FML_Assignment3

Atshaya Suresh

2023-10-13

SUMMARY

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?

Ans) Using the information from the dataset, if an accident has just been reported and no further information is available, the prediction is "INJURY = Yes". The reason behind this is that, P(INJURY = "Yes") = 0.50878 and P(INJURY = "No") = 0.49121. Since Probability of INJURY = "Yes" is greater than that of INJURY = "No", it is safe to predict the same.

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.

Ans) The Pivot table is as follows: Refer to Line 151 and 152 (Variables: dt1 and dt2)

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

Ans) Below are the probabilities: (Note: INJURY = Yes is I, WEATHER_R is W and TRAF_CON_R is T) P(I|W=1,T=0)= 0.666667 P(I|W=1,T=1)= 0 P(I|W=1,T=2)= 0 P(I|W=2,T=0)= 0.1818182 P(I|W=2,T=1)= 0 P(I|W=2,T=2)= 1

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

Ans) Refer to Line 154 for the classifications of the 24 accidents with a cut off of 0.5 from using the probabilities above.

Variable name: accidents24$prob_injury

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

Ans) The answer is 0.

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?

Ans) Refer to Line185 to Line224

Dataset for reference: accidents24_sorted

The probabilities of naives Bayes and exact Bayes are not equivalent and classifications possibly can not be equivalent for a cut off value of 0.5. However, there is a very high correlation in terms of their ranking. Except for 2 records, (3rd observation and 24th observation), after sorting the records in ascending order as per the prob_injury(exact Bayes value), the ranks of the observations are very similar or more or less equivalent.

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

Ans) The data is partitioned into 2 copies of 2 sets (a)train.df(60%) and valid.df(40%) (b)training_set(60%) and validation_set(40%)

The purpose of choosing 2 sets of training and validation sets is to differentiate how the dimensions taken into account for building a model influences the model itself.

Method 1: We are using only a subset of the features which are, "INJURY", "HOUR_I_R", "ALIGN_I", "WRK_ZONE", "WKDY_I_R","INT_HWY", "LGTCON_I_R", "PROFIL_I_R", "SPD_LIM", "SUR_COND","TRAF_CON_R", "TRAF_WAY", "WEATHER_R". which is assumed to be relevant predictors for the target variable.

Method 2: We are using all the features except MAX_SEV_IR (which is classified as Yes/No in the INJURY column) and SPD_LIM (which is a numeric variable), since it is mentioned that, the predictors are categorical.

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.

Ans) Method 1: Confusion Matrix

```
Method_1 <- matrix(c(5574,6972,4829,7934),
             nrow = 2, byrow = TRUE)
colnames(Method_1)<-c("Actual Yes", "Actual No")
rownames(Method_1)<-c("Predicted Yes","Predicted No")
tab<-as.table(Method_1)
Method_1
```

```
##               Actual Yes Actual No
## Predicted Yes       5574      6972
## Predicted No        4829      7934
```

Method 2: Confusion Matrix

```
Method_2 <- matrix(c(9030,0,0,9310),
             nrow = 2, byrow = TRUE)
colnames(Method_2)<-c("Actual Yes", "Actual No")
rownames(Method_2)<-c("Predicted Yes","Predicted No")
tab<-as.table(Method_2)
Method_2
```

```
##               Actual Yes Actual No
## Predicted Yes       9030         0
## Predicted No           0      9310
```

3.2 What is the overall error of the validation set?

Ans) The overall errors are: Method 1: 0.4663 Method 2: 0

```
options(repos = "https://cran.stat.ucla.edu/")
```

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).

Loading the necessary Packages and Libraries

```r
install.packages("e1071")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'

## Warning: package 'e1071' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```r
install.packages("caret")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'

## Warning: package 'caret' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```r
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'

## Warning: package 'dplyr' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

Importing the data

```
accidents <- read.csv("accidentsFull.csv")
```

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."

```
accidents$INJURY <-  ifelse(accidents$MAX_SEV_IR>0, "yes", "no")
table(accidents$INJURY)
```

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

---

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?

