Assignment_3

MadhusudhanMasineni

3/4/2022

Naive Bayes

This is Assignment 3 for Naive Bayes Classifier.

```
library("readr")
library("caret")
```

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

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

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyr")
library("reshape2")
```

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

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library("e1071")


rm(list=ls())
uni_bank <- read.csv("C:\\Users\\madhu\\OneDrive\\Desktop\\MS\\1stSem\\2.Fundamentals of ML\\Modules\\M
uni_bank$Personal.Loan <- as.factor(uni_bank$Personal.Loan)
uni_bank$Online <- as.factor(uni_bank$Online)
uni_bank$CreditCard <- as.factor(uni_bank$CreditCard)

set.seed(64060)

train_index = createDataPartition(uni_bank$Personal.Loan, p=.6, list = F) #60% train
train_df = uni_bank[train_index,]
validation_df = uni_bank[-train_index,]
```

1. Create a pivot table for the training data with Online as a column variable, CC as a row variable, and Loan as a secondary row variable. The values inside the table should convey the count. In R use functions melt() and cast(), or function table()

```
melt_uni_bank = melt(train_df, id=c("CreditCard","Personal.Loan"), variable = "Online") # elongated DF
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
dcast_uni_bank = dcast(melt_uni_bank, CreditCard+Personal.Loan~Online)
```

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

```
dcast_uni_bank[,c(1:2,14)] # CreditCard, Personal.Loan, Online DF
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1924
## 2          0             1    195
## 3          1             0    788
## 4          1             1     93
```

2. 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? [This is the probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and being an active user of online banking services (Online = 1)].

the probability that this customer will accept the loan :: 93/3000=3.1%

3. 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.

```
loan_melt_unibank = melt(train_df, id=c("Personal.Loan"), variable = "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
cc_melt_unibank = melt(train_df, id=c("CreditCard"), variable = "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
dcast_loan_unibank = dcast(loan_melt_unibank, Personal.Loan~Online)
```

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

```
dcast_cc_unibank = dcast(cc_melt_unibank, CreditCard~Online)
```

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

```
dcast_loan_unibank[,c(1,13)]
```

```
##   Personal.Loan Online
## 1             0   2712
## 2             1    288
```

```
dcast_cc_unibank[,c(1,14)]
```

```
##   CreditCard Online
## 1          0   2119
## 2          1    881
```

4. 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) 93/(93+195) = 32.29%

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

```
##            Personal.Loan
## CreditCard    0    1
##          0 1924  195
##          1  788   93
```

II. P(Online = 1 | Loan = 1) 179/(179+109) = 62.15%

```
table(train_df[,c(13,10)])
```

```
##        Personal.Loan
## Online    0    1
##      0 1081  109
##      1 1631  179
```

III. P(Loan = 1) 288/(288+2712) = 9.6%

```
table(train_df[c(10)])
```

```
##
##    0    1
## 2712  288
```

IV. P(CC = 1 | Loan = 0) 788/(788+1924) = 29%

V. P(Online = 1 | Loan = 0) 1631/(1631+1081) = 60%

VI. P(Loan = 0) 2712/(2712+288) = 90.4%

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

((93/(93+195))* (179/(179+109))* (288/(288+2712)))/((93/(93+195))* (179/(179+109))* (288/(288+2712))) + ((788/(788+1924)) * (1631/(1631+1081)) * (2712/(2712+288))) = 0.1087106

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

0.1 nearly same as calculated method in 5th answer. The calculated method needed independent variable to predict, where as Naive Bayes not needed the independent variable.

7. Which of the entries in this table are needed for computing P(Loan = 1 | CC = 1, Online = 1)? 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).

```
train_nb = train_df[,c(10,13:14)]
nb = naiveBayes(Personal.Loan~., data=train_nb)
nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
## 0.904 0.096
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.3985988 0.6014012
##   1 0.3784722 0.6215278
##
##    CreditCard
## Y           0         1
##   0 0.7094395 0.2905605
##   1 0.6770833 0.3229167
```

(0.32 * 0.62 * 0.09) / (0.32 * 0.62 * 0.09) + (0.29 * 0.60 * 0.90) = 0.098

We received the nearly same out put for Naive Bayes in above.