Assignment 3

Kristian Lavinder

Kent State University

BA-64060-002: FUNDAMENTALS OF MACHINE LEARNING

Dr. Li Liu, Ph.D.

October 15th, 2023

https://github.com/klavinde/64060_klavinde/tree/main/klavinde_assignment_3

A. 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(). In Python, use panda dataframe methods melt() and pivot().

```
> table(Train.m1)
, , variable = Online, value = 0

          Personal.Loan
CreditCard    0    1
         0  785   65
         1  317   34

, , variable = Online, value = 1

          Personal.Loan
CreditCard    0    1
         0 1145  122
         1  475   57

> Train.c1
  CreditCard    0   1 (all)
1          0 1145 122  1267
2          1  475  57   532
3      (all) 1620 179  1799
> 
```

```
Train.m1                 3000 obs. of 4 variables
```

B. 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)].

  i. p.cc1loan1online1 <-57/3000*100
  ii. **1.9%**

C. 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.

```
> table(Train.Loan2Online)
, , value = 0

              variable
Personal.Loan Online
            0   1102
            1     99

, , value = 1

              variable
Personal.Loan Online
            0   1620
            1    179
```

```
> table(Train.Loan2CC)
, , value = 0

              variable
Personal.Loan CreditCard
            0       1930
            1        187

, , value = 1

              variable
Personal.Loan CreditCard
            0        792
            1         91
```

D. 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)
  - **3.0333%**

ii. P(Online = 1 | Loan = 1)
  - **5.9666%**

iii. P(Loan = 1) (the proportion of loan acceptors)
  - **9.26666%**

iv. P(CC = 1 | Loan = 0)
  - **26.4%**

v. P(Online = 1 | Loan = 0)
  - **54%**

vi. P(Loan = 0)
  - **90.733%**

Values	
p.cc1.loan0	26.4
p.cc1.loan1	3.03333333333333
p.cc1loan1online1	1.9
p.loan0	90.7333333333333
p.loan1	9.26666666666667
p.online1.loan0	54
p.online1.loan1	5.96666666666667

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         0          1
0.90733333 0.09266667

Conditional probabilities:
   Online
Y        [,1]      [,2]
  0 0.5951506 0.4909531
  1 0.6438849 0.4797134

   CreditCard
Y        [,1]      [,2]
  0 0.2909625 0.4542897
  1 0.3273381 0.4700881
```

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

   i. Calculated: **1.95289%,** Cross Table Validation = 1.01%

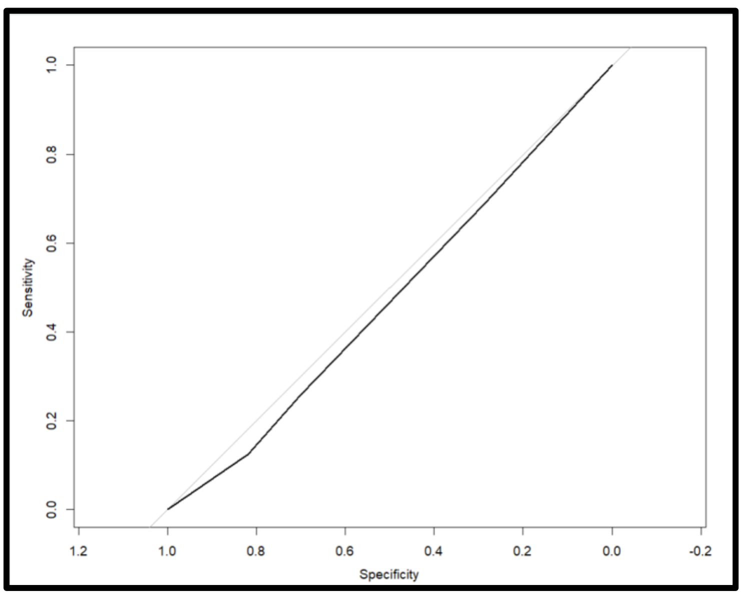
```
Total Observations in Table:  2000

