BA_64060_Assignment3

Durga Prasad Gandi

2023-10-14

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."

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?

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 12 records. Use all three variables in the pivot table as rows/columns.

- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
- Classify the 24 accidents using these probabilities and a cutoff of 0.5.
- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.
- 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?

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

- 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.
- What is the overall error of the validation set?

Summary

Data Input and Cleaning

Load the required libraries and read the input file

```
library(e1071)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(klaR)

## Loading required package: MASS

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

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

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

accidents = read.csv("C:/Users/gdurg/Documents/FML
ASSIGNMENTS/accidentsFull.csv")

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

head(accidents)

##   HOUR_I_R ALCHL_I ALIGN_I STRATUM_R WRK_ZONE WKDY_I_R INT_HWY LGTCON_I_R
## 1        0       2       2         1        0        1       0          3
## 2        1       2       1         0        0        1       1          3
## 3        1       2       1         0        0        1       0          3
## 4        1       2       1         1        0        0       0          3
## 5        1       1       1         0        0        1       0          3
## 6        1       2       1         1        0        1       0          3
##   MANCOL_I_R PED_ACC_R RELJCT_I_R REL_RWY_R PROFIL_I_R SPD_LIM SUR_COND
## 1          0         0          1         0          1      40        4
## 2          2         0          1         1          1      70        4
## 3          2         0          1         1          1      35        4
```

```
## 4          2          0          1          1          1      35          4
## 5          2          0          0          1          1      25          4
## 6          0          0          1          0          1      70          4
##   TRAF_CON_R TRAF_WAY VEH_INVL WEATHER_R INJURY_CRASH NO_INJ_I
PRPTYDMG_CRASH
## 1          0        3        1         1            1        1
0
## 2          0        3        2         2            0        0
1
## 3          1        2        2         2            0        0
1
## 4          1        2        2         1            0        0
1
## 5          0        2        3         1            0        0
1
## 6          0        2        1         2            1        1
0
##   FATALITIES MAX_SEV_IR INJURY
## 1          0          1    yes
## 2          0          0     no
## 3          0          0     no
## 4          0          0     no
## 5          0          0     no
## 6          0          1    yes
```

```r
# Convert variables to factor
for (i in c(1:dim(accidents)[2])){
  accidents[,i] = as.factor(accidents[,i])
}
head(accidents,n=24)
```

```
##    HOUR_I_R ALCHL_I ALIGN_I STRATUM_R WRK_ZONE WKDY_I_R INT_HWY LGTCON_I_R
## 1         0       2       2         1        0        1       0          3
## 2         1       2       1         0        0        1       1          3
## 3         1       2       1         0        0        1       0          3
## 4         1       2       1         1        0        0       0          3
## 5         1       1       1         0        0        1       0          3
## 6         1       2       1         1        0        1       0          3
## 7         1       2       1         0        0        1       1          3
## 8         1       2       1         1        0        1       0          3
## 9         1       2       1         1        0        1       0          3
## 10        0       2       1         0        0        0       0          3
## 11        1       2       1         0        0        1       0          3
## 12        1       2       1         1        0        1       0          3
## 13        1       2       1         1        0        1       0          3
## 14        1       2       2         0        0        1       0          3
## 15        1       2       2         1        0        1       0          3
## 16        1       2       2         1        0        1       0          3
## 17        1       2       1         1        0        1       0          3
## 18        1       2       1         1        0        0       0          3
```

```
## 19        1       2       1         1        0        1       0           3
## 20        1       2       1         0        0        1       0           3
## 21        1       2       1         1        0        1       0           3
## 22        1       2       2         0        0        1       0           3
## 23        1       2       1         0        0        1       0           3
## 24        1       2       1         1        0        1       9           3
##    MANCOL_I_R PED_ACC_R RELJCT_I_R REL_RWY_R PROFIL_I_R SPD_LIM SUR_COND
## 1           0         0          1         0          1      40        4
## 2           2         0          1         1          1      70        4
## 3           2         0          1         1          1      35        4
## 4           2         0          1         1          1      35        4
## 5           2         0          0         1          1      25        4
## 6           0         0          1         0          1      70        4
## 7           0         0          0         0          1      70        4
## 8           0         0          0         0          1      35        4
## 9           0         0          1         0          1      30        4
## 10          0         0          1         0          1      25        4
## 11          0         0          0         0          1      55        4
## 12          2         0          0         1          1      40        4
## 13          1         0          0         1          1      40        4
## 14          0         0          0         0          1      25        4
## 15          0         0          0         0          1      35        4
## 16          0         0          0         0          1      45        4
## 17          0         0          0         0          1      20        4
## 18          0         0          0         0          1      50        4
## 19          0         0          0         0          1      55        4
## 20          0         0          1         1          1      55        4
## 21          0         0          1         0          0      45        4
## 22          0         0          1         0          0      65        4
## 23          0         0          0         0          0      65        4
## 24          2         0          1         1          0      55        4
##    TRAF_CON_R TRAF_WAY VEH_INVL WEATHER_R INJURY_CRASH NO_INJ_I
PRPTYDMG_CRASH
## 1           0        3        1         1            1        1
0
## 2           0        3        2         2            0        0
1
## 3           1        2        2         2            0        0
1
## 4           1        2        2         1            0        0
1
## 5           0        2        3         1            0        0
1
## 6           0        2        1         2            1        1
0
## 7           0        2        1         2            0        0
1
## 8           0        1        1         1            1        1
0
## 9           0        1        1         2            0        0
```

```
1
## 10          0        1        1         2            0        0
1
## 11          0        1        1         2            0        0
1
## 12          2        1        2         1            0        0
1
## 13          0        1        4         1            1        2
0
## 14          0        1        1         1            0        0
1
## 15          0        1        1         1            1        1
0
## 16          0        1        1         1            1        1
0
## 17          0        1        1         2            0        0
1
## 18          0        1        1         2            0        0
1
## 19          0        1        1         2            0        0
1
## 20          0        1        1         2            0        0
1
## 21          0        3        1         1            1        1
0
## 22          0        3        1         1            0        0
1
## 23          2        2        1         2            1        2
0
## 24          0        2        2         2            1        1
0
##    FATALITIES MAX_SEV_IR INJURY
## 1           0          1    yes
## 2           0          0     no
## 3           0          0     no
## 4           0          0     no
## 5           0          0     no
## 6           0          1    yes
## 7           0          0     no
## 8           0          1    yes
## 9           0          0     no
## 10          0          0     no
## 11          0          0     no
## 12          0          0     no
## 13          0          1    yes
## 14          0          0     no
## 15          0          1    yes
## 16          0          1    yes
## 17          0          0     no
## 18          0          0     no
```

```
## 19          0          0     no
## 20          0          0     no
## 21          0          1    yes
## 22          0          0     no
## 23          0          1    yes
## 24          0          1    yes
```

