Assignment2

SivaReddy

2/20/2022

colnames are arranged according CSV file

```
library(readr)
library(caret)
```

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

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

```
library(class)
universalbank <- read_csv("universalbank.csv")
```

```
## Rows: 5000 Columns: 14
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education, M...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(universalbank)<-c('ID', 'Age','Experience','Income','ZIP Code','Family','CCAvg','Education','M
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
```

```
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
```

#need to remove variables which are not included in the model with help of NUll Function.we should also transform character attributes to factors

```
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
```

#before running the model we should normalise the whole data here we used the z statistics #Question 1 #we have used KNN algoritham we are anble to get value as a zero

```
Train_Index = createDataPartition(universalbank$Personal.Loan,p=0.6, list=FALSE)
Train.df=universalbank_norm[Train_Index,]
Validation.df=universalbank_norm[-Train_Index,]
To_Predict=data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education= 1,Mortg

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)
```

```
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  0.9493333  0.6765816
##    2  0.9450000  0.6414392
##    3  0.9515000  0.6700428
##    4  0.9480000  0.6432075
##    5  0.9493333  0.6391731
##    6  0.9471667  0.6202557
##    7  0.9465000  0.6085459
##    8  0.9451667  0.5972825
##    9  0.9446667  0.5881896
##   10  0.9425000  0.5684346
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

#Question3 #Accuarcy 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 1799   66
##          1    9  126
##
##                Accuracy : 0.9625
##                  95% CI : (0.9532, 0.9704)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7509
##
##  Mcnemar's Test P-Value : 1.004e-10
##
##             Sensitivity : 0.9950
##             Specificity : 0.6562
##          Pos Pred Value : 0.9646
##          Neg Pred Value : 0.9333
##              Prevalence : 0.9040
##          Detection Rate : 0.8995
##    Detection Prevalence : 0.9325
##       Balanced Accuracy : 0.8256
##
##        'Positive' Class : 0
##
```

#Question4

```
To_Predict=data.frame(Age=40,Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education = 0, Mortga

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

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

#Question5

```
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, Education= 1,Mortgage

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)

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  0.9535000  0.7041302
##    2  0.9500000  0.6788207
##    3  0.9551667  0.6962717
##    4  0.9523333  0.6717114
##    5  0.9483333  0.6298623
##    6  0.9463333  0.6135520
##    7  0.9455000  0.5973830
##    8  0.9445000  0.5880937
##    9  0.9441667  0.5798294
```

```
##   10  0.9426667  0.5690234
##
## 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 1799   66
##          1    9  126
##
##                Accuracy : 0.9625
##                  95% CI : (0.9532, 0.9704)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7509
##
##  Mcnemar's Test P-Value : 1.004e-10
##
##             Sensitivity : 0.9950
##             Specificity : 0.6562
##          Pos Pred Value : 0.9646
##          Neg Pred Value : 0.9333
##              Prevalence : 0.9040
##          Detection Rate : 0.8995
##    Detection Prevalence : 0.9325
##       Balanced Accuracy : 0.8256
##
##        'Positive' Class : 0
##
```