Assignment 2

Snehitha Anpur

2022-10-01

Import the Universal Bank data set into the working environment:

```
library(readr)
univbank <- read.csv("UniversalBank.csv")
summary(univbank)
```

```
##        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  
```

Check if the data set has any null values:

```
any(is.na(univbank))
```

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

Prepare the data set according to the requirements given in the problem statement:

```
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
```

```
m_univbank <- select(univbank,-ID,-ZIP.Code) # Select the required variables

class(m_univbank$Education) = "character" # Convert the class of Education to character as it is in num
class(m_univbank$Education)
```

```
## [1] "character"
```

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

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

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

```
#Create dummy Variables for the categorical variables where the levels are more than two

dummyModel <- dummyVars(~Education,data=m_univbank) # create the model using dummyVars in Caret package
educationDummy <- predict(dummyModel,m_univbank)  # apply it to the data set
head(educationDummy)
```

```
##   Education1 Education2 Education3
## 1          1          0          0
## 2          1          0          0
## 3          1          0          0
## 4          0          1          0
## 5          0          1          0
## 6          0          1          0
```

Append the Education dummy variables to the original data set and remove the numeric Education variable:

```
m_univbank <- select(m_univbank,-Education) # Remove the numeric Education variable

m_univbank_dummy <- cbind(m_univbank[,-13],educationDummy) # Append the dummy variables for education i
head(m_univbank_dummy)
```

```
##   Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account
## 1  25          1     49      4   1.6        0             0                  1
## 2  45         19     34      3   1.5        0             0                  1
## 3  39         15     11      1   1.0        0             0                  0
## 4  35          9    100      1   2.7        0             0                  0
## 5  35          8     45      4   1.0        0             0                  0
## 6  37         13     29      4   0.4      155             0                  0
##   CD.Account Online CreditCard Education1 Education2 Education3
## 1          0      0          0          1          0          0
## 2          0      0          0          1          0          0
## 3          0      0          0          1          0          0
## 4          0      0          0          0          1          0
## 5          0      0          1          0          1          0
## 6          0      1          0          0          1          0
```

```
m_univbank_dummy <- m_univbank_dummy %>% select(Personal.Loan, everything()) # Place the dependent vari
m_univbank_dummy$Personal.Loan = as.factor(m_univbank_dummy$Personal.Loan) # Convert the data type into
head(m_univbank_dummy)
```

```
##   Personal.Loan Age Experience Income Family CCAvg Mortgage Securities.Account
## 1             0  25          1     49      4   1.6        0                  1
## 2             0  45         19     34      3   1.5        0                  1
## 3             0  39         15     11      1   1.0        0                  0
## 4             0  35          9    100      1   2.7        0                  0
## 5             0  35          8     45      4   1.0        0                  0
## 6             0  37         13     29      4   0.4      155                  0
##   CD.Account Online CreditCard Education1 Education2 Education3
## 1          0      0          0          1          0          0
## 2          0      0          0          1          0          0
## 3          0      0          0          1          0          0
## 4          0      0          0          0          1          0
## 5          0      0          1          0          1          0
## 6          0      1          0          0          1          0
```