Questions

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: If there is no information available whether accident will result in INJURY(Yes or No), then we caculate probabilty of INJURY = YES and, NO and compare both which ever has highest value we can consider that as outcome of the accident.

Example code,

```
yes = accidents %>% filter(accidents$INJURY=="yes") %>% summarise(count= n())
p_yes =  yes / nrow(accidents)
p_yes$count

## [1] 0.5087832

no = accidents %>% filter(accidents$INJURY=="no") %>% summarise(count= n())
p_no =  no / nrow(accidents)
p_no$count

## [1] 0.4912168
```

As you can see probability for yes is 0.5087832 and probability for no is 0.4912168. So, we can consider outcome of the accident as INJURY = Yes.

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 12 records. Use all three variables in the pivot table as rows/columns.

- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
- Classify the 24 accidents using these probabilities and a cutoff of 0.5.
- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.
- 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?

```
accidents24 = accidents[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
```

```
dt1 = ftable(accidents24)
dt1

##                   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

dt2 = ftable(accidents24[,-1]) # print table only for conditions
dt2

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

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 12 records. Use all three variables in the pivot table as rows/columns.

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

```
# Injury = yes
p1 = dt1[3,1] / dt2[1,1] # Injury, Weather=1 and Traf=0
p2 = dt1[4,1] / dt2[2,1] # Injury, Weather=2, Traf=0
p3 = dt1[3,2] / dt2[1,2] # Injury, W=1, T=1
p4 = dt1[4,2] / dt2[2,2] # I, W=2,T=1
p5 = dt1[3,3] / dt2[1,3] # I, W=1,T=2
p6 = dt1[4,3]/ dt2[2,3] #I,W=2,T=2

# Injury = no
n1 = dt1[1,1] / dt2[1,1] # Weather=1 and Traf=0
n2 = dt1[2,1] / dt2[2,1] # Weather=2, Traf=0
n3 = dt1[1,2] / dt2[1,2] # W=1, T=1
n4 = dt1[2,2] / dt2[2,2] # W=2,T=1
n5 = dt1[1,3] / dt2[1,3] # W=1,T=2
n6 = dt1[2,3] / dt2[2,3] # W=2,T=2
print(c(p1,p2,p3,p4,p5,p6))

## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000

print(c(n1,n2,n3,n4,n5,n6))

## [1] 0.3333333 0.8181818 1.0000000 1.0000000 1.0000000 0.0000000
```

2. Let us now compute

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

