NaiveBayes_Assignment_ML

Abhishek_Sau

07/03/2021

#Setting up Working Directory, importing dataset & changing to factors

```
setwd("D:\\Study\\Assignments\\MachLearn\\MachLearnAssignment3_Naive_Bayes")
unibank_main<-read.csv("UniversalBank.csv")

unibank_main$Personal.Loan <- as.factor(unibank_main$Personal.Loan)
unibank_main$CreditCard <- as.factor(unibank_main$CreditCard)
unibank_main$Online <- as.factor(unibank_main$Online)
```

Splitting the data

```
library(caret)
```

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

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

```
set.seed(111)

split_index <- createDataPartition(unibank_main$Age,1, p=0.6, list = FALSE)

Train_UB <- unibank_main[split_index,]
Test_UB <- unibank_main[-split_index,]
```

#Q1 - Pivot - Online as a column variable, CC as a row variable, and Loan as a secondary row variable.

```
library(reshape)

train_melt <- melt(Train_UB, id=c("CreditCard", "Personal.Loan"), measure.vars = "Online")
train_cast <- cast(train_melt, CreditCard+Personal.Loan~variable, value = "Online")
```

```
## Aggregation requires fun.aggregate: length used as default
```

The result of this melted and casted training set shows the frequency of customers who use Online banking services in different combinations of if or not they have a CC and Personal.loan

#Q2 - Consider the task of classifying a customer who owns a bank credit card and is actively using online banking services. Looking at the pivot table, what is the probability that this customer will accept the loan offer?

Answer - 79/79+782 = 0.09175377468 or 9.175%.

9.17% is the probability of a customer who has a bank CC and actively uses Online banking services, as per the pivot table created in the above step.

#Q3 Create two separate pivot tables for the training data. One will have Loan (rows) as a function of Online (columns) and the other will have Loan (rows) as a function of CC.

```
train_melt1<- melt(Train_UB, id="Personal.Loan", measure.vars = "Online")
train_melt2<- melt(Train_UB, id="CreditCard", measure.vars="Online")

train_cast1 <- cast(train_melt1,Personal.Loan~variable, value = "Online" )
```

```
## Aggregation requires fun.aggregate: length used as default
```

```
train_cast2 <- cast(train_melt2, CreditCard~variable, value = "Online")
```

```
## Aggregation requires fun.aggregate: length used as default
```

#Q4 - Compute the following quantities [P(A | B) means "the probability ofA given B"]:

```
# i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)

table(Train_UB[,c(10,14)])
```

```
##              CreditCard
## Personal.Loan    0    1
##             0 1948  782
##             1  192   79
```

```
#Answer - P(CC=1|Loan=1) = 79/(79+192)= 0.2915 or 29.15%

# ii. P(Online = 1 | Loan = 1)
table(Train_UB[,c(10,13)])
```

```
##              Online
## Personal.Loan    0    1
##             0 1141 1589
##             1  107  164
```

```
#Answer-  P(Online=1|Loan=1)= 164/(107+164)= 0.6051 or 60.51%

# iii. P(Loan = 1) (the proportion of loan acceptors)

table(Train_UB[,10])
```

```
##
##    0    1
## 2730  271
```

```
#Answer- P(Loan = 1) = 271/3001 = 0.0903 or 9.03%

#iv. P(CC = 1 | Loan = 0)
table(Train_UB[,c(10,14)])
```

```
##              CreditCard
## Personal.Loan    0    1
##             0 1948  782
##             1  192   79
```

```
#Answer-  P(CC=1|Loan=0)= 782/(782+1948) = 0.2864 or 28.64%

#v. P(Online = 1 | Loan = 0)
table(Train_UB[,c(10,13)])
```

```
##              Online
## Personal.Loan    0    1
##             0 1141 1589
##             1  107  164
```

```
#Answer- P(Online = 1 | Loan = 0) = 1589/(1589+1141) = 0.5821 or 58.21%

#vi. P(Loan = 0)

table(Train_UB[,10])
```

```
##
##    0    1
## 2730  271
```

```
#Answer- P(Loan = 0) = 2730/(2730+271)= 0.9097 or 90.97%
```

#Q5 - Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1).

