FML Assignment-3

Saride Lakshmi Priya

2023-10-15

```r
#Loading the libraries that are required for the task
library(class)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(e1071)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
#Loading the data set and assigning it to buried variable.
Accidentss_Full <- read.csv("C:\\Users\\DELL\\Downloads\\accidentsFull.csv")
dim(Accidentss_Full)
```

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

```r
Accidentss_Full$INJURY = ifelse(Accidentss_Full$MAX_SEV_IR %in% c(1,2),"yes","no")
table(Accidentss_Full$INJURY) # as yes is greater then no
```

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

```r
t(t(names(Accidentss_Full)))
```

```
##       [,1]
##  [1,] "HOUR_I_R"
##  [2,] "ALCHL_I"
##  [3,] "ALIGN_I"
##  [4,] "STRATUM_R"
##  [5,] "WRK_ZONE"
##  [6,] "WKDY_I_R"
##  [7,] "INT_HWY"
##  [8,] "LGTCON_I_R"
##  [9,] "MANCOL_I_R"
## [10,] "PED_ACC_R"
## [11,] "RELJCT_I_R"
## [12,] "REL_RWY_R"
## [13,] "PROFIL_I_R"
## [14,] "SPD_LIM"
## [15,] "SUR_COND"
## [16,] "TRAF_CON_R"
## [17,] "TRAF_WAY"
## [18,] "VEH_INVL"
## [19,] "WEATHER_R"
## [20,] "INJURY_CRASH"
## [21,] "NO_INJ_I"
## [22,] "PRPTYDMG_CRASH"
## [23,] "FATALITIES"
## [24,] "MAX_SEV_IR"
## [25,] "INJURY"
```

```
#Creating the pivot tables
sub_Accidentss_Full <- Accidentss_Full[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
sub_Accidentss_Full
```

```
##    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
## 7      no         2          0
## 8     yes         1          0
## 9      no         2          0
## 10     no         2          0
## 11     no         2          0
## 12     no         1          2
## 13    yes         1          0
## 14     no         1          0
## 15    yes         1          0
## 16    yes         1          0
## 17     no         2          0
## 18     no         2          0
## 19     no         2          0
## 20     no         2          0
## 21    yes         1          0
## 22     no         1          0
## 23    yes         2          2
## 24    yes         2          0
```

```
pvr_table1 <- ftable(sub_Accidentss_Full)
pvr_table1
```

```
##                   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
```

```
pvr_table2 <- ftable(sub_Accidentss_Full[,-1])
pvr_table2
```

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

##2.1

```
#bayes
#INJURY = YES
pair_k = pvr_table1[3,1]/pvr_table2[1,1]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):", pair_k, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0): 0.6666667
```

```
pair_s = pvr_table1[3,2]/pvr_table2[1,2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", pair_s, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
pair_u = pvr_table1[3,3]/pvr_table2[1,3]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):", pair_u, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2): 0
```

```
pair_v = pvr_table1[4,1]/pvr_table2[2,1]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):", pair_v, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0): 0.1818182
```

```
pair_w = pvr_table1[4,2]/pvr_table2[2,2]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):", pair_w, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1): 0
```

```r
pair_x = pvr_table1[4,3]/pvr_table2[2,3]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):", pair_x, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2): 1
```

```r
#Now we check the condition whether Injury = no

dual_k = pvr_table1[1,1]/pvr_table2[1,1]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0):", dual_k, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0): 0.3333333
```

```r
dual_s = pvr_table1[1,2]/pvr_table2[1,2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", dual_s, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

```r
dual_u = pvr_table1[1,3]/pvr_table2[1,3]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2):", dual_u, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2): 1
```

```r
dual_v = pvr_table1[2,1]/pvr_table2[2,1]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0):", dual_v, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0): 0.8181818
```

```r
dual_w = pvr_table1[2,2]/pvr_table2[2,2]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1):", dual_w, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1): 1
```

```r
dual_x = pvr_table1[2,3]/pvr_table2[2,3]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2):", dual_x, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2): 0
```

```r
#Now probability of the total occurences.
```

```r
#cutoff is 0.5 and for 24 records
# Assuming you have calculated the conditional probabilities already, you can use them to classify the
# Let's say you have a data frame named 'new_data' containing these 24 records.