```
p_yes <- mean(accidents$INJURY == "yes")
p_no <- mean(accidents$INJURY == "no")
p_yes
```

```
## [1] 0.5087832
```

```
p_no
```

```
## [1] 0.4912168
```

Converting all the features except SPD_LIM to factor

```
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
```

```
accidents <- accidents %>%
  mutate_at(vars(-SPD_LIM), as.factor)
```

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.

For getting the first 24 records from the Dataframe

```
accidents24 <- accidents[1:24, c("INJURY", "WEATHER_R", "TRAF_CON_R")]
```

For Generating the Pivot Table

```
dt1 <- ftable(accidents24)
dt2 <- ftable(accidents24[,-1])
dt1
```

```
##                   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
```

```
dt2
```

```
##            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.

For Bayes Conditional Probability (Consider only the last 6 probabilities)

```
pn10 <- dt1[1,1]/dt2[1,1]
pn11 <- dt1[1,2]/dt2[1,2]
pn12 <- dt1[1,3]/dt2[1,3]
pn20 <- dt1[2,1]/dt2[2,1]
pn21 <- dt1[2,2]/dt2[2,2]
pn22 <- dt1[2,3]/dt2[2,3]
py10 <- dt1[3,1]/dt2[1,1]
py11 <- dt1[3,2]/dt2[1,2]
py12 <- dt1[3,3]/dt2[1,3]
py20 <- dt1[4,1]/dt2[2,1]
py21 <- dt1[4,2]/dt2[2,2]
py22 <- dt1[4,3]/dt2[2,3]
##P(Injury = No| Weather = 1, Traffic = 0)
pn10
```

```
## [1] 0.3333333
```

```
##P(Injury = No| Weather = 1, Traffic = 1)
pn11
```

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

```
##P(Injury = No| Weather = 1, Traffic = 2)
pn12
```

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

```
##P(Injury = No| Weather = 2, Traffic = 0)
pn20
```

```
## [1] 0.8181818
```

```
##P(Injury = No| Weather = 2, Traffic = 1)
pn21
```

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

```
##P(Injury = No| Weather = 2, Traffic = 2)
pn22
```

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

```
##P(Injury = Yes| Weather = 1, Traffic = 0)
py10
```

```
## [1] 0.6666667
```

```
##P(Injury = Yes| Weather = 1, Traffic = 1)
py11
```

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

```
##P(Injury = Yes| Weather = 1, Traffic = 2)
py12
```

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

```
##P(Injury = Yes| Weather = 2, Traffic = 0)
py20
```

```
## [1] 0.1818182
```

```
##P(Injury = Yes| Weather = 2, Traffic = 1)
py21
```

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

```
##P(Injury = Yes| Weather = 2, Traffic = 2)
py22
```

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

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

```
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library(dplyr)
accidents24 <- accidents24 %>%
  mutate(prob_injury = case_when(
    WEATHER_R == 1 & TRAF_CON_R == 0 ~ py10,
    WEATHER_R == 1 & TRAF_CON_R == 1 ~ py11,
    WEATHER_R == 1 & TRAF_CON_R == 2 ~ py12,
    WEATHER_R == 2 & TRAF_CON_R == 0 ~ py20,
    WEATHER_R == 2 & TRAF_CON_R == 1 ~ py21,
    WEATHER_R == 2 & TRAF_CON_R == 2 ~ py22,
    TRUE ~ NA_real_  # default case if none of the above conditions are met
  ))
accidents24 <- accidents24 %>%
  mutate(classification = if_else(prob_injury > 0.5, "Yes", "No"))
accidents24
```

```
##    INJURY WEATHER_R TRAF_CON_R prob_injury classification
## 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
## 7      no         2          0   0.1818182             No
## 8     yes         1          0   0.6666667            Yes
## 9      no         2          0   0.1818182             No
## 10     no         2          0   0.1818182             No
## 11     no         2          0   0.1818182             No
## 12     no         1          2   0.0000000             No
## 13    yes         1          0   0.6666667            Yes
## 14     no         1          0   0.6666667            Yes
## 15    yes         1          0   0.6666667            Yes
## 16    yes         1          0   0.6666667            Yes
## 17     no         2          0   0.1818182             No
## 18     no         2          0   0.1818182             No
## 19     no         2          0   0.1818182             No
## 20    no         2          0   0.1818182            No
## 21   yes         1          0   0.6666667           Yes
## 22    no         1          0   0.6666667           Yes
## 23   yes         2          2   1.0000000           Yes
## 24   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.

Calculating Naive Bayes manually from the values: P(Injury=Yes|WEATHER_R=1,TRAF_CON_R=1)

```
nbpy11 = ((6/9)*(0/9)*(9/24))/(((6/9)*(0/9)*(9/24))+((5/15)*(2/15)*(15/24)))
nbpy11
```

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

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?

