Assignment 2 Updated

Pranay Kumar Kodeboyina

2023-03-01

# importing the libraries and including caret package  
library("caret")

## Loading required package: ggplot2

## Loading required package: lattice

# Loading required library: ISLR  
library('ISLR')  
  
# Loading required library: dplyr  
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

# Loading required library: class  
library('class')  
  
#Extracting the current working directory  
getwd()

## [1] "/Users/kodeboyina/Documents/Kent State/Sem1/Fundamentals of ML"

#setting the working directory to the Assignment Folder  
setwd("/Users/kodeboyina/Documents/Kent State/Sem1/Fundamentals of ML")  
  
#Loading Universal csv data Import the data set into R  
UniBank.df <- read.csv("Assignment2/data/UniversalBank.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)  
  
#Converting the "Education" and "Personal Loan" variable to a factor value that responses as "yes" or "no."  
UniBank.df$Education = as.factor(UniBank.df$Education)  
UniBank.df$Personal.Loan <- factor(UniBank.df$Personal.Loan,levels=c('0','1'),labels=c('No','Yes'))  
  
#Drop ID and Zip Code columns(classification with all predictors except ID and ZIP code)   
UniBank.df$ID <- NULL  
UniBank.df$ZIP.Code <- NULL  
  
#Observing the first 10 observations of the data set post removing ID and ZIP code  
head(UniBank.df, n=10L)

## Age Experience Income Family CCAvg Education Mortgage Personal.Loan  
## 1 25 1 49 4 1.6 1 0 No  
## 2 45 19 34 3 1.5 1 0 No  
## 3 39 15 11 1 1.0 1 0 No  
## 4 35 9 100 1 2.7 2 0 No  
## 5 35 8 45 4 1.0 2 0 No  
## 6 37 13 29 4 0.4 2 155 No  
## 7 53 27 72 2 1.5 2 0 No  
## 8 50 24 22 1 0.3 3 0 No  
## 9 35 10 81 3 0.6 2 104 No  
## 10 34 9 180 1 8.9 3 0 Yes  
## Securities.Account CD.Account Online CreditCard  
## 1 1 0 0 0  
## 2 1 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 1  
## 6 0 0 1 0  
## 7 0 0 1 0  
## 8 0 0 0 1  
## 9 0 0 1 0  
## 10 0 0 0 0

#Priniting the Structure of the data post removing the ID and Zip Code  
str(UniBank.df)

## 'data.frame': 5000 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 : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...  
## $ 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 of data for the observations  
summary(UniBank.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 Personal.Loan Securities.Account  
## Min. : 0.000 1:2096 Min. : 0.0 No :4520 Min. :0.0000   
## 1st Qu.: 0.700 2:1403 1st Qu.: 0.0 Yes: 480 1st Qu.:0.0000   
## Median : 1.500 3:1501 Median : 0.0 Median :0.0000   
## Mean : 1.938 Mean : 56.5 Mean :0.1044   
## 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   
## Min. :0.0000 Min. :0.0000 Min. :0.000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000   
## Median :0.0000 Median :1.0000 Median :0.000   
## Mean :0.0604 Mean :0.5968 Mean :0.294   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.000   
## Max. :1.0000 Max. :1.0000 Max. :1.000

#Creating Dummy columns for the Education data and spliting them into 3 columns "Education\_1 to \_3" using Ifelse loop .   
UniBank.df$Education\_1 <- ifelse(UniBank.df$Education == 1, 1, 0)  
UniBank.df$Education\_2 <- ifelse(UniBank.df$Education == 2, 1, 0)  
UniBank.df$Education\_3 <- ifelse(UniBank.df$Education == 3, 1, 0)  
  
#Dropping the Original education data post creation of the dummy columns  
UniBank.df$Education <- NULL  
  
#Displaying the names of the data columns post columns  
names(UniBank.df)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Mortgage"   
## [7] "Personal.Loan" "Securities.Account" "CD.Account"   
## [10] "Online" "CreditCard" "Education\_1"   
## [13] "Education\_2" "Education\_3"

#Randomization of the data and setting same random sequence  
set.seed(123)  
  
## Seperating 60% of data as Training set and remaining 40% as validation set  
train.index <-sample(row.names(UniBank.df), 0.6\*dim(UniBank.df)[1])  
valid.index <-setdiff(row.names(UniBank.df), train.index)   
  
