Assignment 3

Steven Kalinoff

2023-03-01

# ctl + shift + c to uncomment block for intial install of packages.  
# install.packages("class")  
# install.packages("caret")  
# install.packages("readr")  
# install.packages("e1071")  
# install.packages("reshape")  
# install.packages("reshape2")  
# install.packages("dplyr")  
# install.packages("Stats")  
# install.packages("ISLR")  
# install.packages("gmodels")  
# install.packages("fastDummies")

library(class)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(readr)  
library(e1071)  
library(reshape)

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:class':  
##   
## condense

library(reshape2)

##   
## Attaching package: 'reshape2'

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

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:reshape':  
##   
## rename

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

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

library(stats)  
library(ISLR)  
library(gmodels)   
library(ggplot2)  
library(lattice)  
library(fastDummies)  
library(readxl)

getwd()

## [1] "C:/Users/marin/Documents"

NB = read.csv("UniversalBank.csv")  
head(NB)

## ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage  
## 1 1 25 1 49 91107 4 1.6 1 0  
## 2 2 45 19 34 90089 3 1.5 1 0  
## 3 3 39 15 11 94720 1 1.0 1 0  
## 4 4 35 9 100 94112 1 2.7 2 0  
## 5 5 35 8 45 91330 4 1.0 2 0  
## 6 6 37 13 29 92121 4 0.4 2 155  
## Personal.Loan Securities.Account CD.Account Online CreditCard  
## 1 0 1 0 0 0  
## 2 0 1 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 1  
## 6 0 0 0 1 0

#123  
set.seed(123)  
NB$Online = as.factor(NB$Online)  
NB$CreditCard = as.factor(NB$CreditCard)  
NB$Personal.Loan = as.factor(NB$Personal.Loan)  
  
train.index = sample(row.names(NB), 0.6\*dim(NB)[1])  
test.index = setdiff(row.names(NB), train.index)  
train.df = NB[train.index,]  
test.df = NB[test.index,]  
train = NB[train.index,]  
test = NB[train.index,]  
  
# A: creating a table with 2 var and outcome  
melt.NB = melt(train,id=c("CreditCard","Personal.Loan"),variable = "Online")

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

recast.NB=dcast(melt.NB,CreditCard+Personal.Loan~Online)

## Aggregation function missing: defaulting to length

recast.NB[c(1:2,14)]

## CreditCard Personal.Loan Online  
## 1 0 0 1930  
## 2 0 1 187  
## 3 1 0 792  
## 4 1 1 91

# B: Customers with 1=yes CC and 1=yes have 91/792 = 11.5% chance of having an online accnt.   
  
# C: pivot tables of training data %60   
NB\_O=melt(train,id=c("Personal.Loan"),variable="Online")

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

recast.NB\_O=dcast(NB\_O,Personal.Loan~Online)

## Aggregation function missing: defaulting to length

Loan=recast.NB\_O[,c(1,13)]  
Loan

## Personal.Loan Online  
## 1 0 2722  
## 2 1 278

NB\_CC=melt(train,id=c("CreditCard"),variable="Online")

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

recast.NB\_CC=dcast(NB\_CC,CreditCard~Online)

## Aggregation function missing: defaulting to length

Loan2=recast.NB\_CC[,c(1,14)]  
Loan2

## CreditCard Online  
## 1 0 2117  
## 2 1 883

table(train[,c(14,10)])

## Personal.Loan  
## CreditCard 0 1  
## 0 1930 187  
## 1 792 91

table(train[,c(13,10)])

## Personal.Loan  
## Online 0 1  
## 0 1102 99  
## 1 1620 179

table(train[,c(10)])

##   
## 0 1   
## 2722 278

x1=table(train[,c(14,10)])  
x2=table(train[,c(13,10)])  
x3=table(train[,c(10)])  
#D:  
## P(CC=1)|Loan=1)   
#\* i: 91/(187+91)= 32.7%  
A= x1[2,2]/(x1[2,2]+x1[1,2])  
A

## [1] 0.3273381

#\* ii: 179/(179+99) = %64.4  
B = x2[2,2]/(x2[2,2]+x2[1,2])  
B

## [1] 0.6438849

#\* iii: 278/(2722+278) = 9.3%  
C = x3[2]/(x3[1]+x3[2])  
C

## 1   
## 0.09266667

#\* iv: 883/(2117+883) = 29.1%  
D = x1[2,1]/(x1[1,1]+x1[2,1])  
D

## [1] 0.2909625

#\* v: 1620/(1620+1102) = 59.5%  
E= x2[2,1]/(x2[1,1]+x2[2,1])  
E

## [1] 0.5951506

#\* vi: 2722/(278+2722) = 91.0%  
F = x3[1]/(x3[1]+x3[2])  
F

## 0   
## 0.9073333

#E:   
#NBANS P(A/B) = (P(B/A)\*P(A))/P(B) =11.1%  
NBANS = ((A\*B\*C)/((A\*B\*C)+(D\*E\*F)))  
NBANS

## 1   
## 0.1105637

#F:   
#\* Customers Using the split pivot have a roughly similar chance of being approved at 11.1% opposed to the not split data at 11.5% in the joined approved training data. This shows that eider method is a reliable method classifying method with the training data provided.

NB.train = train.df[,c(10,13:14)]  
NB.test = test.df[,c(10,13:14)]  
NBTRUE= naiveBayes(Personal.Loan~.,data=NB.train)  
NBTRUE

##   
## 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 0 1  
## 0 0.4048494 0.5951506  
## 1 0.3561151 0.6438849  
##   
## CreditCard  
## Y 0 1  
## 0 0.7090375 0.2909625  
## 1 0.6726619 0.3273381

NBLONG = (.327\*.644\*.093)/((.327\*.644\*.093)+(.291\*.595\*.907))  
NBLONG

## [1] 0.1108815

# Manual method = 11.05% NB method yields 11.08% a .03% difference likely to rounding of results from NB matrix. This long hand method is sufficiently precise enough use in place of the naivebayes method provided in the library or vis-versa.