FML ASSIGN_3

ALLEN RICHARDS

2023-10-15

Summary :

```
This below problem shows the data about automobile accidents in USA. It requires analyzing in
formation from the dataset such as day of the week, weather conditions, and road type. In the
statement the goal is clearly mentioned to predict whether these accidents will ends in an in
jury or not or based on the data which is available in datafile.
There are some steps I need to take before I initiate the code.
1.Importing Data
2.Preparation (processing,cleaning or removing unwanted informations)
3.Analyze the information (to find the relationship or pattern between one to another)
4.Creation of Variables (according to data and to get better insight)
5.Generating Result (output)
```


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?

In these types of situation we can make predictions according to pattern or relationship between the information. So we can finalize as INJURY = YES or INJURY = NO (baseline assumption). I have created a “INJURY” variable based on “MAX_SEV_IR” also it is set to 1 or 2 for this variable as “YES” or else “NO”. It shows injury level 1 or 2 and there will be no injury if it set to “0”. These details were based by dataset information. If there is limited information about accident in baseline assumption then the result will be “no”

These predictions based on available source of information and it may affect the severity of other variable such as location, weather conditions etc.

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 indexing and column section used by the first 24 records and the relevant columns “INJURY,” “WEATHER_R,” and “TRAF_CON_R”. I created a pivot table using the functions It specifies the variable to summarize (INJURY). I can used other functions to compensate required missing values.

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

The calculation of prior probabilities of each predictor combinations were made to identify the occurrence value for “WEATHER_R” and “TRAF_CON_R.” data frame will contain the Bayes conditional probabilities of an injury (INJURY = Yes)

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

The resulting classified_data data frame will include the classification of each accident as “Yes” (injury) or “No” (no injury) based on the calculated Bayes conditional probabilities and the specified cutoff of 0.5 for each combination of the predictors “WEATHER_R” and “TRAF_CON_R.” Classify the 24 accidents using these probabilities and a cutoff of 0.5

I define a function classify_accident that takes “WEATHER_R” and “TRAF_CON_R” as input, looks up the corresponding probability from final_probabilities, and then compares the probability to the cutoff of 0.5 to classify the accident as “Yes” or “No.”

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

In this scenario I made a formulations to identify the required values of probabilities “WEATHER_R = 1” and “TRAF_CON_R = 1” the exact calculation depends on the specific probabilities I have in dataset. It need to substitute these probabilities into the formula to compute the Naive Bayes conditional probability of an injury

given “WEATHER_R = 1” and “TRAF_CON_R = 1.”

(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?

The resulting classifications and ranking of observations are equivalent between the Naive Bayes and exact Bayes methods It makes certain independence assumptions, so the results may not be exactly the same as the exact Bayes classification, especially if there are dependencies between the predictors.

```
The probability is 0.6666667    INJURY yes
The probability is 0.1818182    INJURY no
```

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

To split the data randomly while maintaining the specified proportions. I used the caret package, which provides a convenient function for data splitting.

(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.

The Naive Bayes classifier on the complete training set with categorical predictors and “INJURY” as the response variable, you can use the naiveBayes function from the e1071 package and then generate a confusion matrix to evaluate the classifier’s performance.

```
     Accuracy : 0.3
             95% CI : (0.0667, 0.6525)
No Information Rate : 0.6
P-Value [Acc > NIR] : 0.9877
              Kappa : -0.2069
```

Mcnemar's Test P-Value : 0.1306
Sensitivity : 0.7500
Specificity : 0.0000
Pos Pred Value : 0.3333
Neg Pred Value : 0.0000
Prevalence : 0.4000
Detection Rate : 0.3000
Detection Prevalence : 0.9000
Balanced Accuracy : 0.3750
'Positive' Class : no

(B)What is the overall error of the validation set?

