FML - Assignment 3 - Automobile Accidents

Gaurav Kudeshia

2023-10-15

Summary

- Predicting injury with no information:-

  When only initial accident reports are available, the most probable outcome is the presence of an injury. This is because the dataset shows that out of 42,183 accidents, 21,462 resulted in injuries, translating to a 50.88% chance.

- Converting variables to factors:-

  The Naive Bayes algorithm requires categorical variables to be in the factor class. Therefore, at the beginning of the analysis, the variables in the dataset were converted to factors.

- Naive Bayes conditional probability for WEATHER_R = 1 and TRAF_CON_R = 1:-

  Upon manually computing the Naive Bayes conditional probability of an injury when WEATHER_R = 1 and TRAF_CON_R = 1, the probability was determined to be “0.” This means that there were no accidents with WEATHER_R = 1 and TRAF_CON_R = 1 in the training dataset.

- Disparities between Naive Bayes and exact Bayes classifications:-

  The exact Bayes classifier makes no simplifying assumptions about the data. In contrast, the Naive Bayes classifier assumes that the predictors are independent of each other. This assumption may not always hold true in real-world data, which can explain the disparities between the Naive Bayes and exact Bayes classifications.

- Performance on the training dataset:-

  When the Naive Bayes classifier was applied to the entire training dataset with all relevant predictors, it achieved an accuracy rate of 53.8%. This means that the classifier correctly predicted the outcome of 53.8% of the accidents in the training set.

- Performance on the validation set:-

  The overall error of the validation set was computed as “46.3.” This means that the classifier incorrectly predicted the outcome of 46.3% of the accidents in the validation set.

- Performed Manually:-

  When using the provided data with `WEATHER_R = 1` and `TRAF_CON_R = 1` and manually calculating the Naive Bayes conditional probability of injury=yes, the numerator was determined to be “0.” This was because the probability term `Probability(TRAF_CON_R=1|Injury=Yes)` in the formula resulted in 0/9, leading to the entire answer being “0”.

Conclusion

- The Naive Bayes classifier demonstrated an accuracy rate of 53.8% on the training dataset. However, its performance significantly declined on the validation set, resulting in an error rate of 46.3%. This discrepancy hints at potential over fitting of the classifier to the training data.
- Initially, the Naive Bayes classifier was applied to predict injury outcomes in a subset of 24 records and later extended to the entire data set, utilizing two predictors in both instances.
- Utilizing the exact Bayes classifier on the initial 24 records revealed that when WEATHER_CON=2 and TRAF_CON=0, the probability of injury is maximized, denoted as “1.” This indicates that this specific combination poses the highest risk for drivers.
- On the training set, the model achieved an accuracy of 53.7%, accompanied by a validation error rate of 46.3%, suggesting moderate predictive capability:-
    - However, it’s essential to note that the Naive Bayes classifier assumes independence between predictor variables, a condition not always met in real-world scenarios.
    - Consequently, this assumption can lead to inaccuracies. Nevertheless, the Naive Bayes classifier remains valuable for ranking and classification tasks.
- While Naive Bayes offers a straightforward and effective approach for injury outcome prediction, it is imperative to recognize and account for its inherent limitations.

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.

   2.1:- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors. 2.2:- Classify the 24 accidents using these probabilities and a cutoff of 0.5. 2.3:- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1. 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?

3. Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%). 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. 3.2:- What is the overall error of the validation set?

Loaded datasets from the necessary libraries

```
library(klaR)
```

```
## Loading required package: MASS
```

```
library(e1071)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(klaR)
library(ggplot2)
```

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?

Installed the dataset from “accidentsFull.csv”, computed the counts for occurrences of “MAX_SEV_IR = yes or no,” and generated a concise summary table

```
accidents <- read.csv("/Users/gauravkudeshia/Downloads/accidentsFull.csv")
accidents$INJURY = ifelse(accidents$MAX_SEV_IR>0,"yes","no")
injury_table <- table(accidents$INJURY)
injury_table
```

```
##
##    no   yes
## 20721 21462
```

Predicted “INJURY” status in the absence of additional information

```
probability_injury <- (injury_table["yes"] / sum(injury_table))*100
probability_injury
```

```
##      yes
## 50.87832
```

Transformed variables into the factor data type

```
for (i in c(1:dim(accidents)[2])){
  accidents[,i] <- as.factor(accidents[,i])
}
head(accidents,n=24)
```

1:- (INJURY = Yes or No?) Why?

Ans 1:- Based on the available dataset, if an accident has just been reported with no additional information, there is a 50.88% chance that an injury occurred. This conclusion is drawn from the data, which shows that 21,462 out of 42,183 accidents resulted in an injury.

