FML Assignment 3

2023-10-15

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."

```
#load the required libraries
library(caret)
```

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

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

```
library(e1071)
```

```
#load the data
accident= read.csv("C://Users//Siri//Downloads//accidentsFull.csv")
head(accident) #displays first 6 records
```

```
##   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
## 1          0          1
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          1
```

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?

Create a dummy variable called 'INJURY'.

```
#value "yes" if MAX_SEV_IR = 1 or 2, and otherwise "no."
accident$INJURY= ifelse(accident$MAX_SEV_IR>0, "YES", "NO")
table(accident$INJURY)
```

```
## 
##    NO   YES 
## 20721 21462
```

If an accident has just been reported and there is no information available, it is predicted that there might be injuries i.e (INJURY = Yes). The goal is to predict whether an accident will involve an injury (MAX_SEV_IR = 1 or 2) or not (MAX_SEV_IR = 0).So, if you have no specific information about a new accident and you want to make an initial prediction, it would be reasonable to predict that there is a possibility of injury ("INJURY" = "Yes") because a proportion of accidents in the historical data resulted in injuries.

- There are total of " 20721 NO and 21462 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 24 records. Use all three variables in the pivot table as rows/columns.

```
#selecting first 24 records and look at response (INJURY) and 2 predictors WEATHER_R and TRAF_CON_R
#CONVERTING THE VARIABLES TO CATEGORICAL TYPE
# IDENTIFYING THE TARGET VARIABLE COLUMN INDEX (ASSUMING IT'S THE LAST COLUMN)
target_col = dim(accident)[2]

#CONVERTING ALL COLUMNS EXCEPT THE TARGRT VARIABLE TO FACTORS
accident[, 1:(target_col - 1)] = lapply(accident[, 1:(target_col - 1)], as.factor)

#create a new subset with only the required records
accident_24 = accident[1:24, c("INJURY", "WEATHER_R", "TRAF_CON_R")]
accident_24
```

```
##    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
## 7      NO         2          0
## 8     YES         1          0
## 9      NO         2          0
## 10     NO         2          0
## 11     NO         2          0
## 12     NO         1          2
## 13    YES         1          0
## 14     NO         1          0
## 15    YES         1          0
## 16    YES         1          0
## 17     NO         2          0
## 18     NO         2          0
## 19     NO         2          0
## 20     NO         2          0
## 21    YES         1          0
## 22     NO         1          0
## 23    YES         2          2
## 24    YES         2          0
```

```
#creating pivot table
t1= ftable(accident_24)
t2= ftable(accident_24[,-1]) # table for fatality
t1
```

```
##                   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
```

```
t2
```

```
##           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.

```
# P(INJURY= YES|WEATHER_R= 1, TRAF_CON_R= 0)
P1= t1[3,1]/t2[1,1]
P1
```

```
## [1] 0.6666667
```

```
# P(INJURY= YES|WEATHER_R= 2, TRAF_CON_R= 0)
P2= t1[4,1]/t2[2,1]
P2
```

```
## [1] 0.1818182
```

```
# P(INJURY= YES|WEATHER_R= 1, TRAF_CON_R= 1)
P3= t1[3,2]/t2[1,2]
P3
```

```
## [1] 0
```

```
# P(INJURY= YES|WEATHER_R= 2, TRAF_CON_R= 1)
P4= t1[4,2]/t2[2,2]
P4
```

```
## [1] 0
```

```
# P(INJURY= YES|WEATHER_R= 1, TRAF_CON_R= 2)
P5= t1[3,3]/t2[1,3]
P5
```

```
## [1] 0
```

```
# P(INJURY= YES|WEATHER_R= 2, TRAF_CON_R= 2)
P6= t1[4,3]/t2[2,3]
P6
```

```
## [1] 1
```

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

```
#Adding probability
accident_24_prob= accident_24
head(accident_24_prob)
```

```
##   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
```

```
probability.injury = c(0.667, 0.167, 0, 0, 0.667, 0.167, 0.167, 0.667, 0.167, 0.167, 0.167, 0)

accident_24_prob$PROB_INJ = rep(probability.injury, length.out = nrow(accident_24_prob))

#Add column for injury prediction based on cutoff of 0.5.
accident_24_prob$PROB_PREDICT=ifelse(accident_24_prob$PROB_INJ>.5,"YES","NO")
accident_24_prob
```

```
##    INJURY WEATHER_R TRAF_CON_R PROB_INJ PROB_PREDICT
## 1     YES         1          0    0.667          YES
## 2      NO         2          0    0.167           NO
## 3      NO         2          1    0.000           NO
## 4      NO         1          1    0.000           NO
## 5      NO         1          0    0.667          YES
## 6     YES         2          0    0.167           NO
## 7      NO         2          0    0.167           NO
## 8     YES         1          0    0.667          YES
## 9      NO         2          0    0.167           NO
## 10     NO         2          0    0.167           NO
## 11     NO         2          0    0.167           NO
## 12     NO         1          2    0.000           NO
## 13    YES         1          0    0.667          YES
## 14     NO         1          0    0.167           NO
## 15    YES         1          0    0.000           NO
## 16    YES         1          0    0.000           NO
## 17     NO         2          0    0.667          YES
## 18     NO         2          0    0.167           NO
## 19     NO         2          0    0.167           NO
## 20     NO         2          0    0.667          YES
## 21    YES         1          0    0.167           NO
## 22     NO         1          0    0.167           NO
## 23    YES         2          2    0.167           NO
## 24    YES         2          0    0.000           NO
```

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

