FML_Assignment3_ Naive Bayes

Anil Kumar Akula

2023-10-14

#Summary:

1) The function creates a binary dummy variable 'INJURY' which has the value "Yes" if 'MAX_SEV_IR' is either 1 or 2, otherwise it has the value "No".

2) It calculates the proportion of accidents in the dataset that resulted in an injury (INJURY = Yes). This proportion is used as a threshold for making predictions.

3) Based on the calculated percentage, it predicts whether there will be an injury for a newly reported accident with no further information. A higher proportion of injuries indicates a higher likelihood of injury. If the proportion of injuries is greater than 50%, the prediction is "Yes." Otherwise, the prediction is "No," suggesting a lower likelihood of injury.

4) This pivot table illustrates the distribution of the variables "INJURY," "WEATHER_R," and "TRAF_CON_R" for the first 24 records in the dataset. By visualizing the frequency of each combination of these variables, it makes it easier to understand their relationships within this subset of data.

5) We present the classification results based on the exact Bayes probabilities and the naive Bayes classifier for the first 24 records, which reveal whether the injury result is "Yes" or "No." For the validation set, the naive Bayes classifier results in an overall error of 0, which is consistent with the exact Bayes classification. This suggests that the naive Bayes model is a reliable method for predicting injury outcomes.

6) As can be seen from the calculated conditional probabilities, the likelihood of injury (INJURY = Yes) varies significantly depending on the combination of predictors. It is approximately 66.67% likely that someone will be injured when the weather condition (WEATHER_R) is 1 and the traffic control condition (TRAF_CON_R) is 0. If the weather condition is 2 and the traffic control condition is 2, however, the probability of injury decreases significantly to around 11.11 percent. In order to understand the impact of different predictors on the likelihood of accidents resulting in injuries, these probabilities are essential.

7) The conditional probabilities also show substantial variations based on the combination of predictors when it comes to no injury probability (INJURY = No). In the case of 2 weather conditions and 0 traffic control, the probability of no injury is 60%, which indicates a relatively safe scenario. The probability of no injury drops to 0% when the weather condition is 1 and traffic control is 2, highlighting the increased risk of injury under such conditions. These insights are valuable for risk assessment and accident prevention.

8) Based on the calculated Bayes conditional probabilities, the code assigns classifications ("Yes" or "No") to each of the 24 accidents. These classifications are based on the combinations of the predictors, including WEATHER_R and TRAF_CON_R, and the defined probabilities (p1 to p6). According to the calculated probabilities and the specified threshold of 0.5, the code prints the classification results, indicating whether each accident will result in injury ("Yes") or not ("No").

9) The naive Bayes model is constructed using the e1071 library's 'naiveBayes' function. Specifically, it calculates the conditional probability of an injury (INJURY = Yes) given the scenario where WEATHER_R is 1 and TRAF_CON_R is 1. The probability is manually computed using the naive Bayes model and is printed. This probability represents the likelihood of injury in a specific situation defined by the values of WEATHER_R and TRAF_CON_R, offering valuable insights for decision-making.A comparison is also made between the results of the exact Bayes method and the Naive Bayes method as a result of the code. As a result of using the same data, a Naive Bayes model is trained, and all 24 records are classified based on a cutoff of 0. The code then compares the classifications with those generated by the exact Bayes method. It also determines whether the two methods have equivalent rankings (orders) of observations. In this comparison, the Naive Bayes model is evaluated in terms of its accuracy and performance in predicting injuries and its consistency with the exact Bayes approach.

10) A two-step analysis is performed and it partitions the dataset into training and validation sets, with 60% being allocated to training and 40% for validation. The second step uses categorical predictors from the dataset to use a Naive Bayes classifier to predict injuries (the "INJURY" response variable). The code calculates a confusion matrix to assess the classifier's predictive performance on the validation data and computes the overall error rate, a key metric to evaluate how accurately the model classifies accidents as either resulting in injuries (Yes) or not (No). A model's overall error rate indicates its predictive accuracy on unseen data, and a lower error rate indicates improved model performance.

#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?

