KNN for Classification

Nisha Chaurasia

2023-02-17

#The goal is to use k-NN to predict whether a new customer will accept a loan offer. This will serve as the basis for the design of a new campaign.

##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/Chaur/OneDrive/Desktop/FML/Assignment\_2\_KNN/UniversalBank.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] "factor"

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 = Education3)  
Bankdata4 <- Bankdata3[,-6]

##Partitioning the data into training (60%) and validation (40%) sets Also showed the summary statistics of both train and test data set.

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 :45.00 Median :20.00 Median : 63.00 Median :2.000   
## Mean :45.35 Mean :20.11 Mean : 73.66 Mean :2.391   
## 3rd Qu.:55.00 3rd Qu.:30.00 3rd Qu.: 99.00 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. : 0.0 0:2712 Min. :0.0000   
## 1st Qu.: 0.700 1st Qu.: 0.0 1: 288 1st Qu.:0.0000   
## Median : 1.600 Median : 0.0 Median :0.0000   
## Mean : 1.943 Mean : 55.1 Mean :0.1023   
## 3rd Qu.: 2.500 3rd Qu.: 99.0 3rd Qu.:0.0000   
## Max. :10.000 Max. :617.0 Max. :1.0000   
## CD.Account Online CreditCard Education1 Education2  
## Min. :0.000 Min. :0.0000 Min. :0.0000 0:1756 0:2143   
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1:1244 1: 857   
## Median :0.000 Median :1.0000 Median :0.0000   
## Mean :0.063 Mean :0.6077 Mean :0.2903   
## 3rd Qu.:0.000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.000 Max. :1.0000 Max. :1.0000   
## Education3  
## 0:2101   
## 1: 899   
##   
##   
##   
##

nrow(Validation\_df)

## [1] 2000

summary(Validation\_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 :45.33 Mean :20.1 Mean : 73.94 Mean :2.404   
## 3rd Qu.:55.00 3rd Qu.:30.0 3rd Qu.: 98.00 3rd Qu.:4.000   
## Max. :67.00 Max. :43.0 Max. :218.00 Max. :4.000   
## CCAvg Mortgage Personal\_Loan Securities.Account  
## Min. : 0.000 Min. : 0.0 0:1808 Min. :0.0000   
## 1st Qu.: 0.700 1st Qu.: 0.0 1: 192 1st Qu.:0.0000   
## Median : 1.500 Median : 0.0 Median :0.0000   
## Mean : 1.930 Mean : 58.6 Mean :0.1075   
## 3rd Qu.: 2.525 3rd Qu.:103.0 3rd Qu.:0.0000   
## Max. :10.000 Max. :635.0 Max. :1.0000   
## CD.Account Online CreditCard Education1 Education2  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 0:1148 0:1454   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1: 852 1: 546   
## Median :0.0000 Median :1.0000 Median :0.0000   
## Mean :0.0565 Mean :0.5805 Mean :0.2995   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Education3  
## 0:1398   
## 1: 602   
##   
##   
##   
##

##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  
## 1 -1.7846455 -1.6778232 -0.53643752 1.4083901 -0.1969860 -0.5520363  
## 3 -0.5566784 -0.4484772 -1.36295297 -1.2181145 -0.5413035 -0.5520363  
## 4 -0.9075262 -0.9753398 0.57283320 -1.2181145 0.4342629 -0.5520363  
## 7 0.6712887 0.6052481 -0.03617818 -0.3426129 -0.2543722 -0.5520363  
## 8 0.4081529 0.3418168 -1.12369850 -1.2181145 -0.9430074 -0.5520363  
## 13 0.2327290 0.2540063 0.87733889 -0.3426129 1.0655117 -0.5520363  
## Personal\_Loan Securities.Account CD.Account Online CreditCard Education1  
## 1 0 2.9612604 -0.2592556 -1.2443218 -0.6395122 1  
## 3 0 -0.3375815 -0.2592556 -1.2443218 -0.6395122 1  
## 4 0 -0.3375815 -0.2592556 -1.2443218 -0.6395122 0  
## 7 0 -0.3375815 -0.2592556 0.8033828 -0.6395122 0  
## 8 0 -0.3375815 -0.2592556 -1.2443218 1.5631705 0  
## 13 0 2.9612604 -0.2592556 -1.2443218 -0.6395122 0  
## Education2 Education3  
## 1 0 0  
## 3 0 0  
## 4 1 0  
## 7 1 0  
## 8 0 1  
## 13 0 1

validation\_norm<-predict(Norm\_model,Validation\_df)  
head(validation\_norm)

