

FML ASSIGNMENT 3 - 811290653

GEETHIKA VULLI

2023-10-15

Summary

It is considered that there may be injuries when an accident has just been reported and no additional information is provided (INJURY = Yes). In order to appropriately portray the accident's highest amount of harm, MAX_SEV_IR, this assumption is made. According to the instructions, if MAX_SEV_IR is 1 or 2, there has been some sort of injury (INJURY = Yes). If MAX_SEV_IR, on the other hand, equals 0, it means that there is no inferred injury (INJURY = No). Therefore, when there is a dearth of additional information regarding the accident, it is fair to infer the presence of some degree of harm until new information demonstrates otherwise.

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

The subsequent actions were conducted in order to produce a new data frame with 24 records and only 3 variables (Injury, Weather, and Traffic):

a pivot table was made with the variables traffic, weather, and injury. The data had to be arranged in a tabular format with these particular columns in this phase.

- The variable Injury was dropped from the data frame because it wouldn't be used in the analysis that would follow.

For each of the first 24 entries in the data frame, Bayes probabilities were estimated to determine the likelihood that an injury will occur.

Accidents that were classified using a cutoff of 0.5. - Based on a 0.5 cutoff criterion, each accident was classified as either likely or not likely to result in injuries using the probabilities derived in Step 3. In order to calculate the naive bayes conditional probability of harm, WEATHER_R and TRAF_CON_R were set to 1 respectively. These are the outcomes.

-If there is an injury, the likelihood is 0.

If there is no injury, the likelihood is 1.

The Naive Bayes model's predictions and the exact Bayes classification have 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 Levels: no 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 Levels: no yes

The records are classified in this situation as either “yes” or “no.” The most important finding is that both categories have the same values at some spots, demonstrating that the two classifications agree on the ranking or order of the observations. This implies that both classes value the factors similarly and have a similar grasp of the data.

The full dataset is split into two sets in the following step: a training set (which will contain 60% of the data) and a validation set (40% of the data). The model will be trained using the training data after the

dataset has been split in order to accomplish this. The full dataset will be utilized to evaluate the model's performance and its capacity to forecast upcoming accidents, and metrics including accuracy, precision, recall, and F1-score will be used for a thorough review.

The next step is to normalize the data after segmenting the data frame. By guaranteeing that each segment is represented as a single row, this normalization procedure enables more precise decision-making. It is crucial that the qualities being studied have constant levels and are either numeric or integer values to ensure the validity of comparisons. This uniformity in attribute levels and data types aids in preventing analytical errors and guarantees that the operations performed on the data produce accurate and significant findings for use in decision-making.

Additionally, you mentioned that the overall error rate for the validation set is approximately 0.47 when expressed as a decimal. This suggests that the Naive Bayes classifier performs quite well and accurately on this dataset.

Here are the confusion matrix and statistics for your classification model:

- Accuracy: The accuracy of your model is 0.5, indicating that 50% of the predictions are correct.
- Sensitivity: Sensitivity, also known as true positive rate or recall, is 0.15635. This means that your model correctly identifies positive cases (e.g., injuries) 15.635% of the time.
- Specificity: Specificity is 0.8708, indicating that your model correctly identifies negative cases (e.g., no injuries) 87.08% of the time.

As a whole, your model seems to function very well, although it might not be very good at accurately forecasting injuries, especially in circumstances where the injuries are positive. The Naive Bayes approach works well, but it oversimplifies the sometimes-untrue assumption of variable independence. Consider the particular conclusions and their consequences in light of your own dataset and goals.

