```r
library(class)
```

```
## Warning: package 'class' was built under R version 4.3.2
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Loading required package: lattice
```

```r
Universal_Bank <- read.csv ("C:\\Users\\pakan\\Desktop\\FML\\UniversalBank.csv")
dim(Universal_Bank)
```

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

```r
head(Universal_Bank)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```r
final_data <- Universal_Bank[,-c(1,5)]
dim(final_data)
```

```
## [1] 5000   12
```

```r
final_data$Education <- as.factor(final_data$Education)
```

```r
#here we are creating dummy variables
dum_vars<- dummyVars(~.,data=final_data)
final_data <- as.data.frame(predict(dum_vars,final_data))
```

```r
#partition the data
set.seed(1)
trainset_data <- sample(rownames(final_data), 0.6*dim(final_data)[1])
validset_data <- setdiff(rownames(final_data),trainset_data)
train <- final_data[trainset_data,]
valid <- final_data[validset_data,]
t(t(names(train)))
```

```
##       [,1]
##  [1,] "Age"
##  [2,] "Experience"
##  [3,] "Income"
##  [4,] "Family"
##  [5,] "CCAvg"
##  [6,] "Education.1"
##  [7,] "Education.2"
##  [8,] "Education.3"
##  [9,] "Mortgage"
## [10,] "Personal.Loan"
## [11,] "Securities.Account"
## [12,] "CD.Account"
## [13,] "Online"
## [14,] "CreditCard"
```

```r
summary(train)
```

```
##       Age          Experience       Income          Family     
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000  
##  1st Qu.:36.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000  
##  Median :45.00   Median :20.00   Median : 63.00   Median :2.000  
##  Mean   :45.43   Mean   :20.19   Mean   : 73.08   Mean   :2.388  
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 98.00   3rd Qu.:3.000  
##  Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000  
##      CCAvg         Education.1      Education.2     Education.3    
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.: 0.700   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000  
##  Median : 1.500   Median :0.0000   Median :0.000   Median :0.0000  
##  Mean   : 1.915   Mean   :0.4173   Mean   :0.285   Mean   :0.2977  
##  3rd Qu.: 2.500   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000  
##  Max.   :10.000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
##     Mortgage      Personal.Loan     Securities.Account   CD.Account     
##  Min.   :  0.00   Min.   :0.00000   Min.   :0.0000     Min.   :0.00000  
##  1st Qu.:  0.00   1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.:0.00000  
##  Median :  0.00   Median :0.00000   Median :0.0000     Median :0.00000  
##  Mean   : 57.34   Mean   :0.09167   Mean   :0.1003     Mean   :0.05367  
##  3rd Qu.:102.00   3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.:0.00000  
##  Max.   :635.00   Max.   :1.00000   Max.   :1.0000     Max.   :1.00000  
##      Online         CreditCard    
##  Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :1.0000   Median :0.0000  
##  Mean   :0.5847   Mean   :0.2927  
##  3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :1.0000
```

```r
cat("The size of the training dataset is:",nrow(train))
```

```
## The size of the training dataset is: 3000
```

```r
summary(valid)
```

```
##       Age         Experience        Income           Family
##  Min.   :23.0   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.0   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.0   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :45.2   Mean   :19.97   Mean   : 74.81   Mean   :2.409
##  3rd Qu.:55.0   3rd Qu.:30.00   3rd Qu.: 99.00   3rd Qu.:3.000
##  Max.   :67.0   Max.   :43.00   Max.   :218.00   Max.   :4.000
##      CCAvg         Education.1     Education.2     Education.3
##  Min.   : 0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
##  Median : 1.600   Median :0.000   Median :0.000   Median :0.000
##  Mean   : 1.973   Mean   :0.422   Mean   :0.274   Mean   :0.304
##  3rd Qu.: 2.600   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :10.000   Max.   :1.000   Max.   :1.000   Max.   :1.000
##     Mortgage      Personal.Loan    Securities.Account   CD.Account
##  Min.   :  0.00   Min.   :0.0000   Min.   :0.0000     Min.   :0.0000
##  1st Qu.:  0.00   1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:0.0000
##  Median :  0.00   Median :0.0000   Median :0.0000     Median :0.0000
##  Mean   : 55.24   Mean   :0.1025   Mean   :0.1105     Mean   :0.0705
##  3rd Qu.: 97.25   3rd Qu.:0.0000   3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   :617.00   Max.   :1.0000   Max.   :1.0000     Max.   :1.0000
##      Online        CreditCard
##  Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.000
##  Median :1.000   Median :0.000
##  Mean   :0.615   Mean   :0.296
##  3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.000
```

```r
cat("The size of the validation dataset is:",nrow(valid))
```

```
## The size of the validation dataset is: 2000
```

```r
trainset_norm <- train[,-10]
validset_norm <- valid[,-10]
norm <- preProcess(train[,-10],method=c("center","scale"))
```

```r
trainset_norm <- predict(norm,train[,-10])
validset_norm <- predict(norm,valid[,-10])
```