prob_injury <- rep(0,24)
for(s in 1:24){
  print(c(sub_Accidentss_Full$WEATHER_R[s],sub_Accidentss_Full$TRAF_CON_R[s]))

  if(sub_Accidentss_Full$WEATHER_R[s] == "1" && sub_Accidentss_Full$TRAF_CON_R[s] == "0"){
    prob_injury[s] = pair_k

  } else if (sub_Accidentss_Full$WEATHER_R[s] == "1" && sub_Accidentss_Full$TRAF_CON_R[s] == "1"){
    prob_injury[s] = pair_s

  } else if (sub_Accidentss_Full$WEATHER_R[s] == "1" && sub_Accidentss_Full$TRAF_CON_R[s] == "2"){
    prob_injury[s] = pair_u

  }
  else if (sub_Accidentss_Full$WEATHER_R[s] == "2" && sub_Accidentss_Full$TRAF_CON_R[s] == "0"){
    prob_injury[s] = pair_v

  } else if (sub_Accidentss_Full$WEATHER_R[s] == "2" && sub_Accidentss_Full$TRAF_CON_R[s] == "1"){
    prob_injury[s] = pair_w

  }
  else if(sub_Accidentss_Full$WEATHER_R[s] == "2" && sub_Accidentss_Full$TRAF_CON_R[s] == "2"){
    prob_injury[s] = pair_x
  }

}
```

```
## [1] 1 0
## [1] 2 0
## [1] 2 1
## [1] 1 1
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 2
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 1 0
## [1] 2 2
## [1] 2 0
```

```
#cutoff 0.5

sub_Accidentss_Full$prob_injury = prob_injury
sub_Accidentss_Full$pred.prob  = ifelse(sub_Accidentss_Full$prob_injury>0.5, "yes","no")
```

```
head(sub_Accidentss_Full)
```

```
##   INJURY WEATHER_R TRAF_CON_R prob_injury pred.prob
## 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
```

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

```
IY = pvr_table1[3,2]/pvr_table2[1,2]
Y = (IY * pvr_table1[3, 2]) / pvr_table2[1, 2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", IY, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
IN = pvr_table1[1,2]/pvr_table2[1,2]
N = (IY * pvr_table1[3, 2]) / pvr_table2[1, 2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", IN, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

##2.4

```
new_k <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = sub_Accidentss_Full)

new_Accidentss_Full <- predict(new_k, newdata = sub_Accidentss_Full,type = "raw")
sub_Accidentss_Full$nbpred.prob <- new_Accidentss_Full[,2]


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

```
## 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_R

## 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_R, WEATHER_R

## 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_R

## 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_R

## 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_R
## 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_R, WEATHER_R

## 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_R, WEATHER_R

## 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_R

## 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_R

## 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_R

## 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_R

## 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_R

## 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_R

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

```r
predict(new_s, newdata = sub_Accidentss_Full[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

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

```r
predict(new_s, newdata = sub_Accidentss_Full[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

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

#Let's go back to the complete dataset now. Divide the data into training and validation (60 and 40%, respectively). Run a naïve Bayes classifier with the pertinent predictors (with INJURY as the response) on the entire training data. Keep in mind that every predictor is categorical. Display the matrix of bewilderment.What is the validation set's overall error?

```r
accident = Accidentss_Full[c(-24)]

set.seed(1)
accs_index = sample(row.names(accident), 0.6*nrow(accident)[1])
valid_index = setdiff(row.names(accident), accs_index)
```

```r
accs_df = accident[accs_index,]
valid_df= accident[valid_index,]

dim(accs_df)
```

```
## [1] 25309    24
```

```r
dim(valid_df)
```

```
## [1] 16874    24
```

```r
norm_values <- preProcess(accs_df[,], method = c("center", "scale"))
accs_norm.df <- predict(norm_values, accs_df[, ])
valid_norm.df <- predict(norm_values, valid_df[, ])

levels(accs_norm.df)
```

```
## NULL
```

```r
class(accs_norm.df$INJURY)
```

```
## [1] "character"
```

```r
accs_norm.df$INJURY <- as.factor(accs_norm.df$INJURY)

class(accs_norm.df$INJURY)
```

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

```r
nb.model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = accs_norm.df)

predictions <- predict(nb.model, newdata = valid_norm.df)

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

# Show the confusion matrix
confusionMatrix(predictions, valid_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
##
```

```r
# Calculate the overall error rate
error_rate <- 1 - sum(predictions == valid_norm.df$INJURY) / nrow(valid_norm.df)
error_rate
```

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

#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.