## Age Experience Income Family CCAvg Mortgage  
## 2 -0.0304068 -0.09723542 -0.8626936 0.5328886 -0.2543722 -0.5520363  
## 5 -0.9075262 -1.06315020 -0.6234391 1.4083901 -0.5413035 -0.5520363  
## 6 -0.7321023 -0.62409803 -0.9714457 1.4083901 -0.8856211 1.0009540  
## 9 -0.9075262 -0.88752933 0.1595755 0.5328886 -0.7708486 0.4899701  
## 10 -0.9952381 -0.97533976 2.3128657 -1.2181145 3.9922109 -0.5520363  
## 11 1.7238319 1.65897326 0.6815852 1.4083901 0.2621041 -0.5520363  
## Personal\_Loan Securities.Account CD.Account Online CreditCard Education1  
## 2 0 2.9612604 -0.2592556 -1.2443218 -0.6395122 1  
## 5 0 -0.3375815 -0.2592556 -1.2443218 1.5631705 0  
## 6 0 -0.3375815 -0.2592556 0.8033828 -0.6395122 0  
## 9 0 -0.3375815 -0.2592556 0.8033828 -0.6395122 0  
## 10 1 -0.3375815 -0.2592556 -1.2443218 -0.6395122 0  
## 11 0 -0.3375815 -0.2592556 -1.2443218 -0.6395122 0  
## Education2 Education3  
## 2 0 0  
## 5 1 0  
## 6 1 0  
## 9 1 0  
## 10 0 1  
## 11 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.Account=0,Online=1,CreditCard=1,Education1=0,Education2=1,Education3=0)  
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 -0.4689665 -0.8875293 0.2248267 -0.3426129 0.03255905 -0.5520363  
## Securities.Account CD.Account Online CreditCard Education1 Education2  
## 1 -0.3375815 -0.2592556 0.8033828 1.563171 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 0.9503421 0.6870157  
## 2 0.9479717 0.6680624  
## 3 0.9465942 0.6500074  
## 4 0.9457450 0.6365158  
## 5 0.9464570 0.6327889  
## 6 0.9458925 0.6215564  
## 7 0.9453041 0.6138299  
## 8 0.9447641 0.6072226  
## 9 0.9438555 0.5957918  
## 10 0.9436452 0.5925648  
## 11 0.9427356 0.5811269  
## 12 0.9416077 0.5706085  
## 13 0.9406017 0.5591549  
## 14 0.9396588 0.5500056  
## 15 0.9399063 0.5498148  
## 16 0.9393705 0.5440672  
## 17 0.9388683 0.5380813  
## 18 0.9387608 0.5353467  
## 19 0.9377910 0.5241475  
## 20 0.9370034 0.5182665  
## 21 0.9370076 0.5167183  
## 22 0.9367851 0.5123197  
## 23 0.9369655 0.5132753  
## 24 0.9357707 0.5013879  
## 25 0.9356925 0.4997720  
## 26 0.9352281 0.4933661  
## 27 0.9343582 0.4848563  
## 28 0.9341369 0.4817121  
## 29 0.9339575 0.4807816  
## 30 0.9329848 0.4715857  
##   
## 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 1784 66  
## 1 24 126  
##   
## Accuracy : 0.955   
## 95% CI : (0.945, 0.9637)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7126   
##   
## Mcnemar's Test P-Value : 1.548e-05   
##   
## Sensitivity : 0.9867   
## Specificity : 0.6562   
## Pos Pred Value : 0.9643   
## Neg Pred Value : 0.8400   
## Prevalence : 0.9040   
## Detection Rate : 0.8920   
## Detection Prevalence : 0.9250   
## Balanced Accuracy : 0.8215   
##   
## '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 | 1784 | 66 | 1850 |   
## | 7.447 | 70.127 | |   
## | 0.964 | 0.036 | 0.925 |   
## | 0.987 | 0.344 | |   
## | 0.892 | 0.033 | |   
## -----------------------|-----------|-----------|-----------|  
## 1 | 24 | 126 | 150 |   
## | 91.848 | 864.900 | |   
## | 0.160 | 0.840 | 0.075 |   
## | 0.013 | 0.656 | |   
## | 0.012 | 0.063 | |   
## -----------------------|-----------|-----------|-----------|  
## 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 remaining data as training  
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.:36.00 1st Qu.:10.00 1st Qu.: 38.75 1st Qu.:1.000   
## Median :46.00 Median :21.00 Median : 62.00 Median :2.000   
## Mean :45.75 Mean :20.48 Mean : 73.20 Mean :2.416   
## 3rd Qu.:56.00 3rd Qu.:30.00 3rd Qu.: 95.00 3rd Qu.:3.000   
## Max. :67.00 Max. :43.00 Max. :218.00 Max. :4.000   
## CCAvg Mortgage Personal\_Loan Securities.Account  
## Min. : 0.000 Min. : 0.0 0:2260 Min. :0.0000   
## 1st Qu.: 0.700 1st Qu.: 0.0 1: 240 1st Qu.:0.0000   
## Median : 1.500 Median : 0.0 Median :0.0000   
## Mean : 1.902 Mean : 56.9 Mean :0.1112   
## 3rd Qu.: 2.500 3rd Qu.:101.0 3rd Qu.:0.0000   
## Max. :10.000 Max. :635.0 Max. :1.0000   
## CD.Account Online CreditCard Education1 Education2  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 0:1451 0:1825   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1:1049 1: 675   
## Median :0.0000 Median :1.0000 Median :0.0000   
## Mean :0.0612 Mean :0.5964 Mean :0.2948   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Education3  
## 0:1724   
## 1: 776   
##   
##   
##   
##