#Assigning the data interms of indexes to tain and Validation sets  
train\_data <- UniBank.df[train.index,]  
val\_data <- UniBank.df[valid.index,]  
  
#Creating New Customer data and passing the value as integer for calculations in Euclidean Distances  
new\_cust = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer(84), Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.integer(0), Securities.Account = as.integer(0), CD.Account = as.integer(0), Online = as.integer(1), CreditCard = as.integer(1), Education\_1 = as.integer(0), Education\_2 = as.integer(1), Education\_3 = as.integer(0))  
  
#Displaying the names of the New customer data  
names(new\_cust)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Mortgage"   
## [7] "Securities.Account" "CD.Account" "Online"   
## [10] "CreditCard" "Education\_1" "Education\_2"   
## [13] "Education\_3"

# Copy the original data to Normalization data frame to perform normalization  
train\_norm\_df <- train\_data  
Val\_norm\_df <- val\_data  
new\_cust\_df <- new\_cust  
  
  
# use preProcess() from the caret package to normalize data using center scale method  
norm.values <- preProcess(train\_data[, 1:6], method=c("center", "scale"))  
  
#Normalizing the Training data for the first 6 columns(As Other columns are already Normalized)   
train\_norm\_df[, 1:6] <- predict(norm.values, train\_data[, 1:6])  
  
# Replace first 6 columns with normalized values of Training and Validation data  
Val\_norm\_df[, 1:6] <- predict(norm.values, val\_data[, 1:6])  
  
  
norm.valuestest <- preProcess(new\_cust[, 1:5], method=c("center", "scale"))

## Warning in preProcess.default(new\_cust[, 1:5], method = c("center", "scale")):  
## Std. deviations could not be computed for: Age, Experience, Income, Family,  
## CCAvg

# Replace first 5 columns with normalized values of Training and Validation data (As Other columns are already Normalized)  
new\_cust\_df[, 1:5] <- predict(norm.valuestest, new\_cust[, 1:5])  
  
  
#Summary of the Normalized training data   
summary(train\_norm\_df)

## Age Experience Income Family   
## Min. :-1.96925 Min. :-2.03259 Min. :-1.4286 Min. :-1.2142   
## 1st Qu.:-0.82784 1st Qu.:-0.89392 1st Qu.:-0.7514 1st Qu.:-1.2142   
## Median : 0.05016 Median :-0.01801 Median :-0.2053 Median :-0.3463   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.84037 3rd Qu.: 0.85789 3rd Qu.: 0.4720 3rd Qu.: 1.3895   
## Max. : 1.89398 Max. : 1.99656 Max. : 3.2901 Max. : 1.3895   
## CCAvg Mortgage Personal.Loan Securities.Account  
## Min. :-1.1157 Min. :-0.5477 No :2722 Min. :0.000   
## 1st Qu.:-0.7098 1st Qu.:-0.5477 Yes: 278 1st Qu.:0.000   
## Median :-0.2459 Median :-0.5477 Median :0.000   
## Mean : 0.0000 Mean : 0.0000 Mean :0.103   
## 3rd Qu.: 0.3921 3rd Qu.: 0.4231 3rd Qu.:0.000   
## Max. : 4.6835 Max. : 5.7429 Max. :1.000   
## CD.Account Online CreditCard Education\_1   
## Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000   
## Median :0.000 Median :1.0000 Median :0.0000 Median :0.000   
## Mean :0.059 Mean :0.5997 Mean :0.2943 Mean :0.424   
## 3rd Qu.:0.000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.000   
## Education\_2 Education\_3   
## Min. :0.000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.000 Median :0.000   
## Mean :0.285 Mean :0.291   
## 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.000

var(train\_norm\_df[, 1:6])

## Age Experience Income Family CCAvg  
## Age 1.000000000 0.994046117 -0.05888630 -0.04751827 -0.06495956  
## Experience 0.994046117 1.000000000 -0.04962719 -0.05207024 -0.06336173  
## Income -0.058886301 -0.049627190 1.00000000 -0.15859307 0.64907373  
## Family -0.047518269 -0.052070243 -0.15859307 1.00000000 -0.10462456  
## CCAvg -0.064959563 -0.063361729 0.64907373 -0.10462456 1.00000000  
## Mortgage -0.007099632 -0.003875213 0.20135911 -0.02046881 0.11566904  
## Mortgage  
## Age -0.007099632  
## Experience -0.003875213  
## Income 0.201359111  
## Family -0.020468810  
## CCAvg 0.115669037  
## Mortgage 1.000000000