Questions

2. Select the first 24 records in the data set 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.

Selected first “24” rows from the “accidents” data set

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

```
##   INJURY WEATHER_R TRAF_CON_R
## 1    yes         1          0
## 2     no         2          0
## 3     no         2          1
## 4     no         1          1
## 5     no         1          0
## 6    yes         2          0
```

Created a Pivot table of “accidents24” data set for calculating the exact Bayes conditional probabilities

```
table_pivot.1 <- ftable(accidents24)
table_pivot.2 <- ftable(accidents24[,-1]) # print table only for conditions
table_pivot.1
```

```
##                   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
```

```
table_pivot.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.

Calculated precise Bayes conditional probabilities for injury occurrence

```
Probability_1 = table_pivot.1[3,1] / table_pivot.2[1,1] # Injury, Weather=1 and Tr
af=0
Probability_2 = table_pivot.1[4,1] / table_pivot.2[2,1] # Injury, Weather=2, Traf=
0
Probability_3 = table_pivot.1[3,2] / table_pivot.2[1,2] # Injury, W=1, T=1
Probability_4 = table_pivot.1[4,2] / table_pivot.2[2,2] # I, W=2,T=1
Probability_5 = table_pivot.1[3,3] / table_pivot.2[1,3] # I, W=1,T=2
Probability_6 = table_pivot.1[4,3]/ table_pivot.2[2,3] #I,W=2,T=2

# Injury = no
Probability_No_1 = table_pivot.1[1,1] / table_pivot.2[1,1] # Weather=1 and Traf=0
Probability_No_2 = table_pivot.1[2,1] / table_pivot.2[2,1] # Weather=2, Traf=0
Probability_No_3 = table_pivot.1[1,2] / table_pivot.2[1,2] # W=1, T=1
Probability_No_4 = table_pivot.1[2,2] / table_pivot.2[2,2] # W=2,T=1
Probability_No_5 = table_pivot.1[1,3] / table_pivot.2[1,3] # W=1,T=2
Probability_No_6 = table_pivot.1[2,3] / table_pivot.2[2,3] # W=2,T=2
print(c(Probability_1,Probability_2,Probability_3,Probability_4,Probability_5,Prob
ability_6))
```

```
## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000
```

```
print(c(Probability_No_1,Probability_No_2,Probability_No_3,Probability_No_4,Probab
ility_No_5,Probability_No_6))
```

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

2. Let us now compute

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

```
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] = Probability_1
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = Probability_3
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = Probability_5
      }
    }
    else {
      if (accidents24$TRAF_CON_R[i]=="0"){
        prob.inj[i] = Probability_2
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = Probability_4
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = Probability_6
      }
    }
  }
```

```
accidents24$prob.inj <- prob.inj
accidents24$prob.inj
```

```
##  [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.6666667 0.1818182 0.1818182
##  [8] 0.6666667 0.1818182 0.1818182 0.1818182 0.0000000 0.6666667 0.6666667
## [15] 0.6666667 0.6666667 0.1818182 0.1818182 0.1818182 0.1818182 0.6666667
## [22] 0.6666667 1.0000000 0.1818182
```

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

```
##  [1] "yes" "no"  "no"  "no"  "yes" "no"  "no"  "yes" "no"  "no"  "no"  "no" 
## [13] "yes" "yes" "yes" "yes" "no"  "no"  "no"  "no"  "yes" "yes" "yes" "no"
```

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

Manual Solution

Printed printed pivot table for manual calculation

```
table_pivot.1
```

```
##                    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
```

Took data from the above table

Probability(Injury=Yes/WEATHER_R=1,TRAF_CON_R=1)

= [Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Yes)] / [Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Yes) + Probability(WEATHER_R=1/Injury=No) * Probability(TRAF_CON_R=1/Injury=No) * Probability(Injury=No)]

= [6/9 * 0/9 * 9/24] / [6/9 * 0/9 * 9/24 + 5/15 * 2/15 * 15/24] = The result will be “0” since the numerator is equal to zero.

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?

Applied a naive Bayes classifier to 24 records.

```
n.B <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                  data = accidents24)
n.B
```

```
## 
## Naive Bayes Classifier for Discrete Predictors
## 
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
## 
## A-priori probabilities:
## Y
##    no   yes 
## 0.625 0.375 
## 
## Conditional probabilities:
##      TRAF_CON_R
## Y              0          1          2
##   no  0.80000000 0.13333333 0.06666667
##   yes 0.88888889 0.00000000 0.11111111
## 
##      WEATHER_R
## Y             1         2
##   no  0.3333333 0.6666667
##   yes 0.6666667 0.3333333
```

```
n.Bt <- predict(n.B, newdata = accidents24,type = "raw")

# Transfered the "Yes" n.B prediction

accidents24$n.B_Predicted_probability <- n.Bt[,2]
accidents24$n.B_Predicted_probability
```

