

# RAJESH FML ASS 3

2023-10-30

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 ( $\text{MAX\_SEV\_IR} = 1$  or  $2$ ) or will not ( $\text{MAX\_SEV\_IR} = 0$ ). For this purpose, create a dummy variable called INJURY that takes the value “yes” if  $\text{MAX\_SEV\_IR} = 1$  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 24 records. Use all three variables in the pivot table as rows/columns.
  - a. Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
  - b. Classify the 24 accidents using these probabilities and a cutoff of 0.5.
  - c. Compute manually the naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1.
  - d. Run a naive Bayes classifier on the 24 records and two predictors. Check the model output to obtain probabilities and classifications for all 24 records. Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?
3. Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%).
  - 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.
  - b. What is the overall error of the validation set?

#loading the required libraries “e1071”, “caret”

```
library(e1071)
library(caret)
```

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

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

#reading the given file and calling first 24 rows

```
accident_data <- read.csv("C:\\\\Users\\\\rajes\\\\OneDrive\\\\Desktop\\\\Business analytics sem1\\\\FML\\\\accidentsFull.csv")
head(accident_data)
```

```
##   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
##   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
##   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
##   FATALITIES MAX_SEV_IR
## 1      0      1
## 2      0      0
## 3      0      0
## 4      0      0
## 5      0      0
## 6      0      1
```

#dimension of the given data which is rows and columns

```
dim(accident_data)
```

```
## [1] 42183    24
```

#creating a dummy variable called INJURy

```
accident_data$INJURY <- ifelse(accident_data$MAX_SEV_IR>0, "yes", "no")
table(accident_data$INJURY)
```

```
##
##      no    yes
## 20721 21462
```

#converting

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

```
##   INJURY WEATHER_R TRAF_CON_R
## 1   yes      1       0
## 2   no       2       0
## 3   no       2       1
## 4   no       1       1
## 5   no       1       0
## 6   yes      2       0
```

#creating a pivot table

```
df1 <- ftable(accidents24)
df2 <- ftable(accidents24[,-1])
#print the tables
df1
```

```
##                                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
```

df2

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

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.

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

```

#if INJURY is yes

X1= df1[3,1]/df2[1,1] #if TRAF_CON_R=0, WEATHER_R=1
X2= df1[3,2]/df2[1,2] #if TRAF_CON_R=1, WEATHER_R=1
X3= df1[3,3]/df2[1,3] #if TRAF_CON_R=2, WEATHER_R=1
X4= df1[4,1]/df2[2,1] #if TRAF_CON_R=0, WEATHER_R=2
X5= df1[4,2]/df2[2,2] #if TRAF_CON_R=1, WEATHER_R=2
X6= df1[4,3]/df2[2,3] #if TRAF_CON_R=2, WEATHER_R=2

#if INJURY is no

Y1= df1[1,1]/df2[1,1] #if TRAD_CON_R=0, WEATHER_R=1
Y2= df1[1,2]/df2[1,2] #if TRAF_CON_R=1, WEATHER_R=1
Y3= df1[1,3]/df2[1,3] #if TRAF_CON_R=2, WEATHER_R=1
Y4= df1[2,1]/df2[2,1] #if TRAF_CON_R=0, WEATHER_R=2
Y5= df1[2,2]/df2[2,2] #if TRAF_CON_R=1, WEATHER_R=2
Y6= df1[2,3]/df2[2,3] #if TRAF_CON_R=2, WEATHER_R=2

print(c(X1,X2,X3,X4,X5,X6))

```

```
## [1] 0.6666667 0.0000000 0.0000000 0.1818182 0.0000000 1.0000000
```

```
print(c(Y1,Y2,Y3,Y4,Y5,Y6))
```

```
## [1] 0.3333333 1.0000000 1.0000000 0.8181818 1.0000000 0.0000000
```

2.

