Untitled

2023-10-02

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
UniversalBank <- read.csv("C:\\Users\\13308\\OneDrive\\Documents\\UniversalBank.csv")

data <- UniversalBank

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

## Loading required package: lattice
```

```r
library(class)
```

```
## Warning: package 'class' was built under R version 4.2.3
```

```r
Loan_normalized <- data[, -1]
Loan_normalized <- Loan_normalized[,-4]

str(Loan_normalized)
```

```
## '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 ...
##  $ 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 ...
```

```r
norm_model <- preProcess(Loan_normalized, method = c('range'))
Loan_normalized <- predict(norm_model,Loan_normalized)


Index_Train <- createDataPartition(Loan_normalized$`Personal.Loan`, p=0.6, list=FALSE)
Train <- Loan_normalized[Index_Train,]
Validation <- Loan_normalized[-Index_Train,]

Train_Predictors <- Train[1:3000, c(1:7, 9:12)]
Val_Predictors <- Validation[1:2000, c(1:7, 9:12)]


Train_labels <- Train[, 8]
Val_labels <- Validation [, 8]

set.seed(123)

Predicted_Val_labels <- knn(Train_Predictors,
                            Val_Predictors,
                            cl=Train_labels,
                            k = 1)

new_data <- data.frame(
  Age = c(40),
  Experience = c(10),
  Income = c(84),
  Family = c(2),
  CCAvg = c(2),
  Education = c(2),
  Mortgage = c(0),
  "Securities Account" = c(0),
  "CD Account" = c(0),
  Online = c(1),
  CreditCard = c(1)
)

names(new_data) <- c("Age","Experience","Income","Family","CCAvg","Education","Mortgage","Securities Ac
new_data
```

```
##   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
```

```r
norm_model2 <- preProcess(new_data, method = c('range'))
```

```
## Warning in preProcess.default(new_data, method = c("range")): No variation for
## for: Age, Experience, Income, Family, CCAvg, Education, Mortgage, Securities
## Account, CD Account, Online, CreditCard
```

```
## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"):
## STATS is longer than the extent of 'dim(x)[MARGIN]'
```

```r
new_data_normalized <- predict(norm_model2, new_data)
```

```
## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of ’dim(x)[MARGIN]’
```

```r
print("any")
```

```
## [1] "any"
```

```r
predicted_loan_status <- knn(
  Train_Predictors,
  new_data_normalized,
  cl = Train_labels,
  k = 1
)

print(predicted_loan_status)
```

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

```r
#1: This customer would accept a loan offer (1).
#2:

k_values <- seq(1, 50, by = 1)

results <- data.frame(k = numeric(length(k_values)), Accuracy = numeric(length(k_values)))


for (i in 1:length(k_values)) {
  set.seed(123)

  Predicted_Test_labels <- knn(
    Train_Predictors,
    Val_Predictors,
    cl = Train_labels,
    k = k_values[i]
  )

  accuracy <- sum(Predicted_Val_labels == Val_labels) / length(Val_labels)

  results[i, ] <- c(k = k_values[i], Accuracy = accuracy)
}

best_k <- results[which.max(results$Accuracy), "k"]
print(paste("Best k value:", best_k))
```

```
## [1] "Best k value: 1"
```

```
print(paste("Best accuracy:", max(results$Accuracy)))
```

```
## [1] "Best accuracy: 0.955"
```

```
#2: Per above, the best k value is 1.
#3:

library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.2.3
```

```
CrossTable(x=Val_labels,y=Predicted_Val_labels, prop.chisq = FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  2000 
## 
## 
##              | Predicted_Val_labels 
##   Val_labels |         0 |         1 | Row Total | 
## -------------|-----------|-----------|-----------|
##            0 |      1787 |        18 |      1805 | 
##              |     0.990 |     0.010 |     0.902 | 
##              |     0.961 |     0.128 |           | 
##              |     0.893 |     0.009 |           | 
## -------------|-----------|-----------|-----------|
##            1 |        72 |       123 |       195 | 
##              |     0.369 |     0.631 |     0.098 | 
##              |     0.039 |     0.872 |           | 
##              |     0.036 |     0.061 |           | 
## -------------|-----------|-----------|-----------|
## Column Total |      1859 |       141 |      2000 | 
##              |     0.929 |     0.070 |           | 
## -------------|-----------|-----------|-----------|
## 
## 
```

```
#4:

new_data <- data.frame(
  Age = c(40),
  Experience = c(10),
  Income = c(84),
  Family = c(2),
  CCAvg = c(2),
  Education = c(2),
  Mortgage = c(0),
  "Securities Account" = c(0),
  "CD Account" = c(0),
  Online = c(1),
  CreditCard = c(1)
)

names(new_data) <- c("Age","Experience","Income","Family","CCAvg","Education","Mortgage","Securities Ac
new_data
```

```
##   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
```

```
norm_model2 <- preProcess(new_data, method = c('range'))
```

```
## Warning in preProcess.default(new_data, method = c("range")): No variation for
## for: Age, Experience, Income, Family, CCAvg, Education, Mortgage, Securities
## Account, CD Account, Online, CreditCard

## Warning in preProcess.default(new_data, method = c("range")): STATS is longer
## than the extent of 'dim(x)[MARGIN]'
```

```
new_data_normalized <- predict(norm_model2, new_data)
```

```
## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'
```

```
print("any")
```

```
## [1] "any"
```

```
predicted_loan_status <- knn(
  Train_Predictors,
  new_data_normalized,
  cl = Train_labels,
  k = 1
)

print(predicted_loan_status)
```

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

```
#4: This customer would also accept a loan offer (1), however it is worth noting that the customer info
```