Assignment 2

Tongxiang Lu

2022-09-29

```
library(caret)
```

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

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

```
library(lattice)
```

```
library(class)
```

```
library(readr)
universalbank <- read.csv("universalbank.csv")
colnames(universalbank) <- c('ID', 'Age','Experience','Income','ZIP Code','Family','CCAvg','Education',
summary(universalbank)
```

```
##        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
```

```
# Question 1: Use "Null" function to remove variables not included in the mode, then transform characte
universalbank$ID <- NULL
universalbank$`ZIP Code` <- NULL
universalbank$`Personal.Loan`= as.factor(universalbank$`Personal.Loan`)
summary(universalbank)
```

```
##       Age          Experience       Income           Family
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.0   Median : 64.00   Median :2.000
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg          Education        Mortgage     Personal.Loan
##  Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4520
##  1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 480
##  Median : 1.500   Median :2.000   Median :  0.0
##  Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :10.000   Max.   :3.000   Max.   :635.0
##  Securities.Account   CD.Account          Online          CreditCard
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000     Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.1044     Mean   :0.0604   Mean   :0.5968   Mean   :0.294
##  3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

```
# Normalize the whole data before running the model.
Norm_model <- preProcess(universalbank, method = c("center", "scale"))
universalbank_norm=predict(Norm_model,universalbank)
summary(universalbank_norm)
```

```
##       Age             Experience            Income            Family
##  Min.   :-1.94871   Min.   :-2.014710   Min.   :-1.4288   Min.   :-1.2167
##  1st Qu.:-0.90188   1st Qu.:-0.881116   1st Qu.:-0.7554   1st Qu.:-1.2167
##  Median :-0.02952   Median :-0.009121   Median :-0.2123   Median :-0.3454
##  Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84284   3rd Qu.: 0.862874   3rd Qu.: 0.5263   3rd Qu.: 0.5259
##  Max.   : 1.88967   Max.   : 1.996468   Max.   : 3.2634   Max.   : 1.3973
##      CCAvg           Education          Mortgage       Personal.Loan
##  Min.   :-1.1089   Min.   :-1.0490   Min.   :-0.5555   0:4520
##  1st Qu.:-0.7083   1st Qu.:-1.0490   1st Qu.:-0.5555   1: 480
##  Median :-0.2506   Median : 0.1417   Median :-0.5555
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3216   3rd Qu.: 1.3324   3rd Qu.: 0.4375
##  Max.   : 4.6131   Max.   : 1.3324   Max.   : 5.6875
##  Securities.Account   CD.Account          Online          CreditCard
##  Min.   :-0.3414    Min.   :-0.2535   Min.   :-1.2165   Min.   :-0.6452
##  1st Qu.:-0.3414    1st Qu.:-0.2535   1st Qu.:-1.2165   1st Qu.:-0.6452
##  Median :-0.3414    Median :-0.2535   Median : 0.8219   Median :-0.6452
##  Mean   : 0.0000    Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.3414    3rd Qu.:-0.2535   3rd Qu.: 0.8219   3rd Qu.: 1.5495
##  Max.   : 2.9286    Max.   : 3.9438   Max.   : 0.8219   Max.   : 1.5495
```

```
universalbank_norm$Personal.Loan = universalbank$Personal.Loan
```

```
# Partition the data
train_index = createDataPartition(universalbank$Personal.Loan,p=0.6, list=FALSE)
train.df=universalbank_norm[train_index,]
Validation.df=universalbank_norm[-train_index,]
# Create the test data
To_Predict=data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,
                      CD.Account = 0, Online = 1,CreditCard = 1)
print(To_Predict)
```

```
##   Age Experience Income Family CCAvg Education Mortgage Securities.Account
## 1  40         10     84      2     2         1        0                  0
##   CD.Account Online CreditCard
## 1          0      1          1
```

```
To_Predict_norm<-predict(Norm_model,To_Predict)
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -1.048973 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

```
Prediction <-knn(train=train.df[,1:7,9:12],
                 test=To_Predict_norm[,1:7,9:12],
                 cl=train.df$Personal.Loan,
                 k=1)
```

```
## Warning in drop && !has.j: 'length(x) = 4 > 1' in coercion to 'logical(1)'

## Warning in drop && length(y) == 1L: 'length(x) = 4 > 1' in coercion to
## 'logical(1)'

## Warning in drop && !mdrop: 'length(x) = 4 > 1' in coercion to 'logical(1)'

## Warning in drop && !has.j: 'length(x) = 4 > 1' in coercion to 'logical(1)'

## Warning in drop && length(y) == 1L: 'length(x) = 4 > 1' in coercion to
## 'logical(1)'

## Warning in drop && !mdrop: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
print(Prediction)
```