##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(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
```

```
accidents <- read.csv("C:\\Users\\anila\\Desktop\\accidentsFull.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
```

##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?

#Reason for Yes:
A dataset of automobile accidents is analyzed to predict whether a newly reported accident will result in an injury (INJURY = Yes) or not (INJURY = No).

#The code accomplishes the following: * The function creates a binary dummy variable 'INJURY' which has the value "Yes" if 'MAX_SEV_IR' is either 1 or 2, otherwise it has the value "No". * It calculates the proportion of accidents in the dataset that resulted in an injury (INJURY = Yes). This proportion is used as a threshold for making predictions. * Based on the calculated percentage, it predicts whether there will be an injury for a newly reported accident with no further information. A higher proportion of injuries indicates a higher likelihood of injury. If the proportion of injuries is greater than 50%, the prediction is "Yes." Otherwise, the prediction is "No," suggesting a lower likelihood of injury.

```r
# Create a dummy variable for injury
accidents$INJURY <- ifelse(accidents$MAX_SEV_IR %in% c("1", "2"), "Yes", "No")

# Compute the proportion of accidents that resulted in an injury
```

```r
proportion_injury <- mean(accidents$INJURY == "Yes", na.rm = TRUE)

# Prediction for a newly reported accident with no further information
prediction <- ifelse(proportion_injury > 0.5, "Yes", "No")

# Print the prediction
print(prediction)
```

```
## [1] "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.

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

```r
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
```

```r
dt2
```

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

#Create a pivot table that examines INJURY as a function of the two predictors WEATHER_R and TRAF_CON_R for the first 24 records.

```r
# Select the first 24 records and relevant columns
subset_data <- accidents[1:24, c("INJURY", "WEATHER_R", "TRAF_CON_R")]

# Create a pivot table examining INJURY as a function of the two predictors
pivot_table <- table(subset_data$INJURY, subset_data$WEATHER_R, subset_data$TRAF_CON_R)
print(pivot_table)
```

```
## , ,  = 0
##
##
##        1 2
##   No   3 9
##   Yes 6 2
##
## , ,  = 1
##
##
##        1 2
##   No   1 1
##   Yes 0 0
##
## , ,  = 2
##
##
##        1 2
##   No   1 0
##   Yes 0 1
```

#2(1).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
```

#Second Apporach

```r
# Injury = Yes
p1 = pivot_table["Yes", "1", "0"] / sum(pivot_table["Yes", , ])
p2 = pivot_table["Yes", "2", "0"] / sum(pivot_table["Yes", , ])
p3 = pivot_table["Yes", "1", "1"] / sum(pivot_table["Yes", , ])
p4 = pivot_table["Yes", "2", "1"] / sum(pivot_table["Yes", , ])
p5 = pivot_table["Yes", "1", "2"] / sum(pivot_table["Yes", , ])
p6 = pivot_table["Yes", "2", "2"] / sum(pivot_table["Yes", , ])

# Injury = No
n1 = pivot_table["No", "1", "0"] / sum(pivot_table["No", , ])
n2 = pivot_table["No", "2", "0"] / sum(pivot_table["No", , ])
n3 = pivot_table["No", "1", "1"] / sum(pivot_table["No", , ])
n4 = pivot_table["No", "2", "1"] / sum(pivot_table["No", , ])
n5 = pivot_table["No", "1", "2"] / sum(pivot_table["No", , ])
n6 = pivot_table["No", "2", "2"] / sum(pivot_table["No", , ])

# Print the conditional probabilities
cat("Conditional Probabilities given INJURY = Yes:\n")
```

```
## Conditional Probabilities given INJURY = Yes:
```

```r
cat(p1, " ", p2, " ", p3, " ", p4, " ", p5, " ", p6, "\n")
```

```
## 0.6666667   0.2222222   0   0   0   0.1111111
```

```r
cat("Conditional Probabilities given INJURY = No:\n")
```

```
## Conditional Probabilities given INJURY = No:
```

```r
cat(n1, " ", n2, " ", n3, " ", n4, " ", n5, " ", n6, "\n")
```

```
## 0.2   0.6   0.06666667   0.06666667   0.06666667   0
```

#2(2). 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")
```

```
# Define a vector to store the classification results
classification_results <- character(24)

# Assign classifications based on the probabilities and a cutoff of 0.5
for (i in 1:24) {
    if (subset_data$WEATHER_R[i] == "1") {
        if (subset_data$TRAF_CON_R[i] == "0") {
            classification_results[i] = ifelse(p1 > 0.5, "Yes", "No")
        } else if (subset_data$TRAF_CON_R[i] == "1") {
            classification_results[i] = ifelse(p3 > 0.5, "Yes", "No")
        } else {
            classification_results[i] = ifelse(p5 > 0.5, "Yes", "No")
        }
    } else {
        if (subset_data$TRAF_CON_R[i] == "0") {
            classification_results[i] = ifelse(p2 > 0.5, "Yes", "No")
        } else if (subset_data$TRAF_CON_R[i] == "1") {
            classification_results[i] = ifelse(p4 > 0.5, "Yes", "No")
        } else {
            classification_results[i] = ifelse(p6 > 0.5, "Yes", "No")
        }
    }
}

# Print the classification results
cat("Classification Results based on Exact Bayes:\n")
```

```
## Classification Results based on Exact Bayes:
```

```
cat(classification_results, sep = " ")
```

```
## Yes No No No Yes No No Yes No No No No Yes Yes Yes Yes No No No No Yes Yes No No
```

