Assignment_2

Gowtham Chakri Mallepaka

2022-10-03

embedded R code chunks within the document. You can embed an R code chunk like this:

```
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(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(gmodels)


#
ubank<-read.csv("c:/Users/reliance digital/Downloads/UniversalBank.csv")

#
ubank_1<-ubank[,-1]
ubank_1<-ubank_1[,-4]
head(ubank_1)
```

```
##   Age Experience Income Family CCAvg Education Mortgage Personal.Loan
## 1  25          1     49      4   1.6         1        0             0
## 2  45         19     34      3   1.5         1        0             0
## 3  39         15     11      1   1.0         1        0             0
## 4  35          9    100      1   2.7         2        0             0
## 5  35          8     45      4   1.0         2        0             0
```

```
## 6  37         13     29      4   0.4         2      155             0
##   Securities.Account CD.Account Online CreditCard
## 1                  1          0      0          0
## 2                  1          0      0          0
## 3                  0          0      0          0
## 4                  0          0      0          0
## 5                  0          0      0          1
## 6                  0          0      1          0
```

```
#
ubank_1$Personal.Loan<-as.factor(ubank_1$Personal.Loan)

#
head(is.na(ubank_1))
```

```
##        Age Experience Income Family CCAvg Education Mortgage Personal.Loan
## [1,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
## [2,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
## [3,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
## [4,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
## [5,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
## [6,] FALSE      FALSE  FALSE  FALSE FALSE     FALSE    FALSE         FALSE
##      Securities.Account CD.Account Online CreditCard
## [1,]              FALSE      FALSE  FALSE      FALSE
## [2,]              FALSE      FALSE  FALSE      FALSE
## [3,]              FALSE      FALSE  FALSE      FALSE
## [4,]              FALSE      FALSE  FALSE      FALSE
## [5,]              FALSE      FALSE  FALSE      FALSE
## [6,]              FALSE      FALSE  FALSE      FALSE
```

```
#
Education<-as.character(ubank_1$Education)

ubank_2<-cbind(ubank_1[,-6],Education)
head(ubank_2)
```

```
##   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 Education
## 1          0      0          0         1
## 2          0      0          0         1
## 3          0      0          0         1
## 4          0      0          0         2
## 5          0      0          1         2
## 6          0      1          0         2
```

```
#
dummy<-dummyVars("~Education",data = ubank_2)
dummyeducation<-data.frame(predict(dummy,ubank_2))

ubank_dummy<-cbind(ubank_2[,-12],dummyeducation)
head(ubank_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
```

```
#
set.seed(8)
train<-createDataPartition(ubank_dummy$Personal.Loan,p=0.6,list = FALSE)
trainingset<-ubank_dummy[train,]
validationset<-ubank_dummy[-train,]
nrow(validationset)
```

```
## [1] 2000
```

```
testingset<-data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,  Mortgage = 0, Se
                       CD.Account = 0, Online = 1,CreditCard = 1,Education1 = 0, Education2 = 1, Education3
summary(trainingset)
```

```
##       Age          Experience       Income          Family
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1st Qu.:36.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.0   Median : 63.00   Median :2.000
##  Mean   :45.42   Mean   :20.2   Mean   : 73.59   Mean   :2.392
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 99.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg            Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:2712        Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 288        1st Qu.:0.000
##  Median : 1.500   Median :  0.00                 Median :0.000
##  Mean   : 1.967   Mean   : 56.04                 Mean   :0.111
##  3rd Qu.: 2.600   3rd Qu.: 99.00                 3rd Qu.:0.000
##  Max.   :10.000   Max.   :635.00                 Max.   :1.000
##    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   :0.06067   Mean   :0.5947   Mean   :0.2897   Mean   :0.4233
##  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.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.275   Mean   :0.3017
##  3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000
```

```
summary(validationset)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :45.21   Mean   :19.96   Mean   : 74.05   Mean   :2.403
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :203.00   Max.   :4.000
##      CCAvg           Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:1808        Min.   :0.0000
##  1st Qu.: 0.670   1st Qu.:  0.00   1: 192        1st Qu.:0.0000
##  Median : 1.600   Median :  0.00                 Median :0.0000
##  Mean   : 1.895   Mean   : 57.18                 Mean   :0.0945
##  3rd Qu.: 2.500   3rd Qu.:103.00                 3rd Qu.:0.0000
##  Max.   :10.000   Max.   :617.00                 Max.   :1.0000
##    CD.Account       Online      CreditCard       Education1      Education2
##  Min.   :0.00   Min.   :0.0   Min.   :0.0000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.00   1st Qu.:0.0   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.000
##  Median :0.00   Median :1.0   Median :0.0000   Median :0.000   Median :0.000
##  Mean   :0.06   Mean   :0.6   Mean   :0.3005   Mean   :0.413   Mean   :0.289
##  3rd Qu.:0.00   3rd Qu.:1.0   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.00   Max.   :1.0   Max.   :1.0000   Max.   :1.000   Max.   :1.000
##    Education3
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.298
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
summary(testingset)
```

```
##       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
##  Securities.Account   CD.Account     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
```

```
#
normalvariables<-c('Age',"Experience","Income","Family","CCAvg","Mortgage","Securities.Account",
                   "CD.Account","Online","CreditCard","Education1","Education2","Education3")

normalization_values<-preProcess(trainingset[,normalvariables],method=c('center','scale'))

trainingset.norm<-predict(normalization_values,trainingset)
validationset.norm<-predict(normalization_values,validationset)
testingset.norm<-predict(normalization_values,testingset)


#1st question
set.seed(8)
grid<-expand.grid(k=1)
model_1<-train(Personal.Loan~.,data=trainingset.norm,method='knn',tuneGrid=grid)
model_1
```

```
## 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:
##
##   Accuracy  Kappa
##   0.949094  0.6851349
##
## Tuning parameter 'k' was held constant at a value of 1
```