#Summary of the Normalized Validation data   
summary(Val\_norm\_df)

## Age Experience Income Family   
## Min. :-1.96925 Min. :-2.03259 Min. :-1.42861 Min. :-1.214194   
## 1st Qu.:-0.91565 1st Qu.:-0.89392 1st Qu.:-0.75140 1st Qu.:-1.214194   
## Median :-0.03764 Median :-0.01801 Median :-0.22710 Median :-0.346293   
## Mean :-0.01981 Mean :-0.02213 Mean : 0.02067 Mean :-0.005641   
## 3rd Qu.: 0.84037 3rd Qu.: 0.85789 3rd Qu.: 0.58119 3rd Qu.: 0.521609   
## Max. : 1.89398 Max. : 1.99656 Max. : 2.85314 Max. : 1.389511   
## CCAvg Mortgage Personal.Loan Securities.Account  
## Min. :-1.11573 Min. :-0.54769 No :1798 Min. :0.0000   
## 1st Qu.:-0.70979 1st Qu.:-0.54769 Yes: 202 1st Qu.:0.0000   
## Median :-0.18786 Median :-0.54769 Median :0.0000   
## Mean : 0.02029 Mean : 0.03003 Mean :0.1065   
## 3rd Qu.: 0.33407 3rd Qu.: 0.47267 3rd Qu.:0.0000   
## Max. : 4.68347 Max. : 5.40606 Max. :1.0000   
## CD.Account Online CreditCard Education\_1   
## 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 :1.0000 Median :0.0000 Median :0.000   
## Mean :0.0625 Mean :0.5925 Mean :0.2935 Mean :0.412   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.000   
## Education\_2 Education\_3   
## Min. :0.000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.000 Median :0.000   
## Mean :0.274 Mean :0.314   
## 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.000

var(Val\_norm\_df[, 1:6])

## Age Experience Income Family CCAvg  
## Age 1.03275513 1.02201827 -0.05130218 -0.04514796 -0.03498247  
## Experience 1.02201827 1.02267578 -0.04292421 -0.05347267 -0.03216072  
## Income -0.05130218 -0.04292421 1.02852108 -0.15653031 0.67246370  
## Family -0.04514796 -0.05347267 -0.15653031 0.98080706 -0.11883497  
## CCAvg -0.03498247 -0.03216072 0.67246370 -0.11883497 1.06824606  
## Mortgage -0.02079095 -0.02057021 0.22158694 -0.02050409 0.10677938  
## Mortgage  
## Age -0.02079095  
## Experience -0.02057021  
## Income 0.22158694  
## Family -0.02050409  
## CCAvg 0.10677938  
## Mortgage 1.03820652

#Summary of the Normalized New Customer data   
summary(new\_cust\_df)

## Age Experience Income Family CCAvg Mortgage  
## Min. :0 Min. :0 Min. :0 Min. :0 Min. :0 Min. :0   
## 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0   
## Median :0 Median :0 Median :0 Median :0 Median :0 Median :0   
## Mean :0 Mean :0 Mean :0 Mean :0 Mean :0 Mean :0   
## 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0   
## Max. :0 Max. :0 Max. :0 Max. :0 Max. :0 Max. :0   
## Securities.Account CD.Account Online CreditCard Education\_1  
## Min. :0 Min. :0 Min. :1 Min. :1 Min. :0   
## 1st Qu.:0 1st Qu.:0 1st Qu.:1 1st Qu.:1 1st Qu.:0   
## Median :0 Median :0 Median :1 Median :1 Median :0   
## Mean :0 Mean :0 Mean :1 Mean :1 Mean :0   
## 3rd Qu.:0 3rd Qu.:0 3rd Qu.:1 3rd Qu.:1 3rd Qu.:0   
## Max. :0 Max. :0 Max. :1 Max. :1 Max. :0   
## Education\_2 Education\_3  
## Min. :1 Min. :0   
## 1st Qu.:1 1st Qu.:0   
## Median :1 Median :0   
## Mean :1 Mean :0   
## 3rd Qu.:1 3rd Qu.:0   
## Max. :1 Max. :0