Split the data into Training and Validation groups:

```
set.seed(46)
Train_Index = createDataPartition(m_univbank_dummy$Personal.Loan,p=0.60, list=FALSE) # 60% of data as T
Train_Data = m_univbank_dummy[Train_Index,]
Validation_Data = m_univbank_dummy[-Train_Index,] # rest as validation

Test_Data <- data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Mortgage=0,SecuritiesAccount=0,

# Check the summary of Train, Validation and Test data sets
summary(Train_Data)
```

```
##  Personal.Loan      Age          Experience        Income           Family     
##  0:2712        Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000  
##  1: 288        1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 38.00   1st Qu.:1.000  
##                Median :45.00   Median :20.00   Median : 64.00   Median :2.000  
##                Mean   :45.37   Mean   :20.18   Mean   : 74.54   Mean   :2.402  
##                3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.:100.00   3rd Qu.:3.000  
##                 Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000
##      CCAvg           Mortgage      Securities.Account   CD.Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.0000     Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.0000     1st Qu.:0.000
##  Median : 1.500   Median :  0.00   Median :0.0000     Median :0.000
##  Mean   : 1.945   Mean   : 56.91   Mean   :0.1043     Mean   :0.059
##  3rd Qu.: 2.500   3rd Qu.:100.00   3rd Qu.:0.0000     3rd Qu.:0.000
##  Max.   :10.000   Max.   :617.00   Max.   :1.0000     Max.   :1.000
##      Online        CreditCard       Education1       Education2
##  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 :1.000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.584   Mean   :0.2833   Mean   :0.4257   Mean   :0.2783
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    Education3
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.296
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
summary(Validation_Data)
```

```
##  Personal.Loan      Age          Experience        Income           Family
##  0:1808        Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1: 192        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   :45.29   Mean   :19.98   Mean   : 72.63   Mean   :2.389
##                3rd Qu.:55.00   3rd Qu.:29.00   3rd Qu.: 94.00   3rd Qu.:3.000
##                Max.   :67.00   Max.   :43.00   Max.   :205.00   Max.   :4.000
##      CCAvg           Mortgage      Securities.Account   CD.Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.0000     Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.0000     1st Qu.:0.0000
##  Median : 1.500   Median :  0.00   Median :0.0000     Median :0.0000
##  Mean   : 1.927   Mean   : 55.88   Mean   :0.1045     Mean   :0.0625
##  3rd Qu.: 2.600   3rd Qu.:101.00   3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   :10.000   Max.   :635.00   Max.   :1.0000     Max.   :1.0000
##      Online        CreditCard     Education1       Education2
##  Min.   :0.000   Min.   :0.00   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.000
##  Median :1.000   Median :0.00   Median :0.0000   Median :0.000
##  Mean   :0.616   Mean   :0.31   Mean   :0.4095   Mean   :0.284
##  3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.00   Max.   :1.0000   Max.   :1.000
##    Education3
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3065
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
summary(Test_Data)
```

```
##       Age       Experience     Income       Family      CCAvg      Mortgage
##  Min.   :40   Min.   :10   Min.   :84   Min.   :2   Min.   :2   Min.   :0  
##  1st Qu.:40   1st Qu.:10   1st Qu.:84   1st Qu.:2   1st Qu.:2   1st Qu.:0  
##  Median :40   Median :10   Median :84   Median :2   Median :2   Median :0  
##  Mean   :40   Mean   :10   Mean   :84   Mean   :2   Mean   :2   Mean   :0  
##  3rd Qu.:40   3rd Qu.:10   3rd Qu.:84   3rd Qu.:2   3rd Qu.:2   3rd Qu.:0  
##  Max.   :40   Max.   :10   Max.   :84   Max.   :2   Max.   :2   Max.   :0  
##  SecuritiesAccount   CDAccount     Online    CreditCard   Education1
##  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  
##    Education2   Education3
##  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  
```

