FML ASSISGNMENT 2

Sushma Palancha

2023-09-26

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

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(e1071)
```

#Read the data.

```
UniversalBank = read.csv("~/Documents/FML/FML ASSIGNMENT 2/UniversalBank.csv")
```

#Drop ID and ZIP

```
UniversalBank = UniversalBank[,-c(1,5)]
```

#Conversion of Factor

```
#Only Education needs to be converted into Factor
UniversalBank$Education = as.factor(UniversalBank$Education)
levels(UniversalBank$Education)
```

```
## [1] "1" "2" "3"
```

```
#Now, Convert Education to Dummy Variables

groups = dummyVars(~.,data = UniversalBank) #This created a dummy varible

UniversalBank.Mod = as.data.frame(predict(groups,UniversalBank))
```

```
set.seed(1) # Important to ensure that we get the same sample if we rerun the code

training.dif = sample(row.names(UniversalBank.Mod),0.6*dim(UniversalBank.Mod)[1])
validation.dif = setdiff(row.names(UniversalBank.Mod),training.dif)
train.diff = UniversalBank.Mod[training.dif,]
valid.diff = UniversalBank.Mod[validation.dif,]
t(t(names(train.diff)))
```

```
##       [,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"
```

```
#Second approach
library(caTools)
set.seed(1)
split <- sample.split(UniversalBank.Mod, SplitRatio = 0.6)
train_set <- subset(UniversalBank.Mod, split == TRUE)
valid_set <- subset(UniversalBank.Mod, split == FALSE)

# Print the sizes of the training and validation sets
print(paste("The size of the training set is:", nrow(train_set)))
```

```
## [1] "The size of the training set is: 2858"
```

```
print(paste("The size of the validation set is:", nrow(valid_set)))
```

```
## [1] "The size of the validation set is: 2142"
```

#Now, let us normalize data

```
train.normal.diff <- train.diff[,-10] # Note that Personal Income is the 10th variable
valid.normal.diff <- valid.diff[,-10]

normal.values <- preProcess(train.diff[, -10], method=c("center", "scale"))
train.normal.diff <- predict(normal.values, train.diff[, -10])
valid.normal.diff <- predict(normal.values, valid.diff[, -10])
```

Now let's solve the given Questions

#Question No:1 1.Consider the following customer: 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?

```r
# We have converted all categorical variables to dummy variables
# Let's create a new sample
New_Customer1 <- 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
New.Cust.normal <- New_Customer1
New.Cust.normal <- predict(normal.values, New.Cust.normal)
```

#Now, let us predict using KNN

```r
KNN.Predct1 <- class::knn(train = train.normal.diff,
                       test = New.Cust.normal,
                       cl = train.diff$Personal.Loan, k = 1)
KNN.Predct1
```

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

#Question No:2 2.What is a choice of K that balances between over-fitting and ignoring the predictor information?

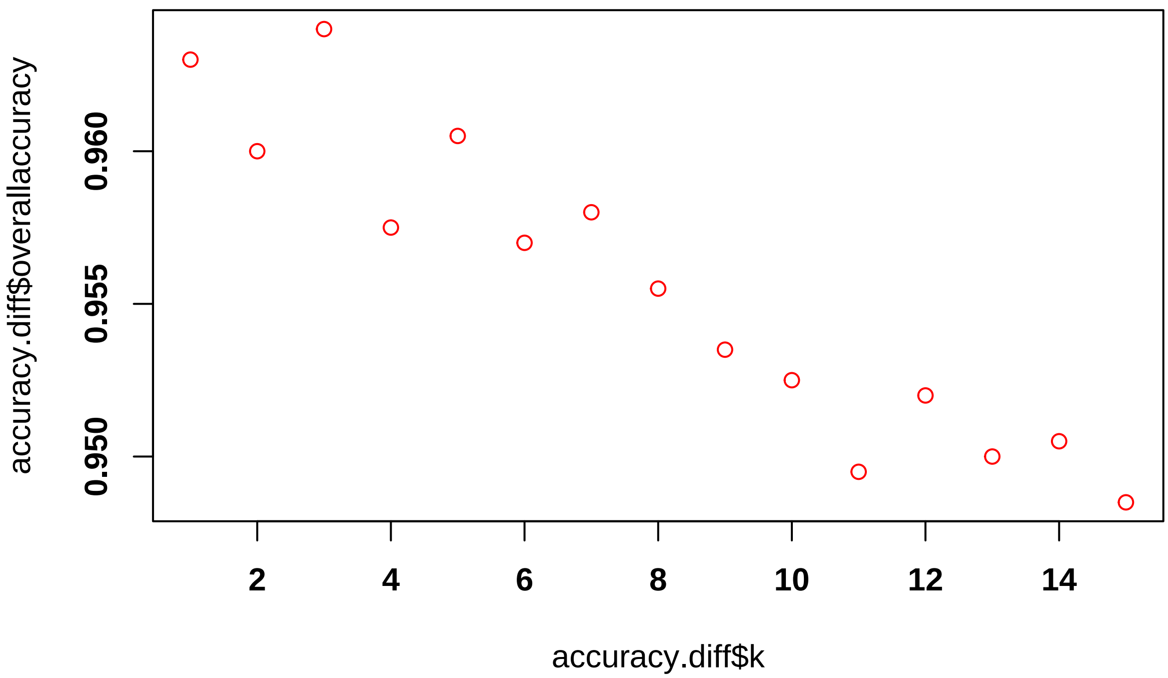
```r
#Calculate the accuracy for each value of k
#Set the range of k values to consider