-In total, there are "20721 NO and yes are 21462".

In order to produce a new data frame with 24 records and only 3 variables (Injury, Weather, and Traffic), the following actions were taken:

a pivot table was made using the variables traffic, weather, and injury. -This stage included setting up the data in a table with these particular columns.

The variable "Injury" was removed. -Injury was taken out of the data frame because it wasn't necessary for the analysis that came next.

Bayes probabilities were computed. -In order to calculate the likelihood of an injury for each of the first 24 records in the data frame, Bayes probabilities were generated.

utilising a cutoff of 0.5 to classify accidents. -Based on a 0.5 cutoff criterion, each accident was classified as either likely to result in an injury or not likely based on the probabilities determined in Step 3. With the provided characteristics WEATHER_R = 1 and TRAF_CON = 1, we calculated the naive Bayes conditional probability of harm. The outcomes are as follows.

-If INJURY = YES, the probability is 0.

-If INJURY - NO , the probability 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 in this situation are categorised as either "yes" or "no." The most important finding is the fact that both categories share values at specific positions, showing that the rankings or ordering of the observations are consistent between the two classifications. This implies that both classes place a similar value on the factors and have a common understanding of the information.

The analysis will then divide the entire dataset into two sets: a training set (which will contain 60% of the data) and a validation set (40% of the data). The goal is to create a model that can forecast accidents in the future, especially those containing fresh or unrecognised data. To accomplish this, the model will be trained using training data that was obtained after the dataset was split. Based on the whole dataset, parameters including accuracy, precision, recall, and F1-score will be used for a thorough evaluation of the model's performance and its capacity to forecast upcoming incidents. A validation set is: In order to validate the data in this set, we use a reference dataset as our training dataset.

The data must first be normalised before being segmented from the data frame. In order to facilitate more precise decision-making, this normalisation method makes sure that each segment is represented as a single row. The properties being analysed must have constant levels and either have numeric or integer values in order for comparisons to be legitimate. This consistency in attribute levels and data types aids in the prevention of analysis errors and makes sure that the operations performed on the data produce accurate and significant results for use in decision-making.

Printing the classifications. -The classifications, which indicate whether each mishap is likely to cause an injury or not, can be printed or displayed for additional research or reporting.

These actions imply that you've conducted a statistical analysis to determine the risk of injuries in accidents based on the supplied variables (Weather and Traffic), and that you've then classified these events using a 0.5 probability cutoff. For determining risks and making decisions in many circumstances, this information might be helpful.

When comparing the precise Bayes and Naive Bayes classification algorithms, you found some disparities in the "Yes" category of classifications. The Naive Bayes classifier's assumption of independence, which might not be valid in all circumstances, may be to blame for this disparity. Though it might be computationally taxing for bigger datasets, the exact Bayes technique is seen to be superior when precise probabilities and conditional dependencies are important.

You also indicated that, when reported as a decimal, the overall error rate for the validation set is roughly 0.47. This shows that, on this dataset, the Naive Bayes classifier performs pretty effectively and accurately.

The classification model's confusion matrix and statistics are as follows:

-Accuracy: Your model's accuracy is 0.5, which means that 50% of the predictions are accurate.

- Sensitivity: Sensitivity, sometimes referred to as the true positive rate or recall, is 0.15635. This indicates that your model accurately detects positive situations (such as injuries) 15.635% of the time.
- Specificity: The specificity of your model is 0.8708, meaning it accurately recognises negative instances (such as no injuries) 87.08% of the time.

In conclusion, your model seems to function quite well, although it may not be reliable in forecasting injuries, particularly for positive situations. The Naive Bayes approach works well, however it oversimplifies the assumption of variable independence, which may not always be true. In light of your unique dataset and aims, the particular results and their implications should be taken into account.