FML ASSIGNMENT 3

Yaswanth Golla

2023-10-15

Summary

When an accident is only reported and no other information is supplied, it is assumed that there may be injuries (INJURY = Yes). This assumption is made in order to accurately depict the accident's maximum level of injury, MAX_SEV_IR. If MAX_SEV_IR is 1 or 2, there has been some form of injury (INJURY = Yes), according to the guidelines. If MAX_SEV_IR equals 0, it indicates that there is no inferred injury (INJURY = No). As a result, where there is a scarcity of more information on the accident, it is reasonable to assume that some degree of injury has occurred until fresh information indicates otherwise.

- There are a total of "20721 NO and yes are 21462".

The steps that followed were carried out in order to create The following steps were taken to create a new data frame with 24 records and just three variables (Injury, Weather, and Traffic):

With the variables traffic, weather, and injury, a pivot table was created. The data had to be formatted in a tabular manner with these exact columns in this step.

- The variable Injury was removed from the data frame since it will not be used in the subsequent analysis.

Bayes probabilities were computed for each of the first 24 elements in the data frame to evaluate the possibility of an injury occurring.

Accidents that were categorised using a 0.5 cutoff. - Each accident was evaluated as likely or not likely to result in death based on a 0.5 cutoff threshold.

-The chance of an injury is zero.

If there is no harm, the probability is 1.

The predictions of the Naive Bayes model and the precise Bayes classification provide the following results:

[1] yes no no yes yes no no yes no no no yes yes yes yes yes no no no no [21] yes, yes, no, no, no, no, no, no, no, Levels: no and yes

[1] yes no no yes yes no no yes no no no yes yes yes yes yes no no no no [21] yes, yes, no, no, no, no, no, no, no, Levels: no and yes

In this case, the records are categorised as "yes" or "no." The most significant discovery is that both groups have the same values in some places, indicating that the two classifications agree on the

In this case, the records are categorised as "yes" or "no." The most noteworthy discovery is that both categories have the same values in certain places, indicating that the two classifications agree on the order or ordering of the data. This means that both classes place the same importance on the components and have a comparable understanding of the facts.

In the following stage, the entire dataset is divided into two sets: a training set (which will comprise 60% of the data) and a validation set (40% of the data). To do this, the model will be trained using the training data after the dataset has been divided. The entire dataset will be used to assess the model's performance and ability to anticipate future events. After segmenting the data frame, the following step is to normalize the data. This normalization process allows for more accurate decision-making by ensuring that each segment is represented as a single row. To ensure the validity of comparisons, the traits being researched must have stable levels and be either numeric or integer values. This consistency in attribute levels and data types assists in the prevention of analytical mistakes and ensures that data operations deliver correct and significant results for use in decision-making.

Furthermore, you said that the overall error rate for the validation set is roughly 0.47 when presented in decimal form. This indicates that the Naive Bayes classifier performs admirably and accurately on this dataset.

Here are the issues: Accuracy: Your model's accuracy is 0.5, suggesting that 50% of the predictions are right.

- Sensitivity is 0.15635, also known as true positive rate or recall. This indicates that 15.635% of the time, your model successfully finds positive situations (e.g., injuries).
- Specificity: Your model's specificity is 0.8708, which means it accurately detects negative instances (e.g., no injuries) 87.08% of the time.

Overall, your model appears to work well, albeit it may not be very excellent at properly projecting injuries, especially when the injuries are positive. The Naive Bayes technique is effective, although it oversimplifies the occasionally incorrect assumption of variable independence. Consider the specific results and their implications in light of your own dataset and aims.

#calling the libraries

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

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(class)
```

#reading of the dataset

```r
accidentsfull <- read.csv("~/Documents/FML/FML ASSIGNMENT 3/accidentsFull (1).csv")