#Question 1  
#KNN classification excluding the personal loan column and predicting the value of Personal Loan for K = 1  
new\_norm\_cust\_pred <- class::knn(train= train\_norm\_df[,-c(7)], test = new\_cust\_df, cl= train\_norm\_df$Personal.Loan, k=1 , prob=TRUE)  
  
#calculate knn for customer prediction  
print(new\_norm\_cust\_pred)

## [1] No  
## attr(,"prob")  
## [1] 1  
## Levels: No Yes

#From the above observation the customer will not accept the loan offer from KNN classification using K = 1

#Question 2 Choice of K for the highest value of accuracy  
#What is a choice of k that balances between over fitting and ignoring the predictor information?  
# performing 10-fold cross-validation and initializing data frame with two columns: k, and accuracy.  
set.seed(11)  
Uni\_Bank\_Acc <- trainControl(method= "repeatedcv", number = 3, repeats = 2)  
searchGrid = expand.grid(k=1:10)  
knn\_Predict = train(Personal.Loan~., data = train\_norm\_df, method = 'knn', tuneGrid = searchGrid,trControl = Uni\_Bank\_Acc)  
knn\_Predict

## k-Nearest Neighbors   
##   
## 3000 samples  
## 13 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold, repeated 2 times)   
## Summary of sample sizes: 2000, 1999, 2001, 2001, 2000, 1999, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.9575011 0.7142331  
## 2 0.9521688 0.6722806  
## 3 0.9583343 0.6993282  
## 4 0.9528343 0.6508396  
## 5 0.9534990 0.6480949  
## 6 0.9515005 0.6283389  
## 7 0.9509996 0.6211979  
## 8 0.9500013 0.6104091  
## 9 0.9495000 0.6046515  
## 10 0.9473330 0.5822959  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 3.

#Question 3 Show the confusion matrix for the validation data that results from using the best k.  
  
knn\_Predict\_bank <- predict(knn\_Predict,Val\_norm\_df)  
confusionMatrix(knn\_Predict\_bank,Val\_norm\_df$Personal.Loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1796 66  
## Yes 2 136  
##   
## Accuracy : 0.966   
## 95% CI : (0.9571, 0.9735)  
## No Information Rate : 0.899   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7821   
##   
## Mcnemar's Test P-Value : 2.174e-14   
##   
## Sensitivity : 0.9989   
## Specificity : 0.6733   
## Pos Pred Value : 0.9646   
## Neg Pred Value : 0.9855   
## Prevalence : 0.8990   
## Detection Rate : 0.8980   
## Detection Prevalence : 0.9310   
## Balanced Accuracy : 0.8361   
##   
## 'Positive' Class : No   
##

#Question 4 Classifying the New customer with the best value of K = 3  
#KNN classification excluding the personal loan column and prediction the Value of Personal Loan for K = 3  
new\_norm\_cust\_pred <- class::knn(train= train\_norm\_df[,-c(7)], test = new\_cust\_df, cl= train\_norm\_df$Personal.Loan, k=3 , prob=TRUE)  
  
#calculate knn for customer prediction for the best value of K i.e 3  
print(new\_norm\_cust\_pred)

## [1] No  
## attr(,"prob")  
## [1] 1  
## Levels: No Yes

#From the above observation the customer will not accept the loan offer from KNN classification using best value i.e K = 3

#Splitting the data in the format(50%: 30%: 20%)  
#Randomization Setting seed to generate the same random sequence  
set.seed(123)  
  
## Separating 50% of data as Training set and 30% as validation set and remaining 20% as Test set  
train.index <-sample(row.names(UniBank.df), 0.5\*dim(UniBank.df)[1])  
valid.index <-sample(setdiff(row.names(UniBank.df), train.index), 0.3\*dim(UniBank.df)[1])   
test.index <-setdiff(row.names(UniBank.df),union(train.index,valid.index))  
  
#Assigning the data indexes to train, validation and test sets  
train\_data <- UniBank.df[train.index,]  
val\_data <- UniBank.df[valid.index,]  
Test\_data <- UniBank.df[test.index,]  
  
#Creating New Customer data and passing the value as integer for calculations in Euclidean distance calculation  
new\_cust = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer(84), Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.integer(0), Securities.Account = as.integer(0), CD.Account = as.integer(0), Online = as.integer(1), CreditCard = as.integer(1), Education\_1 = as.integer(0), Education\_2 = as.integer(1), Education\_3 = as.integer(0))  
  
