Assignment 3 FML

Sudarshan.Rayapati

2023-10-15

```
knitr::opts_chunk$set(echo = TRUE)
```

Summary

1:-According to the given dataset, if an accident has just been reported and no further information is available, there is a 50.88% chance that the injury has taken place, because data suggests that previously out of 42,183 cases, 21,462 cases have reported “injury=yes”.

2.1: The exact Bayes conditional probabilities of an injury (INJURY = Yes) for the six possible combinations of the predictons are:

Predictor combination Probability WEATHER_R = 1 and TRAF_CON_R = 1 0.6666667 WEATHER_R = 2 and TRAF_CON_R = 0 0.1818182 WEATHER_R = 1 and TRAF_CON_R = 1 0.0000000 WEATHER_R = 2 and TRAF_CON_R = 1 0.0000000 WEATHER_R = 1 and TRAF_CON_R = 2 0.0000000 WEATHER_R = 2 and TRAF_CON_R = 2 1.0000000 2.2: The Classification of the 24 accidents using their probabilities and a cutoff of 0.5 quantitatively are:

[0.6666667 0.1818182 0.0000000 0.0000000 0.6666667 0.1818182 0.1818182 0.6666667 0.1818182 0.1818182 0.1818182 0.0000000 0.6666667 0.6666667 0.6666667 0.6666667 0.1818182 0.1818182 0.1818182 0.1818182 0.6666667 0.6666667 1.0000000 0.1818182]

qualitatively are:

[“yes” “no” “no” “no” “yes” “no” “no” “yes” “no” “no” “no” “no” “yes” “yes” “yes” “yes” “no” “no” “no” “no” “yes” “yes” “yes” “no”]

2.3: The Computation of the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1 manually the output is “0”.

2.4: Now, after applying the naïve Bayes classifier to the 24 records and two predictors, checking the model output to get probabilities and classifications for all 24 records, it was identified that the result of classification and ranks were not equal as the Exact Bayes caluclation.

3.1: When we applied the naive Bayes classifier to the full training set with the key predictors and used “INJURY” as the response, the results shows us confusion matrix and statistics. The accuracy of the model turned out to be 53.7%. This tells us

that the model predicts whether an accident causes injury or not.

Confusion Matrix and Statistics

```
      Reference
```

Prediction no yes no 3444 4866 yes 2947 5617

```
           Accuracy : 0.537
```

3.2: the overall error of the validation set is “46.3”.

Problem Statement

The file accidentsFull.csv contains information on 42,183 actual automobile accidents in 2001 in the United States that involved one of three levels of injury: NO INJURY, INJURY, or FATALITY. For each accident, additional information is recorded, such as day of week, weather conditions, and road type. A firm might be interested in developing a system for quickly classifying the severity of an accident based on initial reports and associated data in the system (some of which rely on GPS-assisted reporting).

Our aim is to predict whether an accident just reported will involve an injury (MAX_SEV_IR = 1 or 2) or will not (MAX_SEV_IR = 0). For this purpose, create a dummy variable called INJURY that takes the value “yes” if MAX_SEV_IR = 1 or 2, and otherwise “no.”

Data Input and Cleaning

Loading the required libraries and reading the input file

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

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

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

```
library(ggplot2)
```

Q1. Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?

```
accidents_24 <- read.csv("/Users/sudarshan/Desktop/FML/dataset/accidentsFull.csv")
accidents_24$INJURY = ifelse(accidents_24$MAX_SEV_IR>0,"yes","no")
injury_table <- table(accidents_24$INJURY)
injury_table
```

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

```
head(accidents_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
##   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 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
```

```
probability_injury <- (injury_table["yes"] / sum(injury_table))*100
probability_injury
```

```
##      yes
## 50.87832
```

```
#Converting variables into factor
for (i in c(1:dim(accidents_24)[2])){
  accidents_24[,i] <- as.factor(accidents_24[,i])
}
head(accidents_24,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
```

Q2. Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER_R and TRAF_CON_R. Create a pivot table that examines INJURY as a function of the two predictors for these 12 records. Use all three variables in the pivot table as rows/columns.

```
accidents_24_Df <- accidents_24[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
head(accidents_24_Df)
```

```
##   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
```

```
Pt1 <- ftable(accidents_24_Df)
Pt2 <- ftable(accidents_24_Df[,-1]) # print table only for conditions
Pt1
```

```
##                   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
```

```
Pt2
```

```
##           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.

```
#Injury = yes
p1 = Pt1[3,1] / Pt2[1,1] # Injury, Weather=1 and Traf=0
p2 = Pt1[4,1] / Pt2[2,1] # Injury, Weather=2, Traf=0
p3 = Pt1[3,2] / Pt2[1,2] # Injury, W=1, T=1
p4 = Pt1[4,2] / Pt2[2,2] # I, W=2,T=1
p5 = Pt1[3,3] / Pt2[1,3] # I, W=1,T=2
p6 = Pt1[4,3]/ Pt2[2,3] #I,W=2,T=2

# Injury = no
n1 = Pt1[1,1] / Pt2[1,1] # Weather=1 and Traf=0
n2 = Pt1[2,1] / Pt2[2,1] # Weather=2, Traf=0
n3 = Pt1[1,2] / Pt2[1,2] # W=1, T=1
n4 = Pt1[2,2] / Pt2[2,2] # W=2,T=1
n5 = Pt1[1,3] / Pt2[1,3] # W=1,T=2
n6 = Pt1[2,3] / Pt2[2,3] # W=2,T=2
print(c(p1,p2,p3,p4,p5,p6))
```

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

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

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

2.2:-Classify the 24 accidents using these

probabilities and a cutoff of 0.5.

