FML_Assignment2

Vivek Pamulaparthi

2023-09-30

SUMMARY

Questions & Answers

1. Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1, and Credit Card = 1. Perform a k-NN classification with all predictors except ID and ZIP code using k = 1. Remember to transform categorical predictors with more than two categories into dummy variables first. Specify the success class as 1 (loan acceptance), and use the default cutoff value of 0.5. How would this customer be classified?

Ans: This new customer is classified as 0. The model predicted that the customer would not apply for a personal loan based on the test data that was provided. This new customer would be classified as 0, does not take the personal loan

2. What is a choice of k that balances between overfitting and ignoring the predictor information?

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

3. Show the confusion matrix for the validation data that results from using the best k.

Ans: Using k=3 as we got the best value of K as 3 we got the confusion matrix with True Negative= 1805, True Positive= 123, False Positive= 69 and False Negative= 3 with accuracy of 0.964 and other parameters.

4. Consider the following customer: Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1 and Credit Card = 1. Classify the customer using the best k.

Ans: The model predicted that the customer would not apply for a personal loan based on the best k value, which was determined to be 3.

5. Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%). Apply the k-NN method with the k chosen above. Compare the confusion matrix of the test set with that of the training and validation sets. Comment on the differences and their reason.

Ans: a. The accuracy of the training set is higher than the validation and testing sets with 98.08%, this does not guarantee that the model will apply well to fresh data, either.

b. The k-NN technique and the choice of k are both quite straightforward, which might aid in the model's successful generalization. Larger variances might be seen in more complicated models.

c. The model's performance should remain constant across these sets if the training, validation, and test sets are all representative of the same underlying data distribution and have comparable data quality.

d. A large percentage of the true negatives in the training set demonstrate how well the model shows the non-acceptance for clients who rejected the loan during training.

e. The Test set data is quite similar with the validation set, the number of true positives in the test set is similarly lower than in the training set. This suggests that when the model is applied to the test set, its performance remains consistent.

f. Because it reflects completely fresh, previously unobserved data, the confusion matrix on the test set is the most accurate gauge of your model's performance in real-world scenarios. Any variations or discrepancies between the training and validation sets and the test set are a sign of how well your model generalizes. Here we got not too high or low values in the matrix when compared with Training and Validation sets.

g. Similar model performance would arise if the correlations between client features and loan acceptance were the same across all three groups.

Problem Statement

Universal bank is a young bank growing rapidly in terms of overall customer acquisition. The majority of these customers are liability customers (depositors) with varying sizes of relationship with the bank. The customer base of asset customers (borrowers) is quite small, and the bank is interested in expanding this base rapidly in more loan business. In particular, it wants to explore ways of converting its liability customers to personal loan customers.

A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise smarter campaigns with better target marketing. 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.

The file UniversalBank.csv contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

Partition the data into training (60%) and validation (40%) sets

---

Initially, load all the required libraries

```
library(class)
library(caret)
```

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

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

```
library(e1071)
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
```

#I loaded the UniversalBank dataset given in the assignment, read the data and returning the dimensions of the new dataset.

```
library(readr)
UniBk.df <- read.csv("UniversalBank.csv")
dim(UniBk.df)
```

```
## [1] 5000   14
```

```
any(is.na(UniBk.df))
```

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

```
View(UniBk.df)
```

Removing the columns ID and ZIP from the new dataset created.

```
UniBk.df <- UniBk.df[,-c(1,5)]
```

#After removing the ID and Zip Code Columns:

```
View(UniBk.df)
```

```
class(UniBk.df$Education) = "character"
class(UniBk.df$Education)
```