head(accidentsfull)
```

```
##   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
```

```
str(accidentsfull)
```

```
## ’data.frame’:    42183 obs. of  24 variables:
##  $ HOUR_I_R      : int  0 1 1 1 1 1 1 1 1 0 ...
##  $ ALCHL_I       : int  2 2 2 2 1 2 2 2 2 2 ...
##  $ ALIGN_I       : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ STRATUM_R     : int  1 0 0 1 0 1 0 1 1 0 ...
##  $ WRK_ZONE      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WKDY_I_R      : int  1 1 1 0 1 1 1 1 1 0 ...
##  $ INT_HWY       : int  0 1 0 0 0 0 1 0 0 0 ...
##  $ LGTCON_I_R    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ MANCOL_I_R    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ PED_ACC_R     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RELJCT_I_R    : int  1 1 1 1 0 1 0 0 1 1 ...
##  $ REL_RWY_R     : int  0 1 1 1 1 0 0 0 0 0 ...
##  $ PROFIL_I_R    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ SPD_LIM       : int  40 70 35 35 25 70 70 35 30 25 ...
##  $ SUR_COND      : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ TRAF_CON_R    : int  0 0 1 1 0 0 0 0 0 0 ...
##  $ TRAF_WAY      : int  3 3 2 2 2 2 2 1 1 1 ...
##  $ VEH_INVL      : int  1 2 2 2 3 1 1 1 1 1 ...
##  $ WEATHER_R     : int  1 2 2 1 1 2 2 1 2 2 ...
##  $ INJURY_CRASH  : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ NO_INJ_I      : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ PRPTYDMG_CRASH: int  0 1 1 1 1 0 1 0 1 1 ...
##  $ FATALITIES    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MAX_SEV_IR    : int  1 0 0 0 0 1 0 1 0 0 ...
```

```
accidentsfull$INJURY = ifelse(accidentsfull$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?

```
table(accidentsfull$INJURY)
```

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

#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

```
#Converting the variables to factors

# Convert variables to factor
for (i in c(1:dim(accidentsfull)[2])){
  accidentsfull[,i] <- as.factor(accidentsfull[,i])
}
head(accidentsfull,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
```

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.

```
# Create a dataframe with 24 rows
accidentdata_24 <- accidentsfull[1:24,c("INJURY", "WEATHER_R", "TRAF_CON_R")]
dim(accidentdata_24)
```

```
## [1] 24  3
```

```
#Generate a pivot table from the above dataframe

# Generate a pivot table using ftable function
d1 <- ftable(accidentdata_24) #ftable for creating pivot table
d2 <- ftable(accidentdata_24[,-1]) #pivot table by dropping the first column

# print the table
d1
```

```
##                   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
```

```
d2
```

```
##           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.

```
## When INJURY = YES
# INJURY = YES, when WEATHER_R = 1, TRAF_CON_R = 0
Pt1 <- d1[3,1] / d2[1,1] #INJURY = YES, WEATHER_R = 1, TRAF_CON_R = 0
# Print the data
cat("Probabilty injury=yes when weather=1, traffic=0 is", Pt1,"\n")
```

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

```
# When INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 0
Pt2 <- d1[4,1] / d2[2,1] #INJURY = YES, WEATHER_R = 2, TRAF_CON_R = 0
# Print the data
cat("Probabilty injury=yes when weather=2, traffic=0 is", Pt2,"\n")
```

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

```
#INJURY = YES, when WEATHER_R = 1, TRAF_CON_R = 1
Pt3 <- d1[3,2] / d2[1,2] #INJURY = YES, WEATHER_R = 1, TRAF_CON_R = 1
# Print the data
cat("Probabilty injury=yes when weather=1, traffic=1 is", Pt3,"\n")
```

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

```
# INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 1
Pt4 <- d1[4,2] / d2[2,2] #INJURY = YES, WEATHER_R = 2, TRAF_CON_R = 1
# Print the data
cat("Probabilty injury=yes when weather=2, traffic=1 is", Pt4,"\n")
```

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

```
# INJURY = YES, when WEATHER_R = 1, TRAF_CON_R = 2
Pt5 <- d1[3,3] / d2[1,3] #INJURY = YES, WEATHER_R = 1, TRAF_CON_R = 2
# Print the data
cat("Probabilty injury=yes when weather=1, traffic=2 is", Pt5,"\n")
```

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

```r
# INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 2
Pt6 <- d1[4,3] / d2[2,3] #INJURY = YES, WEATHER_R = 2, TRAF_CON_R = 2
# Print the data
cat("Probabilty injury=yes when weather=2, traffic=2 is", Pt6,"\n")
```

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

```r
# Probabilities when INJURY = Yes
cat("list of probabilities when INJURY = yes", "\n")
```

