fml_assignment_3_811289717

SAKSHI

2023-10-16

R Markdown

```r
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
library(e1071)
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
accidents = read.csv("/Users/sakshibansal/Downloads/accidentsFull.csv")
```

Summary

1. We predicted that injury = yes since the probability of injury happening (0.5087832) is greater than the probability of injury not happening (0.4912168).

2.1. 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.

2.2. With 0.5 as cutoff, the 24 records of accidents were classified by model as 10 "YES" and 14 "NO".

2.3. The naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1 is 0.

2.4. Yes, the resulting classifications and ranking of the observations is equivalent.

3.2 Overall error of validation set is 0.4794951

Conclusions dervived from this assignment

Naive bayes theorem assume that all the variables are independent which is not the case with bayes theorem resulting in different answers.

Naive bayes ranking is identical to bayes when we have sufficient data and same class of variables.

Naive bayes use "Laplace Smoothing" which assigns random non-zero values to zero-value and one-value probabilities.

Questions

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."

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?

Answer 1

```
# Creating a dummy variable called injury

accidents$injury = ifelse(accidents$MAX_SEV_IR > 0, "yes" , "no")

# Finding the count for "yes" and "no"

t=table(accidents$injury)
t
```

```
   no   yes
20721 21462
```

```
# Finding the probability of injury not happening

injuryno = t["no"]/nrow(accidents)
injuryno
```

```
       no
0.4912168
```

```
# Finding the probability of injury happening

injuryyes = t["yes"]/nrow(accidents)
injuryyes
```

```
      yes
0.5087832
```

As we can see that probability of injury happening (0.5087832) is greater than the probability of injury not happening (0.4912168), we can safely assume that if an accident had just been reported and no further information is available , then we can predict that there has been an 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.

```
#Selecting first 24 records in the dataset

df = accidents[1:24,]

#Making a pivot table of all three variables

df = df %>%
  select(injury,WEATHER_R,TRAF_CON_R)

prob.df = ftable(df)
prob.df
```

```
                 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
```

```
prob.df.2 = ftable(df[,-1])
prob.df.2
```

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

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

Below is the probability of an injury = YES when we are considering six possible combinations of the predictors:

TRAF_CON_R = 0, WEATHER_R = 1

```
prob.yes.1 = round(prob.df[3,1]/prob.df.2[1,1],2)
prob.yes.1
```

```
[1] 0.67
```

TRAF_CON_R = 0, WEATHER_R = 2

```
prob.yes.2 = round(prob.df[4,1]/prob.df.2[2,1],2)
prob.yes.2
```

```
[1] 0.18
```

TRAF_CON_R = 1, WEATHER_R = 1

```
prob.yes.3 = prob.df[3,2]/prob.df.2[1,2]
prob.yes.3
```

```
[1] 0
```

TRAF_CON_R = 1, WEATHER_R = 2

```
prob.yes.4 = prob.df[4,2]/prob.df.2[2,2]
prob.yes.4
```

```
[1] 0
```

TRAF_CON_R = 2, WEATHER_R = 1

```
prob.yes.5 = prob.df[3,3]/prob.df.2[1,3]
prob.yes.5
```

```
[1] 0
```

TRAF_CON_R = 2, WEATHER_R = 2

```
prob.yes.6 = prob.df[4,3]/prob.df.2[2,3]
prob.yes.6
```

```
[1] 1
```

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

```
prob.accidents = rep(0,24)

# Creating a new variable and putting loop to determine the values of all probabilities given that belo

prob.injury = prob.accidents
for (i in 1:24) {
      if (df$WEATHER_R[i] == "1") {
      if (df$TRAF_CON_R[i]=="0"){
        prob.injury[i] = prob.yes.1
      }
      else if (df$TRAF_CON_R[i]=="1") {
        prob.injury[i] = prob.yes.3
```

```
      }
      else if (df$TRAF_CON_R[i]=="2") {
        prob.injury[i] = prob.yes.5
      }
    }
    else {
      if (df$TRAF_CON_R[i]=="0"){
        prob.injury[i] = prob.yes.2
      }
      else if (df$TRAF_CON_R[i]=="1") {
        prob.injury[i] = prob.yes.4
      }
      else if (df$TRAF_CON_R[i]=="2") {
        prob.injury[i] = prob.yes.6
      }
    }
}
df$probablity = prob.injury

# Predicting the possibility of accidents

df$prediction = ifelse(df$probablity > 0.5,"yes","no")

head(df , 24)
```

```
   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
11     no         2          0       0.18         no
12     no         1          2       0.00         no
13    yes         1          0       0.67        yes
14     no         1          0       0.67        yes
15    yes         1          0       0.67        yes
16    yes         1          0       0.67        yes
17     no         2          0       0.18         no
18     no         2          0       0.18         no
19     no         2          0       0.18         no
20     no         2          0       0.18         no
21    yes         1          0       0.67        yes
22     no         1          0       0.67        yes
23    yes         2          2       1.00        yes
24    yes         2          0       0.18         no
```

