FML\_Assignment3

Bhargav

2023-10-15

# Summary

# Questions - Answers

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?

A. From the above data I can say that the accidents that were Injured was 21462 and the accidents that were not injured was 20721. using the above information, if an accident has just been reported and no further information is available, I can say or predict that the accident Reported was “INJURED” that means INJURY = YES as the accidents that were injured was high compared to that of not injured.

1. 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.

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

A. The exact Bayes conditional probabilities of an injury (INJURY = Yes) with six possible combinations of the predictors are

Probability injury=yes when weather=1, traffic=0 is 0.6666667

Probability injury=yes when weather=2, traffic=0 is 0.1818182

Probability injury=yes when weather=1, traffic=1 is 0

Probability injury=yes when weather=2, traffic=1 is 0

Probability injury=yes when weather=1, traffic=2 is 0

Probability injury=yes when weather=2, traffic=2 is 1

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

A. Out of 24 rows there are 5 rows that the actual values of Injury from the dataset that does not matches with the predicted values.

Depending on Actual data provided in dataset Accidents that were Injured (INJURED=Yes): 9 Accidents that were not Injured (INJURED=No): 15

Depending on Predicted values: Accidents that were Injured (INJURED=Yes): 10 Accidents that were not Injured (INJURED=No): 14

row5: the actual Injury was No, but the predicted was Yes row6: the actual Injury was Yes, but the predicted was No

row14: the actual Injury was No, but the predicted was Yes

row21: the actual Injury was No, but the predicted was Yes

row 14: the actual Injury was Yes, but the predicted was No

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

A. The naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1 is ‘0’, As 0 accidents with injuries injuries were reported when the weather = 1 and traffic =1

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?

A. After comparing the two predictions from bayes and naiveBayes theorems and taking the cutoff value 0.4 for the naiveBayes, I can say that both the predictions are nearly equal with only difference in 2 of the values for the 24 rows that we have predicted.

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.

A. After splitting the dataset into training set(60%) and validation set(40%) and training the model using training data set and predicting the validation data, the confusion matrix between predicted data and the data from dataset was:

Reference

Prediction no yes

no 1144 811

yes 7104 7815

True Negative = 1144

False positive = 811

False Negative = 7104

True Positive = 7815

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

A. the overall error of the validation set is 0.4690648, it should be low. but here the overall error was high, that might be due to various factors such as over fitting, sample size, data leakage, Noisy data, Insufficient training data etc. the main reason was over fitting.

## 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.”

# Data Import And Cleaning

Load the Required Libraries

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)

data import which was in .csv format

accidents <- read.csv("C:/Users/BHARGAV/OneDrive/Desktop/FML Assignment/Assignment\_3/accidentsFull.csv")  
dim(accidents)

## [1] 42183 24

Create a Dummy variable “INJURY”

# create a new variable INJURY with "yes" or "no" by using the column MAX\_SEV\_IR  
accidents$INJURY = ifelse(accidents$MAX\_SEV\_IR > 0, "yes", "no")

## 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?

# Accidents that are not Injured  
not\_Injured\_accidents <- sum(accidents$INJURY == "no")  
# Print the data  
cat("No.of Accidents that are not Injured are", not\_Injured\_accidents,"\n")

## No.of Accidents that are not Injured are 20721

# Accidents that are Injured  
Injured\_accidents <- sum(accidents$INJURY == "yes")  
# Print the data  
cat("No.of Accidents that are Injured are", Injured\_accidents,"\n")

## No.of Accidents that are Injured are 21462

From the above data, the accidents that are Injured was 21462 and the accidents that are not injured was 20721. by using the above information, if an accident has just been reported and no further information is available,I can predict that the reported accident is “INJURED” that means INJURY = YES.

Converting the variables to factors

# Converting variables of the dataset to factors  
for (i in c(1:dim(accidents)[2])){  
 accidents[,i] <- as.factor(accidents[,i])  
}

1. 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.
2. Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
3. Classify the 24 accidents using these probabilities and a cutoff of 0.5.
4. Compute manually the naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R =
5. 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?

