Assignment-3 (FML)

Dev

2023-10-14

Summary

Q-1:

- We predicted that injury = yes since the probability of getting injury is greater than the probability of not getting injury.

Q-2:

- Following are the Bayes probability of an injury = yes given all possible combination of weather and traffic parameters- 0.67 , 0.18 , 0 , 0 , 0 , 1.
- With 0.5 as cutoff, the 24 records of accidents were classified by model.
- The Naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1 is 0 (calculated manually).

Q-3:

- Baye's and Naive Baye's probablities are not equivalent but both have the same ranking.
- Overall error of validation set is 0.479.

Problem Statement:

The file accidentsFull.csv contains information on 42,183 actual automobile accidents in 2001 in the United States that involved one of three levels of injury: NO INJURY, INJURY, or FATALITY. For each accident, additional information is recorded, such as day of week, weather conditions, and road type. A firm might be interested in developing a system for quickly classifying the severity of an accident based on initial reports and associated data in the system (some of which rely on GPS-assisted reporting).

Our goal here is to predict whether an accident just reported will involve an injury (MAX_SEV_IR = 1 or 2) or will not (MAX_SEV_IR = 0). For this purpose, create a dummy variable called INJURY that takes the value "yes" if MAX_SEV_IR = 1 or 2, and otherwise "no."

Preparing data as required in problem statement

```r
#Loading libraries -
library(e1071)
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v lubridate 1.9.2     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.0

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x purrr::lift()   masks caret::lift()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
#Setting directory and reading the csv file
setwd("/Users/devmarwah/Downloads")
df=read.csv("accidentsFull.csv")
#Making dummy variable injury -
df$INJURY=ifelse(df$MAX_SEV_IR>0,"YES","NO")
```

1. Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?

Whether or not accident caused injury will then depend on the probability of getting injury in accidents.

```r
#Getting number of yes of INJURY variable using for loop and if statements -
y=0
for (i in 1:dim(df)[1]) {
   if(df$INJURY[i]=="YES"){
     y=y+1
   }
}
```

- Probability of getting injured in an accident:

```r
#Calculating probablity manually
round(y/nrow(df),2)
```

```
[1] 0.51
```

- Probability of not getting injured in an accident:

```
#Using table command this time for number of "NO" is injury
#Calculating probability manually
round(table(df$INJURY)["NO"]/nrow(df),2)
```

```
  NO
0.49
```

This suggests that the probability of getting injured is slightly more than that of not getting injured.

Hence, we can predict on the basis of higher probability of injured that YES the accident will cause injury

---

2. Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER_R and TRAF_CON_R. Create a pivot table that examines INJURY as a function of the two predictors for these 24 records. Use all three variables in the pivot table as rows/columns.

The pivot table of injury, weather_r and traf_con_r is -

```
#Selecting first 24 rows
df.2=df[1:24,]
#Making a pivot table of all three variables
df.2=df.2 %>%
  select(INJURY,WEATHER_R,TRAF_CON_R)
#Making pivot table of Injury, Weather_r,Traf_con_r
p.df.2=ftable(df.2)
p.df.2
```

```
                 TRAF_CON_R 0 1 2
INJURY WEATHER_R
NO     1                    3 1 1
       2                    9 1 0
YES    1                    6 0 0
       2                    2 0 1
```

```
#Making a seperate pivot table of weather_r and traf_con_r
p.df.wt=ftable(df.2[,-1])
p.df.wt
```

```
          TRAF_CON_R  0  1  2
WEATHER_R
1                     9  1  1
2                    11  1  1
```

a) Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.

Following are probabilities of all six cases when injury=YES:

- Traf_con_r = 0, weather_r=1

```
p1=round(p.df.2[3,1]/p.df.wt[1,1],2)
p1
```

```
[1] 0.67
```

- Traf_con_r = 0, weather_r=2

```
p2=round(p.df.2[4,1]/p.df.wt[2,1],2)
p2
```

```
[1] 0.18
```

- Traf_con_r = 1, weather_r=1

```
p3=p.df.2[3,2]/p.df.wt[1,2]
p3
```

```
[1] 0
```

- Traf_con_r = 1, weather_r=2

```
p4=p.df.2[4,2]/p.df.wt[2,2]
p4
```

```
[1] 0
```

- Traf_con_r = 2, weather_r=1

```
p5=p.df.2[3,3]/p.df.wt[1,3]
p5
```

```
[1] 0
```

- Traf_con_r = 2, weather_r=2

```
p6=p.df.2[4,3]/p.df.wt[2,3]
p6
```

```
[1] 1
```

b) Classify the 24 accidents using these probabilities and a cutoff of 0.5.

```
prob=rep(0,24)
#Using for loop to get "yes" probabilities for all cases
prob.inj=prob
for (i in 1:24) {
      if (df.2$WEATHER_R[i] == "1") {
      if (df.2$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p1
      }
      else if (df.2$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p3
      }
      else if (df.2$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p5
      }
    }
    else {
      if (df.2$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p2
      }
      else if (df.2$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p4
      }
      else if (df.2$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p6
      }
    }
}
#Storing these probabilities
df.2$probablity=prob.inj
#Predicting injury based on probablities
df.2$prediction=ifelse(df.2$probablity>0.5,"YES","NO")
head(df.2,10)
```

```
   INJURY WEATHER_R TRAF_CON_R probablity prediction
1     YES         1          0       0.67        YES
2      NO         2          0       0.18         NO
3      NO         2          1       0.00         NO
4      NO         1          1       0.00         NO
5      NO         1          0       0.67        YES
6     YES         2          0       0.18         NO
7      NO         2          0       0.18         NO
8     YES         1          0       0.67        YES
9      NO         2          0       0.18         NO
10     NO         2          0       0.18         NO
```

c) Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

Naive Baye's formula for above formula can be formulated as-

P(I=yes/W=1/T=1)={P(W=1/I=yes)P(T=1/I=yes)P(yes)}/P(W=1)P(T=1)

