kNN-Lab.R

oliverhering

2021-10-16

# Oliver Hering  
# kNN Lab   
# Data Mining and Big Data Fall 2021  
  
  
# library and reading data  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

data <- read.csv('UniversalBank.csv', header = TRUE)  
head(data)

## ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage  
## 1 1 25 1 49 91107 4 1.6 1 0  
## 2 2 45 19 34 90089 3 1.5 1 0  
## 3 3 39 15 11 94720 1 1.0 1 0  
## 4 4 35 9 100 94112 1 2.7 2 0  
## 5 5 35 8 45 91330 4 1.0 2 0  
## 6 6 37 13 29 92121 4 0.4 2 155  
## Personal.Loan Securities.Account CD.Account Online CreditCard  
## 1 0 1 0 0 0  
## 2 0 1 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 1  
## 6 0 0 0 1 0

# looking at data   
str(data)

## 'data.frame': 5000 obs. of 14 variables:  
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ ZIP.Code : int 91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : int 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Securities.Account: int 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD.Account : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : int 0 0 0 0 0 1 1 0 1 0 ...  
## $ CreditCard : int 0 0 0 0 1 0 0 1 0 0 ...

names(data)

## [1] "ID" "Age" "Experience"   
## [4] "Income" "ZIP.Code" "Family"   
## [7] "CCAvg" "Education" "Mortgage"   
## [10] "Personal.Loan" "Securities.Account" "CD.Account"   
## [13] "Online" "CreditCard"

# CLEANING DATA-------------------------  
data <- data[, -c(1, 5)]  
names(data)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Education"   
## [7] "Mortgage" "Personal.Loan" "Securities.Account"  
## [10] "CD.Account" "Online" "CreditCard"

# re ordering variables  
data <- data[, c(1:7, 9:12, 8)]  
names(data)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Education"   
## [7] "Mortgage" "Securities.Account" "CD.Account"   
## [10] "Online" "CreditCard" "Personal.Loan"

table(data$Personal.Loan)

##   
## 0 1   
## 4520 480

str(data)

## 'data.frame': 5000 obs. of 12 variables:  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : int 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Securities.Account: int 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD.Account : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : int 0 0 0 0 0 1 1 0 1 0 ...  
## $ CreditCard : int 0 0 0 0 1 0 0 1 0 0 ...  
## $ Personal.Loan : int 0 0 0 0 0 0 0 0 0 1 ...

# setting categorical variables as factors  
  
data$Education <-as.factor(data$Education)  
data$Securities.Account <-as.factor(data$Securities.Account)  
data$CD.Account <-as.factor(data$CD.Account)   
data$Online <-as.factor(data$Online)   
data$CreditCard <-as.factor(data$CreditCard)   
  
data$Personal.Loan <- factor(data$Personal.Loan,  
 levels = c('0', '1'),  
 labels = c('No', 'Yes'))  
str(data)

## 'data.frame': 5000 obs. of 12 variables:  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Securities.Account: Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 1 ...  
## $ CD.Account : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...  
## $ CreditCard : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...  
## $ Personal.Loan : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...

# TRAINING AND VALIDATION SPLIT---------------  
set.seed(666)  
  
train\_index <- sample(1:nrow(data), 0.6 \* nrow(data))  
valid\_index <- setdiff(1:nrow(data), train\_index)  
  
train <- data[train\_index, ]  
valid <- data[valid\_index, ]  
  
str(train)

## 'data.frame': 3000 obs. of 12 variables:  
## $ Age : int 66 49 32 28 45 33 38 56 44 31 ...  
## $ Experience : int 41 23 7 4 21 8 12 32 17 5 ...  
## $ Income : int 11 121 44 58 140 122 29 23 25 79 ...  
## $ Family : int 3 1 4 3 2 1 4 4 3 4 ...  
## $ CCAvg : num 0.1 4.9 0.8 1.5 7.6 0 0.2 0.7 1 2.2 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 3 1 1 1 1 1 1 1 2 2 ...  
## $ Mortgage : int 0 0 0 131 132 0 0 0 0 0 ...  
## $ Securities.Account: Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...  
## $ CD.Account : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Online : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 1 2 1 1 ...  
## $ CreditCard : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 2 2 1 ...  
## $ Personal.Loan : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...