#Displaying the names of the New customer data  
names(new\_cust)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Mortgage"   
## [7] "Securities.Account" "CD.Account" "Online"   
## [10] "CreditCard" "Education\_1" "Education\_2"   
## [13] "Education\_3"

# Copy the original data to Normalized data frame to perform normalization  
train\_norm\_df <- train\_data  
Val\_norm\_df <- val\_data  
Test\_norm\_df <- Test\_data  
  
# use preProcess() from the caret package to normalize data using center scale method  
norm.values <- preProcess(train\_data[, 1:6], method=c("center", "scale"))  
  
#Normalizing the Training data for the first 6 columns(Other columns are already Normalized)   
train\_norm\_df[, 1:6] <- predict(norm.values, train\_data[, 1:6])  
  
# Replace first 6 columns with normalized values of Validation data  
Val\_norm\_df[, 1:6] <- predict(norm.values, val\_data[, 1:6])  
  
# Replace first 6 columns with normalized values of Test data  
Test\_norm\_df[, 1:6] <- predict(norm.values, Test\_data[, 1:6])  
  
#Summary of the Normalized training data   
summary(train\_norm\_df)

## Age Experience Income Family   
## Min. :-1.96620 Min. :-2.028011 Min. :-1.4337 Min. :-1.2072   
## 1st Qu.:-0.91026 1st Qu.:-0.887141 1st Qu.:-0.7553 1st Qu.:-1.2072   
## Median : 0.05769 Median :-0.009548 Median :-0.2081 Median :-0.3442   
## Mean : 0.00000 Mean : 0.000000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.84965 3rd Qu.: 0.868044 3rd Qu.: 0.5360 3rd Qu.: 1.3818   
## Max. : 1.90559 Max. : 2.008915 Max. : 3.2937 Max. : 1.3818   
## CCAvg Mortgage Personal.Loan Securities.Account  
## Min. :-1.1117 Min. :-0.5478 No :2271 Min. :0.000   
## 1st Qu.:-0.7101 1st Qu.:-0.5478 Yes: 229 1st Qu.:0.000   
## Median :-0.2510 Median :-0.5478 Median :0.000   
## Mean : 0.0000 Mean : 0.0000 Mean :0.106   
## 3rd Qu.: 0.3801 3rd Qu.: 0.4302 3rd Qu.:0.000   
## Max. : 4.6261 Max. : 5.7888 Max. :1.000   
## CD.Account Online CreditCard Education\_1   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :1.0000 Median :0.0000 Median :0.0000   
## Mean :0.0576 Mean :0.6004 Mean :0.2976 Mean :0.4268   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Education\_2 Education\_3   
## Min. :0.00 Min. :0.0000   
## 1st Qu.:0.00 1st Qu.:0.0000   
## Median :0.00 Median :0.0000   
## Mean :0.28 Mean :0.2932   
## 3rd Qu.:1.00 3rd Qu.:1.0000   
## Max. :1.00 Max. :1.0000

var(train\_norm\_df[, 1:6])

## Age Experience Income Family CCAvg  
## Age 1.000000000 0.993875393 -0.07315732 -0.05288650 -0.08008489  
## Experience 0.993875393 1.000000000 -0.06362619 -0.05662717 -0.07872158  
## Income -0.073157322 -0.063626187 1.00000000 -0.15774567 0.64293801  
## Family -0.052886503 -0.056627166 -0.15774567 1.00000000 -0.10730808  
## CCAvg -0.080084888 -0.078721584 0.64293801 -0.10730808 1.00000000  
## Mortgage -0.009330027 -0.006271337 0.18868170 -0.02548424 0.08823603  
## Mortgage  
## Age -0.009330027  
## Experience -0.006271337  
## Income 0.188681704  
## Family -0.025484237  
## CCAvg 0.088236029  
## Mortgage 1.000000000

#Summary of the Normalized Validation data   
summary(Val\_norm\_df)

