Assignment_3

Balamanoj Reddy Kommareddy

2023-10-09

#Summary

The data has been stored in the data frame ‘Accident_data’ for convenient access and utilization. Subsequently, a dummy variable named “Injury” has been generated to categorize the maximum severity of injury, Max_SEV_IR is either 1 or 2, indicating some degree of injury (Injury = Yes). Conversely, if Max_SEV _IR equals 0, it implies “No injury”.

1). We have an attribute checked Injury that’s a type of variable with classifiers like yes or no. We just know that an accident was reported, so the predicted accident would be Injury=Yes. That’s because the number of records that say “Injury=yes” is higher than records that say “No”, which means it’s more likely to be seen as an accident.

2). We will be selecting the first 24 records of the dataset and taking into account two predicting factors, WEATHER_R and TRAF_CON_R. The dataset has been stored in a variable name “Sub_accident_data”. To gain a better understanding of the data, we created a pivot table of the records, which will be sorted according to weather and traffic levels.

##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.

2.1). We were able to figure out how likely it is that one of the six injury predictors would be a yes. For different combinations, we got the following values.

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). In this example, we’ve set the cut-off value to 0.5, meaning anything above 0.5 is considered “yes” and anything below 0.5 is “no”. We’ve also added a new attribute to store the injury predicted, so we can compare the actual injury with the predicted injury.

2.3). Let’s see what the naive Bayes conditional probability is for injury looks like. We’ve given it the following values: WEATHER_R: 1 TRAF_CON_R: 1

-If INJURY = YES, the probability is 0.

-If INJURY - NO , the probability is 1.

2.4). Naive Bayes model predictions and exact Bayes classification.

-The first thing to note is that both of these classifications show “yes” at the same indexes. This means that the Ranking(= Ordering) of observations is the same.

-If the rank is equal, then it indicates that both classifications assign the same importance to all factors and have a similar understanding of the data. In this case, models are consistently making decisions about the importance of the data points.

-In conclusion, this evaluation was based on a subset containing only three attributes. In order to obtain an overall model performance and equivalence, the model would typically be evaluated on a dataset as a whole and the standard evaluation metrics are used to gain a better understanding of the classification performance of the model, including accuracy, precision, and recheck, as well as F1-score, which provides a more comprehensive view of the model's performance.

3.The next thing here is we split all our data into training set (60%) and validation set (40%). After analyzing the sets, we train the model with the training data used to identify future crashes (new or unseen data) with the information provided.

-Validation Set: This set is used to validate the data it contains, using a reference as the training set, so that we can know how well our model is trained when they receive unknown data (new data). It classifies the validation set given the training set.

-After partitioning the data frame, we normalize the data so that all the data is on the same line. We perform operations on this normalized data to obtain accurate values that we use to make decisions.

3.1) what are the results of confusion matrix for the validation data.

confusion matrix results:

Miscalculations: 8052

Accuracy : 0.5228

Sensitivity : 0.15635

Specificity : 0.87083

3.2) What is the overall error of the validation set.

overall error of the validation set is 0.4774209

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

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

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

```
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
```

```
library(klaR)
```

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

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
#Loading the data set and assigning it to Accidents_Data variable.
Accidents_Data <- read.csv("accidentsFull.csv")
dim(Accidents_Data)
```

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

#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.

```
Accidents_Data$INJURY = ifelse(Accidents_Data$MAX_SEV_IR %in% c(1,2),"yes","no") # as yes is greater then no
# Summary of the INJURY variable
table(Accidents_Data$INJURY)
```

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

```
#Summary of other variables in the dataset
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.

```
#Creating the pivot tables
# Subset of data
sub_Accidents_Data <- Accidents_Data[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
sub_Accidents_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
```

```
# Creating a pivot table for the subset with all columns
pivot_table1 <- ftable(sub_Accidents_Data)
pivot_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
```

```
# Creating a pivot table for the subset without the "INJURY" column
pivot_table2 <- ftable(sub_Accidents_Data[,-1])
pivot_table2
```

```
##           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.

```
#bayes
#INJURY = YES
part_1 = pivot_table1[3,1]/pivot_table2[1,1]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):", part_1, "\n")
```

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

```
part_2 = pivot_table1[3,2]/pivot_table2[1,2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", part_2, "\n")
```

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

```
part_3 = pivot_table1[3,3]/pivot_table2[1,3]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):", part_3, "\n")
```

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

```
part_4 = pivot_table1[4,1]/pivot_table2[2,1]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):", part_4, "\n")
```

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

```
part_5 = pivot_table1[4,2]/pivot_table2[2,2]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):", part_5, "\n")
```

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

```
part_6 = pivot_table1[4,3]/pivot_table2[2,3]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):", part_6, "\n")
```

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

```
#These probabilities are based on the data which we have, and used the counts from your pivot tables to calculate the condit
ional probabilities. These conditional probabilities can be useful for making Bayesian inferences or predictions based on th
e available data.