As per the values mentioned in data set. Formulations made to identify the required values. After finalizing the entire validation set to Overall Error Rate: 0.7

```
accidents <- read.csv("D:/KSU SEM-1/FUNDAMENTAL OF MACHINE LEARNING/Assignment_3/accidentsFul
l.csv")
accidents$INJURY = ifelse(accidents$MAX_SEV_IR>0,"yes","no")


# 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
```

Question - 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

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

```
dt1 <- ftable(accidents24)
dt2 <- ftable(accidents24[,-1]) # print table only for conditions
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
```

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

Question 2.(A)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
```

Question 2(B) 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] = 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")
```

Question 2(C) Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1

```
# Calculate P(INJURY = "Yes")
p_injury_yes <- sum(accidents24$INJURY == "yes") / nrow(accidents24)

# Calculate P(WEATHER_R = 1 | INJURY = "Yes")
p_weather_1_given_injury_yes <- sum(accidents24$INJURY == "yes" & accidents24$WEATHER_R == 1)
/ sum(accidents24$INJURY == "yes")

# Calculate P(TRAF_CON_R = 1 | INJURY = "Yes")
p_traf_con_1_given_injury_yes <- sum(accidents24$INJURY == "yes" & accidents24$TRAF_CON_R ==
1) / sum(accidents24$INJURY == "yes")

# Calculate the conditional probability
conditional_prob <- p_injury_yes * p_weather_1_given_injury_yes * p_traf_con_1_given_injury_y
es
```

Question 2(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?

```
# Load the e1071 package for Naive Bayes
library(e1071)

# Train a naive Bayes classifier
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = accidents24)

# Make predictions for probabilities of "YES" (INJURY)
nbt <- predict(nb, newdata = accidents24, type = "raw")

# Add the probability of the positive class ("YES") to your dataset
accidents24$nbpred.prob <- nbt[,"yes"]

# View the updated dataset
head(accidents24)
```

```
##   INJURY WEATHER_R TRAF_CON_R  prob.inj pred.prob nbpred.prob
## 1    yes         1          0 0.6666667       yes 0.571428571
## 2     no         2          0 0.1818182        no 0.250000000
## 3     no         2          1 0.0000000        no 0.002244949
## 4     no         1          1 0.0000000        no 0.008919722
## 5     no         1          0 0.6666667       yes 0.571428571
## 6    yes         2          0 0.1818182        no 0.250000000
```

Now use “CARET”

```
library(caret)
```

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

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

```
nb2 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = accidents24, method = "nb")
```

```
## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2
```

```
## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample03: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample05: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample06: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample11: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample12: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample14: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample15: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample17: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample18: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample19: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample20: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

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

```
## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

```
##  [1] no no no no no no no no no no no no no no no no no no no no no no no no no no
## Levels: no yes
```

```
predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

```
##  [1] no no no no no no no no no no no no no no no no no no no no no no no no no
## Levels: no yes
```

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

```
# Set the seed for reproducibility
set.seed(123)

# Define the proportion for the training set (e.g., 60%)
train_proportion <- 0.6

# Generate a vector of indices for the training set
train_indices <- sample(1:nrow(accidents24), size = round(train_proportion * nrow(accidents2
4)))

# Create the training and validation sets based on the indices
trainingData <- accidents24[train_indices, ]
validationData <- accidents24[-train_indices, ]
```

Question 3(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

```
install.packages("e1071")
```

```
## Warning: package 'e1071' is in use and will not be installed
```

```
library(e1071)

# Assuming we have already split our data into training and validation sets
# For this example, we'll use the trainingData and validationData

# Build the naive Bayes model
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = trainingData)

# Make predictions on the validation set
nb_predictions <- predict(nb_model, validationData, type = "class")

# Load the caret package for confusion matrix
library(caret)

# Create a confusion matrix
confusion_matrix <- confusionMatrix(nb_predictions, validationData$INJURY)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no   3   6
##        yes  1   0
##
##                Accuracy : 0.3
##                  95% CI : (0.0667, 0.6525)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.9877
##
##                   Kappa : -0.2069
##
##  Mcnemar's Test P-Value : 0.1306
##
##             Sensitivity : 0.7500
##             Specificity : 0.0000
##          Pos Pred Value : 0.3333
##          Neg Pred Value : 0.0000
##              Prevalence : 0.4000
##          Detection Rate : 0.3000
##    Detection Prevalence : 0.9000
##       Balanced Accuracy : 0.3750
##
##        'Positive' Class : no
##
```

Question 3(B) What is the overall error of the validation set?

```
# Assuming you have already created a confusion matrix named 'confusion_matrix' as shown in t
he previous response

# Calculate the error rate
error_rate <- 1 - confusion_matrix$overall["Accuracy"]
cat("Overall Error Rate: ", error_rate, "\n")
```

```
## Overall Error Rate:  0.7
```