```
##  [1] 0.571428571 0.250000000 0.002244949 0.008919722 0.571428571 0.250000000
##  [7] 0.250000000 0.571428571 0.250000000 0.250000000 0.250000000 0.666666667
## [13] 0.571428571 0.571428571 0.571428571 0.571428571 0.250000000 0.250000000
## [19] 0.250000000 0.250000000 0.571428571 0.571428571 0.333333333 0.250000000
```

Let us use Caret

```
# Imported essential libraries
library(klaR)

# Defined a formula ("F") incorporating all relevant variables
F <- INJURY ~ TRAF_CON_R + WEATHER_R

# Developed the Naive Bayes model using Laplace smoothing
n.B2 <- NaiveBayes(F, data = accidents24, laplace = 1)

# Generated predictions using the trained model
 predict(n.B2, newdata = accidents24[, c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

```
## $class
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
## yes  no  no  no yes  no  no yes  no  no  no yes yes yes yes yes  no  no  no  no 
##  21  22  23  24 
## yes yes  no  no 
## Levels: no yes
## 
## $posterior
##           no         yes
## 1  0.4285714 0.571428571
## 2  0.7500000 0.250000000
## 3  0.9977551 0.002244949
## 4  0.9910803 0.008919722
## 5  0.4285714 0.571428571
## 6  0.7500000 0.250000000
## 7  0.7500000 0.250000000
## 8  0.4285714 0.571428571
## 9  0.7500000 0.250000000
## 10 0.7500000 0.250000000
## 11 0.7500000 0.250000000
## 12 0.3333333 0.666666667
## 13 0.4285714 0.571428571
## 14 0.4285714 0.571428571
## 15 0.4285714 0.571428571
## 16 0.4285714 0.571428571
## 17 0.7500000 0.250000000
## 18 0.7500000 0.250000000
## 19 0.7500000 0.250000000
## 20 0.7500000 0.250000000
## 21 0.4285714 0.571428571
## 22 0.4285714 0.571428571
## 23 0.6666667 0.333333333
## 24 0.7500000 0.250000000
```

```
# Raw probabilities can also be obtained from the model predictions
Prediction <- predict(n.B2, newdata = accidents24[, c("INJURY", "WEATHER_R", "TRAF
_CON_R")], type = "raw")
Prediction
```

```
## $class
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
## yes  no  no  no yes  no  no yes  no  no  no yes yes yes yes yes  no  no  no  no 
##  21  22  23  24 
## yes yes  no  no 
## Levels: no yes
## 
## $posterior
##           no         yes
## 1  0.4285714 0.571428571
## 2  0.7500000 0.250000000
## 3  0.9977551 0.002244949
## 4  0.9910803 0.008919722
## 5  0.4285714 0.571428571
## 6  0.7500000 0.250000000
## 7  0.7500000 0.250000000
## 8  0.4285714 0.571428571
## 9  0.7500000 0.250000000
## 10 0.7500000 0.250000000
## 11 0.7500000 0.250000000
## 12 0.3333333 0.666666667
## 13 0.4285714 0.571428571
## 14 0.4285714 0.571428571
## 15 0.4285714 0.571428571
## 16 0.4285714 0.571428571
## 17 0.7500000 0.250000000
## 18 0.7500000 0.250000000
## 19 0.7500000 0.250000000
## 20 0.7500000 0.250000000
## 21 0.4285714 0.571428571
## 22 0.4285714 0.571428571
## 23 0.6666667 0.333333333
## 24 0.7500000 0.250000000
```

Compared exact Bayes classification

```
# Evaluated the Naive Bayes model against the exact Bayes classification.
classification_match <- all(accidents24$n.B_Predicted_probability == accidents24$p
rob.inj)
probability_match <- all.equal(accidents24$n.B_Predicted_probability, accidents24$
prob.inj)

# Verified if classifications and rankings match
if (classification_match && is.na(probability_match)) {
  cat("The resulting classifications and rankings match perfectly.\n")
} else {
  cat("The resulting classifications and rankings do not align.\n")
}
```

```
## The resulting classifications and rankings do not align.
```

Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

Ans 2.4 - When the naive Bayes classifier was applied to the 24 records with two predictors, the resulting classifications and rankings did not match those obtained from the exact Bayes calculation.