```r
prob.inj = rep(0,24)

for (i in 1:24) {
  print(c(accidents24$WEATHER_R[i],accidents24$TRAF_CON_R[i]))
    if (accidents24$WEATHER_R[i] == "1") {
      if (accidents24$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p1
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p3
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p5
      }
    }
    else {
      if (accidents24$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p2
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p4
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p6
      }
    }
  }

## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 1
## Levels: 1 2 0
## [1] 1 1
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
```

```
## [1] 1 2
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 2
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0

accidents24$prob.inj = prob.inj

accidents24$pred.prob = ifelse(accidents24$prob.inj>0.5, "yes", "no")
```

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

```
new_data = data.frame(WEATHER_R = "1", TRAF_CON_R = "1")


nb = naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = accidents24)


prediction = predict(nb, newdata = new_data, type = "raw")


probability_injury_yes = prediction[, "yes"]

cat("Naive Bayes conditional probability of injury (INJURY = Yes) given
WEATHER_R = 1 and TRAF_CON_R = 1: ", probability_injury_yes, "\n")

## Naive Bayes conditional probability of injury (INJURY = Yes) given
WEATHER_R = 1 and TRAF_CON_R = 1:  0.008919722
```

- 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?

```
nb = naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = accidents24)

nbt = predict(nb, newdata = accidents24)
# accidents24$nbpred.prob = nbt[,2] # Transfer the "Yes" nb prediction

cutoff = 0.5

exact_bayes_classifications = ifelse(c(p1, p2, p3, p4, p5, p6) > cutoff,
"yes", "no")

comparison_result = data.frame(
  "Exact_Bayes_Classification" = exact_bayes_classifications,
  "Naive_Bayes_Probability" = nbt

)

equivalent_classifications = exact_bayes_classifications == nbt

equivalent_ranking = order(-as.numeric(c(p1, p2, p3, p4, p5, p6))) == order(-
as.numeric(nbt))

comparison_result

##    Exact_Bayes_Classification Naive_Bayes_Probability
## 1                         yes                     yes
## 2                          no                      no
## 3                          no                      no
## 4                          no                      no
## 5                          no                     yes
## 6                         yes                      no
## 7                         yes                      no
## 8                          no                     yes
## 9                          no                      no
## 10                         no                      no
## 11                         no                      no
## 12                        yes                     yes
## 13                        yes                     yes
## 14                         no                     yes
## 15                         no                     yes
## 16                         no                     yes
## 17                         no                      no
## 18                        yes                      no
## 19                        yes                      no
## 20                         no                      no
```

```
## 21                          no                       yes
## 22                          no                       yes
## 23                          no                        no
## 24                         yes                        no

cat("Are the resulting classifications equivalent? ",
all(equivalent_classifications), "\n")

## Are the resulting classifications equivalent?  FALSE

cat("Is the ranking of observations equivalent? ", all(equivalent_ranking),
"\n")

## Is the ranking of observations equivalent?  FALSE
```

Let us use Caret

```
# nb2 = train(INJURY ~ TRAF_CON_R + WEATHER_R,
#        data = accidents24, method = "nb")
#
# predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R",
"TRAF_CON_R")])
# predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R",
"TRAF_CON_R")],
#                                      type = "raw")
```

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

- 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.
- What is the overall error of the validation set?

```
set.seed(1)

accidents_new = read.csv("C:/Users/gdurg/Documents/FML
ASSIGNMENTS/accidentsFull.csv")


accidents_new$INJURY = ifelse(accidents_new$MAX_SEV_IR>0,1,0)

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


# splitIndex = createDataPartition(accidents_new$INJURY, p = 0.6,
#                                    list = FALSE)
#
# training_data = accidents_new[splitIndex, ]
#
```

```r
#
# validation_data = accidents_new[-splitIndex, ]

train.split = sample(row.names(accidents_new), 0.6*dim(accidents_new)[1])

valid.split = setdiff(row.names(accidents_new), train.split)

training_data = accidents_new[train.split,]

validation_data = accidents_new[valid.split,]


nb_model = naiveBayes(INJURY ~ ., data = training_data)

nb_predictions = predict(nb_model, validation_data)



confusion_matrix = confusionMatrix(nb_predictions,
as.factor(validation_data$INJURY))

print(confusion_matrix)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 8219  205
##          1    0 8450
##
##                Accuracy : 0.9879
##                  95% CI : (0.9861, 0.9894)
##     No Information Rate : 0.5129
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9757
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.9763
##          Pos Pred Value : 0.9757
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4871
##          Detection Rate : 0.4871
##    Detection Prevalence : 0.4992
##       Balanced Accuracy : 0.9882
##
```

```
##        'Positive' Class : 0
##
```

```r
overall_error_rate = 1 - confusion_matrix$overall["Accuracy"]

cat("overall error of the validation set is", overall_error_rate)
```

```
## overall error of the validation set is 0.01214887
```