FML Assignment 2

2023-10-16

```r
# Loading the required libraries

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
Accidents.data <- read.csv("C:/Users/navat/Downloads/accidentsFull.csv")
```

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?

```r
Accidents.data$INJURY=ifelse(Accidents.data$MAX_SEV_IR%in% c(1,2),"yes","no")
table(Accidents.data$INJURY)
```

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

```r
t(t(names(Accidents.data)))
```

```
##       [,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"
```

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.

```
# Pivot table for the data

Acc.data <- Accidents.data[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]

Acc.data
```

```
##    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
```

```
Piv.table <- ftable(Acc.data)

Piv.table2 <- ftable(Acc.data[,-1])
```

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

```
#If the injury = Yes

Combination1 <- Piv.table[3,1]/Piv.table2[1,1]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=0):",Combination1,"\n")
```

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

```
Combination2 <- Piv.table[3,2]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=1):",Combination2,"\n")
```

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

```
Combination3 <- Piv.table[3,3]/Piv.table2[1,3]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=2):",Combination3,"\n")
```

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

```
Combination4 <- Piv.table[4,1]/Piv.table2[2,1]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=0):",Combination4,"\n")
```

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

```
Combination5 <- Piv.table[4,2]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=1):",Combination5,"\n")
```

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

```
Combination6 <- Piv.table[4,3]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=2):",Combination6,"\n")
```

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

```r
# If the injury = No

SCombination1 <- Piv.table[1,1]/Piv.table2[1,1]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=0):",SCombination1,"\n")
```

```
## P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=0): 0.3333333
```

```r
SCombination2 <- Piv.table[1,2]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=1):",SCombination2,"\n")
```

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

```r
SCombination3 <- Piv.table[1,3]/Piv.table2[1,3]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=2):",SCombination3,"\n")
```

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

```r
SCombination4 <- Piv.table[2,1]/Piv.table2[2,1]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=0):",SCombination4,"\n")
```

```
## P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=0): 0.8181818
```

```r
SCombination5 <- Piv.table[2,2]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=1):",SCombination5,"\n")
```

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

```r
SCombination6 <- Piv.table[2,3]/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=2 and TRAF_CON_R=2):",SCombination6,"\n")
```

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

```r
#We can see the probabilities now
```

2.2) Classify the 24 accidents using these probabilities and a cutoff of 0.5.

```r
#for the cutoff = 0.5 for 24 records

Probability.of.injury <- rep(0,24)
for(i in 1:24){print(c(Acc.data$WEATHER_R[i],Acc.data$TRAF_CON_R[i]))}
```

```
## [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
```

```
if(Acc.data$WEATHER_R[i]=="1"&&Acc.data$TRAF_CON_R[i]=="0"){Probability.of.injury[i]=Combination1
} else if(Acc.data$WEATHER_R[i]=="1"&&Acc.data$TRAF_CON_R[i]=="1"){Probability.of.injury[i]=Combination2
} else if(Acc.data$WEATHER_R[i]=="1"&&Acc.data$TRAF_CON_R[i]=="2"){Probability.of.injury[i]=Combination3
} else if(Acc.data$WEATHER_R[i]=="2"&&Acc.data$TRAF_CON_R[i]=="0"){Probability.of.injury[i]=Combination4
} else if(Acc.data$WEATHER_R[i]=="2"&&Acc.data$TRAF_CON_R[i]=="1"){Probability.of.injury[i]=Combination5
} else if(Acc.data$WEATHER_R[i]=="2"&&Acc.data$TRAF_CON_R[i]=="2"){Probability.of.injury[i]=Combination6
```

```
Acc.data$probability.of.injury = Probability.of.injury
Acc.data$probability.of.prediction = ifelse(Acc.data$probability.of.injury>0.5, "yes","no")

head(Acc.data)
```

```
##   INJURY WEATHER_R TRAF_CON_R probability.of.injury probability.of.prediction
## 1    yes         1          0                     0                        no
## 2     no         2          0                     0                        no
## 3     no         2          1                     0                        no
## 4     no         1          1                     0                        no
## 5     no         1          0                     0                        no
## 6    yes         2          0                     0                        no
```

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

```
Injury.yes=Piv.table[3,2]/Piv.table2[1,2]
I=(Injury.yes*Piv.table[3,2])/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=1):", Injury.yes,"\n")
```

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

```
Injury.No=Piv.table[1,2]/Piv.table2[1,2]
I=(Injury.No*Piv.table[3,2])/Piv.table2[1,2]
cat("P(INJURY=Yes|WEATHER_R=1 and TRAF_CON_R=1):", Injury.No,"\n")
```