accuracy.diff <- data.frame(k = seq(1, 15, 1), overallaccuracy = rep(0, 15))
for(i in 1:15) {
  KNN.Predct <- class::knn(train = train.normal.diff,
                         test = valid.normal.diff,
                         cl = train.diff$Personal.Loan, k = i)
  accuracy.diff[i, 2] <- confusionMatrix(KNN.Predct,
                                     as.factor(valid.diff$Personal.Loan),positive = "1")$overall[1]
}

which(accuracy.diff[,2] == max(accuracy.diff[,2]))
```

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

```r
plot(accuracy.diff$k,accuracy.diff$overallaccuracy,col="red",font=2)
```



***

#Question No:3 3. Show the confusion matrix for the validation data that results from using the best k**

```r
KNN.Predct2 <- class::knn(train = train.normal.diff,
                          test = valid.normal.diff,
                          cl = train.diff$Personal.Loan, k = 3)

confusionMatrix(KNN.Predct2,as.factor(valid.diff$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 No:4 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

```
#Classifying the customer using the best K.

New_Customer2 = 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
)

KNN.Predct3 <- class::knn(train = train.normal.diff,
                         test = New_Customer2,
                         cl = train.diff$Personal.Loan, k = 3)

KNN.Predct3
```

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

```r
#The customer has been classified as approved for personal loan
print("As the out put is 0, The Customer will not accept the Personal Loan offer")
```

```
## [1] "As the out put is 0, The Customer will not accept the Personal Loan offer"
```

#Question No:5

```r
set.seed(2)
#Let's take 50% of the entire modified data as Training data
train.diff2 = sample(row.names(UniversalBank.Mod), 0.5*dim(UniversalBank.Mod)[1])

#Let's take 30% of the data from the remaining 50% as Validation Data
valid.diff2 = sample(setdiff(row.names(UniversalBank.Mod), train.diff2), 0.3*dim(UniversalBank.Mod)[1])

#Let's take remaining 20% of the modified data as Test Data
test.diff2 = setdiff(row.names(UniversalBank.Mod), union(train.diff2,valid.diff2))

train.normal.diff2 = UniversalBank.Mod[train.diff2,]
valid.normal.diff2 = UniversalBank.Mod[valid.diff2,]
test.normal.diff2 = UniversalBank.Mod[test.diff2,]

#transporting the data
t(t(names(train.normal.diff2)))
```

```
##       [,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
# Applying the k-NN method with the chosen K.

trainknn2 = knn(train = train.normal.diff2[,-8], test = train.normal.diff2[,-8], cl = train.normal.diff

validknn2 = knn(train = train.normal.diff2[,-8], test = valid.normal.diff2[,-8], cl = train.normal.diff

testknn2 = knn(train = train.normal.diff2[,-8], test = test.normal.diff2[,-8], cl = train.normal.diff2[
```

#Comparing the confusion matrix of the training set, validation sets and test set

```r
Confusionmatrix_trainknn2 = confusionMatrix(trainknn2, as.factor(train.normal.diff2$Personal.Loan),posi

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```

```r
Confusionmatrix_validknn2 = confusionMatrix(validknn2, as.factor(valid.normal.diff2$Personal.Loan),posi

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
```

```
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```

```r
Confusionmatrix_testknn2 = confusionMatrix(testknn2, as.factor(test.normal.diff2$Personal.Loan),positive

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```

#The sets are not mutually exclusive. ***