Assignment1- KNN for classification

AKHILA KALPURI

2023-02-17

#The aim is to forecast if a new customer will accept a loan offer using k-NN. This will be the starting point for creating a fresh campaign.

```
#installing the packages
#install.packages("gmodels")
```

##loading required library

```
rm(list = ls()) #cleaning the environment
library(readr)
library(caret)
```

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

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

```
library(knitr)
library(class)
library(ggplot2)
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
```

##Import Data "UniversalBank.csv"

```
library(readr)
Bankdata1 <- read.csv("C://Users//vishe//OneDrive//Desktop//FML//Assignment2//UniversalBank (1).csv")
head(Bankdata1)
```

```
##   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
```

##Understand the bank data structure

```
str(Bankdata1)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ 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 ...
```

```
summary(Bankdata1)
```

```
##        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
```

##Cleaning and Preparing the data set ###(1)Remove Zipcode ###(2)Converting Personal_loan to factor because the customer response to the last personal loan campaign is "Personal_Loan" variable and want to covert into category ###(3)creating the dummy variables for Education and converting them to factor

```
Bankdata2 <-Bankdata1[,-c(1,5)]
Bankdata2$Personal.Loan <- as.factor(Bankdata2$Personal.Loan)
class(Bankdata2$Personal_Loan)
```

```
## [1] "NULL"
```

```
Education1 <- ifelse(Bankdata2$Education == 1, 1,0)
Education1 <- as.factor(Education1)
Education2 <- ifelse(Bankdata2$Education == 2, 1,0)
Education2 <- as.factor(Education2)
Education3 <- ifelse(Bankdata2$Education == 3, 1,0)
Education3 <- as.factor(Education3)
Bankdata3 <- data.frame(Bankdata2,Education1 = Education1,Education2 = Education2, Education3 = Educati
Bankdata4 <- Bankdata3[,-6]
```

##Dividing the data into sets for training (60%) and validation (40%), respectively Furthermore displayed the summary statistics for the test and train data sets.

```
Train_Index = createDataPartition(Bankdata4$Personal.Loan,p=0.6, list = FALSE)
Train_df <-Bankdata4[Train_Index,]
Validation_df<-Bankdata4[-Train_Index,]
nrow(Train_df)
```

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

```
summary(Train_df)
```

```
##       Age          Experience        Income          Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :46.00   Median :20.00   Median : 63.00   Median :2.000
##  Mean   :45.45   Mean   :20.21   Mean   : 73.24   Mean   :2.382
##  3rd Qu.:56.00   3rd Qu.:30.00   3rd Qu.: 95.75   3rd Qu.:3.000
```

```
##  Max.   :67.00   Max.   :43.00   Max.   :218.00   Max.   :4.000
##      CCAvg          Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.00   Min.   :  0.00   0:2712        Min.   :0.0000
##  1st Qu.: 0.70   1st Qu.:  0.00   1: 288        1st Qu.:0.0000
##  Median : 1.50   Median :  0.00                 Median :0.0000
##  Mean   : 1.93   Mean   : 56.36                 Mean   :0.1083
##  3rd Qu.: 2.50   3rd Qu.:100.00                 3rd Qu.:0.0000
##  Max.   :10.00   Max.   :635.00                 Max.   :1.0000
##    CD.Account       Online         CreditCard     Education1 Education2
##  Min.   :0.00   Min.   :0.0000   Min.   :0.0000   0:1758     0:2127
##  1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.0000   1:1242     1: 873
##  Median :0.00   Median :1.0000   Median :0.0000
##  Mean   :0.06   Mean   :0.5877   Mean   :0.2883
##  3rd Qu.:0.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00   Max.   :1.0000   Max.   :1.0000
##  Education3
##  0:2115
##  1: 885
##
##
##
##
```

```
nrow(Validation_df)
```

```
## [1] 2000
```

```
summary(Validation_df)
```

```
##       Age          Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:36.00   1st Qu.:11.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :45.16   Mean   :19.94   Mean   : 74.57   Mean   :2.418
##  3rd Qu.:55.00   3rd Qu.:29.00   3rd Qu.:101.00   3rd Qu.:4.000
##  Max.   :67.00   Max.   :42.00   Max.   :224.00   Max.   :4.000
##      CCAvg          Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.00   Min.   :  0.00   0:1808        Min.   :0.0000
##  1st Qu.: 0.70   1st Qu.:  0.00   1: 192        1st Qu.:0.0000
##  Median : 1.60   Median :  0.00                 Median :0.0000
##  Mean   : 1.95   Mean   : 56.71                 Mean   :0.0985
##  3rd Qu.: 2.60   3rd Qu.:103.25                 3rd Qu.:0.0000
##  Max.   :10.00   Max.   :612.00                 Max.   :1.0000
##    CD.Account        Online         CreditCard     Education1 Education2
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   0:1146     0:1470
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1: 854     1: 530
##  Median :0.000   Median :1.0000   Median :0.0000
##  Mean   :0.061   Mean   :0.6105   Mean   :0.3025
##  3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##  Education3
##  0:1384
##  1: 616
```

```
##
##
##
##
```