str(valid)

## 'data.frame': 2000 obs. of 12 variables:  
## $ Age : int 25 39 35 50 34 29 48 59 67 42 ...  
## $ Experience : int 1 15 8 24 9 5 23 32 41 18 ...  
## $ Income : int 49 11 45 22 180 45 114 40 112 81 ...  
## $ Family : int 4 1 4 1 1 3 2 4 1 4 ...  
## $ CCAvg : num 1.6 1 1 0.3 8.9 0.1 3.8 2.5 2 2.4 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 1 1 2 3 3 2 3 2 1 1 ...  
## $ Mortgage : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Securities.Account: Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 2 1 ...  
## $ CD.Account : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 2 1 1 ...  
## $ CreditCard : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 1 1 ...  
## $ Personal.Loan : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 1 1 1 ...

# NORMALIZING  
  
train\_norm <- train  
valid\_norm <- valid  
  
  
# function  
norm\_function <- preProcess(train[, -c(6, 8:12)],  
 method = c('center',  
 'scale'))  
# train\_norm  
train\_norm[, -c(6, 8:12)] <- predict(norm\_function,  
 train[, -c(6, 8:12)])  
  
head(train\_norm)

## Age Experience Income Family CCAvg Education  
## 1598 1.80541475 1.82348943 -1.3615267 0.5125152 -1.0417945 3  
## 4734 0.31381987 0.24536857 1.0396523 -1.2287596 1.7094789 1  
## 873 -1.17777502 -1.15740553 -0.6411730 1.3831527 -0.6405672 1  
## 652 -1.52873852 -1.42042567 -0.3355684 0.5125152 -0.2393398 1  
## 1697 -0.03714364 0.07002181 1.4544014 -0.3581222 3.2570703 1  
## 1074 -1.09003415 -1.06973215 1.0614812 -1.2287596 -1.0991127 1  
## Mortgage Securities.Account CD.Account Online CreditCard Personal.Loan  
## 1598 -0.5495866 1 0 0 1 No  
## 4734 -0.5495866 0 0 1 0 No  
## 873 -0.5495866 0 0 1 0 No  
## 652 0.7604450 0 0 0 0 No  
## 1697 0.7704452 0 0 0 0 No  
## 1074 -0.5495866 0 0 1 1 No

table(train\_norm$Personal.Loan)

##   
## No Yes   
## 2692 308

# valid\_norm  
valid\_norm[, -c(6, 8:12)] <- predict(norm\_function,  
 valid[, -c(6, 8:12)])  
  
head(valid\_norm)

## Age Experience Income Family CCAvg Education Mortgage  
## 1 -1.7919612 -1.6834458 -0.5320285 1.3831527 -0.1820216 1 -0.5495866  
## 3 -0.5635889 -0.4560185 -1.3615267 -1.2287596 -0.5259308 1 -0.5495866  
## 5 -0.9145524 -1.0697321 -0.6193441 1.3831527 -0.5259308 2 -0.5495866  
## 8 0.4015607 0.3330420 -1.1214088 -1.2287596 -0.9271582 3 -0.5495866  
## 10 -1.0022933 -0.9820588 2.3275574 -1.2287596 4.0022069 3 -0.5495866  
## 12 -1.4409976 -1.3327523 -0.6193441 0.5125152 -1.0417945 2 -0.5495866  
## Securities.Account CD.Account Online CreditCard Personal.Loan  
## 1 1 0 0 0 No  
## 3 0 0 0 0 No  
## 5 0 0 0 1 No  
## 8 0 0 0 1 No  
## 10 0 0 0 0 Yes  
## 12 0 0 1 0 No

table(valid\_norm$Personal.Loan)