Create a dataframe by taking the first 24 columns and 3 columns “INJURY”, “WEATHER\_R”, “TRAF\_CON\_R” from actual dataframe(accidents)

# Create a dataframe by taking the first 24 rows  
accidents24 <- accidents[1:24,c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")]  
dim(accidents24)

## [1] 24 3

create a pivot table from the above accidents24

# Create a pivot table using ftable function  
data1 <- ftable(accidents24) #ftable for creating pivot table  
data2 <- ftable(accidents24[,-1]) #pivot table by dropping the first column  
  
# print the table  
data1

## 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

data2

## TRAF\_CON\_R 0 1 2  
## WEATHER\_R   
## 1 9 1 1  
## 2 11 1 1

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

Considering Injury = yes and getting six possible combinations of the predictors.

## By taking the INJURY = YES  
# Probability of INJURY = YES, when WEATHER\_R = 1, TRAF\_CON\_R = 0  
P1 <- data1[3,1] / data2[1,1]   
# Print the data  
cat("Probabilty of injury=yes, when weather=1, traffic=0 is", P1,"\n")

## Probabilty of injury=yes, when weather=1, traffic=0 is 0.6666667

# Probability ofINJURY = YES, when WEATHER\_R = 2, TRAF\_CON\_R = 0  
P2 <- data1[4,1] / data2[2,1]   
# Print the data  
cat("Probabilty of injury=yes when weather=2, traffic=0 is", P2,"\n")

## Probabilty of injury=yes when weather=2, traffic=0 is 0.1818182

# Probability of INJURY = YES, when WEATHER\_R = 1, TRAF\_CON\_R = 1  
P3 <- data1[3,2] / data2[1,2]   
# Print the data  
cat("Probabilty of injury=yes when weather=1, traffic=1 is", P3,"\n")

## Probabilty of injury=yes when weather=1, traffic=1 is 0

# Probability of INJURY = YES, when WEATHER\_R = 2, TRAF\_CON\_R = 1  
P4 <- data1[4,2] / data2[2,2]   
# Print the data  
cat("Probabilty of injury=yes when weather=2, traffic=1 is", P4,"\n")

## Probabilty of injury=yes when weather=2, traffic=1 is 0

# Probability of INJURY = YES, when WEATHER\_R = 1, TRAF\_CON\_R = 2  
P5 <- data1[3,3] / data2[1,3]   
# Print the data  
cat("Probabilty of injury=yes when weather=1, traffic=2 is", P5,"\n")

## Probabilty of injury=yes when weather=1, traffic=2 is 0

# Probability of INJURY = YES, when WEATHER\_R = 2, TRAF\_CON\_R = 2  
P6 <- data1[4,3] / data2[2,3]  
# Print the data  
cat("Probabilty of injury=yes when weather=2, traffic=2 is", P6,"\n")

## Probabilty of injury=yes when weather=2, traffic=2 is 1

# Probabilities of 6 possible combinations when INJURY = YES  
cat("Probabilities of 6 possible comdinations when INJURY = YES", "\n")

## Probabilities of 6 possible comdinations when INJURY = YES

c(P1, P2, P3, P4, P5, P6)

## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000

Considering Injury = no and getting six possible combinations of the predictors.

## By taking INJURY = NO  
# Probability of INJURY = no, when WEATHER\_R = 1, TRAF\_CON\_R = 0  
n1 <- data1[1,1] / data2[1,1]   
# Print the data  
cat("Probabilty of injury=no, when weather=1, traffic=0 is", n1,"\n")

## Probabilty of injury=no, when weather=1, traffic=0 is 0.3333333

# Probability of INJURY = no, when WEATHER\_R = 2, TRAF\_CON\_R = 0  
n2 <- data1[2,1] / data2[2,1]   
# Print the data  
cat("Probabilty of injury=no, when weather=2, traffic=0 is", n2,"\n")

## Probabilty of injury=no, when weather=2, traffic=0 is 0.8181818

# Probability of INJURY = no when WEATHER\_R = 1, TRAF\_CON\_R = 1  
n3 <- data1[1,2] / data2[1,2]   
# Print the data  
cat("Probabilty of injury=no, when weather=1, traffic=1 is", n3,"\n")