Data sets have to be normalized before starting to process the model.

```
colnames(m_univbank_dummy) # Fetch the column names in the data set
```

```
##  [1] "Personal.Loan"      "Age"                "Experience"        
##  [4] "Income"             "Family"             "CCAvg"             
##  [7] "Mortgage"           "Securities.Account" "CD.Account"        
## [10] "Online"             "CreditCard"         "Education1"        
## [13] "Education2"         "Education3"
```

```
norm_var <- c("Age","Experience","Income","Family","CCAvg","Mortgage") # Get all the numeric Variables

train.norm.df <- Train_Data[,norm_var] # Filter the numeric variables in train data
valid.norm.df <- Validation_Data[,norm_var] # Filter the numeric variables in validation data
test.norm.df <- Test_Data[,norm_var] # Filter the numeric variables in test data

norm.values <- preProcess(Train_Data[,norm_var], method=c("center", "scale")) # Using preProcess find o

train.norm.df <- predict(norm.values,Train_Data)
valid.norm.df <- predict(norm.values, Validation_Data)
test.norm.df <- predict(norm.values, test.norm.df)

# Verify the normalized values
summary(train.norm.df)
```

```
##  Personal.Loan      Age             Experience           Income       
##  0:2712        Min.   :-1.94149   Min.   :-2.01060   Min.   :-1.4226  
##  1: 288        1st Qu.:-0.90009   1st Qu.:-0.88324   1st Qu.:-0.7812
##                Median :-0.03225   Median :-0.01604   Median :-0.2253
##                Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##                3rd Qu.: 0.83558   3rd Qu.: 0.85116   3rd Qu.: 0.5444
##                Max.   : 1.87698   Max.   : 1.97852   Max.   : 3.1956
##      Family            CCAvg            Mortgage       Securities.Account
##  Min.   :-1.2147   Min.   :-1.1060   Min.   :-0.5493   Min.   :0.0000
##  1st Qu.:-1.2147   1st Qu.:-0.7080   1st Qu.:-0.5493   1st Qu.:0.0000
##  Median :-0.3481   Median :-0.2530   Median :-0.5493   Median :0.0000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   :0.1043
##  3rd Qu.: 0.5185   3rd Qu.: 0.3157   3rd Qu.: 0.4159   3rd Qu.:0.0000
##  Max.   : 1.3852   Max.   : 4.5808   Max.   : 5.4062   Max.   :1.0000
##    CD.Account        Online       CreditCard       Education1
##  Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :1.000   Median :0.0000   Median :0.0000
##  Mean   :0.059   Mean   :0.584   Mean   :0.2833   Mean   :0.4257
##  3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##    Education2       Education3
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.000
##  Mean   :0.2783   Mean   :0.296
##  3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.000
```

```
summary(valid.norm.df)
```

```
##  Personal.Loan      Age              Experience          Income
##  0:1808        Min.   :-1.941487   Min.   :-2.01060   Min.   :-1.42261
##  1: 192        1st Qu.:-0.900088   1st Qu.:-0.88324   1st Qu.:-0.75982
##                Median :-0.032254   Median :-0.01604   Median :-0.24669
##                Mean   :-0.007217   Mean   :-0.01743   Mean   :-0.04083
##                3rd Qu.: 0.835579   3rd Qu.: 0.76444   3rd Qu.: 0.41611
##                Max.   : 1.876978   Max.   : 1.97852   Max.   : 2.78933
##      Family             CCAvg              Mortgage         Securities.Account
##  Min.   :-1.21474   Min.   :-1.106033   Min.   :-0.549330   Min.   :0.0000
##  1st Qu.:-1.21474   1st Qu.:-0.707957   1st Qu.:-0.549330   1st Qu.:0.0000
##  Median :-0.34810   Median :-0.253012   Median :-0.549330   Median :0.0000
##  Mean   :-0.01141   Mean   :-0.009912   Mean   :-0.009947   Mean   :0.1045
##  3rd Qu.: 0.51854   3rd Qu.: 0.372537   3rd Qu.: 0.425567   3rd Qu.:0.0000
##  Max.   : 1.38518   Max.   : 4.580776   Max.   : 5.579972   Max.   :1.0000
##    CD.Account         Online        CreditCard     Education1
##  Min.   :0.0000   Min.   :0.000   Min.   :0.00   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.0000
##  Median :0.0000   Median :1.000   Median :0.00   Median :0.0000
##  Mean   :0.0625   Mean   :0.616   Mean   :0.31   Mean   :0.4095
##  3rd Qu.:0.0000   3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.00   Max.   :1.0000
##    Education2      Education3
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.284   Mean   :0.3065  
##  3rd Qu.:1.000   3rd Qu.:1.0000  
##  Max.   :1.000   Max.   :1.0000  
```

```
summary(test.norm.df)
```

```
##       Age              Experience          Income            Family       
##  Min.   :-0.4662   Min.   :-0.8832   Min.   :0.2023   Min.   :-0.3481  
##  1st Qu.:-0.4662   1st Qu.:-0.8832   1st Qu.:0.2023   1st Qu.:-0.3481  
##  Median :-0.4662   Median :-0.8832   Median :0.2023   Median :-0.3481  
##  Mean   :-0.4662   Mean   :-0.8832   Mean   :0.2023   Mean   :-0.3481  
##  3rd Qu.:-0.4662   3rd Qu.:-0.8832   3rd Qu.:0.2023   3rd Qu.:-0.3481  
##  Max.   :-0.4662   Max.   :-0.8832   Max.   :0.2023   Max.   :-0.3481  
##      CCAvg              Mortgage      
##  Min.   :0.03133   Min.   :-0.5493  
##  1st Qu.:0.03133   1st Qu.:-0.5493  
##  Median :0.03133   Median :-0.5493  
##  Mean   :0.03133   Mean   :-0.5493  
##  3rd Qu.:0.03133   3rd Qu.:-0.5493  
##  Max.   :0.03133   Max.   :-0.5493  
```