b. Let us compute Classify the 24 accidents using these probabilities and a cutoff of 0.5.

```

prob.inj <- rep(0,24)

for (i in 1:24) {
  if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="0"){
    prob.inj[i]= X1
  }
  else if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="1"){
    prob.inj[i]=X2
  }
  else if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="2"){
    prob.inj[i]=X3
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="0"){
    prob.inj[i]=X4
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="1"){
    prob.inj[i]=X5
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="2"){
    prob.inj[i]=X6
  }
  else if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="0"){
    prob.inj[i]= Y1
  }
  else if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="1"){
    prob.inj[i]=Y2
  }
  else if(accidents24$WEATHER_R[i]=="1" && accidents24$TRAF_CON_R[i]=="2"){
    prob.inj[i]=Y3
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="0"){
    prob.inj[i]=Y4
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="1"){
    prob.inj[i]=Y5
  }
  else if(accidents24$WEATHER_R[i]=="2" && accidents24$TRAF_CON_R[i]=="2"){
    prob.inj[i]=Y6
  }
}
accidents24$prob.inj <- prob.inj
accidents24$predicted.prob <- ifelse(accidents24$prob.inj>0.5, "yes","no")
accidents24$predicted.prob

```

```

## [1] "yes" "no"  "no"  "no"  "yes" "no"  "no"  "yes" "no"  "no"  "no"  "no"
## [13] "yes" "yes" "yes" "yes" "no"  "no"  "no"  "no"  "yes" "yes" "yes" "no"

```

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

```

P_W1_Iy = (df1[3,1]+df1[3,2]+df1[3,3])/(df1[3,1]+df1[3,2]+df1[3,3]+df1[4,1]+df1[4,2]+df1[4,3])
P_T1_Iy = (df1[3,2]+df1[4,2])/(df1[3,1]+df1[3,2]+df1[3,3]+df1[4,1]+df1[4,2]+df1[4,3])
PIy     = (df1[3,1]+df1[3,2]+df1[3,3]+df1[4,1]+df1[4,2]+df1[4,3])/24
P_W1_In = (df1[1,1]+df1[1,2]+df1[1,3])/(df1[1,1]+df1[1,2]+df1[1,3]+df1[2,1]+df1[2,2]+df1[2,3])
P_T1_In = (df1[1,2]+df1[2,2])/(df1[1,1]+df1[1,2]+df1[1,3]+df1[2,1]+df1[2,2]+df1[2,3])
PIN     = (df1[1,1]+df1[1,2]+df1[1,3]+df1[2,1]+df1[2,2]+df1[2,3])/24

P_Iy_W1.T1= (P_W1_Iy*P_T1_Iy*PIy)/((P_W1_Iy*P_T1_Iy*PIy)+(P_W1_In*P_T1_In*PIN))
P_Iy_W1.T1

```

```
## [1] 0
```

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?

```

# 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$nbpred.condition <- ifelse(accidents24$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 prob.inj predicted.prob nbpred.probability
## 1     yes        1          0 0.6666667      yes       0.5262707
## 2      no        2          0 0.1818182      no        0.2207180
## 3      no        2          1 0.0000000      no        0.2271763
## 4      no        1          1 0.0000000      no        0.5355255
## 5      no        1          0 0.6666667      yes       0.5262707
## 6     yes        2          0 0.1818182      no        0.2207180
## 7      no        2          0 0.1818182      no        0.2207180
## 8     yes        1          0 0.6666667      yes       0.5262707
## 9      no        2          0 0.1818182      no        0.2207180
## 10     no        2          0 0.1818182      no        0.2207180
## 11     no        2          0 0.1818182      no        0.2207180
## 12     no        1          2 0.0000000      no        0.6829456
## 13     yes        1          0 0.6666667      yes       0.5262707
## 14     no        1          0 0.6666667      yes       0.5262707
## 15     yes        1          0 0.6666667      yes       0.5262707
## 16     yes        1          0 0.6666667      yes       0.5262707
## 17     no        2          0 0.1818182      no        0.2207180
## 18     no        2          0 0.1818182      no        0.2207180
## 19     no        2          0 0.1818182      no        0.2207180
## 20     no        2          0 0.1818182      no        0.2207180
## 21     yes        1          0 0.6666667      yes       0.5262707
## 22     no        1          0 0.6666667      yes       0.5262707
## 23     yes        2          2 1.0000000      yes       0.3544982
## 24     yes        2          0 0.1818182      no        0.2207180
##      nbpred.probability.condition
## 1                  yes
## 2                  no
## 3                  no
## 4                  yes
## 5                  yes
## 6                  no
## 7                  no
## 8                  yes
## 9                  no
## 10                 no
## 11                 no
## 12                 yes
## 13                 yes
## 14                 yes
## 15                 yes
## 16                 yes
## 17                 no
## 18                 no
## 19                 no
## 20                 no
## 21                 yes
## 22                 yes
## 23                 no
## 24                 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")])
```