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

```
dummyMod <- dummyVars(~Education,data=UniBk.df)
eduDummy <- predict(dummyMod,UniBk.df)  # apply it to the data set
head(eduDummy)
```

```
##   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
```

```
UniBk.df <- select(UniBk.df,-Education)
UniBk.df_dummy <- cbind(UniBk.df[,-13],eduDummy) # Add the education dummy variables to the original da
head(UniBk.df_dummy)   #Here we are printing the 1st few rows of the dataset : UniBk.df_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
```

```
UniBk.df_dummy <- UniBk.df_dummy %>% select(Personal.Loan, everything()) # To use the dependent variabl
UniBk.df_dummy$Personal.Loan = as.factor(UniBk.df_dummy$Personal.Loan) # Converting the Personal.Loan c
head(UniBk.df_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
```

Question 1

Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 =1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account =

0, Online = 1, and Credit Card = 1. Perform a k-NN classification with all predictors except ID and ZIP code using k = 1. Remember to transform categorical predictors with more than two categories into dummy variables first. Specify the success class as 1 (loan acceptance), and use the default cutoff value of 0.5. How would this customer be classified?

#Converting the entire dataset into two parts: 60% Training set and 40% Validation set

```
set.seed(46)
Training_Index = createDataPartition(UniBk.df_dummy$Personal.Loan,p=0.60, list=FALSE) # Training Set(60
Training_Data = UniBk.df_dummy[Training_Index,]
Validate_Data = UniBk.df_dummy[-Training_Index,] # Validation(40%)
Testing_Data <- data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Mortgage=0,SecuritiesAccount=
```

#Printing the summary of Training Data, Validation Data and Testing Data

```
summary(Training_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(Validate_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(Testing_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
```

```
colnames(UniBk.df_dummy)
```

```
##  [1] "Personal.Loan"      "Age"                "Experience"
##  [4] "Income"             "Family"             "CCAvg"
##  [7] "Mortgage"           "Securities.Account" "CD.Account"
```

```
## [10] "Online"             "CreditCard"         "Education1"
## [13] "Education2"         "Education3"
```

#Normalizing the data:

```
normal_var <- c("Age","Experience","Income","Family","CCAvg","Mortgage") # Getting the numeric Variable
training_labels <- Training_Data[,normal_var] # In Training data, filtering the numerical variables.
validate_labels <- Validate_Data[,normal_var] # In Validation data, filtering the numerical variables.
testing_normalize <- Testing_Data[,normal_var] # In Testing data, filtering the numerical variables.
normalize_data <- preProcess(Training_Data[,normal_var], method=c("center", "scale")) # Discovering the

training_labels <- predict(normalize_data,Training_Data)
validate_labels <- predict(normalize_data, Validate_Data)
testing_normalize <- predict(normalize_data, testing_normalize)
```

#Summary of Training, Validation, Testing Tables after Normalizing the data

```
summary(training_labels)
```

```
##  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(validate_labels)
```

```
##  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(testing_normalize)
```

```
##       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
```