```
prob.inj <- rep(0,24)

for (i in 1:24) {
  print(c(accidents_24_Df$WEATHER_R[i],accidents_24_Df$TRAF_CON_R[i]))
    if (accidents_24_Df$WEATHER_R[i] == "1") {
      if (accidents_24_Df$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p1
      }
      else if (accidents_24_Df$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p3
      }
      else if (accidents_24_Df$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p5
      }
    }
    else {
      if (accidents_24_Df$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p2
      }
      else if (accidents_24_Df$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p4
      }
      else if (accidents_24_Df$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p6
      }
    }
  }
```

```
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 1
## Levels: 1 2 0
## [1] 1 1
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
```

```
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 2
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 2
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
```

```
accidents_24_Df$prob.inj <- prob.inj
accidents_24_Df$prob.inj
```

```
##  [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.6666667 0.1818182 0.1818182
##  [8] 0.6666667 0.1818182 0.1818182 0.1818182 0.0000000 0.6666667 0.6666667
## [15] 0.6666667 0.6666667 0.1818182 0.1818182 0.1818182 0.1818182 0.6666667
## [22] 0.6666667 1.0000000 0.1818182
```

```
accidents_24_Df$pred.prob <- ifelse(accidents_24_Df$prob.inj>0.5, "yes", "no")
accidents_24_Df$pred.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"
```

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

Answer:- Probability(Injury=Yes/WEATHER_R=1,TRAF_CON_R=1)

= [Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Yes)] / [Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Yes) + Probability(WEATHER_R=1/Injury=No) * Probability(TRAF_CON_R=1/Injury=No) * Probability(Injury=No)]

= [6/9 * 0/9 * 9/24] / [6/9 * 0/9 * 9/24 + 5/15 * 2/15 * 15/24] = The result will be “0” Because the numerator is zero.

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?

```
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = accidents_24_Df)

nbt <- predict(nb, newdata = accidents_24_Df,type = "raw")
accidents_24_Df$nbpred.prob <- nbt[,2] # Transfer the "Yes" nb prediction
accidents_24_Df$nbpred.prob
```

```
##  [1] 0.571428571 0.250000000 0.002244949 0.008919722 0.571428571 0.250000000
##  [7] 0.250000000 0.571428571 0.250000000 0.250000000 0.250000000 0.666666667
## [13] 0.571428571 0.571428571 0.571428571 0.571428571 0.250000000 0.250000000
## [19] 0.250000000 0.250000000 0.571428571 0.571428571 0.333333333 0.250000000
```

```
library(klaR)
```

```
## Loading required package: MASS
```

```
#Loading the klaR package

# Creating a variable named formula
formula <- INJURY ~ TRAF_CON_R + WEATHER_R
# Training the Naive Bayes model with Laplace

accidents_24_Df$INJURY <- as.factor(accidents_24_Df$INJURY)
nb2 <- NaiveBayes(formula,data = accidents_24_Df, laplace = 1)

# Making predictions with the model
 predict(nb2, newdata = accidents_24_Df[, c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

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

```
predict(nb2, newdata = accidents_24_Df[, c("INJURY", "WEATHER_R", "TRAF_CON_R")], typ
e = "raw")
```

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

```
#predictions
#raw_probabilities
```

```
# Comparing Both naive Bayes model and exact Bayes classification
classification_match <- all(accidents_24_Df$nbpred.prob == accidents_24_Df$prob.inj)
probability_match <- all.equal(accidents_24_Df$nbpred.prob, accidents_24_Df$prob.inj)

# Checking if classifications and rankings are equal
if (classification_match && is.na(probability_match)) {
  cat("The resulting classifications and rankings are equal.\n")
} else {
  cat("The resulting classifications and rankings are not equal.\n")
}
```

```
## The resulting classifications and rankings are not equal.
```

