FM2

Vijay

2022-21-02

```
getwd()
```

```
## [1] "C:/Users/vijay/Documents"
```

##Here I am Importing the Dataset

```
bank.df= read.csv("universalBank.csv")
```

```
summary(bank.df)
```

```
##        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
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(class)
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
```

```
library(ISLR)
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(FNN)
```

```
##
## Attaching package: 'FNN'

## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

```
library(lattice)
```

##Now I am going to Remove ID , ZIP Code

```
bank.df$ID <- NULL
bank.df$ZIP.Code <- NULL
bank.df$Education = as.factor(bank.df$Education)
```

##later I am creating a dummy dataset

```
dummyvari <- as.data.frame(dummy.code(bank.df$Education))
```

```
names(dummyvari) <- c("Education_1", "Education_2","Education_3")
```

##Now I am setting education to NULL

```
bank.df$Education <- NULL
```

```
bank_Final <- cbind(bank.df, dummyvari)
```

##Here I am going to divide the dataset into train and test

```
set.seed(1)
train.index <- createDataPartition(bank_Final$Personal.Loan, p= 0.6 , list=FALSE)
valid.index <- setdiff(row.names(bank_Final), train.index)
train.df <- bank_Final[train.index,]
valid.df <- bank_Final[valid.index,]
```

##Now we are generating the Test data

```
new_customer <- data.frame(Age = 40,
                        Experience = 10,
                        Income = 84,
                        Family = 2,
                        CCAvg = 2,
                        Mortgage = 0,
                        Securities.Account = 0,
                        CD.Account = 0,
                        Online = 1,
                        CreditCard = 1,
                        Education_1 = 0,
                        Education_2 = 1,
                        Education_3 = 0)
```

normalisation

```
train.norm.df <- train.df[,-7]
valid.norm.df <- valid.df[,-7]
new_customer.norm <- new_customer

norm.values <- preProcess(train.df[, -7], method=c("center", "scale"))
train.norm.df <- predict(norm.values, train.df[, -7])
valid.norm.df <- predict(norm.values, valid.df[, -7])
new_customer.norm <- predict(norm.values, new_customer.norm)
```

```
summary(train.norm.df)
```

```
##       Age              Experience          Income            Family
```

```
##  Min.   :-1.97257   Min.   :-2.03718   Min.   :-1.4240   Min.   :-1.2058
##  1st Qu.:-0.82922   1st Qu.:-0.89531   1st Qu.:-0.7457   1st Qu.:-1.2058
##  Median :-0.03767   Median :-0.01695   Median :-0.2206   Median :-0.3368
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84183   3rd Qu.: 0.86141   3rd Qu.: 0.5452   3rd Qu.: 0.5321
##  Max.   : 1.89723   Max.   : 2.00328   Max.   : 3.3022   Max.   : 1.4010
##      CCAvg            Mortgage       Securities.Account   CD.Account
##  Min.   :-1.1059   Min.   :-0.5679   Min.   :-0.3339    Min.   :-0.2381
##  1st Qu.:-0.7016   1st Qu.:-0.5679   1st Qu.:-0.3339    1st Qu.:-0.2381
##  Median :-0.2396   Median :-0.5679   Median :-0.3339    Median :-0.2381
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000    Mean   : 0.0000
##  3rd Qu.: 0.3380   3rd Qu.: 0.4423   3rd Qu.:-0.3339    3rd Qu.:-0.2381
##  Max.   : 4.6700   Max.   : 5.7216   Max.   : 2.9940    Max.   : 4.1985
##      Online          CreditCard       Education_1       Education_2
##  Min.   :-1.1863   Min.   :-0.6431   Min.   :-0.8462   Min.   :-0.6509
##  1st Qu.:-1.1863   1st Qu.:-0.6431   1st Qu.:-0.8462   1st Qu.:-0.6509
##  Median : 0.8427   Median :-0.6431   Median :-0.8462   Median :-0.6509
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8427   3rd Qu.: 1.5544   3rd Qu.: 1.1814   3rd Qu.: 1.5358
##  Max.   : 0.8427   Max.   : 1.5544   Max.   : 1.1814   Max.   : 1.5358
##   Education_3
##  Min.   :-0.6312
##  1st Qu.:-0.6312
##  Median :-0.6312
##  Mean   : 0.0000
##  3rd Qu.: 1.5836
##  Max.   : 1.5836
```

##Here I am Performing Knn classification, using K=1

```
outputrate <- class::knn(train = train.norm.df,test = new_customer.norm,
                         cl = train.df$Personal.Loan, k = 1)

print(outputrate)
```

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

##we are going to find best K

```
accuracy.df <- data.frame(k = seq(1, 10, 1), accuracy = rep(0, 10))
```

```
for(i in 1:10) {
  knn.prediction <- class::knn(train = train.norm.df,
                           test = valid.norm.df,
                           cl = train.df$Personal.Loan, k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.prediction,
                                       as.factor(valid.df$Personal.Loan))$overall[1]
}
which(accuracy.df[,2] == max(accuracy.df[,2]))
```

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