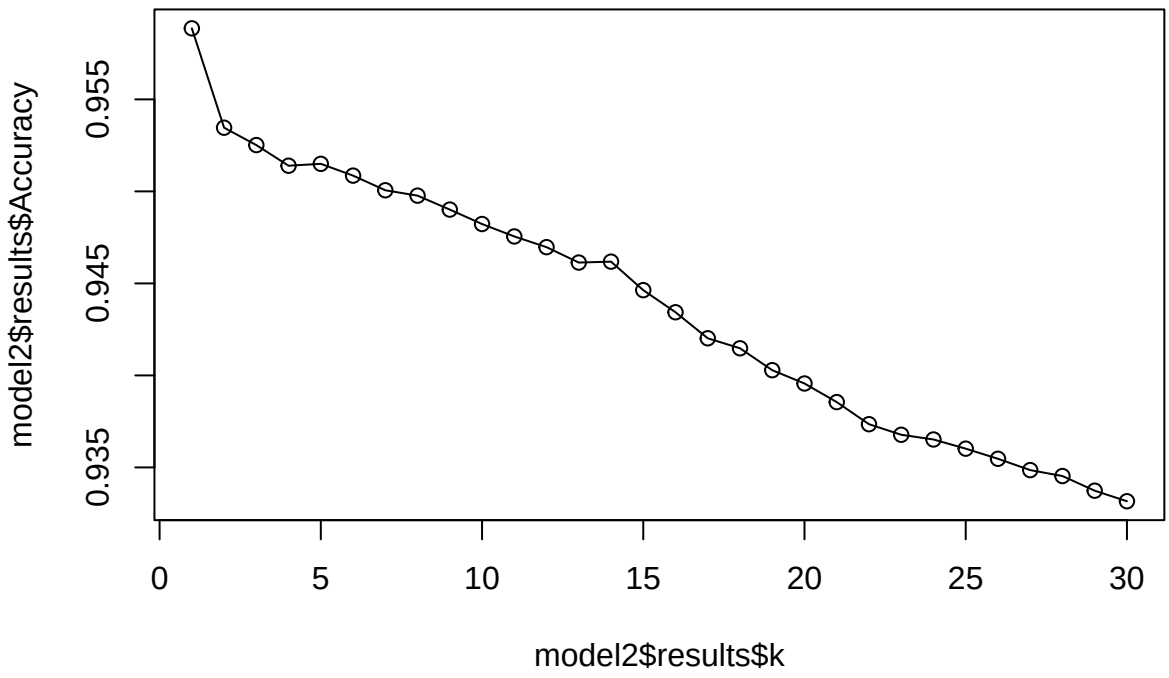
Model: Using knn method for the train method using Caret package

```
set.seed(624)
Grd <- expand.grid(k=seq(1:30))
model2 <- train(Personal.Loan~.,data=training_labels,method="knn",tuneGrid=Grd)
model2
```

```
## 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.
```

```
plot(model2$results$k,model2$results$Accuracy, type = 'o')
```




```
Ideal_k <- model2$bestTune[[1]] # saves the best k
Ideal_k # Here the best k turned out to be 1 using the training data
```

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

Model 2: From the Class Package, we now use the KNN function.

```
library(class)
Training_Predictors <- select(training_labels,-Personal.Loan)
Testing_Predictors <- cbind(testing_normalize,Testing_Data[,7:13])
Validate_Predictors <- select(validate_labels,-Personal.Loan)
Training_Labels <- training_labels[,1]
Validate_Labels <- validate_labels[,1]

#Now Predicting using KNN model

Predicted_Validate_Labels <- knn(Training_Predictors,Validate_Predictors,cl = Training_Labels,k=1)
head(Predicted_Validate_Labels)
```

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

```
Predicted_Testing_Labels <- knn(Training_Predictors,Testing_Predictors,cl = Training_Labels,k=1)
head(Predicted_Testing_Labels)
```

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

1: The model predicted that the customer would not apply for a personal loan based on the test data that was provided.

Question 2

What is a choice of k that balances between overfitting and ignoring the predictor information?

```
exact <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))

for(i in 1:14) {
  knn.predict <- knn(Training_Predictors,Validate_Predictors,cl = Training_Labels,k=i)
  exact[i, 2] <- confusionMatrix(knn.predict, Validate_Labels)$overall[1]
}
exact
```

```
##     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
```

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

Question 3

Show the confusion matrix for the validation data that results from using the best k.

```
#Installed the library gmodels using the console
library(gmodels)
Predicted_Validate_Labels <- knn(Training_Predictors,Validate_Predictors,cl = Training_Labels,k=3)
head(Predicted_Validate_Labels)
```

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

Confusion Matrix for the validation data

```
CrossTable(x = Validate_Labels,y = Predicted_Validate_Labels,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##                  | Predicted_Validate_Labels
## Validate_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 |           |
## ----------------|-----------|-----------|-----------|
##
##
```

3: Using k=3 as we got the best value of K as 3, the above created confusion represents the confusion matrix for the validation data. ***

Question 4

Consider the following customer: Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1 and Credit Card = 1. Classify the customer using the best k.

```
Predicted_Testing_Labels <- knn(Training_Predictors,Testing_Predictors,cl = Training_Labels,k=3)
head(Predicted_Testing_Labels)
```

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

4: The model predicted that the customer would not apply for a personal loan based on the best k value, which was determined to be 3. ***

Question 5

Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%). Apply the k-NN method with the k chosen above. Compare the confusion matrix of the test set with that of the training and validation sets. Comment on the differences and their reason.