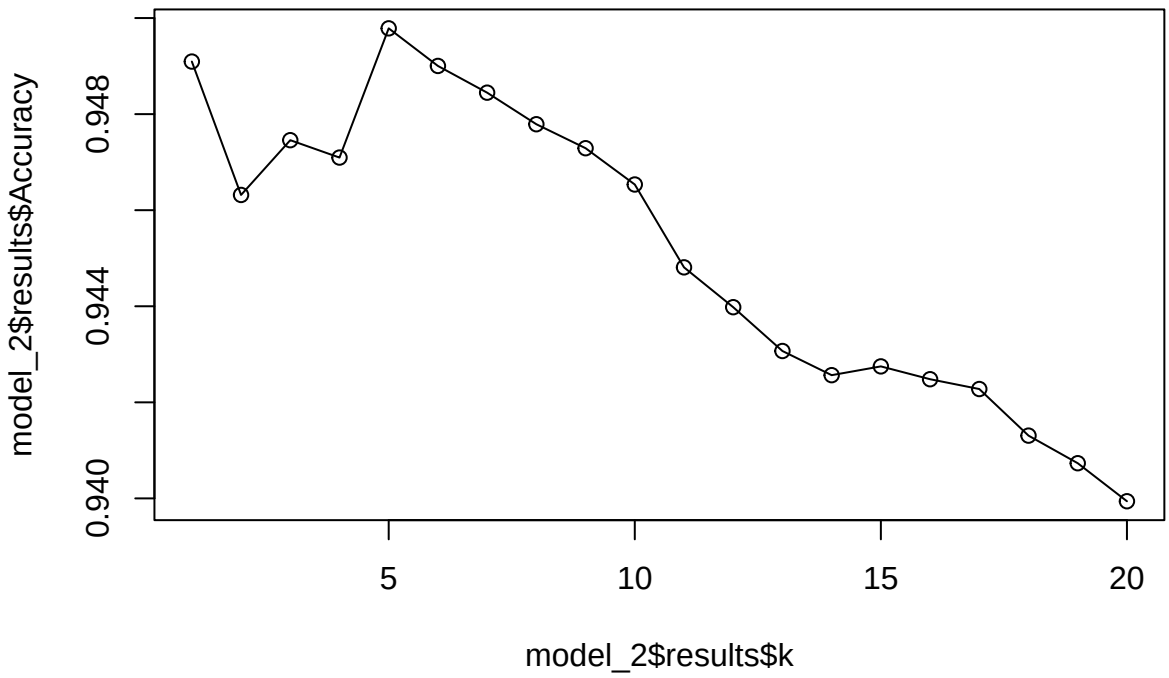
```
customer<-predict(model_1,testingset.norm)
customer
```