#2(3). Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1

```r
# You should load the 'e1071' library to use naiveBayes
library(e1071)

# Create a naive Bayes model
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = subset_data)

# Specify the data for which we want to compute the probability
new_data <- data.frame(WEATHER_R = "1", TRAF_CON_R = "1")

# Predict the probability of "Yes" class
naive_bayes_prob <- predict(nb_model, newdata = new_data, type = "raw")
injury_prob_naive_bayes <- naive_bayes_prob[1, "Yes"]

# Print the probability
cat("Naive Bayes Conditional Probability for WEATHER_R = 1 and TRAF_CON_R = 1:\n")
```

```
## Naive Bayes Conditional Probability for WEATHER_R = 1 and TRAF_CON_R = 1:
```

```r
cat(injury_prob_naive_bayes, "\n")
```

```
## 0.008919722
```

#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?

```r
# Load the e1071 library for naiveBayes
library(e1071)

# Create a naive Bayes model for the 24 records and two predictors
nb_model_24 <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = subset_data)

# Predict using the naive Bayes model with the same data
naive_bayes_predictions_24 <- predict(nb_model_24, subset_data)

# Extract the probability of "Yes" class for each record
injury_prob_naive_bayes_24 <- attr(naive_bayes_predictions_24, "probabilities")[, "Yes"]

# Create a vector of classifications based on a cutoff of 0.5
classification_results_naive_bayes_24 <- ifelse(injury_prob_naive_bayes_24 > 0.5, "Yes", "No")

# Print the classification results
cat("Classification Results based on Naive Bayes for 24 records:\n")
```

```
## Classification Results based on Naive Bayes for 24 records:
```

```r
cat(classification_results_naive_bayes_24, sep = " ")

# Check if the resulting classifications are equivalent to the exact Bayes classification
```

```r
equivalent_classifications <- classification_results_naive_bayes_24 == classification_results

# Check if the ranking (= ordering) of observations is equivalent
equivalent_ranking <- all.equal(injury_prob_naive_bayes_24, as.numeric(pivot_table["Yes", , ]))

# Print the results of the comparison
cat("\nAre the resulting classifications equivalent? ", all(equivalent_classifications))
```

```
##
## Are the resulting classifications equivalent?  TRUE
```

```r
cat("\nIs the ranking (= ordering) of observations equivalent? ", equivalent_ranking)
```

```
##
## Is the ranking (= ordering) of observations equivalent?  target is NULL, current is numeric
```

#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

```r
# Load required libraries
library(e1071)
library(caret)

# Read the dataset
accidents <- read.csv("C:\\Users\\anila\\Desktop\\accidentsFull.csv")

# Create a dummy variable for injury
accidents$INJURY <- ifelse(accidents$MAX_SEV_IR > 0, "Yes", "No")

# Convert variables to factor
for (i in 1:ncol(accidents)) {
  accidents[[i]] <- as.factor(accidents[[i]])
}

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

# Split the data into training (60%) and validation (40%) sets
split_index <- createDataPartition(accidents$INJURY, p = 0.6, list = FALSE)
training_data <- accidents[split_index, ]
validation_data <- accidents[-split_index, ]

# Create a naive Bayes model on the training data
nb_model <- naiveBayes(INJURY ~ ., data = training_data)

# Predict on the validation set
nb_predictions <- predict(nb_model, validation_data)
```

```
# Create a confusion matrix
confusion_matrix <- table(Actual = validation_data$INJURY, Predicted = nb_predictions)

# Print the confusion matrix
print(confusion_matrix)
```

```
##       Predicted
## Actual   No  Yes
##    No  8288    0
##    Yes    0 8584
```

#3(2)What is the overall error of the validation set?

```
# Calculate the overall error
error_rate <- 1 - sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Overall error of the validation set:", error_rate, "\n")
```

```
## Overall error of the validation set: 0
```