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

```
library(splitTools)

#Data should be partitioned
set.seed(5346)
Newdata <- partition(UniBk.df_dummy$Age, p = c(train = 0.5, valid = 0.3, test = 0.2))
str(data)
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL, verbose = getOption("verbose"),
##     envir = .GlobalEnv, overwrite = TRUE)
```

```
training..nm <- UniBk.df_dummy[Newdata$train, ]
validate..nm <- UniBk.df_dummy[Newdata$valid, ]
testing..nm <- UniBk.df_dummy[Newdata$test, ]
```

Normalize the data using train data set:

```
#normal_var <- c("Age","Experience","Income","Family","CCAvg","Mortgage") # Get all the numeric Variabl
training.normal..nm <- training..nm[,normal_var] #In Training data, filtering the numerical variables.
validate.normal..nm <- validate..nm[,normal_var] # In Validation data, filtering the numerical variable
testing.normal..nm <- testing..nm[,normal_var] # In Testing data, filtering the numerical variables.
normalize_data..nm <- preProcess(training..nm[,normal_var], method=c("center", "scale"))
```

```
#Discovering the normalized values of the numerical variables in the train data and use preProcess to a

training.normal..nm <- predict(normalize_data..nm,training..nm)
validate.normal..nm <- predict(normalize_data..nm, validate..nm)
testing.normal..nm <- predict(normalize_data..nm, testing..nm)
```

Normalized value Summary of Training, Validation and Testing Data

```
summary(training.normal..nm)
```

```
##  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
```

```
summary(validate.normal..nm)
```

```
##  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(testing.normal..nm)
```

```
##  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
```

Predicted Values of Training, Validation and Testing data

```
Training_Predictors..nm <- select(training.normal..nm,-Personal.Loan) #Predicting the training values
Validate_Predictors..nm <- select(validate.normal..nm,-Personal.Loan) #Predicting the validation values
Testing_Predictors..nm <- select(testing.normal..nm,-Personal.Loan) ##Predicting the Testing values
Training_Labels_Ub <- training.normal..nm[,1]
Validate_Labels_Ub <- validate.normal..nm[,1]
Testing_Labels_Ub <- testing.normal..nm[,1]

Predicted_Training_Labels_Ub <- knn(Training_Predictors..nm,Training_Predictors..nm,cl = Training_Labels
head(Predicted_Training_Labels_Ub)
```

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

```
Predicted_Validate_Labels_Ub <- knn(Training_Predictors..nm,Validate_Predictors..nm,cl = Training_Labels
head(Predicted_Validate_Labels_Ub)
```

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

```
Predicted_Testing_Labels_Ub <- knn(Training_Predictors..nm,Testing_Predictors..nm,cl = Training_Labels_U
head(Predicted_Testing_Labels_Ub)
```

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

Confusion Matrix for the Training set

```
confusionMatrix(Predicted_Training_Labels_Ub,Training_Labels_Ub,positive = "1") #This displays the conf
```

```
## 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
##
```

Confusion Matrix for the Validation set

```
confusionMatrix(Predicted_Validate_Labels_Ub,Validate_Labels_Ub,positive = "1")  #This displays the con
```

```
## 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
##
```

Confusion Matrix for the Testing set

```
confusionMatrix(Predicted_Testing_Labels_Ub,Testing_Labels_Ub,positive = "1")  #This displays the confu
```

```
## 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
##
```

5:The accuracy of the training set is higher than the validation and testing sets with 98.08%, this does not guarantee that the model will apply well to fresh data, either.
