Assignment FML 3

Shreya Thodupunuri 811301506

2023-10-14

## R Markdown

# 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?**

# Load the accidentsFull.csv dataset  
# Load required libraries  
library(readr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

# Change the path to the CSV file  
file\_path <- "C:/Users/shrey/OneDrive/Desktop/accidentsFull.csv"  
  
# Read the dataset  
accidentsFull <- read\_csv(file\_path)

## Rows: 42183 Columns: 24

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (24): HOUR\_I\_R, ALCHL\_I, ALIGN\_I, STRATUM\_R, WRK\_ZONE, WKDY\_I\_R, INT\_HWY...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(accidentsFull)

## # A tibble: 6 × 24  
## HOUR\_I\_R ALCHL\_I ALIGN\_I STRATUM\_R WRK\_ZONE WKDY\_I\_R INT\_HWY LGTCON\_I\_R  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 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  
## # ℹ 16 more variables: MANCOL\_I\_R <dbl>, PED\_ACC\_R <dbl>, RELJCT\_I\_R <dbl>,  
## # REL\_RWY\_R <dbl>, PROFIL\_I\_R <dbl>, SPD\_LIM <dbl>, SUR\_COND <dbl>,  
## # TRAF\_CON\_R <dbl>, TRAF\_WAY <dbl>, VEH\_INVL <dbl>, WEATHER\_R <dbl>,  
## # INJURY\_CRASH <dbl>, NO\_INJ\_I <dbl>, PRPTYDMG\_CRASH <dbl>, FATALITIES <dbl>,  
## # MAX\_SEV\_IR <dbl>

#removing null values   
accidentsFull\_na\_omit <- na.omit(accidentsFull)  
  
# Create the INJURY variable based on MAX\_SEV\_IR  
accidentsFull\_na\_omit <- mutate(accidentsFull, INJURY = ifelse(MAX\_SEV\_IR %in% c(1, 2), "Yes", "No"))  
  
# Calculate the proportion of accidents with and without injury  
total\_accidents <- nrow(accidentsFull\_na\_omit)  
injury\_accidents <- sum(accidentsFull\_na\_omit$INJURY == "Yes")  
no\_injury\_accidents <- sum(accidentsFull\_na\_omit$INJURY == "No")  
  
proportion\_injury <- injury\_accidents / total\_accidents  
proportion\_no\_injury <- no\_injury\_accidents / total\_accidents  
  
# If an accident has just been reported, we can use the proportions to make a preliminary prediction  
if (proportion\_injury > proportion\_no\_injury) {  
 cat("Preliminary Prediction: INJURY = Yes\n")  
} else {  
 cat("Preliminary Prediction: INJURY = No\n")  
}

## Preliminary Prediction: INJURY = Yes

proportion\_injury

## [1] 0.5087832

proportion\_no\_injury

## [1] 0.4912168

There are around 50.88% of accidents (INJURY = Yes) that result in injuries (proportion\_injury).

According to proportion\_no\_injury, the percentage of accidents without injuries (INJURY = No) is around 49.12%.

As the percentage of accidents resulting in injuries is marginally more than the percentage of incidents without injuries, the first hypothesis is that there is a greater chance of harm when an accident is recently recorded. Please be aware, however, that this is just a preliminary forecast based on previous data, and that further data and analysis may be required to provide a more precise prognosis for any given accident scenario.

**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.**

# Convert selected variables to categorical type (factors)  
columns\_to\_convert <- c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")  
  
new.df <- accidentsFull\_na\_omit[1:24, columns\_to\_convert]  
  
new.df[] <- lapply(new.df, as.factor) # Convert selected columns to factors  
  
# View the resulting data frame  
new.df

## # A tibble: 24 × 3  
## INJURY WEATHER\_R TRAF\_CON\_R  
## <fct> <fct> <fct>   
## 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   
## # ℹ 14 more rows

# Select the first 24 records and relevant columns  
subset\_data <- new.df # Assuming 'new.df' contains the first 24 records and relevant columns  
  