##normalization of the data.

```
Norm_model <- preProcess(Train_df, method = c("center", "scale"))
training_norm<-predict(Norm_model,Train_df)
head(training_norm)
```

```
##           Age Experience     Income     Family      CCAvg   Mortgage
## 4  -0.8993322 -0.9629404  0.5794783 -1.2134695  0.4368124 -0.5531201
## 7   0.6491640  0.5828941 -0.0269273 -0.3352045 -0.2437781 -0.5531201
## 9  -0.8993322 -0.8770607  0.1679888  0.5430605 -0.7542209  0.4675959
## 11  1.6814948  1.6134504  0.6877650  1.4213255  0.2666648 -0.5531201
## 14  1.1653294  1.0122926 -0.7199623  1.4213255  0.3233807 -0.5531201
## 16  1.2513569  0.8405332 -1.1097944 -1.2134695 -0.2437781 -0.5531201
##    Personal.Loan Securities.Account CD.Account     Online CreditCard Education1
## 4              0         -0.3485037 -0.2526035 -1.1936278 -0.6364096          0
## 7              0         -0.3485037 -0.2526035  0.8375029 -0.6364096          0
## 9              0         -0.3485037 -0.2526035  0.8375029 -0.6364096          0
## 11             0         -0.3485037 -0.2526035 -1.1936278 -0.6364096          0
## 14             0         -0.3485037 -0.2526035  0.8375029 -0.6364096          0
## 16             0         -0.3485037 -0.2526035  0.8375029  1.5707913          0
##    Education2 Education3
## 4           1          0
## 7           1          0
## 9           1          0
## 11          0          1
## 14          1          0
## 16          0          1
```

```
validation_norm<-predict(Norm_model,Validation_df)
head(validation_norm)
```

```
##           Age Experience     Income     Family      CCAvg   Mortgage
## 1 -1.75960779 -1.6499779 -0.5250462  1.4213255 -0.1870622 -0.5531201
## 2 -0.03905651 -0.1041434 -0.8499063  0.5430605 -0.2437781 -0.5531201
## 3 -0.55522190 -0.4476622 -1.3480252 -1.2134695 -0.5273574 -0.5531201
## 5 -0.89933215 -1.0488201 -0.6116755  1.4213255 -0.5273574 -0.5531201
## 6 -0.72727702 -0.6194216 -0.9581930  1.4213255 -0.8676527  0.9681393
## 8  0.39108131  0.3252550 -1.1097944 -1.2134695 -0.9243685 -0.5531201
##   Personal.Loan Securities.Account CD.Account     Online CreditCard Education1
## 1             0          2.8684535 -0.2526035 -1.1936278 -0.6364096          1
## 2             0          2.8684535 -0.2526035 -1.1936278 -0.6364096          1
## 3             0         -0.3485037 -0.2526035 -1.1936278 -0.6364096          1
## 5             0         -0.3485037 -0.2526035 -1.1936278  1.5707913          0
## 6             0         -0.3485037 -0.2526035  0.8375029 -0.6364096          0
## 8             0         -0.3485037 -0.2526035 -1.1936278  1.5707913          0
##   Education2 Education3
## 1          0          0
```

```
## 2          0          0
## 3          0          0
## 5          1          0
## 6          1          0
## 8          0          1
```