```
# P(INJURY= YES|WEATHER_R = 1, TRAF_CON_R = 1)
# let INJURY= YES -> Y,INJURY= NO -> N, WEATHER_R -> W,  TRAF_CON_R -> T
# [P(W=1|Y)*P(T=1|Y)*P(Y)] / [P(W=1,T=1)]
# [P(W=1|Y)*P(T=1|Y)*P(Y)] /                                    [P(W=1|Y)*P(T=1|Y)*P(Y)]+[P(W=1|N)*P(T=1|N)*P
(N)]

numerator= 6/9 * 0 * 9/24
denominator= (6/9 * 0 * 9/24)+(5/15 * 2/15 * 15/24)
naive_bayes= numerator/denominator
naive_bayes
```

```
## [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?

```
library(e1071)
library(klaR)
```

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

```
library(caret)
nb=naiveBayes(INJURY ~ ., data =accident_24 )
predict(nb, newdata = accident_24,type = "raw")
```

```
##              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
```

```
#Check the model with caret package
library(caret)
x= accident_24[,-3]
y=accident_24$INJURY
model = train(x,y,'nb', trControl = trainControl(method = 'cv',number=10))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
#Now the generated classification model can be used for prediction
model.pred=predict(model$finalModel,x)


#Creating a confusion matrix to visualize classification errors.
table(model.pred$class,y)
```

```
##      y
##       NO YES
##   NO  15   0
##   YES  0   9
```

```
#Comparing
accident_24_prob$PREDICT_PROB_NB<-model.pred$class
accident_24_prob
```

```
##    INJURY WEATHER_R TRAF_CON_R PROB_INJ PROB_PREDICT PREDICT_PROB_NB
## 1     YES         1          0    0.667          YES             YES
## 2      NO         2          0    0.167           NO              NO
## 3      NO         2          1    0.000           NO              NO
## 4      NO         1          1    0.000           NO              NO
## 5      NO         1          0    0.667          YES              NO
## 6     YES         2          0    0.167           NO             YES
## 7      NO         2          0    0.167           NO              NO
## 8     YES         1          0    0.667          YES             YES
## 9      NO         2          0    0.167           NO              NO
## 10     NO         2          0    0.167           NO              NO
## 11     NO         2          0    0.167           NO              NO
## 12     NO         1          2    0.000           NO              NO
## 13    YES         1          0    0.667          YES             YES
## 14     NO         1          0    0.167           NO              NO
## 15    YES         1          0    0.000           NO             YES
## 16    YES         1          0    0.000           NO             YES
## 17     NO         2          0    0.667          YES              NO
## 18     NO         2          0    0.167           NO              NO
## 19     NO         2          0    0.167           NO              NO
## 20     NO         2          0    0.667          YES              NO
## 21    YES         1          0    0.167           NO             YES
## 22     NO         1          0    0.167           NO              NO
## 23    YES         2          2    0.167           NO             YES
## 24    YES         2          0    0.000           NO             YES
```

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

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.

```
# Load the necessary libraries
library(e1071)
library(caret)

# Assuming your dataset is loaded and named "accidentsFull"

# Set the seed for reproducibility
set.seed(223)

# Split the data into training and validation sets
trainIndex <- createDataPartition(accident$INJURY, p = 0.6, list = FALSE)
train_data <- accident[trainIndex, ]
validation_data <- accident[-trainIndex, ]

# Run a Naive Bayes classifier on the training set
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = train_data)

# Use the model to predict on the validation set
predictions <- predict(nb_model, newdata = validation_data)

# Create the confusion matrix
confusionMatrix(as.factor(validation_data$INJURY),predictions)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   NO  YES
##        NO  1271 7017
##        YES 1119 7465
##                                           
##                Accuracy : 0.5178          
##                  95% CI : (0.5102, 0.5253)
##     No Information Rate : 0.8583          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.0233          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.53180         
##             Specificity : 0.51547         
##          Pos Pred Value : 0.15335         
##          Neg Pred Value : 0.86964         
##              Prevalence : 0.14165         
##          Detection Rate : 0.07533         
##    Detection Prevalence : 0.49123         
##       Balanced Accuracy : 0.52363         
##                                           
##        'Positive' Class : NO              
## 
```

```
conf_matrix <- table(predictions, validation_data$INJURY)
print(conf_matrix)
```

```
## 
## predictions   NO  YES
##         NO  1271 1119
##         YES 7017 7465
```

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

```
# Calculate the overall error
error_rate <- 1 - sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Overall error of the validation set:", error_rate))
```

```
## [1] "Overall error of the validation set: 0.48221906116643"
```

The overall error rate on the validation set is approximately 0.48 when expressed as a decimal. It indicates that the Naive Bayes classifier performs very well on this dataset, with a high level of accuracy.