##   
## No Yes   
## 1828 172

# THE KNN MODEL----------------  
  
# k = 3  
knn\_model\_3 <- caret::knn3(Personal.Loan ~ ., data = train\_norm, k = 3)  
knn\_model\_3

## 3-nearest neighbor model  
## Training set outcome distribution:  
##   
## No Yes   
## 2692 308

# prediction and evaluation - training  
knn\_3\_predict\_train <- predict(knn\_model\_3, newdata = train\_norm[, -c(12)],  
 type = 'class')  
table(knn\_3\_predict\_train)

## knn\_3\_predict\_train  
## No Yes   
## 2758 242

confusionMatrix(knn\_3\_predict\_train, as.factor(train\_norm[, 12]),  
 positive = 'Yes')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2688 70  
## Yes 4 238  
##   
## Accuracy : 0.9753   
## 95% CI : (0.9691, 0.9806)  
## No Information Rate : 0.8973   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8521   
##   
## Mcnemar's Test P-Value : 4.153e-14   
##   
## Sensitivity : 0.77273   
## Specificity : 0.99851   
## Pos Pred Value : 0.98347   
## Neg Pred Value : 0.97462   
## Prevalence : 0.10267   
## Detection Rate : 0.07933   
## Detection Prevalence : 0.08067   
## Balanced Accuracy : 0.88562   
##   
## 'Positive' Class : Yes   
##

# k = 5  
knn\_model\_5 <- caret::knn3(Personal.Loan ~ ., data = train\_norm, k = 5)  
knn\_model\_5

## 5-nearest neighbor model  
## Training set outcome distribution:  
##   
## No Yes   
## 2692 308

# prediction and evaluation - training  
knn\_5\_predict\_train <- predict(knn\_model\_5, newdata = train\_norm[, -c(12)],  
 type = 'class')  
table(knn\_5\_predict\_train)

## knn\_5\_predict\_train  
## No Yes   
## 2782 218

confusionMatrix(knn\_5\_predict\_train, as.factor(train\_norm[, 12]))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2687 95  
## Yes 5 213  
##   
## Accuracy : 0.9667   
## 95% CI : (0.9596, 0.9728)  
## No Information Rate : 0.8973   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7922   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9981   
## Specificity : 0.6916   
## Pos Pred Value : 0.9659   
## Neg Pred Value : 0.9771   
## Prevalence : 0.8973   
## Detection Rate : 0.8957   
## Detection Prevalence : 0.9273   
## Balanced Accuracy : 0.8449   
##   
## 'Positive' Class : No   
##

# k = 7  
knn\_model\_7 <- caret::knn3(Personal.Loan ~ ., data = train\_norm, k = 7)  
knn\_model\_7

## 7-nearest neighbor model  
## Training set outcome distribution:  
##   
## No Yes   
## 2692 308

# prediction and evaluation - training  
knn\_7\_predict\_train <- predict(knn\_model\_7, newdata = train\_norm[, -c(12)],  
 type = 'class')  
table(knn\_7\_predict\_train)

## knn\_7\_predict\_train  
## No Yes   
## 2814 186

confusionMatrix(knn\_7\_predict\_train, as.factor(train\_norm[, 12]))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 2691 123  
## Yes 1 185  
##   
## Accuracy : 0.9587   
## 95% CI : (0.9509, 0.9655)  
## No Information Rate : 0.8973   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.728   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9996   
## Specificity : 0.6006   
## Pos Pred Value : 0.9563   
## Neg Pred Value : 0.9946   
## Prevalence : 0.8973   
## Detection Rate : 0.8970   
## Detection Prevalence : 0.9380   
## Balanced Accuracy : 0.8001   
##   
## 'Positive' Class : No   
##

