer will not take out the personal loan. Accuracy from the prior matrix to this one has increased.