```
## list of probabilities when INJURY = yes
```

```r
c(Pt1, Pt2, Pt3, Pt4, Pt5, Pt6)
```

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

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

## When INJURY = NO
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 0
no1 <- d1[1,1] / d2[1,1] #INJURY = no, WEATHER_R = 1, TRAF_CON_R = 0
# Print the data
cat("Probabilty  ijury=no when weather=1, traffic=0 is", no1,"\n")
```

```
## Probabilty  ijury=no when weather=1, traffic=0 is 0.3333333
```

```r
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 0
no2 <- d1[2,1] / d2[2,1] #INJURY = no, WEATHER_R = 2, TRAF_CON_R = 0
# Print the data
cat("Probabilty injury=no when weather=2, traffic=0 is", no2,"\n")
```

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

```r
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 1
no3 <- d1[1,2] / d2[1,2] #INJURY = no, WEATHER_R = 1, TRAF_CON_R = 1
# Print the data
cat("Probabilty injury=no when weather=1, traffic=1 is", no3,"\n")
```

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

```r
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 1
no4 <- d1[2,2] / d2[2,2] #INJURY = no, WEATHER_R = 2, TRAF_CON_R = 1
# Print the data
cat("Probabilty injury=no when weather=2, traffic=1 is", no4,"\n")
```

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

```
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 2
no5 <- d1[1,3] / d2[1,3] #INJURY = no, WEATHER_R = 1, TRAF_CON_R = 2
# Print the data
cat("Probabilty injury=no when weather=1, traffic=2 is", no5,"\n")
```

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

```
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 2
no6 <- d1[2,3] / d2[2,3] #INJURY = no, WEATHER_R = 2, TRAF_CON_R = 2
# Print the data
cat("Probabilty injury=no when weather=2, traffic=2 is", no6,"\n")
```

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

```
# Probabilities when INJURY = No
cat("list of probabilities when INJURY = NO", "\n")
```

```
## list of probabilities when INJURY = NO
```

```
c(no1, no2, no3, no4, no5, no6)
```

```
## [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.

# Taking the values from 0 to 24
probability.injury <- rep(0,24)

# for loop considering iterations from 1 to 24
for(i in 1:24){
  # when weather=1;
  if (accidentdata_24$WEATHER_R[i] == "1") {
      # when Traffic = 0
      if (accidentdata_24$TRAF_CON_R[i]=="0"){
        probability.injury[i] = Pt1
      }
      # when Traffic = 1
      else if (accidentdata_24$TRAF_CON_R[i]=="1") {
        probability.injury[i] = Pt3
      }
      # when Traffic = 2
      else if (accidentdata_24$TRAF_CON_R[i]=="2") {
        probability.injury[i] = Pt5
      }
    }
    # when weather = 2
    else {
      # when Traffic = 0
      if (accidentdata_24$TRAF_CON_R[i]=="0"){
        probability.injury[i] = Pt2
      }
      # when Traffic = 1
      else if (accidentdata_24$TRAF_CON_R[i]=="1") {
        probability.injury[i] = Pt4
      }
      # when Traffic = 2
      else if (accidentdata_24$TRAF_CON_R[i]=="2") {
        probability.injury[i] = Pt6
      }
    }
  }

# Inserting the probabilities to the table
accidentdata_24$probability.injury <- probability.injury
# print the table
head(accidentdata_24)
```

```
##   INJURY WEATHER_R TRAF_CON_R probability.injury
## 1    yes         1          0          0.6666667
## 2     no         2          0          0.1818182
## 3     no         2          1          0.0000000
## 4     no         1          1          0.0000000
## 5     no         1          0          0.6666667
## 6    yes         2          0          0.1818182
```

```
# Classifying the 24 accidents by cutoff value 0.5 that means if probability was greater than 0.5 the I
accidentdata_24$pred.probability <-
                            ifelse(accidentdata_24$probability.injury>0.5, "yes", "no")
# print the table
head(accidentdata_24)
```

```
##   INJURY WEATHER_R TRAF_CON_R probability.injury 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
```

2.3.Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

```
# Probability of getting Injured when WEATHER_R = 1
PIW <-  (d1[3,1] + d1[3,2] + d1[3,3]) / (d1[3,1] + d1[3,2] + d1[3,3] + d1[4,1] + d1[4,2] + d1[4,3])
PIW
```

```
## [1] 0.6666667
```

```
# Probability of getting Injured when TRAF_CON_R = 1
PIT <- (d1[3,2] + d1[4,2]) / (d1[3,1] + d1[3,2] + d1[3,3] + d1[4,1] + d1[4,2] + d1[4,3])
PIT
```