Question1

```
Brandnew<- data.frame(
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,
CreditCard = 1
)

# Normalize the new customer dataset
customer_setnorm <- predict(norm, Brandnew)
```

```
summary(customer_setnorm)
```

```
##       Age           Experience          Income           Family
##  Min.   :-0.4774   Min.   :-0.8953   Min.   :0.2389   Min.   :-0.3368
##  1st Qu.:-0.4774   1st Qu.:-0.8953   1st Qu.:0.2389   1st Qu.:-0.3368
##  Median :-0.4774   Median :-0.8953   Median :0.2389   Median :-0.3368
##  Mean   :-0.4774   Mean   :-0.8953   Mean   :0.2389   Mean   :-0.3368
##  3rd Qu.:-0.4774   3rd Qu.:-0.8953   3rd Qu.:0.2389   3rd Qu.:-0.3368
##  Max.   :-0.4774   Max.   :-0.8953   Max.   :0.2389   Max.   :-0.3368
##      CCAvg          Education.1       Education.2     Education.3
##  Min.   :0.04924   Min.   :-0.8462   Min.   :1.584   Min.   :-0.6509
##  1st Qu.:0.04924   1st Qu.:-0.8462   1st Qu.:1.584   1st Qu.:-0.6509
##  Median :0.04924   Median :-0.8462   Median :1.584   Median :-0.6509
##  Mean   :0.04924   Mean   :-0.8462   Mean   :1.584   Mean   :-0.6509
##  3rd Qu.:0.04924   3rd Qu.:-0.8462   3rd Qu.:1.584   3rd Qu.:-0.6509
##  Max.   :0.04924   Max.   :-0.8462   Max.   :1.584   Max.   :-0.6509
##     Mortgage       Securities.Account   CD.Account          Online
##  Min.   :-0.5679   Min.   :-0.3339    Min.   :-0.2381   Min.   :0.8427
##  1st Qu.:-0.5679   1st Qu.:-0.3339    1st Qu.:-0.2381   1st Qu.:0.8427
##  Median :-0.5679   Median :-0.3339    Median :-0.2381   Median :0.8427
##  Mean   :-0.5679   Mean   :-0.3339    Mean   :-0.2381   Mean   :0.8427
##  3rd Qu.:-0.5679   3rd Qu.:-0.3339    3rd Qu.:-0.2381   3rd Qu.:0.8427
##  Max.   :-0.5679   Max.   :-0.3339    Max.   :-0.2381   Max.   :0.8427
##    CreditCard
##  Min.   :1.554
##  1st Qu.:1.554
##  Median :1.554
##  Mean   :1.554
##  3rd Qu.:1.554
##  Max.   :1.554
```

```
prediction <- class::knn(train = trainset_norm,
test = customer_setnorm,
cl = train$Personal.Loan, k = 1)
prediction
```

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

According to the k-NN classification model with k = 1 and utilizing all predictors except ID and ZIP code, this customer is predicted to not accept the loan (class 0).