#calling the libraries

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

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

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

```
library(class)
```

```
#reading of the dataset
```

```
accident_data <- read.csv("C:\\Users\\geeth\\Downloads\\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
```

```
str(accident_data)
```

```
## '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 ...
```

```
accident_data$INJURY = ifelse(accident_data$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(accident_data$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(accident_data)[2])){
  accident_data[,i] <- as.factor(accident_data[,i])
}
head(accident_data,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
accident_data_24 <- accident_data[1:24,c("INJURY", "WEATHER_R", "TRAF_CON_R")]
dim(accident_data_24)
```

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

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

# Generate a pivot table using ftable function
d1 <- ftable(accident_data_24) #ftable for creating pivot table
d2 <- ftable(accident_data_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
P1 <- 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", P1,"\n")
```

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

```
# When INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 0
P2 <- 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", P2,"\n")
```

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

```
#INJURY = YES, when WEATHER_R = 1, TRAF_CON_R = 1
P3 <- 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", P3,"\n")
```

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

```
# INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 1
P4 <- 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", P4,"\n")
```

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

```
# INJURY = YES, when WEATHER_R = 1, TRAF_CON_R = 2
P5 <- 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", P5,"\n")
```

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

```
# INJURY = YES, when WEATHER_R = 2, TRAF_CON_R = 2
P6 <- 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", P6,"\n")
```

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

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

```
## list of probabilities 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.
```

```
## When INJURY = NO
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 0
n1 <- d1[1,1] / d2[1,1] #INJURY = no, WEATHER_R = 1, TRAF_CON_R = 0
# Print the data
cat("Probabilty injury=no when weather=1, traffic=0 is", n1,"\n")
```

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

```
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 0
n2 <- 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", n2,"\n")
```

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

```
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 1
n3 <- 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", n3,"\n")
```

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

```
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 1
n4 <- 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", n4,"\n")
```

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



```
# INJURY = no when WEATHER_R = 1, TRAF_CON_R = 2
n5 <- 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", n5,"\n")
```

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

```
# INJURY = no when WEATHER_R = 2, TRAF_CON_R = 2
n6 <- 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", n6,"\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(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.

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

# for loop considering iterations from 1 to 24
for(i in 1:24){
  # when weather=1;
  if (accident_data_24$WEATHER_R[i] == "1") {
    # when Traffic = 0
    if (accident_data_24$TRAF_CON_R[i]=="0"){
      probability.inj[i] = P1
    }
    # when Traffic = 1
    else if (accident_data_24$TRAF_CON_R[i]=="1") {
      probability.inj[i] = P3
    }
    # when Traffic = 2
    else if (accident_data_24$TRAF_CON_R[i]=="2") {
      probability.inj[i] = P5
    }
  }
  # when weather = 2
  else {
```

```

# when Traffic = 0
if (accident_data_24$TRAF_CON_R[i]=="0"){
  probability.inj[i] = P2
}
# when Traffic = 1
else if (accident_data_24$TRAF_CON_R[i]=="1") {
  probability.inj[i] = P4
}
# when Traffic = 2
else if (accident_data_24$TRAF_CON_R[i]=="2") {
  probability.inj[i] = P6
}
}
}

# Inserting the probabilities to the table
accident_data_24$probability.inj <- probability.inj
# print the table
head(accident_data_24)

```

```

##   INJURY WEATHER_R TRAF_CON_R probability.inj
## 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
accident_data_24$pred.probability <-
  ifelse(accident_data_24$probability.inj>0.5, "yes", "no")
# print the table
head(accident_data_24)

```

```

##   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

```

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", P
```

```
## 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?

```

#Run the naiveBayes model

# Run the naiveBayes model by considering Traffic and weather
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = accident_data_24)

# Predicting the data using naiveBayes model
nbt <- predict(nb, newdata = accident_data_24, type = "raw")

# Adding the newly predicted data to accidents24 dataframe
accident_data_24$nbpred.probability <- nbt[,2] # Transfer the "Yes" nb prediction

new_1 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
               data = accident_data_24, method = "nb")

```

```

## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

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

## Warning: model fit failed for Resample03: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2

## Warning: model fit failed for Resample05: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   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.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

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

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

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

## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   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.default
##   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.default
##   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.default
##   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.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

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

## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2

## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## 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

predict(new_1, newdata = accident_data_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
## Levels: no yes

```

```
predict(new_1, newdata = accident_data_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
## 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(accident_data), 0.6 * dim(accident_data)[1])
valid_data <- setdiff(row.names(accident_data), train_data)
t.df <- accident_data[train_data,]
v.df <- accident_data[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: 16874
```

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?

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

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

# 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.norm.df))

## The overall error of the validation set is : 0.4771838

```