Model 1: using knn method in train method in Caret package

```
#train.norm.df$Personal.Loan = as.factor(train.norm.df$Personal.Loan)

set.seed(624)
searchGrid <- expand.grid(k=seq(1:30))
model <- train(Personal.Loan~.,data=train.norm.df,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  0.9588598  0.7346355
##    2  0.9534606  0.6941981
##    3  0.9525164  0.6818184
##    4  0.9513938  0.6677100
##    5  0.9514967  0.6606969
##    6  0.9508568  0.6505496
##    7  0.9500611  0.6394897
##    8  0.9497680  0.6342322
##    9  0.9490110  0.6255261
##   10  0.9482269  0.6162504
##   11  0.9475511  0.6088863
##   12  0.9469691  0.6017367
##   13  0.9461329  0.5928458
##   14  0.9461817  0.5922214
##   15  0.9446334  0.5759140
##   16  0.9434311  0.5626411
##   17  0.9420178  0.5471427
##   18  0.9414745  0.5407819
##   19  0.9402840  0.5286259
##   20  0.9395610  0.5199144
##   21  0.9385489  0.5092487
##   22  0.9373481  0.4970583
##   23  0.9367744  0.4899727
##   24  0.9365184  0.4865502
##   25  0.9360190  0.4806556
##   26  0.9354693  0.4737199
##   27  0.9348545  0.4662428
##   28  0.9345307  0.4600949
##   29  0.9337360  0.4502829
##   30  0.9331623  0.4433681
##
## 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]] # saves the best k
best_k # Here the best k turned out to be 1 using the training data
```

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

Model 2: using knn function in class package

```
library(class)
Train_Predictors <- select(train.norm.df,-Personal.Loan)
Test_Predictors <- cbind(test.norm.df,Test_Data[,7:13])
Valid_Predictors <- select(valid.norm.df,-Personal.Loan)

Train_Labels <- train.norm.df[,1]
Valid_Labels <- valid.norm.df[,1]

Predicted_Valid_Labels <- knn(Train_Predictors,Valid_Predictors,cl = Train_Labels,k=1)

head(Predicted_Valid_Labels)
```

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

```
Predicted_Test_Labels <- knn(Train_Predictors,Test_Predictors,cl = Train_Labels,k=1)

head(Predicted_Test_Labels) # For the given test data the model gave a result that the Customer would n
```

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

Answer 1: For the given test data the model gave a result that the Customer would not apply for Personal Loan

```
library(caret)
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
# compute knn for different k on validation.
for(i in 1:14) {
  knn.pred <- knn(Train_Predictors,Valid_Predictors,cl = Train_Labels,k=i)
  accuracy.df[i, 2] <- confusionMatrix(knn.pred, Valid_Labels)$overall[1]
}
accuracy.df
```

```
##     k accuracy
## 1   1   0.9630
## 2   2   0.9555
## 3   3   0.9640
## 4   4   0.9620
## 5   5   0.9600
## 6   6   0.9565
## 7   7   0.9575
## 8   8   0.9560
## 9   9   0.9540
## 10 10   0.9530
## 11 11   0.9535
## 12 12   0.9510
## 13 13   0.9510
## 14 14   0.9505
```