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

```r
#2nd question
set.seed(8)
grid2<-expand.grid(k=seq(1:20))
model_2<-train(Personal.Loan~.,data=trainingset.norm,method='knn',tuneGrid=grid2)
model_2
```

```
## 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  0.9490940  0.6851349
##    2  0.9463170  0.6654215
##    3  0.9474575  0.6657001
##    4  0.9470954  0.6575623
##    5  0.9497856  0.6695343
##    6  0.9490031  0.6596269
##    7  0.9484466  0.6507375
##    8  0.9477895  0.6440420
##    9  0.9472902  0.6373873
##   10  0.9465350  0.6291945
##   11  0.9448102  0.6131506
##   12  0.9439806  0.6044008
##   13  0.9430687  0.5954681
##   14  0.9425655  0.5904760
##   15  0.9427483  0.5904143
##   16  0.9424813  0.5867194
##   17  0.9422770  0.5841140
##   18  0.9413069  0.5743801
##   19  0.9407327  0.5681902
##   20  0.9399411  0.5607203
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```r
plot(model_2$results$k,model_2$results$Accuracy, type = 'o')
```




```
bestk<-model_2$bestTune[[1]]
bestk
```

```
## [1] 5
```

```
#
training.label<-trainingset.norm[,7]
validation.label<-validationset.norm[,7]
testing.label<-testingset.norm[,7]

predictedvalidation.label<-knn(trainingset.norm,validationset.norm,cl=training.label,k=bestk)

CrossTable(x=validation.label,y=predictedvalidation.label,prop.chisq = FALSE)
```

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

```
## Total Observations in Table:  2000
##
##
##                  | predictedvalidation.label
## validation.label |         0 |         1 | Row Total |
## -----------------|-----------|-----------|-----------|
##                0 |      1805 |         3 |      1808 |
##                  |     0.998 |     0.002 |     0.904 |
##                  |     0.967 |     0.022 |           |
##                  |     0.902 |     0.002 |           |
## -----------------|-----------|-----------|-----------|
##                1 |        61 |       131 |       192 |
##                  |     0.318 |     0.682 |     0.096 |
##                  |     0.033 |     0.978 |           |
##                  |     0.030 |     0.066 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |      1866 |       134 |      2000 |
##                  |     0.933 |     0.067 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
#4th question
set.seed(8)
gridk<-expand.grid(k=bestk)
model_k<-train(Personal.Loan~.,data=trainingset.norm,method='knn',tuneGrid=gridk)
model_k
```

```
## 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:
##
##   Accuracy   Kappa
##   0.9497157  0.6688587
##
## Tuning parameter 'k' was held constant at a value of 5
```

```
customer_k<-predict(model_k,testingset.norm)
customer_k
```