## Probabilty of injury=no, when weather=1, traffic=1 is 1

# Probability of INJURY = no, when WEATHER\_R = 2, TRAF\_CON\_R = 1  
n4 <- data1[2,2] / data2[2,2]   
# Print the data  
cat("Probabilty of injury=no, when weather=2, traffic=1 is", n4,"\n")

## Probabilty of injury=no, when weather=2, traffic=1 is 1

# Probability of INJURY = no, when WEATHER\_R = 1, TRAF\_CON\_R = 2  
n5 <- data1[1,3] / data2[1,3]   
# Print the data  
cat("Probabilty of injury=no, when weather=1, traffic=2 is", n5,"\n")

## Probabilty of injury=no, when weather=1, traffic=2 is 1

# Probability of INJURY = no, when WEATHER\_R = 2, TRAF\_CON\_R = 2  
n6 <- data1[2,3] / data2[2,3]   
# Print the data  
cat("Probabilty of injury=no, when weather=2, traffic=2 is", n6,"\n")

## Probabilty of injury=no, when weather=2, traffic=2 is 0

# Probabilities of 6 possible combinations, when INJURY = No  
cat("Probabilities of 6 possible combinations, when INJURY = NO", "\n")

## Probabilities of 6 possible combinations, when INJURY = NO

c(n1, n2, n3, n4, n5, n6)

## [1] 0.3333333 0.8181818 1.0000000 1.0000000 1.0000000 0.0000000

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

Assigning the probabilities to the each of the 24rows.

# Considering the data from 0 to 24  
probability.inj <- rep(0,24)  
  
# use for loop considering iterations from 1 to 24  
for(i in 1:24){  
 # when Weather=1;  
 if (accidents24$WEATHER\_R[i] == "1") {  
 # when Traffic = 0;  
 if (accidents24$TRAF\_CON\_R[i]=="0"){  
 probability.inj[i] = P1  
 }  
 # when Traffic = 1;  
 else if (accidents24$TRAF\_CON\_R[i]=="1") {  
 probability.inj[i] = P3  
 }  
 # when Traffic = 2;  
 else if (accidents24$TRAF\_CON\_R[i]=="2") {  
 probability.inj[i] = P5  
 }  
 }  
 # when Weather=2;  
 else {  
 # when Traffic = 0;  
 if (accidents24$TRAF\_CON\_R[i]=="0"){  
 probability.inj[i] = P2  
 }  
 # when Traffic = 1;  
 else if (accidents24$TRAF\_CON\_R[i]=="1") {  
 probability.inj[i] = P4  
 }  
 # when Traffic = 2;  
 else if (accidents24$TRAF\_CON\_R[i]=="2") {  
 probability.inj[i] = P6  
 }  
 }  
 }  
   
# Inserting the probabilities to the dataframe   
accidents24$probability.inj <- probability.inj  
  
# Classifying the accidents by means of cutoff value 0.5   
accidents24$pred.probability <- ifelse(accidents24$probability.inj>0.5, "yes", "no") #if probability was greater than 0.5 the Injury will be yes  
  
# print the table  
accidents24

## INJURY WEATHER\_R TRAF\_CON\_R probability.inj pred.probability  
## 1 yes 1 0 0.6666667 yes  
## 2 no 2 0 0.1818182 no  
## 3 no 2 1 0.0000000 no  
## 4 no 1 1 0.0000000 no  
## 5 no 1 0 0.6666667 yes  
## 6 yes 2 0 0.1818182 no  
## 7 no 2 0 0.1818182 no  
## 8 yes 1 0 0.6666667 yes  
## 9 no 2 0 0.1818182 no  
## 10 no 2 0 0.1818182 no  
## 11 no 2 0 0.1818182 no  
## 12 no 1 2 0.0000000 no  
## 13 yes 1 0 0.6666667 yes  
## 14 no 1 0 0.6666667 yes  
## 15 yes 1 0 0.6666667 yes  
## 16 yes 1 0 0.6666667 yes  
## 17 no 2 0 0.1818182 no  
## 18 no 2 0 0.1818182 no  
## 19 no 2 0 0.1818182 no  
## 20 no 2 0 0.1818182 no  
## 21 yes 1 0 0.6666667 yes  
## 22 no 1 0 0.6666667 yes  
## 23 yes 2 2 1.0000000 yes  
## 24 yes 2 0 0.1818182 no