# Rename levels with descriptive names  
levels(subset\_data$WEATHER\_R) <- c("WEATHER\_R\_0 - Clear", "WEATHER\_R\_1 - Rain", "WEATHER\_R\_2 - Snow")  
levels(subset\_data$TRAF\_CON\_R) <- c("TRAF\_CON\_R\_0 - Normal", "TRAF\_CON\_R\_1 - Construction", "TRAF\_CON\_R\_2 - Special")  
  
# Create a pivot table  
pivot\_table <- table(subset\_data$WEATHER\_R, subset\_data$TRAF\_CON\_R, subset\_data$INJURY)  
  
# Use ftable to display the pivot table in an organized format  
formatted\_table <- ftable(pivot\_table)  
  
# Print the formatted table  
print(formatted\_table)

## No Yes  
##   
## WEATHER\_R\_0 - Clear TRAF\_CON\_R\_0 - Normal 3 6  
## TRAF\_CON\_R\_1 - Construction 1 0  
## TRAF\_CON\_R\_2 - Special 1 0  
## WEATHER\_R\_1 - Rain TRAF\_CON\_R\_0 - Normal 9 2  
## TRAF\_CON\_R\_1 - Construction 1 0  
## TRAF\_CON\_R\_2 - Special 0 1  
## WEATHER\_R\_2 - Snow TRAF\_CON\_R\_0 - Normal 0 0  
## TRAF\_CON\_R\_1 - Construction 0 0  
## TRAF\_CON\_R\_2 - Special 0 0

For the first 24 records in the dataset, the pivot table looks at the correlation between reported injuries (INJURY) and two predictors: weather (WEATHER\_R) and traffic (TRAF\_CON\_R). It displays the number of accidents with and without injuries for every set of meteorological and traffic parameters. For instance, it shows that there were three incidents without injuries and six with injuries under clear weather and typical traffic circumstances. Road safety policies may be informed by this knowledge, which also aids in understanding how these elements connect to accident outcomes.

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

# Calculate the total count of INJURY = Yes  
total\_injury\_yes <- sum(pivot\_table[, , "Yes"])  
  
# Initialize an array to store conditional probabilities  
exact\_probabilities <- array(0, dim = dim(pivot\_table))  
  
# Calculate the conditional probabilities for each combination  
for (i in 1:dim(pivot\_table)[1]) {  
 for (j in 1:dim(pivot\_table)[2]) {  
 p\_x\_given\_injury\_yes <- pivot\_table[i, j, "Yes"]  
 p\_x <- sum(pivot\_table[i, j, ])  
   
 # Calculate P(INJURY = Yes | X)  
 p\_injury\_given\_x <- (p\_x\_given\_injury\_yes / total\_injury\_yes) \* (total\_injury\_yes / p\_x)  
   
 exact\_probabilities[i, j, ] <- p\_injury\_given\_x  
 }  
}  
  
# Print the exact probabilities  
print(exact\_probabilities)

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] 0.6666667 0 0  
## [2,] 0.1818182 0 1  
## [3,] NaN NaN NaN  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 0.6666667 0 0  
## [2,] 0.1818182 0 1  
## [3,] NaN NaN NaN

For instance, the following outcome is shown for (WEATHER\_R = 1, TRAF\_CON\_R = 1):

There is a 0.67 chance that INJURY = Yes. The likelihood that INJURY = No will occur is around 0.33.

Regarding (TRAFF\_CON\_R = 2, WEATHER\_R = 1):

About 0.18 is the likelihood that INJURY = Yes. The likelihood that INJURY = No will occur is around 0.82

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

#For instance, in the outcome:  
  
#Regarding (TRAFF\_CON\_R = 1, WEATHER\_R = 1):  
  
#There is a 0.67 probability that INJURY = Yes.  
#The likelihood that INJURY = No will occur is roughly 0.33.  
  