```
accuracy.df
```

```
##     k accuracy
## 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
```

##choosing k = 3

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

confusionMatrix(knn.prediction, as.factor(valid.df$Personal.Loan), positive = "1")
```

```
## 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 : 0.6927
##             Specificity : 0.9950
##          Pos Pred Value : 0.9404
##          Neg Pred Value : 0.9659
##              Prevalence : 0.1025
##          Detection Rate : 0.0710
##    Detection Prevalence : 0.0755
##       Balanced Accuracy : 0.8438
##
##        'Positive' Class : 1
##
```

##Now Here confusion matrix for the best k value =3

```
newcustomer <- data.frame(Age = 40,
                          Experience = 10,
                          Income = 84,
                          Family = 2,
                          CCAvg = 2,
                          Mortgage = 0,
                          Securities.Account = 0,
                          CD.Account = 0,
                          Online = 1,
                          CreditCard = 1,
                          Education_1 = 0,
                          Education_2 = 1,
                          Education_3 = 0)

fitknn <-class::knn(train = train.norm.df,
                    test = newcustomer,
                    cl = train.df$Personal.Loan, k = 3)

fitknn
```

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

##knn model tells that new customer will accept loan ##reading the dataset

```
bank.df= read.csv("universalBank.csv")
```

##packages

```
library(ISLR)
library(psych)
library(caret)
library(FNN)
library(class)
library(dplyr)
library(lattice)
```

##here i am removing id and zipcode variables from the dataset

```
bank.df$ID <- NULL
bank.df$ZIP.Code <- NULL
bank.df$Education = as.factor(bank.df$Education)
```

##Creating a dummy dataframe

```
dummymod <- as.data.frame(dummy.code(bank.df$Education))
```

##Renaming the data frame

```
names(dummymod) <- c("Education_1", "Education_2","Education_3")
```

##deleting education variable

```
bank.df$Education <- NULL
```

##Main dataset

```
bank_Final <- cbind(bank.df, dummymod)
```

#Partitioning the dataset

```
set.seed(1)
train.index <- createDataPartition(bank_Final$Personal.Loan, p= 0.5 , list=FALSE)
valid.index <- createDataPartition(bank_Final$Personal.Loan, p= 0.3 , list=FALSE)
test.index <- setdiff(row.names(bank_Final), union(train.index, valid.index))

train.df <- bank_Final[train.index, ]
valid.df <- bank_Final[valid.index, ]
test.df <- bank_Final[test.index, ]
```

##Performing normalisation

```
train.norm.df <- train.df[,-7]
valid.norm.df <- valid.df[,-7]
test.norm.df <- test.df[,-7]

norm.values <- preProcess(train.df[, -7], method=c("center", "scale"))
train.norm.df <- predict(norm.values, train.df[, -7])
valid.norm.df <- predict(norm.values, valid.df[, -7])
test.norm.df <- predict(norm.values, test.df[, -7])
```

##Performing Knn classification using K=3

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

knn.train.pred <- class::knn(train = train.norm.df,
                             test = train.norm.df,
                             cl = train.df$Personal.Loan, k = 3)

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

##Confusion matrix for K=3

```
confusionMatrix(knn.test.pred, as.factor(test.df$Personal.Loan), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1590   50
##          1    8  111
##
##                Accuracy : 0.967
##                  95% CI : (0.9576, 0.9749)
##     No Information Rate : 0.9085
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7754
##
##  Mcnemar's Test P-Value : 7.303e-08
##
##             Sensitivity : 0.68944
##             Specificity : 0.99499
##          Pos Pred Value : 0.93277
##          Neg Pred Value : 0.96951
##              Prevalence : 0.09153
##          Detection Rate : 0.06310
##    Detection Prevalence : 0.06765
##       Balanced Accuracy : 0.84222
##
##        'Positive' Class : 1
##
```

```
confusionMatrix(knn.train.pred, as.factor(train.df$Personal.Loan), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2263   54
##          1    5  178
##
##                Accuracy : 0.9764
##                  95% CI : (0.9697, 0.982)
##     No Information Rate : 0.9072
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8452
##
##  Mcnemar's Test P-Value : 4.129e-10
##
##             Sensitivity : 0.7672
##             Specificity : 0.9978
##          Pos Pred Value : 0.9727
##          Neg Pred Value : 0.9767
##              Prevalence : 0.0928
##          Detection Rate : 0.0712
##    Detection Prevalence : 0.0732
##       Balanced Accuracy : 0.8825
```

```
##
##        'Positive' Class : 1
##
```

```
confusionMatrix(knn.valid.pred, as.factor(valid.df$Personal.Loan), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1347   43
##          1    3  107
##
##                Accuracy : 0.9693
##                  95% CI : (0.9593, 0.9775)
##     No Information Rate : 0.9
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8067
##
##  Mcnemar's Test P-Value : 8.912e-09
##
##             Sensitivity : 0.71333
##             Specificity : 0.99778
##          Pos Pred Value : 0.97273
##          Neg Pred Value : 0.96906
##              Prevalence : 0.10000
##          Detection Rate : 0.07133
##    Detection Prevalence : 0.07333
##       Balanced Accuracy : 0.85556
##
##        'Positive' Class : 1
##
```