Out of 24 rows there are 5 rows that the actual values of Injury from the dataset that does not matches with the predicted values. row5: the actual Injury was No, but the predicted was Yes row6: the actual Injury was Yes, but the predicted was No row14: the actual Injury was No, but the predicted was Yes row21: the actual Injury was No, but the predicted was Yes row 14: the actual Injury was Yes, but the predicted was No

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

# Probability of getting Injured when WEATHER\_R = 1 and TRAF\_CON\_R = 1.  
#P11 = Probability of getting Injured when WEATHER\_R = 1 and TRAF\_CON\_R = 1  
#PIW1 = Probability of getting Injured when WEATHER\_R = 1  
#PIT1 = Probability of getting Injured when TRAF\_CON\_R = 1  
#PI = Probability of getting injured  
#PNW1 = Probability of not getting Injured when WEATHER\_R = 1  
#PNT1 = Probability of not getting Injured when TRAF\_CON\_R = 1  
#PI=N = Probability of not getting injured  
  
# Probability of getting Injured when WEATHER\_R = 1  
PIW1 <- (data1[3,1] + data1[3,2] + data1[3,3]) / (data1[3,1] + data1[3,2] + data1[3,3] + data1[4,1] + data1[4,2] + data1[4,3])  
PIW1

## [1] 0.6666667

# Probability of getting Injured when TRAF\_CON\_R = 1  
PIT1 <- (data1[3,2] + data1[4,2]) / (data1[3,1] + data1[3,2] + data1[3,3] + data1[4,1] + data1[4,2] + data1[4,3])  
PIT1

## [1] 0

# Probability of getting Injured  
PI <- (data1[3,1] + data1[3,2] + data1[3,3] + data1[4,1] + data1[4,2] + data1[4,3])/24  
PI

## [1] 0.375

# Probability of not getting Injured when WEATHER\_R = 1  
PNW1 <- (data1[1,1] + data1[1,2] + data1[1,3]) / (data1[1,1] + data1[1,2] + data1[1,3] + data1[2,1] + data1[2,2] + data1[2,3])  
PNW1

## [1] 0.3333333

# Probability of not getting Injured when TRAF\_CON\_R = 1  
PNT1 <- (data1[1,2] + data1[2,2]) / (data1[1,1] + data1[1,2] + data1[1,3] + data1[2,1] + data1[2,2] + data1[2,3])  
PNT1

## [1] 0.1333333

# Probability of not getting Injured  
PN <- (data1[1,1] + data1[1,2] + data1[1,3] + data1[2,1] + data1[2,2] + data1[2,3])/24  
PN

## [1] 0.625

# Probability of getting Injured when WEATHER\_R = 1 and TRAF\_CON\_R = 1  
P11 <- (PIW1 \* PIT1 \* PI)/ ((PIW1 \* PIT1 \* PI) + (PNW1 \* PNT1 \* PN))  
P11

## [1] 0

cat("The naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1 is", P11)

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

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?

Training and Predicting the data

# training the naiveBayes model by considering the predictors, Traffic and weather  
nb <- naiveBayes(INJURY ~ TRAF\_CON\_R + WEATHER\_R, data = accidents24)  
  
# Predicting the data using naiveBayes model  
nbt <- predict(nb, newdata = accidents24, type = "raw")  
  
# Inserting the newly predicted data to accidents24 dataframe  
accidents24$nbpred.probability <- nbt[,2] # Transfer the "Yes" nb prediction  
  
  
# Consider cutoff value 0.4 for naiveBayes predictions  
accidents24$nbpred.probability.condition <- ifelse(accidents24$nbpred.probability>0.4, "yes", "no") #if probability was greater than 0.4 the Injury will be yes  
accidents24