Q3, Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%). 3.1, Run a naive Bayes classifier on the complete training set with the relevant predictors(and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix.

```
set.seed(1)
train.index <- sample(c(1:dim(accidents_24)[1]), dim(accidents_24)[1]*0.6)
train.df <- accidents_24[train.index,]
valid.df <- accidents_24[-train.index,]
#Identifying a variable to use
vars <- c("INJURY", "HOUR_I_R",  "ALIGN_I" ,"WRK_ZONE",  "WKDY_I_R",
          "INT_HWY",  "LGTCON_I_R", "PROFIL_I_R", "SPD_LIM", "SUR_COND",
          "TRAF_CON_R",   "TRAF_WAY",   "WEATHER_R")

nbTotal <- naiveBayes(INJURY~.,data = train.df[,vars])
nbTotal
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        no       yes
## 0.4939745 0.5060255
##
## Conditional probabilities:
##      HOUR_I_R
## Y             0         1
```

```
##   no  0.5689490 0.4310510
##   yes 0.5703131 0.4296869
##
##      ALIGN_I
## Y             1         2
##   no  0.8712206 0.1287794
##   yes 0.8652300 0.1347700
##
##      WRK_ZONE
## Y              0          1
##   no  0.97664374 0.02335626
##   yes 0.97727805 0.02272195
##
##      WKDY_I_R
## Y             0         1
##   no  0.2194049 0.7805951
##   yes 0.2381510 0.7618490
##
##      INT_HWY
## Y                0            1            9
##   no  0.8513837786 0.1481362982 0.0004799232
##   yes 0.8593737800 0.1397673147 0.0008589053
##
##      LGTCON_I_R
## Y             1         2         3
##   no  0.6870101 0.1251000 0.1878899
##   yes 0.7014914 0.1096275 0.1888811
##
##      PROFIL_I_R
## Y             0         1
##   no  0.7531595 0.2468405
##   yes 0.7633326 0.2366674
##
##      SPD_LIM
## Y                5           10           15           20           25
##   no  0.0000799872 0.0004799232 0.0043992961 0.0085586306 0.1121420573
##   yes 0.0001561646 0.0003123292 0.0040602795 0.0039041149 0.0906535488
##      SPD_LIM
## Y               30           35           40           45           50
##   no  0.0860662294 0.1896496561 0.0962246041 0.1553351464 0.0407934730
##   yes 0.0860466932 0.2123057703 0.1068946670 0.1574139143 0.0394315609
##      SPD_LIM
## Y               55           60           65           70           75
##   no  0.1590145577 0.0355143177 0.0645496721 0.0409534474 0.0062390018
##   yes 0.1549152807 0.0430233466 0.0621535098 0.0311548372 0.0075739830
##
##      SUR_COND
```

```
## Y                1           2           3           4           9
##   no  0.774196129 0.176931691 0.016717325 0.028155495 0.003999360
##   yes 0.815725775 0.151245413 0.010697275 0.016709612 0.005621926
##
##      TRAF_CON_R
## Y             0         1         2
##   no  0.6566149 0.1902096 0.1531755
##   yes 0.6213009 0.2191770 0.1595221
##
##      TRAF_WAY
## Y              1          2          3
##   no  0.57998720 0.36690130 0.05311150
##   yes 0.56063090 0.39743890 0.04193019
##
##      WEATHER_R
## Y             1         2
##   no  0.8390657 0.1609343
##   yes 0.8744437 0.1255563
```

```
3
```

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

```
#Creating the confusion matrix using the train.df, the prediction and the classes
confusionMatrix(train.df$INJURY, predict(nbTotal, train.df[, vars]), positive = "ye
s")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5097 7405
##        yes 4230 8577
##
##                Accuracy : 0.5403
##                  95% CI : (0.5341, 0.5464)
##     No Information Rate : 0.6315
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0776
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5367
##             Specificity : 0.5465
##          Pos Pred Value : 0.6697
##          Neg Pred Value : 0.4077
##              Prevalence : 0.6315
##          Detection Rate : 0.3389
##    Detection Prevalence : 0.5060
##       Balanced Accuracy : 0.5416
##
##        'Positive' Class : yes
##
```

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

```
confusionMatrix(valid.df$INJURY, predict(nbTotal, valid.df[, vars]), positive = "ye
s")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  3203 5016
##        yes 2862 5793
##
##                Accuracy : 0.5331
##                  95% CI : (0.5256, 0.5407)
##     No Information Rate : 0.6406
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0594
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5359
##             Specificity : 0.5281
##          Pos Pred Value : 0.6693
##          Neg Pred Value : 0.3897
##              Prevalence : 0.6406
##          Detection Rate : 0.3433
##    Detection Prevalence : 0.5129
##       Balanced Accuracy : 0.5320
##
##        'Positive' Class : yes
##
```

```
#Calculating the overall error

ver=1-0.5331
verp=ver*100
paste("Overall Error: ",verp)
```

```
## [1] "Overall Error:  46.69"
```

###CONCLUSION

Using two predictors both times, the Naive Bayes classifier was applied to first predict injury outcomes in a data set of 24 records and subsequently to the whole data set.

Using the exact Bayes classifier for the first 24 records, we identified that the most risky combination for drivers is WEATHER_CON=2,TRAF_CON=0 since the injury is max at “1” in this case.

The model has a validation error of 46.3% and a training set accuracy of 53.7%, showing a modest level of prediction. However, it makes the assumption that the prediction variables are independent, which may not always be the case in real-world situations and might result in errors.But for classification and ranking, we may use the Naive Bayes classifier.

While Naive Bayes is a simple and effective algorithm for injury outcome prediction, it’s crucial to remember its limitations.