# VALIDATION SET--------------  
knn\_3\_predict\_valid <- predict(knn\_model\_3, newdata = valid\_norm[, -c(12)],  
 type = 'class')  
table(knn\_3\_predict\_valid)

## knn\_3\_predict\_valid  
## No Yes   
## 1897 103

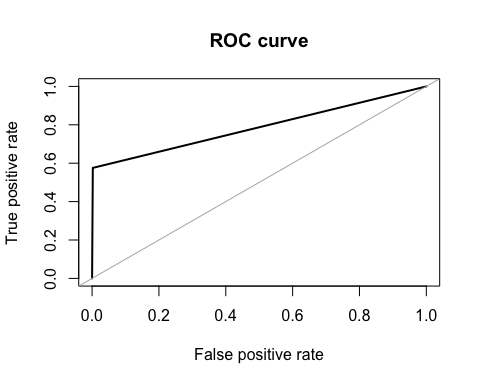
confusionMatrix(knn\_3\_predict\_valid, as.factor(valid\_norm[, 12]),  
 positive = 'Yes')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1824 73  
## Yes 4 99  
##   
## Accuracy : 0.9615   
## 95% CI : (0.9521, 0.9695)  
## No Information Rate : 0.914   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7007   
##   
## Mcnemar's Test P-Value : 9.239e-15   
##   
## Sensitivity : 0.5756   
## Specificity : 0.9978   
## Pos Pred Value : 0.9612   
## Neg Pred Value : 0.9615   
## Prevalence : 0.0860   
## Detection Rate : 0.0495   
## Detection Prevalence : 0.0515   
## Balanced Accuracy : 0.7867   
##   
## 'Positive' Class : Yes   
##

# ROS CURVE--------------  
library(ROSE)

## Loaded ROSE 0.0-4

ROSE::roc.curve(valid\_norm$Personal.Loan,  
 knn\_3\_predict\_valid)



## Area under the curve (AUC): 0.787

# PREDICTION AND EVALUATION NEW CUSTOMER-------------  
  
new\_customer <- data.frame(Age = 40,  
 Experience = 10,  
 Income = 84,  
 Family = 2,  
 CCAvg = 2,  
 Education = 2,  
 Mortgage = 0,  
 Securities.Account = 0,  
 CD.Account = 0,  
 Online = 1,  
 CreditCard = 1)  
new\_customer

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

new\_customer$Education <- as.factor(new\_customer$Education)  
new\_customer$Securities.Account <-as.factor(new\_customer$Securities.Account)  
new\_customer$CD.Account <-as.factor(new\_customer$CD.Account)   
new\_customer$Online <-as.factor(new\_customer$Online)  
new\_customer$CreditCard <-as.factor(new\_customer$CreditCard)  
  
str(new\_customer)

## 'data.frame': 1 obs. of 11 variables:  
## $ Age : num 40  
## $ Experience : num 10  
## $ Income : num 84  
## $ Family : num 2  
## $ CCAvg : num 2  
## $ Education : Factor w/ 1 level "2": 1  
## $ Mortgage : num 0  
## $ Securities.Account: Factor w/ 1 level "0": 1  
## $ CD.Account : Factor w/ 1 level "0": 1  
## $ Online : Factor w/ 1 level "1": 1  
## $ CreditCard : Factor w/ 1 level "1": 1

# normalizing new customer  
new\_customer\_norm <- predict(norm\_function, new\_customer)  
  
# prediction and evaluation  
new\_customer\_predict <- predict(knn\_model\_3,  
 newdata = new\_customer\_norm,  
 type = 'class')  
new\_customer\_predict

## [1] No  
## Levels: No Yes

#INTERPRETATION------------   
# knn = 3 gives the most accurate prediction  
# accuracy of both train and valid sets are high indicating no presence of overfitting  
# new customer will not accept the offer according to our model