                       | Predicted_Test_labels
Valid.df$Personal.Loan |         0 | Row Total |
-----------------------|-----------|-----------|
                     0 |      1798 |      1798 |
                       |     0.899 |           |
-----------------------|-----------|-----------|
                     1 |       202 |       202 |
                       |     0.101 |           |
-----------------------|-----------|-----------|
          Column Total |      2000 |      2000 |
-----------------------|-----------|-----------|
```

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

   i. Naïve Bayes probability is more accurate because it considers multiple probabilities rather than just sample data results. Including the cross table using the validation data gives a slightly lower probability dependent on the partition method.

G. 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).

   i. P(CC = 1 | Loan = 1) ; P(Online = 1 | Loan = 1) ; P(Loan = 1)



CODE BELOW:

```
library(caret)

install.packages("ggplot2")

install.packages("lattice")

library(ISLR)

library(e1071)

library(dplyr)

library(fnn)

universal.bank.df <-read.csv("UniversalBank.csv")

summary(universal.bank.df)

#Isolate Online, Credit Card, and Loan

MyData<-select(universal.bank.df,Personal.Loan,Online,CreditCard)

summary(MyData)

set.seed(123)

#Divide data into test and train

Index_Train<-createDataPartition(MyData$Personal.Loan, p=0.6, list=FALSE)

Train.df <-MyData[Index_Train,]

Valid.df  <-MyData[-Index_Train,]

#create Pivot Table for Online to CC and Loan

summary(Train.df)

install.packages("MASS")

install.packages("reshape2")

install.packages("reshape")

library(MASS)

library(reshape2)

library(reshape)

Train.m1 = melt(Train.df, id=c("CreditCard","Personal.Loan"),

        measure= c("Online"))

Train.m1

Train.c1 = cast(Train.m1, CreditCard ~ Personal.Loan, subset=variable=="Online",
```

```
            margins=c("grand_row","grand_col"), sum)
Train.c1
table(Train.m1)
p.cc1loan1online1 <-57/3000*100
##The Probability that a borrower uses online and has a cc with bank and will accept loan is 1.9%
Train.Loan2Online = melt(Train.df, id=c("Personal.Loan"),
            measure=c("Online"))
table(Train.Loan2Online)
##This table compares personal loan to online user data.
Train.Loan2CC = melt(Train.df, id=c("Personal.Loan"),
            measure=c("CreditCard"))
table(Train.Loan2CC)
##This table compares personal load to credit card user data.
#i. P(CC = 1 | Loan =1)
p.cc1.loan1 <-91/3000*100
# i.probablility is 3.033% for having cc and accepting loan
# ii. P(Online = 1 | Loan =1)
p.online1.loan1 <-179/3000*100
# ii.probablility is 5.966% for using online and accepting loan
#iii. P(Loan =1)
Train.Loan = Train.df$Personal.Loan
table(Train.Loan)
p.loan1 <-278/3000*100
#iii. probability is 9.266% overall that loan is accepted (from training data)
#iv. P(CC = 1 | Loan = 0)
p.cc1.loan0 <-792/3000*100
#iv. probability is 26.4% that have cc but decline loan
#v. P(Online = 1 | Loan = 0)
p.online1.loan0 <-1620/3000*100
#v. probability is 54% for using online but decline loan.
```

```
#vi. P(Loan =0)
p.loan0 <-2722/3000*100
#vi. probability is 90.73% that loan is declined (from training data)
library("gmodels")
install.packages("naivebayes")
nb_model <-naiveBayes(Personal.Loan ~ Online+CreditCard,data=Train.df)
nb_model
Predicted_Test_labels <- predict(nb_model,Valid.df)
library("gmodels")
CrossTable(x=Valid.df$Personal.Loan,y=Predicted_Test_labels, prop.chisq = TRUE)
CrossTable(x=Valid.df$Personal.Loan,y=Predicted_Test_labels, prop.chisq = FALSE)
Predicted_Test_labels <-predict(nb_model,Valid.df, type = "raw")
head(Predicted_Test_labels)
library(pROC)
roc(Valid.df$Personal.Loan, Predicted_Test_labels[,2])
plot.roc(Valid.df$Personal.Loan,Predicted_Test_labels[,2])
```