Assignment\_3

2022-10-17

#Install packages

library(e1071) #package for Naive Bayes classifier  
library(reshape) #package for the melt function  
library(reshape2)#package for the dcast function

##   
## Attaching package: 'reshape2'

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

#Read data

bank.data<-read.csv("Universalbank.csv")

#Data partitioning: 60% training, 40% validation

set.seed(1)  
train.index<-sample(row.names(bank.data), 0.6\*dim(bank.data)[1])  
valid.index <- setdiff(row.names(bank.data), train.index)   
train.df<-bank.data[train.index,]  
valid.df<-bank.data[valid.index,]

#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

#melt to stack columns  
mlt<-melt(train.df, id=c("CreditCard","Personal.Loan"),measure=c("Online"))  
  
#name data for dcast function  
CC<-train.df$CreditCard  
Personal.Loan<-train.df$Personal.Loan  
Online<-train.df$Online  
  
#dcast to create pivot table  
recast.bank=dcast(mlt,CC+Personal.Loan~Online)

## Aggregation function missing: defaulting to length

recast.bank[,c(1:4)]

## CC Personal.Loan 0 1  
## 1 0 0 805 1119  
## 2 0 1 79 119  
## 3 1 0 332 469  
## 4 1 1 30 47

#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?

47/(47+469)

## [1] 0.09108527

#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.

mlt.table1<-melt(train.df, id=c("Personal.Loan"),measure=c("Online"))  
recast.bank.table1=dcast(mlt.table1,Personal.Loan~Online)

## Aggregation function missing: defaulting to length

recast.bank.table1[,c(1:3)]

## Personal.Loan 0 1  
## 1 0 1137 1588  
## 2 1 109 166

mlt.table2<-melt(train.df, id=c("Personal.Loan"),measure=c("CreditCard"))  
recast.bank.table2=dcast(mlt.table2,Personal.Loan~CC)

## Aggregation function missing: defaulting to length

recast.bank.table2[,c(1:3)]

## Personal.Loan 0 1  
## 1 0 1924 801  
## 2 1 198 77

#D. Compute the following quantities

#i   
77/(77+198)

## [1] 0.28

#ii   
166/(166+109)

## [1] 0.6036364

#iii   
275/3000

## [1] 0.09166667

#iv  
801/(801+1924)

## [1] 0.293945

#v  
1588/(1588+1137)

## [1] 0.5827523

#vi  
2725/3000

## [1] 0.9083333

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

(0.28\*0.6036364\*0.09166667)/((0.28\*0.6036364\*0.09166667)+(0.293945\*0.5827523\*0.9083333))

## [1] 0.09055758

#F. Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate? The naive Bayes probability of 0.09055758 and the probability of 0.09108527 from the pivot table are relatively close in value. However, the naive Bayes probability is a more accurate estimate. This is because naive Bayes assumes conditional independence. In other words, it is not required that features of the record to be classified be exactly the same as the features in the training set in order to classify it.

#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).

nb\_model<-naiveBayes(Personal.Loan~CreditCard+Online,data=train.df)  
nb\_model

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.90833333 0.09166667   
##

## Conditional probabilities:  
## CreditCard  
## Y [,1] [,2]  
## 0 0.293945 0.4556506  
## 1 0.280000 0.4498175  
##   
## Online  
## Y [,1] [,2]  
## 0 0.5827523 0.4931950  
## 1 0.6036364 0.4900334

#The naiveBayes model output of 0.09 is the same as the 0.09 output obtained in (E).