#Now we check the condition whether Injury = no

check_1 = pivot_table1[1,1]/pivot_table2[1,1]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0):", check_1, "\n")
```

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

```
check_2 = pivot_table1[1,2]/pivot_table2[1,2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", check_2, "\n")
```

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

```
check_3 = pivot_table1[1,3]/pivot_table2[1,3]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2):", check_3, "\n")
```

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

```
check_4 = pivot_table1[2,1]/pivot_table2[2,1]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0):", check_4, "\n")
```

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

```
check_5 = pivot_table1[2,2]/pivot_table2[2,2]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1):", check_5, "\n")
```

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

```
check_6 = pivot_table1[2,3]/pivot_table2[2,3]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2):", check_6, "\n")
```

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

```
#These probabilities provide insights into the likelihood of "INJURY = no" under different conditions of "WEATHER_R" and "TR
AF_CON_R." These calculations, together with the previous calculations for "INJURY = Yes," can be used for making probabilis
tic inferences or predictions based on our data.

#Now probability of the total occurences.
```

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

```
#cutoff is 0.5 and for 24 records
# Let's say you have a dataframe named 'new_data' containing these 24 records.
# In this code, I assumed that "WEATHER_R" and "TRAF_CON_R" columns contain numeric values.

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

  if(sub_Accidents_Data$WEATHER_R[i] == "1" && sub_Accidents_Data$TRAF_CON_R[i] == "0"){
    prob_injury[i] = part_1

  } else if (sub_Accidents_Data$WEATHER_R[i] == "1" && sub_Accidents_Data$TRAF_CON_R[i] == "1"){
    prob_injury[i] = part_2

  } else if (sub_Accidents_Data$WEATHER_R[i] == "1" && sub_Accidents_Data$TRAF_CON_R[i] == "2"){
    prob_injury[i] = part_3

  }
  else if (sub_Accidents_Data$WEATHER_R[i] == "2" && sub_Accidents_Data$TRAF_CON_R[i] == "0"){
    prob_injury[i] = part_4

  } else if (sub_Accidents_Data$WEATHER_R[i] == "2" && sub_Accidents_Data$TRAF_CON_R[i] == "1"){
    prob_injury[i] = part_5

  }
  else if(sub_Accidents_Data$WEATHER_R[i] == "2" && sub_Accidents_Data$TRAF_CON_R[i] == "2"){
    prob_injury[i] = part_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
# If the probability of "INJURY = Yes" is greater than 0.5, it's classified as "yes"; otherwise, it's classified as "no."

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

head(sub_Accidents_Data)
```

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

```
IY = pivot_table1[3,2]/pivot_table2[1,2]
I = (IY * pivot_table1[3, 2]) / pivot_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 = pivot_table1[1,2]/pivot_table2[1,2]
N = (IY * pivot_table1[3, 2]) / pivot_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 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.

```
new_b <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                data = sub_Accidents_Data)

new_Accidents_Data <- predict(new_b, newdata = sub_Accidents_Data,type = "raw")
sub_Accidents_Data$nbpred.prob <- new_Accidents_Data[,2]


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

```
## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R, WEATHER_R
```

```
## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   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.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample14: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   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.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   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.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = F
ALSE, fL = param$fL, ...) :
##   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.
```

```
predict(new_c, newdata = sub_Accidents_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 yes no  no  no
## [20] no  yes yes no  no
## Levels: no yes
```

```
predict(new_c, newdata = sub_Accidents_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 yes no  no  no
## [20] no  yes yes no  no
## Levels: no yes
```

```
#obtaining class predictions or predicted probabilities for "INJURY" based on the "TRAF_CON_R" and "WEATHER_R" features.
```

#3. 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.What is the overall error of the validation set?

```
#creating a new dataframe accident by excluding the 24th column from the Accidents_Data.
accident = Accidents_Data[c(-24)]

set.seed(1)
acc.index = sample(row.names(accident), 0.6*nrow(accident)[1])
valid.index = setdiff(row.names(accident), acc.index)

#creating two dataframes, acc.df for training and valid.df for validation
acc.df = accident[acc.index,]
valid.df= accident[valid.index,]

dim(acc.df)
```

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

```
dim(valid.df)
```

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

```
norm.values <- preProcess(acc.df[,], method = c("center", "scale"))
acc.norm.df <- predict(norm.values, acc.df[, ])
valid.norm.df <- predict(norm.values, valid.df[, ])

levels(acc.norm.df)
```

```
## NULL
```

```
class(acc.norm.df$INJURY)
```

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

```
acc.norm.df$INJURY <- as.factor(acc.norm.df$INJURY)

class(acc.norm.df$INJURY)
```

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

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

```
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = acc.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(acc.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
##
```

```
# Calculating the overall error rate
error_rate <- 1 - sum(predictions == valid.norm.df$INJURY) / nrow(valid.norm.df)
error_rate
```

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