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

```
# Applying naive bayes formula assuming that weather and injury are independent variables.

prob.injury.2 = ((sum(prob.df[3,])/sum(prob.df[c(3,4),]))*(sum(prob.df[c(3,4),2])/sum(prob.df[c(3,4),])

prob.injury.2
```

```
[1] 0
```

2.4 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 every variable into factors using as.factor

for (i in c(1:dim(accidents)[2])){
  accidents[,i] <- as.factor(accidents[,i])
}

# Adding laplace = 0 in order to to stop R to assign random non-zero values to probability = 0,1 and us

prob.naive <- naiveBayes(injury ~ TRAF_CON_R + WEATHER_R,
                data = df,laplace = 0)

prob.naive.2 <- predict(prob.naive, newdata = df,type = "raw")

prob.naive.2 = round(prob.naive.2,2)

df$prob.naive.3 <- prob.naive.2[,2]

df$prob.naive.4 = ifelse(df$prob.naive.3 > 0.5,"yes","no")

head(df,10) %>%
  select(prediction,prob.naive.4)
```

```
   prediction prob.naive.4
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
```

The laplace was put as 0 because the naive bayes theorem uses laplace smoothing which randomly assigns non-zero values to records with probability of 0,1. In any other case, it would have been an advantage , but here we had to find out the ranking order between naive and bayes theorem, so we had to remove all the zero and one value probabilities since bayes theorem does not follow laplace smoothing and we would have ended up with inequal order.

```r
# Arranging by bayes theorem

bayes.rank = df %>%
  select(probablity,prob.naive.3) %>%
  filter(!probablity==0) %>%
  filter(!probablity==1) %>%
  arrange(probablity)
bayes.rank = rank(bayes.rank)

# Arranging by naive bayes theorem

naive.rank = df %>%
  select(probablity,prob.naive.3) %>%
  filter(!probablity==0) %>%
  filter(!probablity==1) %>%
  arrange(prob.naive.3)
naive.rank = rank(naive.rank)

#Comparison of both ranks

all(bayes.rank == naive.rank)
```

```
[1] TRUE
```

---

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

```r
# Partitioning the data into training set (60%) and validation set (40%).

accident.train = sample(row.names(accidents),0.6*dim(accidents)[1])

accident.valid = setdiff(row.names(accidents),accident.train)

train.df = accidents[accident.train,-24]

valid.df = accidents[accident.valid,-24]
```

3.1 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.

```r
#Using naive bayes theorem

naive.prob <- naiveBayes(injury ~ TRAF_CON_R + WEATHER_R,
                 data = train.df)

naive.prob.2 <- predict(naive.prob , newdata = train.df , type = "raw")

naive.prob.2.pred = ifelse(naive.prob.2[,2] >0.5 ,"yes","no")

naive.prob.2.pred = as.factor(naive.prob.2.pred)
```

```
df.matrix = confusionMatrix(train.df$injury,naive.prob.2.pred,positive = "yes")

df.matrix
```

```
Confusion Matrix and Statistics

          Reference
Prediction    no   yes
       no   1997 10467
       yes  1640 11205

               Accuracy : 0.5216
                 95% CI : (0.5155, 0.5278)
    No Information Rate : 0.8563
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0329

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5170
            Specificity : 0.5491
         Pos Pred Value : 0.8723
         Neg Pred Value : 0.1602
             Prevalence : 0.8563
         Detection Rate : 0.4427
   Detection Prevalence : 0.5075
      Balanced Accuracy : 0.5331

       'Positive' Class : yes
```

3.2 What is the overall error of the validation set?

```
naive.prob.3 <- naiveBayes(injury ~ TRAF_CON_R + WEATHER_R,
                 data = valid.df)

naive.prob.4 <- predict(naive.prob.3,newdata = valid.df,type = "raw")

naive.prob.4.pred = ifelse(naive.prob.4[,2] >0.5 ,"yes","no")

naive.prob.4.pred = as.factor(naive.prob.4.pred)

df.matrix.2 = confusionMatrix(valid.df$injury,naive.prob.4.pred,positive = "yes")
```

Error rate = 1-accuracy

```
Error = 1- df.matrix.2$overall[1]
Error
```

```
 Accuracy
0.4766505
```