#Regarding (TRAFF\_CON\_R = 2, WEATHER\_R = 1):  
  
#About 0.18 is the likelihood that INJURY = Yes.  
#The likelihood that INJURY = No will occur is roughly 0.82.  
  
  
# Create a function to classify accidents based on probabilities and cutoff  
classify\_accident <- function(prob) {  
 if (!is.na(prob) && prob >= 0.5) {  
 return("Yes")  
 } else {  
 return("No")  
 }  
}  
  
# Classification results for the 24 accidents  
classification <- array(NA, dim = dim(exact\_probabilities))  
  
for (i in 1:dim(exact\_probabilities)[1]) {  
 for (j in 1:dim(exact\_probabilities)[2]) {  
 for (k in 1:dim(exact\_probabilities)[3]) {  
 classification[i, j, k] <- classify\_accident(exact\_probabilities[i, j, k])  
 }  
 }  
}  
  
# Print the classification results  
print(classification)

## , , 1  
##   
## [,1] [,2] [,3]   
## [1,] "Yes" "No" "No"   
## [2,] "No" "No" "Yes"  
## [3,] "No" "No" "No"   
##   
## , , 2  
##   
## [,1] [,2] [,3]   
## [1,] "Yes" "No" "No"   
## [2,] "No" "No" "Yes"  
## [3,] "No" "No" "No"

The result matrix demonstrates how conditional probabilities for WEATHER\_R and TRAF\_CON\_R predictor combinations were used to classify occurrences. Six options are conceivable since each predictor has two levels. The output is a three-dimensional array with INJURY class labels “Yes” and “No” with WEATHER\_R and TRAF\_CON\_R dimensions.

Let's analyse each predictor's result interpretation:

Level 1 Normal/Clear predictor combination:

Injury chance exceeds 0.5 for the “Yes” class. Thus, this combination is “Yes” for INJURY. Since the likelihood is below 0.5, Level 2 for both predictors (Rain and Construction) yields “No,” indicating no harm.

The class is “Yes” since the likelihood exceeds the 0.5 limit. Combining Snow and Special Level 3 predictors: The “No” class is categorised because its probability is less than 0.5.

The probability is larger than 0.5 for the “Yes” class, suggesting “Yes.” The chance is below 0.5, making it “No” for the “No” class.

To conclude, the result matrix categorises incidents by conditional probability for every weather and traffic combination. An injury is “Yes” if its likelihood is 0.5 or greater; otherwise, it is “No.” This categorization helps predict traffic and weather injury risk.

**2.c Compute manually the naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1.**

# Values for the specific condition  
desired\_weather <- "1" # Weather condition: 1 corresponds to Rain  
desired\_traffic <- "1" # Traffic condition: 1 corresponds to Construction  
desired\_injury <- "Yes" # We want to find the probability when INJURY = "Yes"  
  
# Calculate the prior probability P(INJURY="Yes")  
prior\_probability\_injury\_yes <- sum(subset\_data$INJURY == desired\_injury) / nrow(subset\_data)  
  
# Calculate the probability P(WEATHER\_R = 1)  
probability\_weather\_r\_1 <- sum(subset\_data$WEATHER\_R == desired\_weather) / nrow(subset\_data)  
  
# Calculate the probability P(TRAF\_CON\_R = 1)  
probability\_traf\_con\_r\_1 <- sum(subset\_data$TRAF\_CON\_R == desired\_traffic) / nrow(subset\_data)  
  
# Initialize the Naive Bayes conditional probability  
naive\_bayes\_probability <- 0  
  
# Check if the denominator is not zero to avoid division by zero  
if (probability\_weather\_r\_1 != 0) {  
 # Calculate the conditional probability P(WEATHER\_R = 1, TRAF\_CON\_R = 1 | INJURY="Yes")  
 probability\_condition\_given\_injury\_yes <- sum(subset\_data$WEATHER\_R == desired\_weather & subset\_data$TRAF\_CON\_R == desired\_traffic & subset\_data$INJURY == desired\_injury) / sum(subset\_data$INJURY == desired\_injury)  
  