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

```
#predicting the data using the model with raw_probabilities
predict(nb2, newdata = accidents24[, c("INJURY", "WEATHER_R", "TRAF_CON_R")], type = "raw")
```

```
##          no      yes
## [1,] 0.4737293 0.5262707
## [2,] 0.7792820 0.2207180
## [3,] 0.7728237 0.2271763
## [4,] 0.4644745 0.5355255
## [5,] 0.4737293 0.5262707
## [6,] 0.7792820 0.2207180
## [7,] 0.7792820 0.2207180
## [8,] 0.4737293 0.5262707
## [9,] 0.7792820 0.2207180
## [10,] 0.7792820 0.2207180
## [11,] 0.7792820 0.2207180
## [12,] 0.3170544 0.6829456
## [13,] 0.4737293 0.5262707
## [14,] 0.4737293 0.5262707
## [15,] 0.4737293 0.5262707
## [16,] 0.4737293 0.5262707
## [17,] 0.7792820 0.2207180
## [18,] 0.7792820 0.2207180
## [19,] 0.7792820 0.2207180
## [20,] 0.7792820 0.2207180
## [21,] 0.4737293 0.5262707
## [22,] 0.4737293 0.5262707
## [23,] 0.6455018 0.3544982
## [24,] 0.7792820 0.2207180
```

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

```
## [1] TRUE
```

```
probability_match <- all.equal(accidents24$nbpred.probability.condition, accidents24$pred.probability)
probability_match
```

```
## [1] "Modes: character, NULL"
## [2] "Lengths: 24, 0"
## [3] "target is character, current is NULL"
```

```
# 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 prob.inj predicted.prob nbpred.probability
## 1     yes        1          0 0.6666667      yes       0.5262707
## 2      no        2          0 0.1818182      no        0.2207180
## 3      no        2          1 0.0000000      no        0.2271763
## 4      no        1          1 0.0000000      no        0.5355255
## 5      no        1          0 0.6666667      yes       0.5262707
## 6     yes        2          0 0.1818182      no        0.2207180
## 7      no        2          0 0.1818182      no        0.2207180
## 8     yes        1          0 0.6666667      yes       0.5262707
## 9      no        2          0 0.1818182      no        0.2207180
## 10     no        2          0 0.1818182      no        0.2207180
## 11     no        2          0 0.1818182      no        0.2207180
## 12     no        1          2 0.0000000      no        0.6829456
## 13     yes        1          0 0.6666667      yes       0.5262707
## 14     no        1          0 0.6666667      yes       0.5262707
## 15     yes        1          0 0.6666667      yes       0.5262707
## 16     yes        1          0 0.6666667      yes       0.5262707
## 17     no        2          0 0.1818182      no        0.2207180
## 18     no        2          0 0.1818182      no        0.2207180
## 19     no        2          0 0.1818182      no        0.2207180
## 20     no        2          0 0.1818182      no        0.2207180
## 21     yes        1          0 0.6666667      yes       0.5262707
## 22     no        1          0 0.6666667      yes       0.5262707
## 23     yes        2          2 1.0000000      yes       0.3544982
## 24     yes        2          0 0.1818182      no        0.2207180
##      nbpred.probability.condition
## 1                  yes
## 2                  no
## 3                  no
## 4                  yes
## 5                  yes
## 6                  no
## 7                  no
## 8                  yes
## 9                  no
## 10                 no
## 11                 no
## 12                 yes
## 13                 yes
## 14                 yes
## 15                 yes
## 16                 yes
## 17                 no
## 18                 no
## 19                 no
## 20                 no
## 21                 yes
## 22                 yes
## 23                 no
## 24                 no