nrow(Validation\_Data\_N)

## [1] 1500

summary(Validation\_Data\_N)

## Age Experience Income Family   
## Min. :23.00 Min. :-2.00 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 :44.99 Mean :19.77 Mean : 74.58 Mean :2.377   
## 3rd Qu.:55.00 3rd Qu.:29.00 3rd Qu.:102.25 3rd Qu.:3.000   
## Max. :67.00 Max. :43.00 Max. :204.00 Max. :4.000   
## CCAvg Mortgage Personal\_Loan Securities.Account  
## Min. : 0.000 Min. : 0.0 0:1356 Min. :0.00000   
## 1st Qu.: 0.700 1st Qu.: 0.0 1: 144 1st Qu.:0.00000   
## Median : 1.600 Median : 0.0 Median :0.00000   
## Mean : 2.000 Mean : 56.3 Mean :0.09667   
## 3rd Qu.: 2.618 3rd Qu.:102.0 3rd Qu.:0.00000   
## Max. :10.000 Max. :601.0 Max. :1.00000   
## CD.Account Online CreditCard Education1 Education2  
## Min. :0.000 Min. :0.0000 Min. :0.0000 0:869 0:1074   
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1:631 1: 426   
## Median :0.000 Median :1.0000 Median :0.0000   
## Mean :0.064 Mean :0.6013 Mean :0.3073   
## 3rd Qu.:0.000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.000 Max. :1.0000 Max. :1.0000   
## Education3  
## 0:1057   
## 1: 443   
##   
##   
##   
##