#creating the test data set and test normalization

```
Test <-data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Mortgage=0,Securities.Account=0,CD.Ac
head(Test)
```

```
##   Age Experience Income Family CCAvg Mortgage Securities.Account CD.Account
## 1  40         10     84      2     2        0                   0          0
##   Online CreditCard Education1 Education2 Education3
## 1      1          1          0          1          0
```

```
test_norm<-predict(Norm_model,Test)
head(test_norm)
```

```
##          Age Experience    Income     Family      CCAvg   Mortgage
## 1 -0.4691943 -0.8770607 0.2329608 -0.3352045 0.03980131 -0.5531201
##   Securities.Account CD.Account    Online CreditCard Education1 Education2
## 1         -0.3485037 -0.2526035 0.8375029   1.570791          0          1
##   Education3
## 1          0
```

#knn algorithm in dataset

```
Train_predictors<-training_norm[,-7]
Train_label<-training_norm[,7]
valid_predictors<-validation_norm[,-7]
Valid_label<-validation_norm[,7]
Predict_test_label<-knn(Train_predictors,test_norm,cl=Train_label,k=1)
Predict_test_label
```

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

```
#Customer will not accept the offer because the value of K = 0
```

#Finding the best value for k by training the model by using train function. Also customizing the grid search

```
set.seed(550)
searchGrid <- expand.grid(k=seq(1:30))
model <- train(Personal.Loan~.,training_norm,method="knn", tuneGrid = searchGrid)
model
```

```
## k-Nearest Neighbors
##
## 3000 samples
```

```
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9504874  0.6954065
##    2  0.9454944  0.6605210
##    3  0.9448470  0.6472661
##    4  0.9440750  0.6325693
##    5  0.9449491  0.6331662
##    6  0.9453525  0.6313597
##    7  0.9449559  0.6255681
##    8  0.9434621  0.6090318
##    9  0.9430240  0.6032396
##   10  0.9420313  0.5915598
##   11  0.9415639  0.5845514
##   12  0.9407055  0.5761703
##   13  0.9401226  0.5690285
##   14  0.9387584  0.5547316
##   15  0.9390852  0.5560834
##   16  0.9384310  0.5481226
##   17  0.9382155  0.5446653
##   18  0.9373548  0.5366975
##   19  0.9372464  0.5347354
##   20  0.9371403  0.5334305
##   21  0.9363024  0.5236409
##   22  0.9352219  0.5116686
##   23  0.9354004  0.5120097
##   24  0.9342828  0.5005311
##   25  0.9336324  0.4944041
##   26  0.9339128  0.4971997
##   27  0.9333734  0.4911904
##   28  0.9321783  0.4793494
##   29  0.9320722  0.4777692
##   30  0.9318220  0.4751550
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
best_k <- model$bestTune[[1]]
#K = 1 will give the best value for K
```

#the confusion matrix using both the functions

```
library(gmodels)
Validation_data_best_k<-predict(model,validation_norm[,-7])
confusionMatrix(Validation_data_best_k ,Valid_label)
```