```

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

```
## [1] TRUE
```

```
probability_match <- all.equal(accidents24$nbpred.probability.condition, accidents24$pred.probability)
probability_match
```

```
## [1] "Modes: character, NULL"
## [2] "Lengths: 24, 0"
## [3] "target is character, current is NULL"
```

```
nbc <- naiveBayes(INJURY ~ WEATHER_R+ TRAF_CON_R, data=accidents24)
nbt1 <- predict(nbc, newdata=accidents24, type="raw")
accidents24$nbcpred.prob <- nbt1[,2]
accidents24$nbcpred.prob
```

```
## [1] 0.5262707 0.2207180 0.2271763 0.5355255 0.5262707 0.2207180 0.2207180
## [8] 0.5262707 0.2207180 0.2207180 0.2207180 0.6829456 0.5262707 0.5262707
## [15] 0.5262707 0.5262707 0.2207180 0.2207180 0.2207180 0.2207180 0.5262707
## [22] 0.5262707 0.3544982 0.2207180
```

```
#let us use caret
```

```
n1<- train(INJURY ~ TRAF_CON_R + WEATHER_R,
            data = accidents24, method= "nb")
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
pred.n1 <- predict(n1, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])  
pred1.n1 <- predict(n1, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")], type =  
"raw")  
pred.n1
```

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

pred1.n1

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

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

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.

```
set.seed(1)  
train_data1 <- sample(row.names(accident_data), 0.6 * dim(accident_data)[1])  
valid_data1 <- setdiff(row.names(accident_data), train_data1)  
  
train.df <- accident_data[train_data1,]  
valid.df <- accident_data[valid_data1,]
```

```
nb_c <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = train.df)  
nb_p <- predict(nb_c, newdata = valid.df)  
nb_p
```

```

## [1] yes no no yes yes yes no no yes no no no no no yes no no no
## [19] no no yes no no no no no no no yes no no yes yes yes
## [37] no no no yes yes no yes yes yes no no yes no no no no yes no
## [55] no no yes no yes yes yes no yes yes no no yes yes yes no no no
## [73] no no no no no no yes no yes no no yes no no no no no no no
## [91] no no no no no yes no no no yes no yes no yes no no no no no
## [109] no yes yes yes yes no no no no yes no no no no no yes no
## [127] no yes yes no no no no no no no yes yes yes yes yes no no
## [145] no yes yes yes yes yes yes no no no no no no yes yes yes no
## [163] no no no yes yes yes no no no yes yes yes yes yes yes yes yes
## [181] yes yes
## [199] yes yes
## [217] yes yes
## [235] yes yes
## [253] yes yes
## [271] yes yes
## [289] yes yes
## [307] yes yes
## [325] yes yes
## [343] yes yes
## [361] yes yes
## [379] yes yes
## [397] yes yes
## [415] yes yes yes no yes yes
## [433] yes yes
## [451] yes yes
## [469] yes yes
## [487] yes yes
## [505] yes yes
## [523] yes yes
## [541] yes yes
## [559] yes yes
## [577] yes yes
## [595] yes no no yes yes
## [613] yes yes
## [631] yes yes yes yes no no no yes yes no yes yes no no yes no no
## [649] no no yes no yes no yes no no yes no no yes no yes yes no
## [667] yes no no no yes no no yes yes yes yes no yes no no no yes no
## [685] no no no no yes yes yes no no no no no no no yes no
## [703] no yes yes no no no no yes no no no yes no no no yes no no no
## [721] yes no no no no no no no no yes yes no no yes no yes
## [739] no no no yes no no yes yes no no no no no no no no yes
## [757] no no yes no no no no no no no no yes no no no yes
## [775] yes yes yes no no no no yes no no no no no no no no yes no
## [793] no no no yes no no yes yes yes no no no no yes yes yes no no
## [811] no no no no yes no no yes no no
## [829] yes yes no no no yes yes no no
## [847] no no yes no yes no no yes no yes no no no no no no yes yes
## [865] yes no yes no yes no no no yes no yes no no no no no no no no
## [883] no no no no no no yes yes yes no yes no no no no no no no no
## [901] no no yes no yes yes yes no no
## [919] no yes yes no no