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 :44.50 Median :20.00 Median : 65.00 Median :2.000   
## Mean :44.83 Mean :19.67 Mean : 74.00 Mean :2.377   
## 3rd Qu.:55.00 3rd Qu.:29.00 3rd Qu.: 95.75 3rd Qu.:3.000   
## Max. :67.00 Max. :42.00 Max. :224.00 Max. :4.000   
## CCAvg Mortgage Personal\_Loan Securities.Account  
## Min. : 0.000 Min. : 0.0 0:904 Min. :0.000   
## 1st Qu.: 0.700 1st Qu.: 0.0 1: 96 1st Qu.:0.000   
## Median : 1.600 Median : 0.0 Median :0.000   
## Mean : 1.935 Mean : 55.8 Mean :0.099   
## 3rd Qu.: 2.500 3rd Qu.: 98.0 3rd Qu.:0.000   
## Max. :10.000 Max. :612.0 Max. :1.000   
## CD.Account Online CreditCard Education1 Education2  
## Min. :0.000 Min. :0.000 Min. :0.000 0:584 0:698   
## 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.000 1:416 1:302   
## Median :0.000 Median :1.000 Median :0.000   
## Mean :0.053 Mean :0.591 Mean :0.272   
## 3rd Qu.:0.000 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.000 Max. :1.000   
## Education3  
## 0:718   
## 1:282   
##   
##   
##   
##

##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 -1.8062167 -1.6979819 -0.52329477 1.3839793 -0.1758672 -0.5588272  
## 3 -0.5875175 -0.4778918 -1.34510598 -1.2365319 -0.5250042 -0.5588272  
## 5 -0.9357173 -1.0879369 -0.60980121 1.3839793 -0.5250042 -0.5588272  
## 7 0.6311817 0.5678996 -0.02588273 -0.3630282 -0.2340567 -0.5588272  
## 8 0.3700319 0.3064518 -1.10721326 -1.2365319 -0.9323306 -0.5588272  
## 9 -0.9357173 -0.9136383 0.16875677 0.5104756 -0.7577621 0.4626583  
## Personal\_Loan Securities.Account CD.Account Online CreditCard Education1  
## 1 0 2.8265896 -0.2552715 -1.2153640 -0.6464288 1  
## 3 0 -0.3536417 -0.2552715 -1.2153640 -0.6464288 1  
## 5 0 -0.3536417 -0.2552715 -1.2153640 1.5463419 0  
## 7 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 0  
## 8 0 -0.3536417 -0.2552715 -1.2153640 1.5463419 0  
## 9 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 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 -0.9357173 -1.0007876 0.5796624 -1.2365319 0.4642171 -0.5588272  
## 10 -1.0227672 -1.0007876 2.3097912 -1.2365319 4.0719652 -0.5588272  
## 11 1.6757811 1.6136911 0.6877954 1.3839793 0.2896486 -0.5588272  
## 12 -1.4580169 -1.3493847 -0.6098012 0.5104756 -1.0487095 -0.5588272  
## 21 0.8923316 0.9164968 -1.0423334 1.3839793 -0.5831936 0.5314121  
## 22 0.9793815 0.5678996 -0.2205222 0.5104756 0.0568907 -0.5588272  
## Personal\_Loan Securities.Account CD.Account Online CreditCard Education1  
## 4 0 -0.3536417 -0.2552715 -1.2153640 -0.6464288 0  
## 10 1 -0.3536417 -0.2552715 -1.2153640 -0.6464288 0  
## 11 0 -0.3536417 -0.2552715 -1.2153640 -0.6464288 0  
## 12 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 0  
## 21 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 0  
## 22 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 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 -0.06521782 -0.12929468 -0.8476939 0.5104756 -0.2340567 -0.5588272  
## 6 -0.76161738 -0.65219041 -0.9558270 1.3839793 -0.8741411 0.9635791  
## 14 1.15348142 1.00364606 -0.7179343 1.3839793 0.3478381 -0.5588272  
## 16 1.24053136 0.82934748 -1.1072133 -1.2365319 -0.2340567 -0.5588272  
## 17 -0.67456744 -0.56504112 1.2284607 1.3839793 1.6280068 0.7573176  
## 19 0.02183213 0.04500389 2.5909372 -0.3630282 3.6064493 -0.5588272  
## Personal\_Loan Securities.Account CD.Account Online CreditCard Education1  
## 2 0 2.8265896 -0.2552715 -1.2153640 -0.6464288 1  
## 6 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 0  
## 14 0 -0.3536417 -0.2552715 0.8224697 -0.6464288 0  
## 16 0 -0.3536417 -0.2552715 0.8224697 1.5463419 0  
## 17 1 -0.3536417 -0.2552715 -1.2153640 -0.6464288 0  
## 19 1 -0.3536417 -0.2552715 -1.2153640 -0.6464288 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 47  
## 1 13 97  
##   
## Accuracy : 0.96   
## 95% CI : (0.9488, 0.9693)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7424   
##   
## Mcnemar's Test P-Value : 2.042e-05   
##   
## Sensitivity : 0.9904   
## Specificity : 0.6736   
## Pos Pred Value : 0.9662   
## Neg Pred Value : 0.8818   
## Prevalence : 0.9040   
## Detection Rate : 0.8953   
## Detection Prevalence : 0.9267   
## Balanced Accuracy : 0.8320   
##   
## '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 | 47 | 1390 |   
## | 5.946 | 55.994 | |   
## | 0.966 | 0.034 | 0.927 |   
## | 0.990 | 0.326 | |   
## | 0.895 | 0.031 | |   
## ------------------------|-----------|-----------|-----------|  
## 1 | 13 | 97 | 110 |   
## | 75.140 | 707.564 | |   
## | 0.118 | 0.882 | 0.073 |   
## | 0.010 | 0.674 | |   
## | 0.009 | 0.065 | |   
## ------------------------|-----------|-----------|-----------|  
## Column Total | 1356 | 144 | 1500 |   
## | 0.904 | 0.096 | |   
## ------------------------|-----------|-----------|-----------|  
##   
##