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

```
##
##           Reference
## Prediction    0    1
##          0 1782   59
##          1   26  133
##
##                Accuracy : 0.9575
##                  95% CI : (0.9477, 0.9659)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7348
##
##  Mcnemar's Test P-Value : 0.0005187
##
##             Sensitivity : 0.9856
##             Specificity : 0.6927
##          Pos Pred Value : 0.9680
##          Neg Pred Value : 0.8365
##              Prevalence : 0.9040
##          Detection Rate : 0.8910
##    Detection Prevalence : 0.9205
##       Balanced Accuracy : 0.8392
##
##        'Positive' Class : 0
##
```

```
CrossTable(Validation_data_best_k,Valid_label)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##                         | Valid_label
## Validation_data_best_k |         0 |         1 | Row Total |
## -----------------------|-----------|-----------|-----------|
##                      0 |      1782 |        59 |      1841 |
##                        |     8.329 |    78.432 |           |
##                        |     0.968 |     0.032 |     0.920 |
##                        |     0.986 |     0.307 |           |
##                        |     0.891 |     0.029 |           |
## -----------------------|-----------|-----------|-----------|
##                      1 |        26 |       133 |       159 |
```

```
##                          |    96.439 |   908.135 |           |
##                          |     0.164 |     0.836 |     0.080 |
##                          |     0.014 |     0.693 |           |
##                          |     0.013 |     0.066 |           |
## -------------------------|-----------|-----------|-----------|
##            Column Total |      1808 |       192 |      2000 |
##                          |     0.904 |     0.096 |           |
## -------------------------|-----------|-----------|-----------|
##
##
```

#Classifying the customer using the best k

```
Prediction_new<-knn(Train_predictors,test_norm,cl=Train_label,k=best_k)
Prediction_new
```

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

```
#Customer using the new K value will also not accept the loan offer because again K = 0
```

#Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%).

```
Test_Index_N = createDataPartition(Bankdata4$Personal.Loan,p=0.2, list=FALSE) # 20% reserved for Test
Test_Data_N = Bankdata4[Test_Index_N,]
TrainAndValid_Data = Bankdata4[-Test_Index_N,] # Validation and Training data is rest
Train_Index_N = createDataPartition(TrainAndValid_Data$Personal.Loan,p=25/40, list=FALSE) # 50% of rema
Train_Data_N = TrainAndValid_Data[Train_Index_N,]
Validation_Data_N = TrainAndValid_Data[-Train_Index_N,] # rest as validation
nrow(Train_Data_N)
```

```
## [1] 2500
```

```
summary(Train_Data_N)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 38.00   1st Qu.:1.000
##  Median :46.00   Median :21.00   Median : 62.00   Median :2.000
##  Mean   :45.46   Mean   :20.21   Mean   : 73.59   Mean   :2.392
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 95.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :205.00   Max.   :4.000
##      CCAvg            Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:2260        Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 240        1st Qu.:0.0000
##  Median : 1.500   Median :  0.00                 Median :0.0000
##  Mean   : 1.909   Mean   : 57.31                 Mean   :0.1084
##  3rd Qu.: 2.500   3rd Qu.:101.00                 3rd Qu.:0.0000
##  Max.   :10.000   Max.   :635.00                 Max.   :1.0000
##    CD.Account         Online         CreditCard    Education1 Education2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   0:1436     0:1812
```

```
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1:1064     1: 688
##  Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.0608   Mean   :0.5992   Mean   :0.292
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000
##  Education3
##  0:1752
##  1: 748
##
##
##
##
```

```
nrow(Validation_Data_N)
```

```
## [1] 1500
```

```
summary(Validation_Data_N)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 65.00   Median :2.000
##  Mean   :45.38   Mean   :20.13   Mean   : 74.11   Mean   :2.409
##  3rd Qu.:56.00   3rd Qu.:30.00   3rd Qu.:102.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :218.00   Max.   :4.000
##      CCAvg           Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:1356        Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 144        1st Qu.:0.0000
##  Median : 1.600   Median :  0.00                 Median :0.0000
##  Mean   : 2.002   Mean   : 56.83                 Mean   :0.1047
##  3rd Qu.: 2.600   3rd Qu.:102.00                 3rd Qu.:0.0000
##  Max.   :10.000   Max.   :612.00                 Max.   :1.0000
##    CD.Account          Online         CreditCard     Education1 Education2
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   0:874      0:1098
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1:626      1: 402
##  Median :0.00000   Median :1.0000   Median :0.0000
##  Mean   :0.06733   Mean   :0.5973   Mean   :0.3047
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##  Education3
##  0:1028
##  1: 472
##
##
##
##
```

```
nrow(Test_Data_N)
```

```
## [1] 1000
```

```
summary(Test_Data_N)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 65.00   Median :2.000
##  Mean   :44.96   Mean   :19.81   Mean   : 73.72   Mean   :2.387
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 94.25   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000
##      CCAvg           Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:904         Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 96         1st Qu.:0.000
##  Median : 1.500   Median :  0.00                 Median :0.000
##  Mean   : 1.915   Mean   : 53.97                 Mean   :0.094
##  3rd Qu.: 2.500   3rd Qu.: 98.00                 3rd Qu.:0.000
##  Max.   :10.000   Max.   :582.00                 Max.   :1.000
##    CD.Account        Online       CreditCard    Education1 Education2
##  Min.   :0.000   Min.   :0.00   Min.   :0.000   0:594      0:687
##  1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.000   1:406      1:313
##  Median :0.000   Median :1.00   Median :0.000
##  Mean   :0.049   Mean   :0.59   Mean   :0.283
##  3rd Qu.:0.000   3rd Qu.:1.00   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.00   Max.   :1.000
##  Education3
##  0:719
##  1:281
##
##
##
##
```

