assignment3_machine-learning-811223821

yada akhil

2022-10-06

#setting default values to get a clean output

#loading all the required packages.

```
library(e1071)
library(ISLR)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(FNN)
library(gmodels)

library(pivottabler)

library(reshape)

library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

#reading the data as well as data partiton

```
UniversalBank <- read.csv("C:/Users/sudhakar/Downloads/UniversalBank (1).csv")

set.seed(15)


Index <- createDataPartition(UniversalBank$Income, p = 0.6, list = FALSE)

trainingData <- UniversalBank[Index,]

set.seed(15)
```

```
testData <- UniversalBank[-Index,]
```

A.create a pivot for the training data with online as a column variable,CC as a row variabe, and loan as a secondary row variable.

```
set.seed(15)
Melt_training = melt(trainingData,id=c("CreditCard","Personal.Loan"),variable= "Online")

cast_training=dcast(Melt_training,CreditCard+Personal.Loan~Online)
```

```
## Aggregation function missing: defaulting to length
```

```
set.seed(15)
cast_training <-cast_training[c(1,2,14)]

cast_training
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1904
## 2          0             1    200
## 3          1             0    814
## 4          1             1     84
```

```
set.seed(15)
```

#b. Creation of pivot tables for the training data where one will loan (rows) as a function of online(columns)and the other will have loan (rows) as a function of CC (columns)

```
set.seed(15)

Melt_training1 <- melt(trainingData,id=c("Personal.Loan"),variable = "Online")

cast_training1  <- dcast(Melt_training1,Personal.Loan~Online)
```

```
## Aggregation function missing: defaulting to length
```

```
cast_training1 <-cast_training1[c(1,13)]

cast_training1
```

```
##   Personal.Loan Online
## 1             0   2718
## 2             1    284
```

```r
set.seed(15)

Melt_training2 <- melt(trainingData,id=c("CreditCard"),variable = "Online")

cast_training2 <- dcast(Melt_training2,CreditCard~Online)
```

```
## Aggregation function missing: defaulting to length
```

```r
cast_training2 <-cast_training2[c(1,14)]

cast_training2
```

```
##   CreditCard Online
## 1          0   2104
## 2          1    898
```

#D.complete the following quantities p(A|B)

```r
set.seed(15)
trainingData1 <- trainingData[c(13,10,14)]



table(trainingData1[,c(3,2)])
```

```
##            Personal.Loan
## CreditCard    0    1
##          0 1904  200
##          1  814   84
```

```r
#I) 84/(84+200) =  0.29577464788

#IV)  814/(814+1904) = 0.29948491537



table(trainingData1[,c(1,2)])
```

```
##        Personal.Loan
## Online    0    1
##      0 1108  112
##      1 1610  172
```

```r
#II)  172/(112+172) = 0.60563380281

#V) 1610/(1610+1108) = 0.5923473142



table(trainingData1[,c(2)])
```

```
##
##    0    1
## 2718  284
```

```
#III) 284/(284+2718) = 0.0946035976

#VI)  2718/(284+2693) = 0.91299966409

set.seed(15)
```

E. Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1).

```
# calculation using above values as below

#0.29577464788 * 0.60563380281 * 0.0946035976/((0.0946035976 * 0.60563380281 * 0.29577464788) +( 0.9129
```

#f. by comparing the value obtained by using teh navie bays probabilty the value obtained in step b #g.run the navie bayes model

```
set.seed(15)

naivebayes = naiveBayes(Personal.Loan~.,data=trainingData1)

naivebayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.9053964 0.0946036
##
## Conditional probabilities:
##    Online
## Y        [,1]      [,2]
##   0 0.5923473 0.4914884
##   1 0.6056338 0.4895768
##
##    CreditCard
## Y        [,1]      [,2]
##   0 0.2994849 0.4581167
##   1 0.2957746 0.4571958
```

```
set.seed(15)
```

0.2957746 * 0.6056338 * 0.0946036 /((0.0946035976 * 0.60563380281 * 0.29577464788) +(0.2994849 * 0.5923473 * 0.9053964))

which gives 0.09543910129

Hence we can say that the results of naive bayes are similar to those of the previous methods i.e 0.09543910129 & which is the same as the previous response 0.09471959475 in E.).