confusionMatrix(Test\_prediction\_N,Test\_label\_N)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 895 39  
## 1 9 57  
##   
## Accuracy : 0.952   
## 95% CI : (0.9369, 0.9644)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : 1.264e-08   
##   
## Kappa : 0.6786   
##   
## Mcnemar's Test P-Value : 2.842e-05   
##   
## Sensitivity : 0.9900   
## Specificity : 0.5938   
## Pos Pred Value : 0.9582   
## Neg Pred Value : 0.8636   
## Prevalence : 0.9040   
## Detection Rate : 0.8950   
## Detection Prevalence : 0.9340   
## Balanced Accuracy : 0.7919   
##   
## '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 | 895 | 39 | 934 |   
## | 3.040 | 28.627 | |   
## | 0.958 | 0.042 | 0.934 |   
## | 0.990 | 0.406 | |   
## | 0.895 | 0.039 | |   
## ------------------|-----------|-----------|-----------|  
## 1 | 9 | 57 | 66 |   
## | 43.022 | 405.120 | |   
## | 0.136 | 0.864 | 0.066 |   
## | 0.010 | 0.594 | |   
## | 0.009 | 0.057 | |   
## ------------------|-----------|-----------|-----------|  
## Column Total | 904 | 96 | 1000 |   
## | 0.904 | 0.096 | |   
## ------------------|-----------|-----------|-----------|  
##   
##

##Compare the confusion matrix of the test set with that of the training and validation sets. ##The confusion matrix were created for the trianing set, validation set, and the test set. Firstly, as always expected for KNN models, the training set confusion matrix shows 100% accuracy with k=1 because the values are already seen by the model. The validation set confusion matrix shows an overall accuracy of 95.47% and a high sensitivity of 98.89% but a low specificity of 63.19%. This confusion matrix reveals that the model is not as accurate in correctly predicting customers who will accept the loan (out of the 144 customers who accepted the loan, the model only predicted that 91 of those customers would accept the loan, hence giving a low specificity of 63.19%). On the other hand, this model is very accurate in correctly predicting customers who will Not accept the loan, hence giving a high sensitivity. The test set confusion matrix shows an overall accuracy of 95% and a sensitivity of 99% and a specificity of 57.29%. This test set confusion matrix trends very similarly to the validation set confusion matrix which is a good thing.