##normalization of all 3 datas.

```
Norm_model_N <- preProcess(Train_Data_N, method = c("center", "scale"))
training_norm_N<-predict(Norm_model_N,Train_Data_N)
head(training_norm_N)
```

```
##          Age Experience      Income    Family      CCAvg   Mortgage
## 1 -1.8046984 -1.6935662 -0.52114002  1.396408 -0.1790084 -0.5607791
## 3 -0.5699326 -0.4590085 -1.32636254 -1.209479 -0.5267236 -0.5607791
## 5 -0.9227229 -1.0762874 -0.60590028  1.396408 -0.5267236 -0.5607791
## 7  0.6648332  0.5991837 -0.03376849 -0.340850 -0.2369609 -0.5607791
## 8  0.4002405  0.3346357 -1.09327181 -1.209479 -0.9323913 -0.5607791
## 9 -0.9227229 -0.8999220  0.15694211  0.527779 -0.7585337  0.4568338
##   Personal.Loan Securities.Account CD.Account     Online CreditCard Education1
## 1             0          2.8673685 -0.2543817 -1.2224614 -0.6420782          1
## 3             0         -0.3486123 -0.2543817 -1.2224614 -0.6420782          1
## 5             0         -0.3486123 -0.2543817 -1.2224614  1.5568197          0
## 7             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
## 8             0         -0.3486123 -0.2543817 -1.2224614  1.5568197          0
## 9             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
##   Education2 Education3
```

```
## 1          0          0
## 3          0          0
## 5          1          0
## 7          1          0
## 8          0          1
## 9          1          0
```

```
validation_norm_N<-predict(Norm_model_N,Validation_Data_N)
head(validation_norm_N)
```

```
##           Age Experience     Income    Family       CCAvg   Mortgage
## 4  -0.9227229 -0.9881047  0.5595534 -1.209479  0.45846943 -0.5607791
## 10 -1.0109204 -0.9881047  2.2547587 -1.209479  4.05152637 -0.5607791
## 11  1.7232039  1.6573760  0.6655037  1.396408  0.28461184 -0.5607791
## 12 -1.4519082 -1.3408354 -0.6059003  0.527779 -1.04829638 -0.5607791
## 21  0.9294259  0.9519145 -1.0297016  1.396408 -0.58467613  0.5253270
## 22  1.0176234  0.5991837 -0.2244791  0.527779  0.05280171 -0.5607791
##    Personal.Loan Securities.Account CD.Account     Online CreditCard Education1
## 4              0         -0.3486123 -0.2543817 -1.2224614 -0.6420782          0
## 10             1         -0.3486123 -0.2543817 -1.2224614 -0.6420782          0
## 11             0         -0.3486123 -0.2543817 -1.2224614 -0.6420782          0
## 12             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
## 21             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
## 22             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
##    Education2 Education3
## 4           1          0
## 10          0          1
## 11          0          1
## 12          1          0
## 21          1          0
## 22          0          1
```

```
Test_norm_N<-predict(Norm_model_N,Test_Data_N)
head(Test_norm_N)
```

```
##           Age Experience     Income    Family      CCAvg   Mortgage
## 2  -0.04074727 -0.1062778 -0.8389910  0.527779 -0.2369609 -0.5607791
## 6  -0.74632774 -0.6353739 -0.9449413  1.396408 -0.8744388  0.9558556
## 14  1.19401855  1.0400972 -0.7118506  1.396408  0.3425644 -0.5607791
## 16  1.28221611  0.8637318 -1.0932718 -1.209479 -0.2369609 -0.5607791
## 17 -0.65813018 -0.5471912  1.1952554  1.396408  1.6175201  0.7503760
## 19  0.04745029  0.0700876  2.5302295 -0.340850  3.5879061 -0.5607791
##    Personal.Loan Securities.Account CD.Account     Online CreditCard Education1
## 2              0          2.8673685 -0.2543817 -1.2224614 -0.6420782          1
## 6              0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
## 14             0         -0.3486123 -0.2543817  0.8176945 -0.6420782          0
## 16             0         -0.3486123 -0.2543817  0.8176945  1.5568197          0
## 17             1         -0.3486123 -0.2543817 -1.2224614 -0.6420782          0
## 19             1         -0.3486123 -0.2543817 -1.2224614 -0.6420782          0
##    Education2 Education3
## 2           0          0
## 6           1          0
## 14          1          0
```

```
## 16          0          1
## 17          0          1
## 19          0          1
```

