FML_Assign-3

MICHAEL BALLAMUDI

2023-10-30

Summary

-Noting that both of these classifications display "yes" at the same indices is the first and most crucial thing to do. This indicates that the observations' Ranking (= Ordering) is consistent.

-If the rank is equivalent, it means that both categories comprehend the data similarly and give equal weight to each factor. In this instance, judgements regarding the significance of the data points are consistently made by the models.

-To sum up, this assessment was predicated on a subset with just three characteristics. The model would normally be evaluated on a dataset as a whole in order to obtain an overall model performance and equivalency. The standard evaluation metrics, such as accuracy, precision, and recall, as well as F1-score, which offers a more comprehensive view of the model's performance, are used to better understand the classification performance of the model.

-We now divide all of our data into two sets: a training set (60%) and a validation set (40%). Following the analysis of the sets, we use the training data to train the model in order to use the information to identify future crashes (new or unseen data).

-Validation Set: This set is used to validate the data it includes, using a reference as the training set, so that we may know how effectively our model is trained when they get unknown data (new data). Given the training set, it categorizes the validation set.

-We normalize the data to put all of the data on the same line after partitioning the data frame. We operate on this normalized data to provide precise numbers that we utilize in our decision-making.

-It is crucial that the characteristics being compared be numbers or integers and have the same levels to prevent errors.

#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 goal here 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."

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?
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 12 records. Use all three variables in the pivot table as rows/columns.

- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
- Classify the 24 accidents using these probabilities and a cutoff of 0.5.
- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.
- 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?

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

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

We can access the information at any time because I've stored it in a data frame named Accident_data. Next, I created a dummy variable named Injury that indicates the extent of the injury. We presume there is some sort of harm if it is greater than one or two. We consider there to be no injury if it is less than zero.

1. One sort of variable we have is named Injury, and it has classifiers like yes or no. Since we just know that an accident was reported, INJURY=YES would be the expected accident. This is because there are more records with the notation "Injury=yes" than with the notation "No," indicating a higher likelihood of an accident.

2: Using WEATHER_R and TRAF_CON_{R} as two predictive parameters, we will pick the top 24 entries in the collection. The "Sub_accident_data" variable contains the dataset. We may organize the information into a pivot table and arrange them based on traffic volume and weather in order to better comprehend the data. The following is the pivot table:

```
              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
```

#Bayes Theorem : P(A/B) = (P(B/A)P(A))/P(B) where P(A),P(B) are events and P(B) not equal to 0.

We could determine the probability that one of the six injury predictors would be positive. The following values are what we obtained for various combinations.

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

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

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

The other 3 combinations pf probability of injury=yes is 0.

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

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

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

2.2: Since the cut-off value in this example is set to 0.5, everything above 0.5 is seen as "yes," while anything below 0.5 is regarded as "no." In order to compare the anticipated injury with the actual injury, we have also created a new characteristic to hold the predicted injury.

2.3: Now let's examine the injury's naive Bayes conditional probability. The values we've assigned it are as follows: WEATHER_R: 1 TRAF_Con_R: 1

-If INJURY = YES, the probability is 0.

-If INJURY - NO , the probability is 1.

2.4: The following are the exact Bayes classification and predictions from the Naive Bayes model: [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

Each record is categorized as "yes" or "no".

-Noting that both of these classifications display "yes" at the same indices is the first and most crucial thing to do. This indicates that the observations' Ranking (= Ordering) is consistent.

-If the rank is equivalent, it means that both categories comprehend the data similarly and give equal weight to each factor. In this instance, judgements regarding the significance of the data points are consistently made by the models.

-To sum up, this assessment was predicated on a subset with just three characteristics. The model would normally be evaluated on a dataset as a whole in order to obtain an overall model performance and equivalency. The standard evaluation metrics, such as accuracy, precision, and recall, as well as F1-score, which offers a more comprehensive view of the model's performance, are used to better understand the classification performance of the model.

-We now divide all of our data into two sets: a training set (60%) and a validation set (40%). Following the analysis of the sets, we use the training data to train the model in order to use the information to identify future crashes (new or unseen data).

-Validation Set: This set is used to validate the data it includes, using a reference as the training set, so that we may know how effectively our model is trained when they get unknown data (new data). Given the training set, it categorizes the validation set.

-We normalize the data to put all of the data on the same line after partitioning the data frame. We operate on this normalized data to provide precise numbers that we utilize in our decision-making.

-It is crucial that the characteristics being compared be numbers or integers and have the same levels to prevent errors.

```r
#Load the library
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
#Assign the dataset
Accidents <- read.csv("C:/Users/micha/OneDrive/Desktop/FML/accidentsFull.csv")
dim(Accidents)
```

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

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

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

```r
t(t(names(Accidents)))
```

```
##       [,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"
```

```
#Create Pivot table
Accidents_Pivot <- Accidents[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
Accidents_Pivot
```

```
##    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
```

```
pivot_table_a <- ftable(Accidents_Pivot)
pivot_table_a
```

```
##                  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
```

```
pivot_table_b <- ftable(Accidents_Pivot[,-1])
pivot_table_b
```

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

#2.1:

```
#Exact Bayes
#Conditional Probability of an injury(INJURY = YES)
cond_prob_1 = pivot_table_a[3,1]/pivot_table_b[1,1]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):", cond_prob_1 , "\n")
```

