MichaelBasta_Assignement_2

Michael Basta

2022-10-02

```
# Question 1
UB <- read.csv("C:\\Kent State\\Fall 2022\\Fundamentals of Machine Leanring\\Module 4\\UniversalBank.cs

#install.packages("fastDummies")
library(class)
library(ISLR)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.2.1
```

```
summary(UB)
```

```
##        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
```

```
# Converting Education to dummy variable
UB <- dummy_cols(UB, select_columns = "Education")
Age <- 40
Experience <- 10
Income <- 84
Family <- 2
CCAvg <- 2
Mortgage <- 0
Securities_Acc <- 0
CD_Account <- 0
Online <- 1
Credit_Card <- 1
Education_1 <- 0
Education_2 <- 1
Education_3 <- 0

# Adding Values to be predicted at the top row of data
UB[1,] <- c(1, Age, Experience, Income, 0 ,Family, CCAvg, 0, Mortgage, 0, Securities_Acc, CD_Account, Or
norm_model <- preProcess(UB, method = c('range'))
UB_normalized <- predict(norm_model, UB)


# Drop Columns ID, Zip Code, Original "Education" not needed after converting to dummy variable
UB_normalized <- UB_normalized[,-c(1,5,8)]

Index_Train <- createDataPartition(UB_normalized$Personal.Loan, p=0.6, list = FALSE)
Train <- UB_normalized[Index_Train,]
Test <- UB_normalized[-Index_Train,]

Train_Predictors <- Train[,c(1:6,8:14)]
Test_Predictors <- Test[,c(1:6,8:14)]

Train_labels <- Train[,7]
Test_labels <- Test[,7]

Predicted_Test_labels <- knn(Train_Predictors, Test_Predictors, cl=Train_labels, k=1, prob = TRUE)
class_prob <- attr(Predicted_Test_labels, 'prob')

# The first value is the one needs to be predicted
head(class_prob)
```

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

```
paste("Customer will accept loan offer")
```

```
## [1] "Customer will accept loan offer"
```

```
# Question 2
set.seed(123)
model <- train(Personal.Loan~Age+Experience+Income+Family+CCAvg+Mortgage+Securities.Account+CD.Account+
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
model
```

```
## k-Nearest Neighbors
##
## 5000 samples
##   13 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 5000, 5000, 5000, 5000, 5000, 5000, ...
## Resampling results across tuning parameters:
##
##   k  RMSE       Rsquared   MAE
##   5  0.1906194  0.5908494  0.05314157
##   7  0.1932165  0.5845541  0.05798578
##   9  0.1955883  0.5803150  0.06181414
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 5.
```

```
# Question 3
#install.packages("gmodels")
library("gmodels")
```

```
## Warning: package 'gmodels' was built under R version 4.2.1
```

```
CrossTable(x=Test_labels, y=Predicted_Test_labels, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | Predicted_Test_labels
##  Test_labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1809 |        18 |      1827 |
##              |     0.990 |     0.010 |     0.913 |
##              |     0.972 |     0.129 |           |
##              |     0.904 |     0.009 |           |
## -------------|-----------|-----------|-----------|
##            1 |        52 |       121 |       173 |
##              |     0.301 |     0.699 |     0.086 |
##              |     0.028 |     0.871 |           |
##              |     0.026 |     0.060 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1861 |       139 |      2000 |
##              |     0.930 |     0.070 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```r
# Question 4

Predicted_Test_labels_bestK <- knn(Train_Predictors, Test_Predictors, cl=Train_labels, k=5, prob = TRUE)
class_prob_bestK <- attr(Predicted_Test_labels_bestK, 'prob')

# The first value is the one needs to be predicted
head(class_prob_bestK)
```

```
## [1] 1.0 1.0 1.0 0.8 1.0 1.0
```

```r
paste("Customer will accept loan offer")
```

```
## [1] "Customer will accept loan offer"
```

```r
# Question 5

# Partitioning the data into
# 50% training 30% Validation 20% Testing

# Taking the test portion from the data to apply the model
# 20% * 5000 = 1000
UB_Test_Normalized <- UB_normalized[4000:5000,]
UB_normalized <- UB_normalized[1:4000,]

# training is 2500
# 2500 / 4000 = 0.625
Index_Train <- createDataPartition(UB_normalized$Personal.Loan, p=0.625, list = FALSE)
Train <- UB_normalized[Index_Train,]
Validation <- UB_normalized[-Index_Train,]

Train_Predictors <- Train[,c(1:6,8:14)]
Validation_Predictors <- Validation[,c(1:6,8:14)]

Test_Predictors <- UB_Test_Normalized[,c(1:6,8:14)]
```

```
Train_labels <- Train[,7]
Validation_labels <- Validation[,7]

Test_labels <- UB_Test_Normalized[,7]

Predicted_Validation_labels <- knn(Train_Predictors, Validation_Predictors, cl=Train_labels, k=5, prob =
CrossTable(x=Validation_labels, y=Predicted_Validation_labels, prop.chisq = FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  1500 
## 
## 
##                   | Predicted_Validation_labels 
## Validation_labels |         0 |         1 | Row Total | 
## ------------------|-----------|-----------|-----------|
##                 0 |      1339 |         7 |      1346 | 
##                   |     0.995 |     0.005 |     0.897 | 
##                   |     0.944 |     0.085 |           | 
##                   |     0.893 |     0.005 |           | 
## ------------------|-----------|-----------|-----------|
##                 1 |        79 |        75 |       154 | 
##                   |     0.513 |     0.487 |     0.103 | 
##                   |     0.056 |     0.915 |           | 
##                   |     0.053 |     0.050 |           | 
## ------------------|-----------|-----------|-----------|
##      Column Total |      1418 |        82 |      1500 | 
##                   |     0.945 |     0.055 |           | 
## ------------------|-----------|-----------|-----------|
## 
## 
```

```
Predicted_Test_Labels<- knn(Train_Predictors, Test_Predictors, cl=Train_labels, k=5, prob = TRUE)
CrossTable(x=Test_labels, y=Predicted_Test_Labels, prop.chisq = FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1001
##
##
##              | Predicted_Test_Labels
##  Test_labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       916 |         2 |       918 |
##              |     0.998 |     0.002 |     0.917 |
##              |     0.955 |     0.048 |           |
##              |     0.915 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |        43 |        40 |        83 |
##              |     0.518 |     0.482 |     0.083 |
##              |     0.045 |     0.952 |           |
##              |     0.043 |     0.040 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       959 |        42 |      1001 |
##              |     0.958 |     0.042 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
paste("It looks like there's way less misclassified cases when we applied it on the test data than the 
```

```
## [1] "It looks like there’s way less misclassified cases when we applied it on the test data than the
```