assignment\_3

phani varshitha

2023-10-14

#summary Summary: Created a dummy variable named “INJURY” based on the “MAX\_SEV\_IR>0” variable. If “MAX\_SEV\_IR” is 1 or 2, If no further information is available for a recently reported accident, the initial prediction should be based on the overall proportion of accidents resulting in injury (value “Yes” in the “INJURY” variable) in the dataset. Utilized the first 24 records in the dataset and focus on “INJURY,” “WEATHER\_R,” and “TRAF\_CON\_R.” Create a pivot table that examines “INJURY” concerning the two predictors for these 24 records.Computed the exact Bayes conditional probabilities of an injury (“INJURY” = Yes) for each of the six possible combinations of the predictors.Classifed the 24 accidents using these probabilities with a cutoff of 0.5. Computed the Naive Bayes conditional probability of an injury given “WEATHER\_R” = 1 and “TRAF\_CON\_R” = 1 manually.as we got the same values for the manual and navie bayes as 0. Runs a Naive Bayes classifier on the 24 records and two predictors. model output to obtain probabilities and classifications for all 24 records. Split the dataset into training (60%) and validation (40%) sets. Apply a Naive Bayes classifier on the complete training set with the relevant predictors, including “INJURY” as the response. All predictors in this case are categorical. as required we got the confusion matrix as below predicted no yes no 11 7 yes 0 0 and got the over\_error rate is 0.3888889. By performing these tasks, we can gain insights into the predictive capabilities of the Naive Bayes classifier for accident injury prediction.

#loading the required libraries and reading the accident\_data set

library(e1071)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

accidents\_data = read.csv("C:/Users/varshitha/Downloads/accidentsFull.csv")  
accidents\_data$INJURY = ifelse(accidents\_data$MAX\_SEV\_IR>0,"yes","no")

# Convert variables to factor  
for (i in c(1:dim(accidents\_data)[2])){  
 accidents\_data[,i] <- as.factor(accidents\_data[,i])  
}  
head(accidents\_data,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

#selecting the first 24 rows of the data and selecting the required varaibles

data2 = accidents\_data[1:24,c("INJURY","WEATHER\_R","TRAF\_CON\_R")]  
head(data2)

## 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

#making the data into a table

pivot\_Table1 <- ftable(data2)  
pivot\_Table2 <- ftable(data2[,-1]) # print table only for conditions  
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

pivot\_Table2

## TRAF\_CON\_R 0 1 2  
## WEATHER\_R   
## 1 9 1 1  
## 2 11 1 1

#presenting the possibiltes of 6 conditional probabilties

# Injury = yes  
y1 = pivot\_Table1[3,1] / pivot\_Table2[1,1] # Injury, Weather=1 and Traf=0  
y2 = pivot\_Table1[4,1] / pivot\_Table2[2,1] # Injury, Weather=2, Traf=0  
y3 = pivot\_Table1[3,2] / pivot\_Table2[1,2] # Injury, W=1, T=1  
y4 = pivot\_Table1[4,2] / pivot\_Table2[2,2] # I, W=2,T=1  
y5 = pivot\_Table1[3,3] / pivot\_Table2[1,3] # I, W=1,T=2  
y6 = pivot\_Table1[4,3]/ pivot\_Table2[2,3] #I,W=2,T=2  
  
# Injury = no  
n1 = pivot\_Table1[1,1] / pivot\_Table2[1,1] # Weather=1 and Traf=0  
n2 = pivot\_Table1[2,1] / pivot\_Table2[2,1] # Weather=2, Traf=0  
n3 = pivot\_Table1[1,2] / pivot\_Table2[1,2] # W=1, T=1  
n4 = pivot\_Table1[2,2] / pivot\_Table2[2,2] # W=2,T=1  
n5 = pivot\_Table1[1,3] / pivot\_Table2[1,3] # W=1,T=2  
n6 = pivot\_Table1[2,3] / pivot\_Table2[2,3] # W=2,T=2  
print(c(y1,y2,y3,y4,y5,y6))

## [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

#computing the values for 24 accidents applying the conditon of cutoff probability = 0.5

prob.inj <- rep(0,24)  
  
for (i in 1:24) {  
 print(c(data2$WEATHER\_R[i],data2$TRAF\_CON\_R[i]))  
 if (data2$WEATHER\_R[i] == "1") {  
 if (data2$TRAF\_CON\_R[i]=="0"){  
 prob.inj[i] = y1  
 }  
 else if (data2$TRAF\_CON\_R[i]=="1") {  
 prob.inj[i] = y3  
 }  
 else if (data2$TRAF\_CON\_R[i]=="2") {  
 prob.inj[i] = y5  
 }  
 }  
 else {  
 if (data2$TRAF\_CON\_R[i]=="0"){  
 prob.inj[i] = y2  
 }  
 else if (data2$TRAF\_CON\_R[i]=="1") {  
 prob.inj[i] = y4  
 }  
 else if (data2$TRAF\_CON\_R[i]=="2") {  
 prob.inj[i] = y6  
 }  
 }  
 }

## [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

data2$prob.inj <- prob.inj  
  
data2$pred.prob <- ifelse(data2$prob.inj>0.5, "yes", "no")

#computing the manually naive Bayes conditional probability of an injury given WEATHER\_R = 1 and TRAF\_CON\_R = 1.

p(I=Y|W=1,T=1) = P(I=Y | W=1,T=1)/ P(W=1,T=1)

=(0/24)/(1/24) = 0/1 =0

#2. 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 = data2)  
  
nbt <- predict(nb, newdata = data2,type = "raw")  
data2$nbpred.prob <- nbt[,2] # Transfer the "Yes" nb prediction

library(klaR)

## Loading required package: MASS

nb2 <- train(INJURY ~ TRAF\_CON\_R + WEATHER\_R,   
 data = data2, method = "nb")

## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample03: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample05: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample06: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample11: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample12: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample14: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample15: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample17: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample18: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample19: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample20: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1

## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :   
## Zero variances for at least one class in variables: TRAF\_CON\_R1, TRAF\_CON\_R2

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

## Warning in train.default(x, y, weights = w, ...): missing values found in  
## aggregated results

predict(nb2, newdata = data2[,c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")])

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

predict(nb2, newdata = data2[,c("INJURY", "WEATHER\_R", "TRAF\_CON\_R")],  
 type = "raw")

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

#Partitioning the data into training and validation sets

set.seed(2808) # For reproducibility  
train\_indices <- createDataPartition(data2,p=0.60,list = FALSE)

## Warning in createDataPartition(data2, p = 0.6, list = FALSE): Some classes have  
## no records ( ) and these will be ignored

## Warning in createDataPartition(data2, p = 0.6, list = FALSE): Some classes have  
## a single record ( ) and these will be selected for the sample

train\_data <- data2[train\_indices, ]  
validation\_data <- data2[-train\_indices, ]

#Running Naive Bayes classifier on the complete training set

nb\_model <- naiveBayes(INJURY ~ WEATHER\_R + TRAF\_CON\_R, data = train\_data)  
# Predict on the validation set  
predicted <- predict(nb\_model, newdata = validation\_data)

# Computing the confusion matrix

conf\_matrix <- table(predicted, validation\_data$INJURY)

#Computing the overall error of the validation set

overall\_error <- mean(predicted != validation\_data$INJURY)

#Print the confusion matrix and overall error  
print(conf\_matrix)

##   
## predicted no yes  
## no 11 7  
## yes 0 0

print(overall\_error)

## [1] 0.3888889