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

```
cond_prob_2 = pivot_table_a[3,2]/pivot_table_b[1,2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):",cond_prob_2, "\n")
```

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

```
cond_prob_3 = pivot_table_a[3,3]/pivot_table_b[1,3]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):", cond_prob_3, "\n")
```

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

```
cond_prob_4 = pivot_table_a[4,1]/pivot_table_b[2,1]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):", cond_prob_4, "\n")
```

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

```
cond_prob_5 = pivot_table_a[4,2]/pivot_table_b[2,2]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):", cond_prob_5, "\n")
```

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

```
cond_prob_6 = pivot_table_a[4,3]/pivot_table_b[2,3]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):", cond_prob_6, "\n")
```

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

```
#Conditional Probability of an injury(INJURY=NO)

var_a = pivot_table_a[1,1]/pivot_table_b[1,1]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0):", var_a, "\n")
```

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

```
var_b = pivot_table_a[1,2]/pivot_table_b[1,2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", var_b, "\n")
```

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

```
var_c = pivot_table_a[1,3]/pivot_table_b[1,3]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2):", var_c, "\n")
```

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

```
var_d = pivot_table_a[2,1]/pivot_table_b[2,1]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0):", var_d, "\n")
```

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

```
var_e = pivot_table_a[2,2]/pivot_table_b[2,2]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1):", var_e, "\n")
```

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

```
var_f = pivot_table_a[2,3]/pivot_table_b[2,3]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2):", var_f, "\n")
```

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

```
#Conditional probability of all possible combinations.
```

#2.2:

```
#cutoff for 24 records is 0.5.
# You can categorize the 24 accidents using the conditional probabilities, assuming you have already co
# Suppose you have these 24 records in a dataframe called "newdata."

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

  if(Accidents_Pivot$WEATHER_R[i] == "1" && Accidents_Pivot$TRAF_CON_R[i] == "0"){
    prob_injury[i] = cond_prob_1

  } else if (Accidents_Pivot$WEATHER_R[i] == "1" && Accidents_Pivot$TRAF_CON_R[i] == "1"){
    prob_injury[i] =cond_prob_2

  } else if (Accidents_Pivot$WEATHER_R[i] == "1" && Accidents_Pivot$TRAF_CON_R[i] == "2"){
    prob_injury[i] = cond_prob_3

  }
  else if (Accidents_Pivot$WEATHER_R[i] == "2" && Accidents_Pivot$TRAF_CON_R[i] == "0"){
    prob_injury[i] = cond_prob_4

  } else if (Accidents_Pivot$WEATHER_R[i] == "2" && Accidents_Pivot$TRAF_CON_R[i] == "1"){
    prob_injury[i] = cond_prob_5

  }
  else if(Accidents_Pivot$WEATHER_R[i] == "2" && Accidents_Pivot$TRAF_CON_R[i] == "2"){
    prob_injury[i] = cond_prob_6
  }

}
```

```
## [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

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

head(Accidents_Pivot)
```

```
##   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
```

#2.3: #Calculate by hand the conditional likelihood of an injury using naive Bayes WEATHER_R = 1 and TRAF_CON_R = 1.

```
Injury_Yes = pivot_table_a[3,2]/pivot_table_b[1,2]
InjuryYes = (Injury_Yes * pivot_table_a[3, 2]) / pivot_table_b[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
```

```r
Injury_No = pivot_table_a[1,2]/pivot_table_b[1,2]
InjuryNo = (Injury_No  * pivot_table_a[3, 2]) / pivot_table_b[1, 2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", Injury_No , "\n")
```

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

#2.4:

```r
NB <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = Accidents_Pivot)

NB_Accidents <- predict(NB, newdata = Accidents_Pivot,type = "raw")
Accidents_Pivot$nbpred.prob <- NB_Accidents[,2]
```

#3.1: #Now, let's go back to the complete dataset. Divide the data into 40% for validation and 60% for training. Using all of the training data and the pertinent predictors, run a naïve Bayes classifier (with INJURY as the response). Keep in mind that each prediction is categorized. Display the matrix of bewilderment.What is the validation set's total error?

```r
Accident_DS = Accidents[c(-24)]

set.seed(1)
Training = sample(row.names(Accident_DS), 0.6*nrow(Accident_DS)[1])
Validation = setdiff(row.names(Accident_DS), Training)


Training_df = Accident_DS[Training,]
Validation_df= Accident_DS[Validation,]

dim(Training_df)
```

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

```r
dim(Validation_df)
```

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

```r
CM <- preProcess(Training_df[,], method = c("center", "scale"))
CM_Training <- predict(CM, Training_df[, ])
CM_Validation <- predict(CM, Validation_df[, ])

levels(CM_Training)
```

```
## NULL
```

```r
class(CM_Training$INJURY)
```

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

```
CM_Training$INJURY <- as.factor(CM_Training$INJURY)

class(CM_Training$INJURY)
```

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

#3.2:

```
NBT <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = CM_Training)

Pre <- predict(NBT, newdata = CM_Validation)

#Make that the training dataset's factor levels correspond with the validation dataset's.
CM_Validation$INJURY <- factor(CM_Validation$INJURY, levels = levels(CM_Training$INJURY))

# Confusion matrix
confusionMatrix(Pre, CM_Validation$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 overall error rate
error_rate <- 1 - sum(Pre == CM_Validation$INJURY) / nrow(CM_Validation)
error_rate
```

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