 # Calculate the Naive Bayes conditional probability  
 naive\_bayes\_probability <- (probability\_condition\_given\_injury\_yes \* prior\_probability\_injury\_yes) / probability\_weather\_r\_1  
}  
  
# Print the result  
cat("The Naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1 is:", naive\_bayes\_probability, "\n")

## The Naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1 is: 0

We apply the Bayes theorem to determine the Naive Bayes conditional probability of injury given Rain (WEATHER\_R = 1) and Construction (TRAF\_CON\_R = 1). This probability’s formula is as follows:

P(WEATHER\_R=1, TRAF\_CON\_R=1) = P(INJURY=“Yes” | WEATHER\_R=1, TRAF\_CON\_R=1) \* P(INJURY=“Yes” | TRAF\_CON\_R=1) \* P(TRAF\_CON\_R=1) \* P(WEATHER\_R=1) / (P(INJURY=“Yes”))

In other words computations is provided here:

P(INJURY=“Yes”): There was a 1/4 prior likelihood of harm. P(WEATHER\_R=1): There is an 11/24 chance of rain. P(TRAF\_CON\_R=1): There is a 1/8 chance of construction. P(TRAFF\_CON\_R=1, WEATHER\_R=1) | INJURY=“Yes”): Given an injury, the conditional probability of both construction and rain is 1/24. Putting these values together now:

P(INJURY=“Yes” | WEATHER\_R=1, TRAF\_CON\_R=1) = (1/24) / [(11/24) \* (1/8) \* (1/4)] = 1/11.

So, the Naive Bayes conditional probability of injury given Rain and Construction is 1/11.

**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 required library  
library(naivebayes)

## naivebayes 0.9.7 loaded

# Assuming you have your dataset loaded and preprocessed, and the factors set as described earlier  
  
# Create a vector to store Naive Bayes classifications  
classifications <- rep(NA, dim(exact\_probabilities)[1] \* dim(exact\_probabilities)[2] \* dim(exact\_probabilities)[3])  
  
# Fill the vector with Naive Bayes classifications  
idx <- 1  
for (i in 1:dim(exact\_probabilities)[1]) {  
 for (j in 1:dim(exact\_probabilities)[2]) {  
 for (k in 1:dim(exact\_probabilities)[3]) {  
 classifications[idx] <- classify\_accident(exact\_probabilities[i, j, k])  
 idx <- idx + 1  
 }  
 }  
}  
  
# Compare the two sets of classifications  
comparison <- classifications == classification  
  
# Print the comparison results  
print("Comparison of Naive Bayes and Exact Bayes Classifications:")

## [1] "Comparison of Naive Bayes and Exact Bayes Classifications:"

print(comparison)

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] TRUE TRUE TRUE  
## [2,] FALSE TRUE FALSE  
## [3,] TRUE TRUE TRUE  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] FALSE TRUE TRUE  
## [2,] FALSE TRUE FALSE  
## [3,] FALSE TRUE TRUE

In the output grid, "TRUE" and "FALSE" values are mixed. "TRUE" means the two classification approaches are similar for certain predictor combinations, whereas "FALSE" means results differ.

Compare Naive Bayes with precise Bayes classifications to show where they agree (TRUE) and disagree (FALSE). Data and Naive Bayes model features determine the agreement or disagreement. If most grid cells indicate "TRUE," the Naive Bayes classifier is most likely using the exact Bayes approach for most pairings.

A majority of grid cells showing "FALSE," indicate considerable disparities between the two categorization systems for this batch of data.

The results show how the Naive Bayes classifier performed against the exact Bayes classification technique for the dataset and predictor variables. A high degree of agreement (many "TRUE" values) implies a good Naive Bayes model for this data set. Significant disparities (many "FALSE" results) may indicate places where the Naive Bayes model needs refinement or where precise Bayes approaches are better.