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

```
# Probability of getting Injured
PII <- (d1[3,1] + d1[3,2] + d1[3,3] + d1[4,1] + d1[4,2] + d1[4,3])/24
PII
```

```
## [1] 0.375
```

```
# Probability of not getting Injured when WEATHER_R = 1
PNW <- (d1[1,1] + d1[1,2] + d1[1,3]) / (d1[1,1] + d1[1,2] + d1[1,3] + d1[2,1] + d1[2,2] + d1[2,3])
PNW
```

```
## [1] 0.3333333
```

```
# Probability of not getting Injured when TRAF_CON_R = 1
PNT <- (d1[1,2] + d1[2,2]) / (d1[1,1] + d1[1,2] + d1[1,3] + d1[2,1] + d1[2,2] + d1[2,3])
PNT
```

```
## [1] 0.1333333
```

```
# Probability of not getting Injured
PNI <- (d1[1,1] + d1[1,2] + d1[1,3] + d1[2,1] + d1[2,2] + d1[2,3])/24
PNI
```

```
## [1] 0.625
```

```
# Probability of getting Injured when WEATHER_R = 1 and TRAF_CON_R = 1
PIWT1 <- (PIW * PIT * PII)/ ((PIW * PIT * PII) + (PNW * PNT * PNI))
PIWT1
```

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

```
cat("The naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1 is", PI
```

```
## 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?

```r
#Run the naiveBayes model

# Run the naiveBayes model by considering Traffic and weather
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = accidentdata_24)

# Predicting the data using naiveBayes model
nbt <- predict(nb, newdata = accidentdata_24, type = "raw")

# Adding the newly predicted data to  accidents24 dataframe
accidentdata_24$nbpred.probability <- nbt[,2] # Transfer the "Yes" nb prediction


new_1 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
         data = accidentdata_24, method = "nb")
```

```
## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample03: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample05: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample06: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample11: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample12: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample14: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample15: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample17: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample18: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample19: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample20: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

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

## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```r
predict(new_1, newdata = accidentdata_24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

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

```
predict(new_1, newdata = accidentdata_24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

```
##  [1] no no no no no no no no no no no no no no no no no no no 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%).

```
set.seed(1)
train_data <- sample(row.names(accidentsfull),0.6*dim(accidentsfull)[1])
valid_data <- setdiff(row.names(accidentdata_24),train_data)
t.df <- accidentsfull[train_data,]
v.df <- accidentsfull[valid_data,]

cat("The size of training data is:",nrow(t.df))
```

```
## The size of training data is: 25309
```

```
cat("The size of validation data is:",nrow(v.df))
```

```
## The size of validation data is: 10
```

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.

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

```
train <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data =t.df)
validation_class <- predict(train, v.df)

values <- preProcess(t.df[,], method = c("center", "scale"))
```

```
## Warning in pre_process_options(method, column_types): The following
## pre-processing methods were eliminated: 'center', 'scale'
```

```
a.norm.df <- predict(values, t.df[, ])
v.norm.df <- predict(values, v.df[, ])

levels(a.norm.df)
```

```
## NULL
```

```
class(a.norm.df$INJURY)
```

```
## [1] "factor"
```

```
a.norm.df$INJURY <- as.factor(a.norm.df$INJURY)
class(a.norm.df$INJURY)
```

```
## [1] "factor"
```

```
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = a.norm.df)

prediction <- predict(nb_model, newdata = v.norm.df)

#Ensure that factor levels in validation dataset match those in training dataset
v.norm.df$INJURY <- factor(v.norm.df$INJURY, levels = levels(a.norm.df$INJURY))

# Show the confusion matrix
confusionMatrix(prediction, v.norm.df$INJURY)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no   4   1
##        yes  2   3
##
##                Accuracy : 0.7
##                  95% CI : (0.3475, 0.9333)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.3823
##
##                   Kappa : 0.4
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.6667
##             Specificity : 0.7500
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.6000
##              Prevalence : 0.6000
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7083
##
##        'Positive' Class : no
##
```

```
# Calculate the overall error rate
error_rate <- 1 - sum(prediction == v.norm.df$INJURY) / nrow(v.norm.df)

cat("The overall error of the validation set is :",1 - sum(prediction == v.norm.df$INJURY) / nrow(v.nor
```

```
## The overall error of the validation set is : 0.3
```