## Age Experience Income Family   
## Min. :-1.966201 Min. :-1.940252 Min. :-1.43373 Min. :-1.20717   
## 1st Qu.:-0.910258 1st Qu.:-0.887141 1st Qu.:-0.75526 1st Qu.:-1.20717   
## Median :-0.030306 Median :-0.009548 Median :-0.22999 Median :-0.34416   
## Mean : 0.005479 Mean : 0.008589 Mean : 0.00752 Mean :-0.02946   
## 3rd Qu.: 0.849647 3rd Qu.: 0.868044 3rd Qu.: 0.54150 3rd Qu.: 0.51884   
## Max. : 1.905589 Max. : 1.833396 Max. : 2.87786 Max. : 1.38184   
## CCAvg Mortgage Personal.Loan Securities.Account  
## Min. :-1.1117 Min. :-0.54777 No :1357 Min. :0.0000   
## 1st Qu.:-0.7101 1st Qu.:-0.54777 Yes: 143 1st Qu.:0.0000   
## Median :-0.2510 Median :-0.54777 Median :0.0000   
## Mean :-0.0057 Mean : 0.04763 Mean :0.1047   
## 3rd Qu.: 0.3227 3rd Qu.: 0.48006 3rd Qu.:0.0000   
## Max. : 4.0524 Max. : 5.44953 Max. :1.0000   
## CD.Account Online CreditCard Education\_1   
## Min. :0.00000 Min. :0.0000 Min. :0.000 Min. :0.000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.00000 Median :1.0000 Median :0.000 Median :0.000   
## Mean :0.06333 Mean :0.5867 Mean :0.302 Mean :0.422   
## 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.00000 Max. :1.0000 Max. :1.000 Max. :1.000   
## Education\_2 Education\_3   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000   
## Mean :0.2707 Mean :0.3073   
## 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000

var(Val\_norm\_df[, 1:6])

## Age Experience Income Family CCAvg  
## Age 1.034390729 1.020937362 -0.02152102 -0.08229513 -0.009366446  
## Experience 1.020937362 1.018204680 -0.01638716 -0.09185600 -0.007441807  
## Income -0.021521017 -0.016387163 1.02210143 -0.14263613 0.644671543  
## Family -0.082295131 -0.091855996 -0.14263613 0.96165678 -0.108245622  
## CCAvg -0.009366446 -0.007441807 0.64467154 -0.10824562 1.006053439  
## Mortgage -0.006476600 -0.005513505 0.23502186 -0.01098045 0.122849332  
## Mortgage  
## Age -0.006476600  
## Experience -0.005513505  
## Income 0.235021855  
## Family -0.010980453  
## CCAvg 0.122849332  
## Mortgage 1.103049365

#Summary of the Normalized Test data   
summary(Test\_norm\_df)

## Age Experience Income Family   
## Min. :-1.96620 Min. :-2.028011 Min. :-1.43373 Min. :-1.20717   
## 1st Qu.:-0.91026 1st Qu.:-0.887141 1st Qu.:-0.77715 1st Qu.:-1.20717   
## Median :-0.03031 Median :-0.009548 Median :-0.22999 Median :-0.34416   
## Mean :-0.01086 Mean :-0.014726 Mean : 0.01781 Mean : 0.03383   
## 3rd Qu.: 0.84965 3rd Qu.: 0.780285 3rd Qu.: 0.60169 3rd Qu.: 0.51884   
## Max. : 1.90559 Max. : 2.008915 Max. : 2.83409 Max. : 1.38184   
## CCAvg Mortgage Personal.Loan Securities.Account  
## Min. :-1.111724 Min. :-0.547768 No :892 Min. :0.0   
## 1st Qu.:-0.727287 1st Qu.:-0.547768 Yes:108 1st Qu.:0.0   
## Median :-0.193665 Median :-0.547768 Median :0.0   
## Mean : 0.009749 Mean : 0.008684 Mean :0.1   
## 3rd Qu.: 0.322744 3rd Qu.: 0.460098 3rd Qu.:0.0   
## Max. : 4.626149 Max. : 5.329786 Max. :1.0   
## CD.Account Online CreditCard Education\_1   
## Min. :0.000 Min. :0.000 Min. :0.000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.000 Median :1.000 Median :0.000 Median :0.000   
## Mean :0.063 Mean :0.603 Mean :0.273 Mean :0.396   
## 3rd Qu.:0.000 3rd Qu.:1.000 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.000 Max. :1.000 Max. :1.000   
## Education\_2 Education\_3   
## Min. :0.000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.000 Median :0.000   
## Mean :0.297 Mean :0.307   
## 3rd Qu.:1.000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.000

var(Test\_norm\_df[, 1:6])