Naive Bayes probability =

$P(\text{Loan} = 1 \mid \text{CC} = 1, \text{Online} = 1) = P(\text{CC}{=}1|\text{Loan}{=}1)\,P(\text{Online}{=}1|\text{Loan}{=}1)P(\text{Loan}{=}1)/[(P(\text{CC}{=}1|\text{Loan}{=}1)\,P(\text{Online}{=}1|\text{Loan}{=}1)P(\text{Loan}{=}1))+(P(\text{CC}{=}1|\text{Loan}{=}0$

$= 0.2915\,0.6051\,0.0903/(0.2915\,0.6051\,0.0903)+(0.2864\,0.5821\,0.9097)$

$= 0.09504$ or 9.50%

#Q6 - Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate?

Answer - The value from the pivot table is 9.175% and the value computed from Naive Bayes probability is 9.504%. We can see here the different is significant.The difference is because of the assumption of Conditional Independence in the Naive Bayes formula.For a smaller dataset, the exact values are easy to be calculated. but for bigger chunks of data Naive bayes probability will be preferred based on theinsignificant difference in the probabilities from the Pivot and Naive Bayes formula.

#Q7 - Which of the entries in this table are needed for computing P (Loan = 1 | CC = 1, Online = 1)? In R, run naive Bayes on the data. Examine the model output on training data, and find the entry that corresponds to P (Loan = 1 | CC = 1, Online = 1).Compare this to the number you obtained in (e).

```
library(e1071)

NBmodel <- naiveBayes(Personal.Loan~.,Train_UB)
NBmodel
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90969677 0.09030323
##
## Conditional probabilities:
##    ID
## Y       [,1]     [,2]
##   0 2509.278 1450.056
##   1 2469.624 1377.653
##
##    Age
## Y       [,1]     [,2]
##   0 45.47839 11.45708
##   1 43.74908 11.55774
##
##    Experience
## Y       [,1]     [,2]
##   0 20.23040 11.44330
##   1 18.51661 11.57674
##
##    Income
## Y        [,1]     [,2]
##   0  66.49121 41.01906
##   1 145.88192 31.97095
##
##    ZIP.Code
## Y       [,1]     [,2]
##   0 93115.13 2377.586
##   1 93239.73 1790.063
##
##    Family
## Y       [,1]     [,2]
##   0 2.345421 1.154161
##   1 2.623616 1.128107
##
##    CCAvg
## Y       [,1]     [,2]
##   0 1.744403 1.579009
##   1 3.730369 2.167107
##
##    Education
## Y       [,1]      [,2]
##   0 1.857143 0.8479678
##   1 2.258303 0.7550296
##
##    Mortgage
## Y       [,1]      [,2]
```

```
##   0 51.80110  93.82882
##   1 99.46494 160.63763
##
##    Securities.Account
## Y        [,1]      [,2]
##   0 0.1043956 0.3058291
##   1 0.1217712 0.3276264
##
##    CD.Account
## Y         [,1]      [,2]
##   0 0.03223443 0.1766545
##   1 0.28782288 0.4535858
##
##    Online
## Y           0         1
##   0 0.4179487 0.5820513
##   1 0.3948339 0.6051661
##
##    CreditCard
## Y           0         1
##   0 0.7135531 0.2864469
##   1 0.7084871 0.2915129
```

```
pred_Test <- predict(NBmodel, Test_UB)

library("gmodels")

#Confusion Matrix of the Naive bayes model

CrossTable(Test_UB$Personal.Loan, pred_Test, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1999
##
##
##                      | pred_Test
## Test_UB$Personal.Loan |         0 |         1 | Row Total |
## ---------------------|-----------|-----------|-----------|
##                    0 |      1640 |       150 |      1790 |
##                      |     0.916 |     0.084 |     0.895 |
##                      |     0.949 |     0.554 |           |
##                      |     0.820 |     0.075 |           |
## ---------------------|-----------|-----------|-----------|
##                    1 |        88 |       121 |       209 |
##                      |     0.421 |     0.579 |     0.105 |
```

```
##                       |     0.051 |     0.446 |           |
##                       |     0.044 |     0.061 |           |
## ----------------------|-----------|-----------|-----------|
##          Column Total |      1728 |       271 |      1999 |
##                       |     0.864 |     0.136 |           |
## ----------------------|-----------|-----------|-----------|
##
##
```

```
# We misclassified 238 cases.
```