```





```
## [2809] yes  
## [2827] yes  
## [2845] yes  
## [2863] yes  
## [2881] yes  
## [2899] yes  
## [2917] yes  
## [2935] yes  
## [2953] yes  
## [2971] yes  
## [2989] yes  
## [3007] yes  
## [3025] yes  
## [3043] yes  
## [3061] yes  
## [3079] yes  
## [3097] yes  
## [3115] yes yes yes no no yes no no no no no no no no yes yes yes  
## [3133] yes no no no no yes no yes no no yes yes yes yes yes yes  
## [3151] yes yes no no no no no no no no yes no no no yes no no yes no  
## [3169] no no no yes yes yes yes yes no  
## [3187] no no yes no no no yes  
## [3205] yes no  
## [3223] no yes yes no no no no no no no yes no no no no no no no yes  
## [3241] no no no yes no no no no no no yes no no no yes no yes no yes  
## [3259] yes yes no no no no no no no yes yes yes no yes yes no yes yes no  
## [3277] yes yes no no yes no no no yes no no no yes no no no no no no  
## [3295] no no no no yes yes yes yes no  
## [3313] no no no no no no no yes yes yes yes no yes yes no no no no no  
## [3331] no no yes yes yes no no no yes yes yes yes yes yes yes yes yes  
## [3349] yes no no  
## [3367] no  
## [3385] no yes yes no yes yes no yes  
## [3403] yes no yes yes no yes no no no yes yes no no no no no no no no  
## [3421] no yes no no no yes no no no no no no no no yes no yes no yes  
## [3439] yes no yes yes no no no no no no no no no yes no yes no no yes  
## [3457] no no no no yes yes yes no no no no no no yes yes yes yes yes  
## [3475] yes yes yes yes yes yes yes yes no  
## [3493] no yes yes yes yes yes  
## [3511] yes no no yes yes no no yes no no no no yes no no no yes no  
## [3529] no yes yes no no yes yes yes yes no no no yes yes yes no no no  
## [3547] no no yes yes yes yes yes yes no no no yes yes yes yes yes yes  
## [3565] yes  
## [3583] yes no no no no no no no no  
## [3601] no  
## [3619] no  
## [3637] no  
## [3655] no no no no yes  
## [3673] yes  
## [3691] yes  
## [3709] yes  
## [3727] yes yes
```



```
## [4681] yes  
## [4699] yes  
## [4717] yes  
## [4735] yes no yes  
## [4753] yes  
## [4771] yes  
## [4789] yes  
## [4807] yes  
## [4825] yes  
## [4843] yes  
## [4861] yes  
## [4879] yes  
## [4897] yes  
## [4915] yes  
## [4933] yes  
## [4951] yes  
## [4969] yes  
## [4987] yes  
## [5005] yes  
## [5023] yes  
## [5041] yes  
## [5059] yes  
## [5077] yes  
## [5095] yes  
## [5113] yes  
## [5131] yes no no no no  
## [5149] no no yes yes yes no yes yes yes no yes no no yes no no no no  
## [5167] no yes no yes no yes no yes yes no no no yes yes no no no yes  
## [5185] yes no no no no no no no  
## [5203] no no no no no yes no no yes yes no yes no no no yes no no no yes  
## [5221] yes yes yes no yes no no yes no no no no yes yes no no yes no  
## [5239] no no yes no no no yes yes yes yes no no no no no no no no no  
## [5257] no no no no yes no yes no no no yes yes no no no no yes yes yes  
## [5275] no yes no no no no no no  
## [5293] no no yes yes yes yes yes yes no yes yes yes yes yes yes yes yes  
## [5311] no no no no no no no no yes yes no no no no no no no yes  
## [5329] no no no yes no no no no no yes yes no no no no no no no no  
## [5347] no no no no no no no no no yes no no no no no no no no no  
## [5365] no no yes no no no no yes no no no no no no no no yes no  
## [5383] no no no yes no no no no no no no no yes yes no yes no  
## [5401] yes no no no no no no yes no no yes no yes no yes no no no no  
## [5419] no no no no no yes no yes yes yes no no no no no no no no no  
## [5437] no yes no no  
## [5455] no no no no yes no yes no no  
## [5473] no no no yes no no no yes yes no  
## [5491] no no no no yes yes yes yes no no no no no no yes yes yes no  
## [5509] no yes  
## [5527] yes yes no no no no yes yes no yes  
## [5545] no no yes no yes yes no no  
## [5563] no no no no no no yes no yes  
## [5581] yes yes yes no no no no no yes yes yes no no no no no no no  
## [5599] no no no no no no yes yes no no no no no no no no no no
```

