FML_Assignment3

Dilip Kumar

2023-10-16

#Summary:

1) If "MAX_SEV_IR" is either 1 or 2, the code creates a binary dummy variable called "INJURY" with the value "Yes," else it has the value "No."

2) The percentage of accidents in the dataset that caused an injury is calculated (INJURY = Yes). Predictions are made using this percentage as a threshold.

3) The determined percentage is used to determine if there will be injuries in a recently reported accident for which there is no additional information. A higher percentage of injuries suggests a higher risk of injury. If the percentage of injuries is more than 50%, the forecast is "Yes." Otherwise, the prediction is "No," indicating a reduced risk of harm.

4) The distribution of the variables "INJURY," "WEATHER_R," and "TRAF_CON_R" for the dataset's first 24 entries is shown in this pivot table. It is simpler to comprehend the links between these variables within this sample of data by viewing the frequency of each combination of these variables.

5) For the first 24 records, we provide the classification outcomes using the precise Bayes probability and the naive Bayes classifier, which show whether the injury result is "Yes" or "No." The naive Bayes classifier produces an overall error for the validation set of 0, which is compatible with the precise Bayes classification. This implies that using the naive Bayes model to predict injury outcomes is a trustworthy technique.

6) The estimated conditional probabilities show that, depending on the combination of predictors, the likelihood of damage (damage = Yes) fluctuates dramatically. When the weather condition (WEATHER_R) is 1 and the traffic control condition (TRAF_CON_R) is 0, there is a 66.67% chance that someone will get hurt. The likelihood of an injury, however, drops to just 11.11 percent if both the weather and traffic control conditions are 2. These probabilities are crucial for comprehending how various factors affect the possibility that accidents would result in injury.

7) When it comes to the chance of no injury (INJURY = No), the conditional probabilities similarly exhibit significant differences based on the mix of variables. The probability of no injuries is 60% in the case of 2 weather conditions and 0 traffic control, indicating a relatively safe situation. When the weather is condition 1 and the traffic control is condition 2, the likelihood of no harm reduces to 0%, showing the higher danger of injury in such circumstances. These perceptions are helpful for risk analysis and accident avoidance.

8) The code classifies each of the 24 accidents as "Yes" or "No" based on the obtained Bayes conditional probabilities. These categorizations are based on the combinations of the defined probabilities (p1 to p6) and the predictors, such as WEATHER_R and TRAF_CON_R. The code prints the classification results, showing whether each accident will cause injury ("Yes") or not ("No"), in accordance with the calculated probability and the chosen threshold of 0.5.

9) The 'naiveBayes' function of the e1071 library is used to build the naive Bayes model. In particular, it determines the conditional probability of an injury (INJURY = Yes) under the assumption that WEATHER_R and TRAF_CON_R are both 1. The naive Bayes model is used to manually calculate the probability, which is then reported. This probability offers useful information for decision-making since it shows the possibility of harm in a particular circumstance determined by the values of WEATHER_R and TRAF_CON_R.As a result of the code, a comparison is also done between the outcomes of the precise Bayes technique and the Naive Bayes approach. Consequently, a Naive Bayes model is trained using the identical data, and all 24 records are categorised using a threshold of 0.

10) The dataset is split into training and validation sets in a two-step analysis, with 60% of the dataset going to training and 40% to validation. The "INJURY" response variable is predicted using a Naive Bayes classifier in the second phase using categorical predictors from the dataset. The code computes an overall error rate, a crucial metric to examine how effectively the model classifies events as either resulting in injuries (Yes) or not (No), and computes a confusion matrix to analyze the classifier's predictive accuracy on the validation data. A model's overall error rate shows how accurate it is at predicting unknown variables, and a lower error rate means the model is performing better.

#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/user/Downloads/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:
In order to determine if a newly reported collision would cause an injury (INJURY = Yes) or not (INJURY = No), a dataset of automotive accidents is analyzed.

The following is what the code does: * It generates a binary fake variable called "INJURY" whose value is "Yes" if "MAX_SEV_IR" is either 1 or 2, and "No" otherwise. * It determines the percentage of collisions in the dataset that caused injuries (INJURY = Yes). As a threshold, this percentage is used to make forecasts.

- Based on the computed percentage, it forecasts if there will be an injury for a recently reported accident with no additional details. Injury probability is higher when there are more injuries overall. The forecast is "Yes" if the percentage of injuries is more than 50%. Otherwise, the forecast is "No," indicating a decreased chance of harm.

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

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

# Printing 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

```
# 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:
```

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

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

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

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

```
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.

```
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

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

```r
# 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
equivalent_classifications <- classification_results_naive_bayes_24 == classification_results
```

```r
# 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/user/Downloads/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)
```

```r
# 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?

```r
# 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
```