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

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.

```
set.seed(123)
Tr_Index <- sample(c(1:dim(accidents)[1]), dim(accidents)[1]*0.6)
Tr_Index_Data <- accidents[Tr_Index,]
Va_Index_Data <- accidents[-Tr_Index,]

V <- c("INJURY", "HOUR_I_R",  "ALIGN_I" ,"WRK_ZONE",  "WKDY_I_R",
          "INT_HWY",  "LGTCON_I_R", "PROFIL_I_R", "SPD_LIM", "SUR_COND",
          "TRAF_CON_R",   "TRAF_WAY",   "WEATHER_R")

n.BTotal <- naiveBayes(INJURY~.,data = Tr_Index_Data[,V])
n.BTotal
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        no       yes
## 0.4903789 0.5096211
##
## Conditional probabilities:
##      HOUR_I_R
## Y             0         1
##   no  0.5690919 0.4309081
##   yes 0.5690029 0.4309971
##
##      ALIGN_I
## Y             1         2
##   no  0.8726936 0.1273064
##   yes 0.8696697 0.1303303
##
```

```
##      WRK_ZONE
## Y              0          1
##   no  0.97502216 0.02497784
##   yes 0.97883393 0.02116607
##
##      WKDY_I_R
## Y             0         1
##   no  0.2190798 0.7809202
##   yes 0.2384091 0.7615909
##
##      INT_HWY
## Y                0            1            9
##   no  0.8491660624 0.1501087745 0.0007251632
##   yes 0.8617615134 0.1374631726 0.0007753140
##
##      LGTCON_I_R
## Y             1         2         3
##   no  0.6871324 0.1285150 0.1843526
##   yes 0.6957668 0.1131958 0.1910374
##
##      PROFIL_I_R
## Y             0         1
##   no  0.7555394 0.2444606
##   yes 0.7617460 0.2382540
##
##      SPD_LIM
## Y                5           10           15           20           25
##   no  8.057368e-05 7.251632e-04 4.673274e-03 8.299090e-03 1.099831e-01
##   yes 7.753140e-05 3.876570e-04 4.419290e-03 4.729415e-03 9.094433e-02
##      SPD_LIM
## Y               30           35           40           45           50
##   no  8.726130e-02 1.892676e-01 9.411006e-02 1.560712e-01 4.101201e-02
##   yes 8.885098e-02 2.163901e-01 1.076911e-01 1.554505e-01 3.806792e-02
##      SPD_LIM
## Y               55           60           65           70           75
##   no  1.604222e-01 3.545242e-02 6.711788e-02 3.948111e-02 6.043026e-03
##   yes 1.532020e-01 4.209955e-02 6.179253e-02 2.876415e-02 7.132889e-03
##
##      SUR_COND
## Y               1           2           3           4           9
##   no  0.778341794 0.173797438 0.015550721 0.028120216 0.004189832
##   yes 0.815475267 0.153434641 0.011397116 0.015118623 0.004574353
##
##      TRAF_CON_R
## Y             0         1         2
##   no  0.6581259 0.1907985 0.1510757
##   yes 0.6217243 0.2203442 0.1579315
##
##      TRAF_WAY
## Y              1          2          3
##   no  0.57360406 0.37426477 0.05213117
##   yes 0.56419600 0.39471236 0.04109164
##
##      WEATHER_R
## Y             1         2
##   no  0.8411893 0.1588107
##   yes 0.8717631 0.1282369
```

```
#Created a confusion matrix using the Tr_Index_Data, the predictions, and the corr
esponding classes.
Confu_m <- confusionMatrix(Tr_Index_Data$INJURY, predict(n.BTotal, Tr_Index_Data[,
V]), positive = "yes")
Confu_m
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5214 7197
##        yes 4475 8423
##
##                Accuracy : 0.5388
##                  95% CI : (0.5327, 0.545)
##     No Information Rate : 0.6172
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0735
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5392
##             Specificity : 0.5381
##          Pos Pred Value : 0.6530
##          Neg Pred Value : 0.4201
##              Prevalence : 0.6172
##          Detection Rate : 0.3328
##    Detection Prevalence : 0.5096
##       Balanced Accuracy : 0.5387
##
##        'Positive' Class : yes
##
```

3.2, What is the overall error of the validation set?

```
Confu_M <- confusionMatrix(Va_Index_Data$INJURY, predict(n.BTotal, Va_Index_Data[,
V]), positive = "yes")
Confu_M
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   no  yes
##        no  3444 4866
##        yes 2947 5617
##                                           
##                Accuracy : 0.537           
##                  95% CI : (0.5294, 0.5445)
##     No Information Rate : 0.6213          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.0706          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.5358          
##             Specificity : 0.5389          
##          Pos Pred Value : 0.6559          
##          Neg Pred Value : 0.4144          
##              Prevalence : 0.6213          
##          Detection Rate : 0.3329          
##    Detection Prevalence : 0.5075          
##       Balanced Accuracy : 0.5374          
##                                           
##        'Positive' Class : yes             
## 
```

Calculated overall error

```
Error <- 1-0.537
Error_per <- Error*100
paste("Overall Error percentage : ",Error_per)
```

```
## [1] "Overall Error percentage :  46.3"
```