Question2

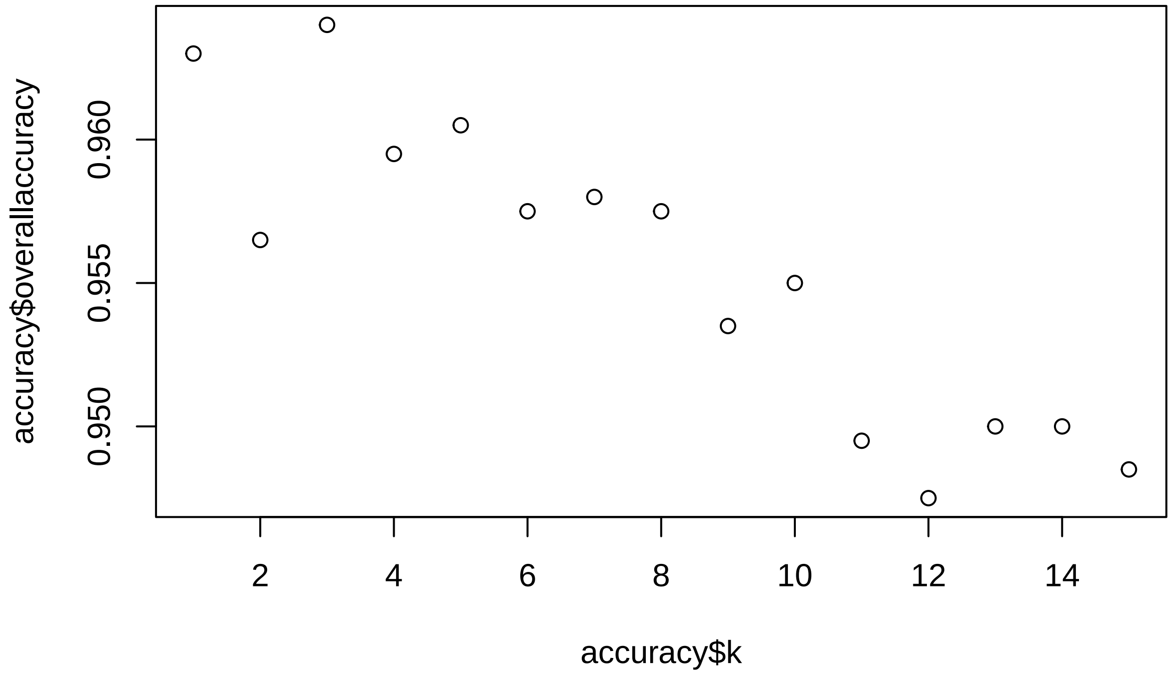
```
 #Calculate the accuracy for each value of k
# Set the range of k values to consider
accuracy <- data.frame(k = seq(1, 15, 1), overallaccuracy = rep(0, 15))
for(i in 1:15) {
kn <- class::knn(train = trainset_norm,
test = validset_norm,
cl = train$Personal.Loan, k = i)
accuracy[i, 2] <- confusionMatrix(kn,
as.factor(valid$Personal.Loan),positive = "1")$overall[1]
}
which(accuracy[,2] == max(accuracy[,2]))
```

```
## [1] 3
```

```
accuracy
```

```
##     k overallaccuracy
## 1   1          0.9630
## 2   2          0.9565
## 3   3          0.9640
## 4   4          0.9595
## 5   5          0.9605
## 6   6          0.9575
## 7   7          0.9580
## 8   8          0.9575
## 9   9          0.9535
## 10 10          0.9550
## 11 11          0.9495
## 12 12          0.9475
## 13 13          0.9500
## 14 14          0.9500
## 15 15          0.9485
```

```
plot(accuracy$k,accuracy$overallaccuracy)
```



the accurate value of k is k=3 which gives greater performance among all the values of k because it meet the balence between over fitting and rejecting forecasts.

Question:3

```
prediction <- class::knn(train = trainset_norm,
test = validset_norm,
cl = train$Personal.Loan, k=3)
confusionMatrix(prediction,as.factor(valid$Personal.Loan))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1786   63
##          1    9  142
##
##                Accuracy : 0.964
##                  95% CI : (0.9549, 0.9717)
##     No Information Rate : 0.8975
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7785
##
##  Mcnemar's Test P-Value : 4.208e-10
##
##             Sensitivity : 0.9950
##             Specificity : 0.6927
##          Pos Pred Value : 0.9659
##          Neg Pred Value : 0.9404
##              Prevalence : 0.8975
##          Detection Rate : 0.8930
##    Detection Prevalence : 0.9245
##       Balanced Accuracy : 0.8438
##
##        'Positive' Class : 0
##
```

question:4

```
customer_set2 <- data.frame(
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,
CreditCard = 1)
#Normalizing the 2nd client dataset
customer_set2_norm <- predict(norm , customer_set2)
```

```
prediction <- class::knn(train = trainset_norm,
test = customer_set2_norm,
cl = train$Personal.Loan, k = 3)
prediction
```