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

```
#5th question
set.seed(8)
```

```
train2<-createDataPartition(ubank_dummy$Personal.Loan,p=0.5,list = FALSE)
trainingset2<-ubank_dummy[train2,]
x<-ubank_dummy[-train2,]
train3<-createDataPartition(x$Personal.Loan,p=0.6,list = FALSE)
validationset2<-x[train3,]
testingset2<-x[-train3,]
nrow(trainingset2)
```

```
## [1] 2500
```

```
nrow(testingset2)
```

```
## [1] 1000
```

```
nrow(validationset2)
```

```
## [1] 1500
```

```
normalvariables<-c('Age',"Experience","Income","Family","CCAvg","Mortgage",
                   "Securities.Account","CD.Account","Online","CreditCard",
                   "Education1","Education2","Education3")

normalization_values2<-preProcess(trainingset2[,normalvariables],method=c('center','scale'))

trainingset.norm2<-predict(normalization_values2,trainingset2)
validationset.norm2<-predict(normalization_values2,validationset2)
testingset.norm2<-predict(normalization_values2,testingset2)


training.label2<-trainingset.norm2[,7]
validation.label2<-validationset.norm2[,7]
testing.label2<-testingset.norm2[,7]

predictedvalidation.label2<-knn(trainingset.norm2,validationset.norm2,cl=training.label2,k=bestk)
predictedtesting.label2<-knn(trainingset.norm2,testingset.norm2,cl=training.label2,k=bestk)

CrossTable(x=validation.label2,y=predictedvalidation.label2,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1500
##
```

```
##
##                    | predictedvalidation.label2
## validation.label2 |         0 |         1 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |      1354 |         2 |      1356 |
##                   |     0.999 |     0.001 |     0.904 |
##                   |     0.966 |     0.020 |           |
##                   |     0.903 |     0.001 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |        48 |        96 |       144 |
##                   |     0.333 |     0.667 |     0.096 |
##                   |     0.034 |     0.980 |           |
##                   |     0.032 |     0.064 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |      1402 |        98 |      1500 |
##                   |     0.935 |     0.065 |           |
## ------------------|-----------|-----------|-----------|
##
##
```

```
CrossTable(x=testing.label2,y=predictedtesting.label2,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                | predictedtesting.label2
## testing.label2 |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |       904 |         0 |       904 |
##                |     1.000 |     0.000 |     0.904 |
##                |     0.962 |     0.000 |           |
##                |     0.904 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
##              1 |        36 |        60 |        96 |
##                |     0.375 |     0.625 |     0.096 |
##                |     0.038 |     1.000 |           |
##                |     0.036 |     0.060 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       940 |        60 |      1000 |
##                |     0.940 |     0.060 |           |
## ---------------|-----------|-----------|-----------|
##
##
```

```
table_validation<-table(validation.label2,predictedvalidation.label2)
confusionMatrix(table_validation)
```

```
## Confusion Matrix and Statistics
##
##                   predictedvalidation.label2
## validation.label2    0    1
##                 0 1354    2
##                 1   48   96
##
##                Accuracy : 0.9667
##                  95% CI : (0.9563, 0.9752)
##     No Information Rate : 0.9347
##     P-Value [Acc > NIR] : 2.878e-08
##
##                   Kappa : 0.776
##
##  Mcnemar's Test P-Value : 1.966e-10
##
##             Sensitivity : 0.9658
##             Specificity : 0.9796
##          Pos Pred Value : 0.9985
##          Neg Pred Value : 0.6667
##              Prevalence : 0.9347
##          Detection Rate : 0.9027
##    Detection Prevalence : 0.9040
##       Balanced Accuracy : 0.9727
##
##        'Positive' Class : 0
##
```

```
table_testing<-table(testing.label2,predictedtesting.label2)
confusionMatrix(table_testing)
```

```
## Confusion Matrix and Statistics
##
##               predictedtesting.label2
## testing.label2   0   1
##              0 904   0
##              1  36  60
##
##                Accuracy : 0.964
##                  95% CI : (0.9505, 0.9747)
##     No Information Rate : 0.94
##     P-Value [Acc > NIR] : 0.0004188
##
##                   Kappa : 0.7508
##
##  Mcnemar's Test P-Value : 5.433e-09
##
##             Sensitivity : 0.9617
##             Specificity : 1.0000
```

```
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6250
##              Prevalence : 0.9400
##          Detection Rate : 0.9040
##    Detection Prevalence : 0.9040
##       Balanced Accuracy : 0.9809
##
##        'Positive' Class : 0
##
```

```
#accuracy and sensitivity of validation set is greater than testing set
```