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

# Load your dataset (replace 'data.csv' with your actual file path)  
accidentsFull <- read.csv("C:/Users/shrey/OneDrive/Desktop/accidentsFull.csv")  
  
# Set a random seed for reproducibility  
set.seed(123)  
  
# Create an index vector for the training set (60%)  
train\_index <- sample(1:nrow(accidentsFull), 0.6 \* nrow(accidentsFull))  
  
# Split the dataset into training and validation sets  
train\_data <- accidentsFull[train\_index, ]  
validation\_data <- accidentsFull[-train\_index, ]

# Load the e1071 library for the Naive Bayes classifier  
library(e1071)  
  
  
# Split the data into predictors and response  
predictors <- train\_data[, c("HOUR\_I\_R", "ALCHL\_I", "ALIGN\_I", "STRATUM\_R", "WRK\_ZONE", "WKDY\_I\_R", "INT\_HWY", "LGTCON\_I\_R", "MANCOL\_I\_R", "PED\_ACC\_R", "RELJCT\_I\_R", "REL\_RWY\_R", "PROFIL\_I\_R", "SPD\_LIM", "SUR\_COND", "TRAF\_CON\_R", "TRAF\_WAY", "VEH\_INVL", "WEATHER\_R")]  
response <- train\_data$INJURY\_CRASH  
  
# Train the Naive Bayes classifier  
naive\_bayes\_model <- naiveBayes(predictors, response)  
  
# Make predictions on the training data  
predictions <- predict(naive\_bayes\_model, predictors)  
  
# Create a confusion matrix  
confusion\_matrix <- table(Actual = response, Predicted = predictions)  
print(confusion\_matrix)

## Predicted  
## Actual 0 1  
## 0 11752 942  
## 1 10386 2229

# Split the validation data into predictors and response  
predictors\_valid <- validation\_data[, c("HOUR\_I\_R", "ALCHL\_I", "ALIGN\_I", "STRATUM\_R", "WRK\_ZONE", "WKDY\_I\_R", "INT\_HWY", "LGTCON\_I\_R", "MANCOL\_I\_R", "PED\_ACC\_R", "RELJCT\_I\_R", "REL\_RWY\_R", "PROFIL\_I\_R", "SPD\_LIM", "SUR\_COND", "TRAF\_CON\_R", "TRAF\_WAY", "VEH\_INVL", "WEATHER\_R")]  
response\_valid <- validation\_data$INJURY\_CRASH  
  
# Use the trained Naive Bayes model to make predictions on the validation data  
predictions\_valid <- predict(naive\_bayes\_model, predictors\_valid)  
  
# Calculate the overall error on the validation set  
error\_rate <- 1 - sum(predictions\_valid == response\_valid) / length(response\_valid)  
print(paste("Overall Error Rate on Validation Set:", error\_rate))

The confusion matrix assesses the performance of a Naive Bayes classifier. It has four cells, each representing different outcomes:

True Positives (TP): Instances where the classifier correctly predicts injury crashes (1). In this case, it correctly predicted 11752 injury crashes.

False Positives (FP): Instances where the classifier incorrectly predicts injury crashes when there were none (0). There were 942 such instances where it made a false positive prediction.

False Negatives (FN): Instances where the classifier failed to predict injury crashes when they did occur. In this case, there were 10386 false negatives.

True Negatives (TN): Instances where the classifier correctly predicts no injury crashes (0). It correctly predicted 2229 no injury crashes.

These values help evaluate the classifier's accuracy, precision, recall, and F1-score, providing insights into its effectiveness in predicting injury crashes and no injury crashes.

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

# Calculate the overall error on the validation set  
error\_rate <- 1 - sum(diag(confusion\_matrix)) / sum(confusion\_matrix)  
cat("Overall Error on Validation Set:", error\_rate, "\n")

## Overall Error on Validation Set: 0.4475878

Validation set errors average 44.76%. In the validation set, 44.76% of Naive Bayes classifier predictions were wrong. Lower mistake rates indicate more precision.