```
## [5617] no no no yes yes no no no no no no no no no yes yes yes no
## [5635] no yes yes yes yes
## [5653] yes yes no no
## [5671] no yes yes yes no no no no no yes
## [5689] yes no yes yes no no no yes no yes yes no no no no yes yes no
## [5707] no no no yes no no no no yes no yes no yes no yes yes yes no
## [5725] no no no no no no no no yes yes no no no no no no no no no
## [5743] no no no yes yes yes no no no yes no no yes no no no no yes
## [5761] no yes no no no no no
## [5779] no no
## [5797] no no no no no no yes yes yes yes yes yes yes no no no no yes yes yes
## [5815] yes no no no no no no
## [5833] no no
## [5851] yes yes
## [5869] yes no no no no no no
## [5887] no no
## [5905] no no
## [5923] no no
## [5941] no no no yes no yes no yes no no yes yes no yes yes yes no no
## [5959] no yes no no no yes no no no no yes yes yes yes yes no no no yes
## [5977] yes no no no no no yes yes no no no yes yes yes yes yes yes no no
## [5995] no no yes no no no no yes no yes yes yes yes yes no no no no
## [6013] no no no no no no no yes yes no no no no no no no no no no
## [6031] no no no yes yes yes yes no no no no no yes yes no yes no yes no
## [6049] no no no yes yes yes no no no no no no no yes yes yes no yes no
## [6067] no no no no no no no yes no no
## [6085] no no no yes yes yes yes yes no no no no no no yes yes yes yes yes
## [6103] yes yes yes no no
## [6121] no no yes no no no no no no no
## [6139] no no
## [6157] no no no yes yes yes yes yes no no
## [6175] no no no no yes yes yes yes yes yes yes no no no no no no no no
## [6193] no yes no no no no no no no
## [6211] no no
## [6229] no no yes yes
## [6247] yes no no
## [6265] no yes yes
## [6283] yes yes yes yes yes yes no no
## [6301] no no no no no no yes yes yes yes yes yes yes no no no no
## [6319] no no yes yes
## [6337] no no
## [6355] no no no no no no yes no yes yes yes yes yes yes yes yes yes yes
## [6373] yes yes yes yes no no
## [6391] no no
## [6409] no no no no no no no no yes yes yes yes yes yes yes yes yes yes
## [6427] yes yes yes yes no no
## [6445] no no
## [6463] no no
## [6481] no no no no no yes yes yes yes yes yes no no yes yes yes yes
## [6499] yes yes yes yes yes no no
## [6517] no yes no yes yes yes yes no no
## [6535] no no no no yes yes yes yes no no
```























```
## [16849] yes  
## [16867] yes no yes yes yes yes yes yes  
## Levels: no yes
```

```
valid.df$INJURY <- as.factor(valid.df$INJURY)  
con_mat <- confusionMatrix(nb_p,valid.df$INJURY)  
con_mat
```

```
## Confusion Matrix and Statistics  
##  
##          Reference  
## Prediction   no  yes  
##          no 1285 1118  
##          yes 6934 7537  
##  
##          Accuracy : 0.5228  
##                  95% CI : (0.5152, 0.5304)  
##      No Information Rate : 0.5129  
##      P-Value [Acc > NIR] : 0.005162  
##  
##          Kappa : 0.0277  
##  
##  Mcnemar's Test P-Value : < 2.2e-16  
##  
##          Sensitivity : 0.15635  
##          Specificity : 0.87083  
##          Pos Pred Value : 0.53475  
##          Neg Pred Value : 0.52083  
##          Prevalence : 0.48708  
##          Detection Rate : 0.07615  
##          Detection Prevalence : 0.14241  
##          Balanced Accuracy : 0.51359  
##  
##          'Positive' Class : no  
##
```