## INJURY WEATHER\_R TRAF\_CON\_R probability.inj pred.probability  
## 1 yes 1 0 0.6666667 yes  
## 2 no 2 0 0.1818182 no  
## 3 no 2 1 0.0000000 no  
## 4 no 1 1 0.0000000 no  
## 5 no 1 0 0.6666667 yes  
## 6 yes 2 0 0.1818182 no  
## 7 no 2 0 0.1818182 no  
## 8 yes 1 0 0.6666667 yes  
## 9 no 2 0 0.1818182 no  
## 10 no 2 0 0.1818182 no  
## 11 no 2 0 0.1818182 no  
## 12 no 1 2 0.0000000 no  
## 13 yes 1 0 0.6666667 yes  
## 14 no 1 0 0.6666667 yes  
## 15 yes 1 0 0.6666667 yes  
## 16 yes 1 0 0.6666667 yes  
## 17 no 2 0 0.1818182 no  
## 18 no 2 0 0.1818182 no  
## 19 no 2 0 0.1818182 no  
## 20 no 2 0 0.1818182 no  
## 21 yes 1 0 0.6666667 yes  
## 22 no 1 0 0.6666667 yes  
## 23 yes 2 2 1.0000000 yes  
## 24 yes 2 0 0.1818182 no  
## nbpred.probability nbpred.probability.condition  
## 1 0.571428571 yes  
## 2 0.250000000 no  
## 3 0.002244949 no  
## 4 0.008919722 no  
## 5 0.571428571 yes  
## 6 0.250000000 no  
## 7 0.250000000 no  
## 8 0.571428571 yes  
## 9 0.250000000 no  
## 10 0.250000000 no  
## 11 0.250000000 no  
## 12 0.666666667 yes  
## 13 0.571428571 yes  
## 14 0.571428571 yes  
## 15 0.571428571 yes  
## 16 0.571428571 yes  
## 17 0.250000000 no  
## 18 0.250000000 no  
## 19 0.250000000 no  
## 20 0.250000000 no  
## 21 0.571428571 yes  
## 22 0.571428571 yes  
## 23 0.333333333 no  
## 24 0.250000000 no

#Loading the klaR package for Naive Bayes  
library(klaR)

## Loading required package: MASS

# Training the Naive Bayes model with Laplace  
nb2 <- NaiveBayes(INJURY ~ TRAF\_CON\_R + WEATHER\_R, data = accidents24, laplace = 1)  
  
# predicting the data using the model  
predict(nb2, newdata = accidents24[, c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")])

## $class  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## yes no no no yes no no yes no no no yes yes yes yes yes no no no no   
## 21 22 23 24   
## yes yes no no   
## Levels: no yes  
##   
## $posterior  
## no yes  
## 1 0.4285714 0.571428571  
## 2 0.7500000 0.250000000  
## 3 0.9977551 0.002244949  
## 4 0.9910803 0.008919722  
## 5 0.4285714 0.571428571  
## 6 0.7500000 0.250000000  
## 7 0.7500000 0.250000000  
## 8 0.4285714 0.571428571  
## 9 0.7500000 0.250000000  
## 10 0.7500000 0.250000000  
## 11 0.7500000 0.250000000  
## 12 0.3333333 0.666666667  
## 13 0.4285714 0.571428571  
## 14 0.4285714 0.571428571  
## 15 0.4285714 0.571428571  
## 16 0.4285714 0.571428571  
## 17 0.7500000 0.250000000  
## 18 0.7500000 0.250000000  
## 19 0.7500000 0.250000000  
## 20 0.7500000 0.250000000  
## 21 0.4285714 0.571428571  
## 22 0.4285714 0.571428571  
## 23 0.6666667 0.333333333  
## 24 0.7500000 0.250000000

#predicting the data using the model with raw\_probabilities  
predict(nb2, newdata = accidents24[, c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")], type = "raw")