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

question5

```
set.seed(500)
Trainset_Index <- sample(row.names(final_data), .5*dim(final_data)[1])
```

```
#create validation index
Valid_Index <- sample(setdiff(row.names(final_data),Trainset_Index),.3*dim(final_data)[1])
Test_Index =setdiff(row.names(final_data),union(Trainset_Index,Valid_Index))#create test index
train.df <- final_data[Trainset_Index,]
cat("The size of training dataset is:", nrow(train.df))
```

```
## The size of training dataset is: 2500
```

```
valid.df <- final_data[Valid_Index, ]
cat("The size of validation dataset is:", nrow(valid.df))
```

```
## The size of validation dataset is: 1500
```

```
test.df <- final_data[Test_Index, ]
cat("The size of the new test dataset is:", nrow(test.df))
```

```
## The size of the new test dataset is: 1000
```

```
normvalues <- preProcess(train.df[, -10], method=c("center", "scale"))
train.df.norm <- predict(norm, train.df[, -10])
valid.df.norm <- predict(norm, valid.df[, -10])
test.df.norm <- predict(norm ,test.df[,-10])
```

```
prediction_3 <- class::knn(train = train.df.norm,
test = test.df.norm,
cl = train.df$Personal.Loan, k=3)
confusionMatrix(prediction_3,as.factor(test.df$Personal.Loan))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 898  36
##          1   6  60
##
##                Accuracy : 0.958
##                  95% CI : (0.9436, 0.9696)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 9.200e-11
##
##                   Kappa : 0.7187
##
##  Mcnemar’s Test P-Value : 7.648e-06
##
##             Sensitivity : 0.9934
##             Specificity : 0.6250
##          Pos Pred Value : 0.9615
##          Neg Pred Value : 0.9091
##              Prevalence : 0.9040
##          Detection Rate : 0.8980
##    Detection Prevalence : 0.9340
##       Balanced Accuracy : 0.8092
##
##        ’Positive’ Class : 0
##
```

```
prediction_4 <- class::knn(train = train.df.norm,
test = valid.df.norm,
cl = train.df$Personal.Loan, k=3)
confusionMatrix(prediction_4,as.factor(valid.df$Personal.Loan))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1332   65
##          1    8   95
##
##                Accuracy : 0.9513
##                  95% CI : (0.9392, 0.9617)
##     No Information Rate : 0.8933
##     P-Value [Acc > NIR] : 6.496e-16
##
##                   Kappa : 0.6971
##
##  Mcnemar’s Test P-Value : 5.590e-11
##
##             Sensitivity : 0.9940
##             Specificity : 0.5938
##          Pos Pred Value : 0.9535
##          Neg Pred Value : 0.9223
##              Prevalence : 0.8933
##          Detection Rate : 0.8880
##    Detection Prevalence : 0.9313
##       Balanced Accuracy : 0.7939
##
##        ’Positive’ Class : 0
##
```

```
prediction_5 <- class::knn(train = train.df.norm,
test = train.df.norm,
cl = train.df$Personal.Loan, k=3)
confusionMatrix(prediction_5,as.factor(train.df$Personal.Loan))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2273   53
##          1    3  171
##
##                Accuracy : 0.9776
##                  95% CI : (0.971, 0.983)
##     No Information Rate : 0.9104
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8473
##
##  Mcnemar’s Test P-Value : 5.835e-11
##
##             Sensitivity : 0.9987
##             Specificity : 0.7634
##          Pos Pred Value : 0.9772
##          Neg Pred Value : 0.9828
##              Prevalence : 0.9104
##          Detection Rate : 0.9092
##    Detection Prevalence : 0.9304
##       Balanced Accuracy : 0.8810
##
##        'Positive' Class : 0
##
```