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

```
#Question 2   k=1 got highest accuracy of 0.953.
fitControl <- trainControl(method = "repeatedcv",
                           number = 3,
                           repeats = 2)
searchGrid = expand.grid(k = 1:10)
Knn.model = train(Personal.Loan~.,
                data = train.df,
                method = 'knn',
                tuneGrid = searchGrid,
                trControl = fitControl,)
Knn.model
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   11 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 2 times)
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9505000  0.6784999
##    2  0.9455000  0.6475715
##    3  0.9531667  0.6806574
##    4  0.9530000  0.6738620
##    5  0.9533333  0.6730544
##    6  0.9508333  0.6511946
##    7  0.9501667  0.6435187
##    8  0.9478333  0.6227036
##    9  0.9488333  0.6287298
##   10  0.9491667  0.6275120
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
#Question3        Accuracy recoded as 0.958.
predictions <- predict(Knn.model,Validation.df)
confusionMatrix(predictions,Validation.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1801   85
##          1    7  107
##
##                Accuracy : 0.954
##                  95% CI : (0.9439, 0.9628)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6762
##
##  Mcnemar’s Test P-Value : 9.923e-16
##
##             Sensitivity : 0.9961
##             Specificity : 0.5573
##          Pos Pred Value : 0.9549
##          Neg Pred Value : 0.9386
##              Prevalence : 0.9040
##          Detection Rate : 0.9005
##    Detection Prevalence : 0.9430
##       Balanced Accuracy : 0.7767
##
##        ’Positive’ Class : 0
##
```

```
#Question 4
To_Predict = data.frame(Age = 40,Experience = 10, Income = 84, Family = 2, CCAvg = 2,

To_Predict_norm = predict(Norm_model,To_Predict)
predict(Knn.model,To_Predict_norm)
```

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

```
#Question 5
splitSample <- sample(1:3, size=nrow(universalbank_norm), prob=c(0.5,0.3,0.2), replace = TRUE)
train_data <- universalbank_norm[splitSample==1,]
valid_data <- universalbank_norm[splitSample==2,]
test_data <- universalbank_norm[splitSample==3,]
Predict = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,
                     CD.Account = 0, Online = 1, CreditCard = 1)
print(Predict)
```

```
##   Age Experience Income Family CCAvg Education Mortgage Securities.Account
## 1  40         10     84      2     2         1        0                  0
##   CD.Account Online CreditCard
## 1          0      1          1
```

```
Predict_norm<-predict(Norm_model,Predict)
print(Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -1.048973 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

```
Prediction_newsplit <-knn(train=train.df[,1:7,9:12],
                          test=To_Predict_norm[,1:7,9:12],
                          cl=train.df$Personal.Loan,
                          k=1)
```

```
## Warning in drop && !has.j: ’length(x) = 4 > 1’ in coercion to ’logical(1)’

## Warning in drop && length(y) == 1L: ’length(x) = 4 > 1’ in coercion to
## ’logical(1)’

## Warning in drop && !mdrop: ’length(x) = 4 > 1’ in coercion to ’logical(1)’

## Warning in drop && !has.j: ’length(x) = 4 > 1’ in coercion to ’logical(1)’

## Warning in drop && length(y) == 1L: ’length(x) = 4 > 1’ in coercion to
## ’logical(1)’

## Warning in drop && !mdrop: ’length(x) = 4 > 1’ in coercion to ’logical(1)’
```

```
print(Prediction_newsplit)
```

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

```
fitControl2 <- trainControl(method = "repeatedcv",
                            number = 3,
                            repeats = 2)
                            searchGrid=expand.grid(k = 1:10)
Knn.model2 = train(Personal.Loan~.,
                  data=train.df,
                  method='knn',
                  tuneGrid=searchGrid,
                  trControl = fitControl2,)
Knn.model2
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   11 predictor
##    2 classes: ’0’, ’1’
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 2 times)
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9528333  0.6878298
##    2  0.9460000  0.6539164
##    3  0.9531667  0.6815989
##    4  0.9520000  0.6715692
##    5  0.9528333  0.6676317
##    6  0.9511667  0.6536142
##    7  0.9508333  0.6458572
##    8  0.9481667  0.6223505
##    9  0.9475000  0.6155112
##   10  0.9453333  0.5947697
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

```
predictions2 <- predict(Knn.model2,Validation.df)
confusionMatrix(predictions2,Validation.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1796   75
##          1   12  117
##
##                Accuracy : 0.9565
##                  95% CI : (0.9466, 0.965)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7063
##
##  Mcnemar's Test P-Value : 2.989e-11
##
##             Sensitivity : 0.9934
##             Specificity : 0.6094
##          Pos Pred Value : 0.9599
##          Neg Pred Value : 0.9070
##              Prevalence : 0.9040
##          Detection Rate : 0.8980
##    Detection Prevalence : 0.9355
##       Balanced Accuracy : 0.8014
##
##        'Positive' Class : 0
##
```