#Classifying the customer from all 3 set (training,validation and testing) using the best k

```
Train_predictors_N <-training_norm_N[,-7]
Train_label_N<-training_norm_N[,7]
valid_predictors_N<-validation_norm_N[,-7]
Valid_label_N<-validation_norm_N[,7]
Test_predictors_N<-Test_norm_N[,-7]
Test_label_N<-Test_norm_N[,7]
training_prediction_N <-knn(Train_predictors_N,Train_predictors_N,cl=Train_label_N,k=best_k)
head(training_prediction_N)
```

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

```
validation_prediction_N <-knn(Train_predictors_N,valid_predictors_N,cl=Train_label_N,k=best_k)
head(validation_prediction_N)
```

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

```
Test_prediction_N <-knn(Train_predictors_N,Test_predictors_N,cl=Train_label_N,k=best_k)
head(Test_prediction_N)
```

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

#the confusion matrix using both the functions for all 3 datasets Training, Validation and Test

```
confusionMatrix(training_prediction_N,Train_label_N)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2260    0
##          1    0  240
##
##                Accuracy : 1
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.000
```

```
##             Specificity : 1.000
##          Pos Pred Value : 1.000
##          Neg Pred Value : 1.000
##              Prevalence : 0.904
##          Detection Rate : 0.904
##    Detection Prevalence : 0.904
##       Balanced Accuracy : 1.000
##
##        'Positive' Class : 0
##
```

```
CrossTable(training_prediction_N,Train_label_N)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2500
##
##
##                      | Train_label_N
## training_prediction_N |         0 |         1 | Row Total |
## ----------------------|-----------|-----------|-----------|
##                     0 |      2260 |         0 |      2260 |
##                       |    23.040 |   216.960 |           |
##                       |     1.000 |     0.000 |     0.904 |
##                       |     1.000 |     0.000 |           |
##                       |     0.904 |     0.000 |           |
## ----------------------|-----------|-----------|-----------|
##                     1 |         0 |       240 |       240 |
##                       |   216.960 |  2043.040 |           |
##                       |     0.000 |     1.000 |     0.096 |
##                       |     0.000 |     1.000 |           |
##                       |     0.000 |     0.096 |           |
## ----------------------|-----------|-----------|-----------|
##          Column Total |      2260 |       240 |      2500 |
##                       |     0.904 |     0.096 |           |
## ----------------------|-----------|-----------|-----------|
##
##
```

```
confusionMatrix(validation_prediction_N,Valid_label_N)
```