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

```
Overall_error <- (con_mat$table[1,2]+con_mat$table[2,1])/sum(con_mat$table)  
Overall_error
```

```
## [1] 0.4771838
```

#summary

Accuracy:0.5228. This indicates that the model correctly predicted the class labels for approximately 52.28% of the instances.

95% Confidence Interval (CI): (0.5152, 0.5304). This is the range in which the true accuracy of the model is likely to lie with 95% confidence.

3. No Information Rate (NIR): 0.5129. NIR is the accuracy that could be achieved by always predicting the majority class.
4. P-Value [Acc > NIR]: 0.005162. This p-value tests whether the accuracy is significantly different from the No Information Rate. In this case, the p-value is less than the typical alpha level of 0.05, suggesting that the difference in accuracy is statistically significant.
5. Kappa: 0.0277. Kappa statistic measures the agreement between the observed accuracy and the expected accuracy (chance agreement). A kappa of 0.0277 indicates low agreement beyond chance.
6. McNemar's Test P-Value: < 2.2e-16. McNemar's test is used to compare the performance of two classifiers on a binary classification problem. In this case, the p-value is extremely low, indicating a significant difference between the models being compared.
7. Sensitivity: 0.15635. Sensitivity, also known as True Positive Rate or Recall, measures the proportion of actual positives that are correctly identified by the model.
8. Specificity: 0.87083. Specificity measures the proportion of actual negatives that are correctly identified by the model.
9. Positive Predictive Value (Pos Pred Value): 0.53475. Positive Predictive Value is the proportion of instances predicted as positive by the model that are actually positive.
10. Negative Predictive Value (Neg Pred Value): 0.52083. Negative Predictive Value is the proportion of instances predicted as negative by the model that are actually negative.
11. Prevalence: 0.48708. Prevalence is the proportion of actual positive cases in the dataset.
12. Detection Rate: 0.07615. Detection Rate is the proportion of actual positive cases that are correctly identified by the model.
13. Detection Prevalence: 0.14241. Detection Prevalence is the proportion of predicted positive cases by the model.
14. **Balanced Accuracy:** 0.51359. Balanced Accuracy is the arithmetic mean of sensitivity and specificity, and it is a better metric when dealing with imbalanced classes.
15. 'Positive' Class: The positive class in this context is labeled as 'no'.

In this, the model has a relatively low accuracy and sensitivity, indicating that it struggles to correctly identify positive cases. The specificity and positive predictive value are higher, suggesting that when the model predicts a positive case, it is often correct. However, the overall performance of the model seems to be subpar, as indicated by the low accuracy and kappa statistic.