Answer 2: Based on the above result the best k for this data set is 3 as it has the highest accuracy of 96.40%

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

Predicted_Valid_Labels <- knn(Train_Predictors,Valid_Predictors,cl = Train_Labels,k=3)

head(Predicted_Valid_Labels)
```

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

```
CrossTable(x = Valid_Labels,y = Predicted_Valid_Labels,prop.chisq = FALSE)
```

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

```
##
## Total Observations in Table:  2000
##
##
##              | Predicted_Valid_Labels
## Valid_Labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1805 |         3 |      1808 |
##              |     0.998 |     0.002 |     0.904 |
##              |     0.963 |     0.024 |           |
##              |     0.902 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |        69 |       123 |       192 |
##              |     0.359 |     0.641 |     0.096 |
##              |     0.037 |     0.976 |           |
##              |     0.034 |     0.061 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1874 |       126 |      2000 |
##              |     0.937 |     0.063 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Answer 3:

Using k=3, above result shows the confusion matrix of validation data set

```
Predicted_Test_Labels <- knn(Train_Predictors,Test_Predictors,cl = Train_Labels,k=3)

head(Predicted_Test_Labels) # For the given test data the model gave a result that the Customer would n
```

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

Answer 4:

Based on k=3 which is the best k value, the model gave a result that the Customer would not apply for Personal Loan

Now, split the data into train, validation and test data sets by the proportions of 50%, 30% and 20% respectively

```
#install.packages("splitTools")
#install.packages("ranger")
library(splitTools)
library(ranger)

# Split data into partitions
set.seed(5346)
inds <- partition(m_univbank_dummy$Age, p = c(train = 0.5, valid = 0.3, test = 0.2))
str(inds)
```

```
## List of 3
##  $ train: int [1:2497] 1 4 8 10 14 16 18 19 20 24 ...
##  $ valid: int [1:1502] 2 3 6 11 13 15 17 22 27 29 ...
##  $ test : int [1:1001] 5 7 9 12 21 23 26 28 45 48 ...
```

```r
train_ub <- m_univbank_dummy[inds$train, ]
valid_ub <- m_univbank_dummy[inds$valid, ]
test_ub <- m_univbank_dummy[inds$test, ]
```