## Age Experience Income Family CCAvg  
## Age 1.03772413 1.03224255 -0.0656540 0.02434046 -0.04872649  
## Experience 1.03224255 1.03880649 -0.0524094 0.01818293 -0.04458844  
## Income -0.06565400 -0.05240940 1.0440164 -0.17815927 0.69009667  
## Family 0.02434046 0.01818293 -0.1781593 0.96183186 -0.11281821  
## CCAvg -0.04872649 -0.04458844 0.6900967 -0.11281821 1.02068200  
## Mortgage -0.03152048 -0.03064381 0.2333915 -0.02097509 0.15504543  
## Mortgage  
## Age -0.03152048  
## Experience -0.03064381  
## Income 0.23339149  
## Family -0.02097509  
## CCAvg 0.15504543  
## Mortgage 0.99631223

## KNN prediction for Training Set of data  
knn\_train\_data <- knn(train=train\_norm\_df[,-c(7)],test=train\_norm\_df[,-c(7)],cl=train\_norm\_df[,c(7)], k=3, prob=TRUE)  
  
## KNN prediction for Vailidation Set of data  
knn\_val\_data<- knn(train=train\_norm\_df[,-c(7)],test=Val\_norm\_df[,-c(7)],cl=train\_norm\_df[,c(7)],k=3, prob=TRUE)  
  
## KNN prediction for Test Set of data  
knn\_test\_data<- knn(train=train\_norm\_df[,-c(7)],test=Test\_norm\_df[,-c(7)],cl=train\_norm\_df[,c(7)],k=3, prob=TRUE)  
  
#display the confusion matrices for Train data set  
confusionMatrix(knn\_train\_data,train\_norm\_df[,c(7)], positive="Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2267 60  
## Yes 4 169  
##   
## Accuracy : 0.9744   
## 95% CI : (0.9674, 0.9802)  
## No Information Rate : 0.9084   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8272   
##   
## Mcnemar's Test P-Value : 6.199e-12   
##   
## Sensitivity : 0.7380   
## Specificity : 0.9982   
## Pos Pred Value : 0.9769   
## Neg Pred Value : 0.9742   
## Prevalence : 0.0916   
## Detection Rate : 0.0676   
## Detection Prevalence : 0.0692   
## Balanced Accuracy : 0.8681   
##   
## 'Positive' Class : Yes   
##

#display the confusion matrices for Validation data set  
confusionMatrix(knn\_val\_data,Val\_norm\_df[,c(7)], positive="Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1352 51  
## Yes 5 92  
##   
## Accuracy : 0.9627   
## 95% CI : (0.9518, 0.9717)  
## No Information Rate : 0.9047   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7472   
##   
## Mcnemar's Test P-Value : 1.817e-09   
##   
## Sensitivity : 0.64336   
## Specificity : 0.99632   
## Pos Pred Value : 0.94845   
## Neg Pred Value : 0.96365   
## Prevalence : 0.09533   
## Detection Rate : 0.06133   
## Detection Prevalence : 0.06467   
## Balanced Accuracy : 0.81984   
##   
## 'Positive' Class : Yes   
##

#display the confusion matrices for Test data set  
confusionMatrix(knn\_test\_data,Test\_norm\_df[,c(7)], positive="Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 891 39  
## Yes 1 69  
##   
## Accuracy : 0.96   
## 95% CI : (0.9459, 0.9713)  
## No Information Rate : 0.892   
## P-Value [Acc > NIR] : 4.095e-15   
##   
## Kappa : 0.7544   
##   
## Mcnemar's Test P-Value : 4.909e-09   
##   
## Sensitivity : 0.6389   
## Specificity : 0.9989   
## Pos Pred Value : 0.9857   
## Neg Pred Value : 0.9581   
## Prevalence : 0.1080   
## Detection Rate : 0.0690   
## Detection Prevalence : 0.0700   
## Balanced Accuracy : 0.8189   
##   
## 'Positive' Class : Yes   
##

#Upon Observation the accuracy levels are as as Training data set = 97.44%, Test data set 96.27% and Validation data set = 96%, It seems that the model has high relative accuracy, specificity and low sensitivity   
#i.e This means the model is able to identify negative case more accurately than the positive cases.  
#i.e The accuracy levels of all the data sets are similar hence we can analyze that we have achieved the optimal value of K  
#i.e Overall, the model appears to perform well on all three datasets, with high accuracy and other evaluation metrics indicating good performance.