Using this formula, we get the probability as-

```
#Making Naive Baye's formula manually
P=((sum(p.df.2[3,])/sum(p.df.2[c(3,4),]))*(sum(p.df.2[c(3,4),2])/sum(p.df.2[c(3,4),])))/(sum(p.df.2[c(1
#Printing the value
P
```

```
[1] 0
```

d) Run a naive Bayes classifier on the 24 records and two predictors. Check the model output to obtain probabilities and classifications for all 24 records. Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

```
#Converting everything as factors
for (i in c(1:dim(df)[2])){
  df[,i] <- as.factor(df[,i])
}
#Using Naive Baye's to solve
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = df.2,laplace = 0)

nbt <- predict(nb, newdata = df.2,type = "raw")
nbt=round(nbt,2)
#Storing Naive Baye's probablities
df.2$nbpred.prob <- nbt[,2]
df.2$nb.prediction=ifelse(df.2$nbpred.prob>0.5,"YES","NO")
head(df.2,10) %>%
  select(prediction,nb.prediction)
```

```
   prediction nb.prediction
1         YES           YES
2          NO            NO
3          NO            NO
4          NO           YES
5         YES           YES
6          NO            NO
7          NO            NO
8         YES           YES
9          NO            NO
10         NO            NO
```

We can observe that the predictions are same in most of the cases but for some cases they are different. This means that predictions made by Naive baye's and baye's theorum are not equivalent in this problem. This also means that weather_r and traf_con_r are not independent to each other as naive baye's assumes that all conditional probablities are independent to each other but that's not the case in given problem.

Ordering by baye's and storing ranks

```
brank=df.2 %>%
  select(probablity,nbpred.prob) %>%
  filter(!probablity==0) %>%
  filter(!probablity==1) %>%
  arrange(probablity)
brank=rank(brank)
```

Ordering by naive baye's and storing ranks

```
nrank=df.2 %>%
  select(probablity,nbpred.prob) %>%
  filter(!probablity==0) %>%
  filter(!probablity==1) %>%
  arrange(nbpred.prob)
nrank=rank(nrank)
```

Comparing ranks -

```
all(brank==nrank)
```

```
[1] TRUE
```

Once can observe that ranking by Baye's and Naive Baye's are identical in nature.This is also confirmed by all function.This proves the well known fact that Naive Baye's methog might not give us the correct probabilities,**however, it can still prove useful in ranking.**

One noticeable fact is that while arranging/ordering probabilities we have filtered out all the cases where Baye's probability was zero/one. This is because Naive Baye's classfication machine learning algorithms use a method called *"Laplace Smoothing"*. In laplace smoothing the classification algorithm assigns random non-zero values to all zero-value/one-value probabilities for effective and smooth functioning of algorithm. However, this can pose a problem in ranking since algorithm does not necessarily assign same values to all zero-value/one-value probabilities. Hence, we have removed all zero-value probabilities to avoid confusion in ranking.

---

3. Let us now return to the entire dataset.Partition the data into training (60%) and validation (40%).

```
#Making training and validation sets
train.index=sample(row.names(df),0.6*dim(df)[1])
valid.index=setdiff(row.names(df),train.index)
train.df=df[train.index,-24]
valid.df=df[valid.index,-24]
```

a) Run a naive Bayes classifier on the complete training set with the relevant predictors (and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix.

```
#Using naive baye's classification
nb2 <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                  data = train.df)

nbt2 <- predict(nb2,newdata = train.df,type = "raw")
nbt2.pred=ifelse(nbt2[,2]>0.5,"YES","NO")
#Making confusion matrix
CM=confusionMatrix(train.df$INJURY,as.factor(nbt2.pred),positive = "YES")
CM$table
```

```
          Reference
Prediction    NO   YES
       NO   1986 10492
       YES  1647 11184
```

b) What is the overall error of the validation set?

```
#Using Naive Baye's classification
nb3 <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                  data = valid.df)

nbt3 <- predict(nb3,newdata = valid.df,type = "raw")
nbt3.pred=ifelse(nbt3[,2]>0.5,"YES","NO")
#Making its confusion matrix
CM2=confusionMatrix(valid.df$INJURY,as.factor(nbt3.pred),positive = "YES")
```

Error rate can be calculated as 1-accuracy-

```
#Calculating error as 1- accuracy since accruacy is 1 element in overall of confusion matrix
1-CM2$overall[1]
```

```
 Accuracy
0.4747541
```

LEARNING OUTCOMES:

- Baye's and Naive Baye's theorum are not equivalent and might give different answers if conditional probabilities are not independent.
- Naive Baye's classification machine learning algorithms use *"Laplace Smoothing"* which assigns random non-zero values to zero-value and one-value probabilities for working of algorithm.
- Naive Baye's is useful is useful in ranking as its ranking is identical to that of Baye's.