Normalize the data using train data set:

```r
#norm_var <- c("Age","Experience","Income","Family","CCAvg","Mortgage") # Get all the numeric Variables

train.norm.ub.df <- train_ub[,norm_var] # Filter the numeric variables in train data
valid.norm.ub.df <- valid_ub[,norm_var] # Filter the numeric variables in validation data
test.norm.ub.df <- test_ub[,norm_var] # Filter the numeric variables in test data

norm.values.ub <- preProcess(train_ub[,norm_var], method=c("center", "scale")) # Using preProcess find

train.norm.ub.df <- predict(norm.values.ub,train_ub)
valid.norm.ub.df <- predict(norm.values.ub, valid_ub)
test.norm.ub.df <- predict(norm.values.ub, test_ub)

# Verify the normalized values
summary(train.norm.ub.df)
```

```
##  Personal.Loan      Age             Experience          Income
##  0:2258        Min.   :-1.95294   Min.   :-2.0184   Min.   :-1.4272
##  1: 239        1st Qu.:-0.90478   1st Qu.:-0.8846   1st Qu.:-0.7594
##                Median :-0.03131   Median :-0.0125   Median :-0.2208
##                Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##                3rd Qu.: 0.84216   3rd Qu.: 0.8596   3rd Qu.: 0.5333
##                Max.   : 1.89032   Max.   : 1.9933   Max.   : 3.0970
##     Family           CCAvg            Mortgage       Securities.Account
##  Min.   :-1.1842   Min.   :-1.1097   Min.   :-0.5496   Min.   :0.0000
##  1st Qu.:-1.1842   1st Qu.:-0.7140   1st Qu.:-0.5496   1st Qu.:0.0000
##  Median :-0.3188   Median :-0.2052   Median :-0.5496   Median :0.0000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   :0.1017
##  3rd Qu.: 0.5465   3rd Qu.: 0.3600   3rd Qu.: 0.4413   3rd Qu.:0.0000
##  Max.   : 1.4119   Max.   : 4.5431   Max.   : 5.5639   Max.   :1.0000
##    CD.Account          Online          CreditCard       Education1
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.05847   Mean   :0.5927   Mean   :0.2799   Mean   :0.4301
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    Education2       Education3
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.000
##  Mean   :0.2679   Mean   :0.302
##  3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.000
```

```r
summary(valid.norm.ub.df)
```

```
##  Personal.Loan      Age              Experience            Income
##  0:1353        Min.   :-1.952939   Min.   :-2.018356   Min.   :-1.42725
##  1: 149        1st Qu.:-0.904776   1st Qu.:-0.884613   1st Qu.:-0.75938
##                Median :-0.031308   Median :-0.012504   Median :-0.19924
##                Mean   :-0.002056   Mean   :-0.004897   Mean   :-0.01682
##                3rd Qu.: 0.842161   3rd Qu.: 0.859606   3rd Qu.: 0.51172
##                Max.   : 1.890324   Max.   : 1.993348   Max.   : 2.79538
##      Family           CCAvg            Mortgage       Securities.Account
##  Min.   :-1.18421   Min.   :-1.10969   Min.   :-0.54956   Min.   :0.0000
##  1st Qu.:-1.18421   1st Qu.:-0.71400   1st Qu.:-0.54956   1st Qu.:0.0000
##  Median :-0.31884   Median :-0.20524   Median :-0.54956   Median :0.0000
##  Mean   : 0.05162   Mean   :-0.02032   Mean   : 0.03833   Mean   :0.1119
##  3rd Qu.: 1.41190   3rd Qu.: 0.30351   3rd Qu.: 0.47099   3rd Qu.:0.0000
##  Max.   : 1.41190   Max.   : 4.54312   Max.   : 5.74223   Max.   :1.0000
##    CD.Account          Online         CreditCard       Education1
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.06924   Mean   :0.5912   Mean   :0.3149   Mean   :0.4095
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    Education2       Education3
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2909   Mean   :0.2996
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
summary(test.norm.ub.df)
```

```
##  Personal.Loan      Age              Experience            Income
##  0:909         Min.   :-1.865592   Min.   :-2.018356   Min.   :-1.42725
##  1: 92         1st Qu.:-0.904776   1st Qu.:-0.884613   1st Qu.:-0.78093
##                Median :-0.031308   Median :-0.012504   Median :-0.28541
##                Mean   :-0.005653   Mean   :-0.009541   Mean   :-0.02574
##                3rd Qu.: 0.842161   3rd Qu.: 0.859606   3rd Qu.: 0.51172
##                Max.   : 1.890324   Max.   : 1.906138   Max.   : 3.22627
##      Family           CCAvg            Mortgage         Securities.Account
##  Min.   :-1.18421   Min.   :-1.10969   Min.   :-0.549565   Min.   :0.0000
##  1st Qu.:-1.18421   1st Qu.:-0.77053   1st Qu.:-0.549565   1st Qu.:0.0000
##  Median :-0.31884   Median :-0.26177   Median :-0.549565   Median :0.0000
##  Mean   : 0.04339   Mean   :-0.04052   Mean   :-0.006349   Mean   :0.0999
##  3rd Qu.: 0.54653   3rd Qu.: 0.30351   3rd Qu.: 0.421451   3rd Qu.:0.0000
##  Max.   : 1.41190   Max.   : 3.97784   Max.   : 5.514336   Max.   :1.0000
##    CD.Account          Online         CreditCard       Education1
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.05195   Mean   :0.6154   Mean   :0.2977   Mean   :0.4066
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    Education2       Education3
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2967   Mean   :0.2967
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
Train_Predictors_Ub <- select(train.norm.ub.df,-Personal.Loan)
Valid_Predictors_Ub <- select(valid.norm.ub.df,-Personal.Loan)
Test_Predictors_Ub <- select(test.norm.ub.df,-Personal.Loan)

Train_Labels_Ub <- train.norm.ub.df[,1]
Valid_Labels_Ub <- valid.norm.ub.df[,1]
Test_Labels_Ub <- test.norm.ub.df[,1]

Predicted_Train_Labels_Ub <- knn(Train_Predictors_Ub,Train_Predictors_Ub,cl = Train_Labels_Ub,k=3)

head(Predicted_Train_Labels_Ub)
```