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

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?

```
Bayes_data <- naiveBayes(INJURY~TRAF_CON_R+WEATHER_R,data = Acc.data)

n.Acc.data <- predict(Bayes_data,newdata = Acc.data,type = "raw")
Acc.data$Naive.bayes.prediction.of.probabilities <- n.Acc.data[,2]

Bayes_data1 <- train(INJURY~TRAF_CON_R+WEATHER_R,data = Acc.data,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_R

## 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_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 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 Resample20: 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: 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_R, WEATHER_R

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

```
predict(Bayes_data1,newdata=Acc.data[,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
```

```
predict(Bayes_data1,newdata=Acc.data[,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
```

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

```r
accidents=Acc.data[c(-24)]

set.seed(1)
accidents.index=sample(row.names(accidents),0.6*nrow(accidents)[1])
validation.index=setdiff(row.names(accidents),accidents.index)

accidents.dataframe=accidents[accidents.index,]
validation.dataframe=accidents[validation.index,]

dim(accidents.dataframe)
```

```
## [1] 14  6
```

```r
dim(validation.dataframe)
```

```
## [1] 10  6
```

```r
normalised.values <- preProcess(accidents.dataframe[,],method=c("center","scale"))
```

```
## Warning in preProcess.default(accidents.dataframe[, ], method = c("center", :
## These variables have zero variances: probability.of.injury
```

```r
accidents.normalised <- predict(normalised.values,accidents.dataframe[, ])
validation.normalised.dataframe <- predict(normalised.values,validation.dataframe[, ])

levels(accidents.normalised)
```

```
## NULL
```

```r
class(accidents.normalised$INJURY)
```

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

```r
accidents.normalised$INJURY <- as.factor(accidents.normalised$INJURY)
```

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.

```r
Naive.bayes.model <- naiveBayes(INJURY~WEATHER_R+TRAF_CON_R, data = accidents.normalised)

Predictions.of.nb <- predict(Naive.bayes.model,newdata=validation.normalised.dataframe)

#Factors in validation dataset should match with training dataset.

validation.normalised.dataframe$INJURY <- factor(validation.normalised.dataframe$INJURY,levels = levels

#confusion matrix

confusionMatrix(Predictions.of.nb,validation.normalised.dataframe$INJURY)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no   3   5
##        yes  2   0
##
##                Accuracy : 0.3
##                  95% CI : (0.0667, 0.6525)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.9453
##
##                   Kappa : -0.4
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.600
##             Specificity : 0.000
##          Pos Pred Value : 0.375
##          Neg Pred Value : 0.000
##              Prevalence : 0.500
##          Detection Rate : 0.300
##    Detection Prevalence : 0.800
##       Balanced Accuracy : 0.300
##
##        'Positive' Class : no
##
```

```
#Overall error rate calculation

Overall.error <- 1 - sum(Predictions.of.nb==validation.normalised.dataframe$INJURY)/nrow(validation.norm

Overall.error
```

```
## [1] 0.7
```

#Summary of the above output

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 injury (INJURY = No).

As per the above data, there are 20721 cases with no injury and 21462 cases with injuries.

We now obtain a different dataframe with only variables as injury, weather and traffic.

Created a pivot table only with above variables. Also, calculated the bayes probabilities with all the combinations.

Using the cutoff of 0.5 for all the 24 records of accidents in the above variables with the given attributes of weather and traffic, computed the naive bayes conditional probability of injuries.

The manual predictions of the naive bayes are:

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

The predictions to the exact bayes models and naive bayes models are as follows:

[1] yes no no yes yes no no yes no no no yes yes yes [15] yes yes no no no no yes yes no no Levels: no yes

[1] yes no no yes yes no no yes no no no yes yes yes [15] yes yes no no no no yes yes no no Levels: no yes

We can observe that both the exact bayes and naive bayes have the same results and orders of ranking is consistent between both.

Further, we also split the data to training (60%) and validation (40%), to let the model perdict the future unseen accidents with the model.

These sets have different functions whereas training set is used to train the model with the data inputs based on the past accidents and validation set is used to validate the training set as reference to label the accidents in future cases.

After splitting the data, we have to normalise to avoid errors, normalising is nothing but building consistence within the data types such as numeric varibles or integers.

Here are the statistics of the model as per the output:

Accuracy : 0.3
95% CI : (0.0667, 0.6525) No Information Rate : 0.5
P-Value [Acc > NIR] : 0.9453

```
              Kappa : -0.4
```

Mcnemar's Test P-Value : 0.4497

```
        Sensitivity : 0.600
        Specificity : 0.000
     Pos Pred Value : 0.375
     Neg Pred Value : 0.000
         Prevalence : 0.500
     Detection Rate : 0.300
```

Detection Prevalence : 0.800
Balanced Accuracy : 0.300

```
   'Positive' Class : no
```

Accuracy : 0.3 which says that the 30% of the predictions are correct.

Sensitivity is 0.3 which is true positive rate.

specificity says that how much percent of the time can the model can identify the negative cases which is 0 in this solution.

As per the summary, we can say that the model is not performing well.