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

```
##           Reference
## Prediction    0    1
##          0 1343   51
##          1   13   93
##
##                Accuracy : 0.9573
##                  95% CI : (0.9458, 0.967)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 5.372e-15
##
##                   Kappa : 0.7213
##
##  Mcnemar's Test P-Value : 3.746e-06
##
##             Sensitivity : 0.9904
##             Specificity : 0.6458
##          Pos Pred Value : 0.9634
##          Neg Pred Value : 0.8774
##              Prevalence : 0.9040
##          Detection Rate : 0.8953
##    Detection Prevalence : 0.9293
##       Balanced Accuracy : 0.8181
##
##        'Positive' Class : 0
##
```

```
CrossTable(validation_prediction_N,Valid_label_N)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1500
##
##
##                          | Valid_label_N
## validation_prediction_N |         0 |         1 | Row Total |
## ------------------------|-----------|-----------|-----------|
##                       0 |      1343 |        51 |      1394 |
##                         |     5.444 |    51.260 |           |
##                         |     0.963 |     0.037 |     0.929 |
##                         |     0.990 |     0.354 |           |
##                         |     0.895 |     0.034 |           |
## ------------------------|-----------|-----------|-----------|
##                       1 |        13 |        93 |       106 |
##                         |    71.588 |   674.117 |           |
```

```
##                          |     0.123 |     0.877 |     0.071 |
##                          |     0.010 |     0.646 |           |
##                          |     0.009 |     0.062 |           |
## -------------------------|-----------|-----------|-----------|
##             Column Total |      1356 |       144 |      1500 |
##                          |     0.904 |     0.096 |           |
## -------------------------|-----------|-----------|-----------|
##
##
```

```
confusionMatrix(Test_prediction_N,Test_label_N)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 894  37
##          1  10  59
##
##                Accuracy : 0.953
##                  95% CI : (0.938, 0.9653)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 5.885e-09
##
##                   Kappa : 0.6903
##
##  Mcnemar's Test P-Value : 0.0001491
##
##             Sensitivity : 0.9889
##             Specificity : 0.6146
##          Pos Pred Value : 0.9603
##          Neg Pred Value : 0.8551
##              Prevalence : 0.9040
##          Detection Rate : 0.8940
##    Detection Prevalence : 0.9310
##       Balanced Accuracy : 0.8018
##
##        'Positive' Class : 0
##
```

```
CrossTable(Test_prediction_N,Test_label_N)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
```

```
##
## Total Observations in Table:  1000
##
##
##                    | Test_label_N
## Test_prediction_N |         0 |         1 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |       894 |        37 |       931 |
##                   |     3.259 |    30.693 |           |
##                   |     0.960 |     0.040 |     0.931 |
##                   |     0.989 |     0.385 |           |
##                   |     0.894 |     0.037 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |        10 |        59 |        69 |
##                   |    43.979 |   414.137 |           |
##                   |     0.145 |     0.855 |     0.069 |
##                   |     0.011 |     0.615 |           |
##                   |     0.010 |     0.059 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       904 |        96 |      1000 |
##                   |     0.904 |     0.096 |           |
## ------------------|-----------|-----------|-----------|
##
##
```

##Compare the confusion matrix between the training and validation sets with the test set.

##For the training set, validation set, and test set, confusion matrices were made. The training set confusion matrix displays 100% accuracy with k=1 as is typical for KNN models because the model is already aware of the values. The validation set confusion matrix displays a 95.47% overall accuracy, a 98.89% high sensitivity, and a 63.19% low specificity. This confusion matrix demonstrates that the model is less successful at accurately predicting which customers will accept the loan (out of the 144 customers who accepted the loan, the model only correctly predicted 91 of those customers would accept the loan, resulting in a low specificity of 63.19%). Nonetheless, this model is quite good at properly anticipating future events.