## $class  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## yes no no no yes no no yes no no no yes yes yes yes yes no no no no   
## 21 22 23 24   
## yes yes no no   
## Levels: no yes  
##   
## $posterior  
## no yes  
## 1 0.4285714 0.571428571  
## 2 0.7500000 0.250000000  
## 3 0.9977551 0.002244949  
## 4 0.9910803 0.008919722  
## 5 0.4285714 0.571428571  
## 6 0.7500000 0.250000000  
## 7 0.7500000 0.250000000  
## 8 0.4285714 0.571428571  
## 9 0.7500000 0.250000000  
## 10 0.7500000 0.250000000  
## 11 0.7500000 0.250000000  
## 12 0.3333333 0.666666667  
## 13 0.4285714 0.571428571  
## 14 0.4285714 0.571428571  
## 15 0.4285714 0.571428571  
## 16 0.4285714 0.571428571  
## 17 0.7500000 0.250000000  
## 18 0.7500000 0.250000000  
## 19 0.7500000 0.250000000  
## 20 0.7500000 0.250000000  
## 21 0.4285714 0.571428571  
## 22 0.4285714 0.571428571  
## 23 0.6666667 0.333333333  
## 24 0.7500000 0.250000000

Comparing the naiveBayes model and exactBayes classification

# Compare the naiveBayes model and exactBayes model  
classification\_match <- all(accidents24$nbpred.probability.condition == accidents24$pred.probability)  
classification\_match

## [1] FALSE

probability\_match <- all.equal(accidents24$nbpred.probability.condition, accidents24$pred.probability)  
probability\_match

## [1] "2 string mismatches"

# Checking if classifications and rankings are equivalent  
if (classification\_match && is.na(probability\_match)) {  
 cat("The classifications and rankings are equivalent.\n")  
} else {  
 cat("The classifications and rankings are not equivalent.\n")  
}

## The classifications and rankings are not equivalent.

1. 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.

Splitting the Data into 60% training and 40% validation.

set.seed(159) # Important to ensure that we get the same sample if we rerun the code  
  
train.index <- sample(row.names(accidents), 0.6\*dim(accidents)[1]) # 60% training data  
train.accidents <- accidents[train.index,]  
  
valid.index <- setdiff(row.names(accidents), train.index) # 40% validation data  
valid.accidents <- accidents[valid.index,]  
  
# Print the dimensions of the split datasets  
cat("Training data dimensions:", dim(train.accidents), "\n")

## Training data dimensions: 25309 25

cat("Validation data dimensions:", dim(valid.accidents), "\n")

## Validation data dimensions: 16874 25

Training and Predicting the data by considering the predictors ‘WEATHER\_R’ and ‘TRAF\_CON\_R’ as they are categorial variables

# Training the naiveBayes model  
nb\_data <- naiveBayes(INJURY ~ TRAF\_CON\_R + WEATHER\_R, train.accidents)  
  
# Predict the validation data  
nb\_pred <- predict(nb\_data, valid.accidents)  
  
#confusion matrix for data  
confusion\_matrix <- confusionMatrix(nb\_pred, valid.accidents$INJURY)  
  
#print the matrix  
cat("Confusion Matrix for validation data:", "\n")

## Confusion Matrix for validation data:

print(confusion\_matrix)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1144 811  
## yes 7104 7815  
##   
## Accuracy : 0.5309   
## 95% CI : (0.5234, 0.5385)  
## No Information Rate : 0.5112   
## P-Value [Acc > NIR] : 1.506e-07   
##   
## Kappa : 0.0454   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.1387   
## Specificity : 0.9060   
## Pos Pred Value : 0.5852   
## Neg Pred Value : 0.5238   
## Prevalence : 0.4888   
## Detection Rate : 0.0678   
## Detection Prevalence : 0.1159   
## Balanced Accuracy : 0.5223   
##   
## 'Positive' Class : no   
##

3.2 What is the overall error of the validation set? Overall error of the validation set

# Extracting the values from confusion matrix  
# True Negative  
TN <- confusion\_matrix$table[1,1]  
# False Positive  
FP <- confusion\_matrix$table[1,2]  
# False Negative  
FN <- confusion\_matrix$table[2,1]  
# True Positive  
TP <- confusion\_matrix$table[2,2]  
  
# Calculate the overall error  
overall\_error <- (FN+FP)/ sum(confusion\_matrix$table)  
  
# Print the data  
cat(" the overall rate of the validation set is", overall\_error)

## the overall rate of the validation set is 0.4690648