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

```
Predicted_Valid_Labels_Ub <- knn(Train_Predictors_Ub,Valid_Predictors_Ub,cl = Train_Labels_Ub,k=3)

head(Predicted_Valid_Labels_Ub)
```

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

```
Predicted_Test_Labels_Ub <- knn(Train_Predictors_Ub,Test_Predictors_Ub,cl = Train_Labels_Ub,k=3)

head(Predicted_Test_Labels_Ub)
```

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

```
confusionMatrix(Predicted_Train_Labels_Ub,Train_Labels_Ub,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2257   47
##          1    1  192
##
##                Accuracy : 0.9808
##                  95% CI : (0.9746, 0.9858)
##     No Information Rate : 0.9043
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8785
##
##  Mcnemar's Test P-Value : 8.293e-11
```

```
##
##             Sensitivity : 0.80335
##             Specificity : 0.99956
##          Pos Pred Value : 0.99482
##          Neg Pred Value : 0.97960
##              Prevalence : 0.09571
##          Detection Rate : 0.07689
##    Detection Prevalence : 0.07729
##       Balanced Accuracy : 0.90145
##
##        ’Positive’ Class : 1
##
```

```
confusionMatrix(Predicted_Valid_Labels_Ub,Valid_Labels_Ub,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1351   58
##          1    2   91
##
##                Accuracy : 0.9601
##                  95% CI : (0.9489, 0.9694)
##     No Information Rate : 0.9008
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7316
##
##  Mcnemar’s Test P-Value : 1.243e-12
##
##             Sensitivity : 0.61074
##             Specificity : 0.99852
##          Pos Pred Value : 0.97849
##          Neg Pred Value : 0.95884
##              Prevalence : 0.09920
##          Detection Rate : 0.06059
##    Detection Prevalence : 0.06192
##       Balanced Accuracy : 0.80463
##
##        ’Positive’ Class : 1
##
```

```
confusionMatrix(Predicted_Test_Labels_Ub,Test_Labels_Ub,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 907  33
##          1   2  59
##
##                Accuracy : 0.965
##                  95% CI : (0.9517, 0.9755)
##     No Information Rate : 0.9081
##     P-Value [Acc > NIR] : 1.581e-12
##
##                   Kappa : 0.7532
##
##  Mcnemar's Test P-Value : 3.959e-07
##
##             Sensitivity : 0.64130
##             Specificity : 0.99780
##          Pos Pred Value : 0.96721
##          Neg Pred Value : 0.96489
##              Prevalence : 0.09191
##          Detection Rate : 0.05894
##    Detection Prevalence : 0.06094
##       Balanced Accuracy : 0.81955
##
##        'Positive' Class : 1
##
```

Answer 5: From the above confusion matrices of train, validation and test data sets, it can be seen that the accuracy of test data is 96.5%. It is between the accuracy of train data sets (98.08%) and accuracy of validation (96.01%). The reason for more accuracy in the training data is that the model was built on it whereas the actual accuracy is identified while the model is tested on validation and test data.