```
install.packages("e1071")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'

## Warning: package 'e1071' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```
library(e1071)
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = accidents24)
nbt <- predict(nb, newdata = accidents24,type = "raw")
nbt
```

```
##              no         yes
##  [1,] 0.4285714 0.571428571
##  [2,] 0.7500000 0.250000000
##  [3,] 0.9977551 0.002244949
##  [4,] 0.9910803 0.008919722
##  [5,] 0.4285714 0.571428571
##  [6,] 0.7500000 0.250000000
##  [7,] 0.7500000 0.250000000
##  [8,] 0.4285714 0.571428571
##  [9,] 0.7500000 0.250000000
## [10,] 0.7500000 0.250000000
## [11,] 0.7500000 0.250000000
## [12,] 0.3333333 0.666666667
## [13,] 0.4285714 0.571428571
## [14,] 0.4285714 0.571428571
## [15,] 0.4285714 0.571428571
## [16,] 0.4285714 0.571428571
## [17,] 0.7500000 0.250000000
## [18,] 0.7500000 0.250000000
## [19,] 0.7500000 0.250000000
## [20,] 0.7500000 0.250000000
## [21,] 0.4285714 0.571428571
## [22,] 0.4285714 0.571428571
## [23,] 0.6666667 0.333333333
## [24,] 0.7500000 0.250000000
```

```
accidents24$nbpred.prob <- nbt[,2] # Transfer the "Yes" nb prediction
```

Another method: Let us use Caret

```
install.packages("caret")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'
```

```
## Warning: package 'caret' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```
library(caret)
```

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

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

```
nb2 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = accidents24, 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_R1, TRAF_CON_R2
## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## 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_R1

## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2

## 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_R1, TRAF_CON_R2

## 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_R1

## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## 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_R1, TRAF_CON_R2

## 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_R1, TRAF_CON_R2

## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   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.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2

## 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_R1, TRAF_CON_R2

## 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_R1, TRAF_CON_R2

## 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_R1

## 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_R1, TRAF_CON_R2

## 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_R1

## 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_R1

## Warning: model fit failed for Resample18: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   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.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## 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_R1, TRAF_CON_R2

## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   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.defaul
##   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.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.defaul
##   Zero variances for at least one class in variables: TRAF_CON_R1

## 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 = accidents24[,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 = accidents24[,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
```

```
#For confusion matrix
# Predictions
predicted_values <- predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])

# Confusion Matrix
conf_mat <- confusionMatrix(predicted_values, accidents24$INJURY)

# Display the confusion matrix
conf_mat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no  15   9
##        yes  0   0
##
##                Accuracy : 0.625
##                  95% CI : (0.4059, 0.812)
##     No Information Rate : 0.625
##     P-Value [Acc > NIR] : 0.589845
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 0.007661
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.625
##          Neg Pred Value :   NaN
##              Prevalence : 0.625
##          Detection Rate : 0.625
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : no
##
```

From the below results: Except for the 3rd observation and 24th observation, the rank ordering is same.

```
library(dplyr)
accidents24_sorted <- accidents24 %>% arrange(prob_injury)
accidents24_sorted
```

```
##    INJURY WEATHER_R TRAF_CON_R prob_injury classification nbpred.prob
## 1      no         2          1   0.0000000             No 0.002244949
## 2      no         1          1   0.0000000             No 0.008919722
## 3      no         1          2   0.0000000             No 0.666666667
## 4      no         2          0   0.1818182             No 0.250000000
## 5     yes         2          0   0.1818182             No 0.250000000
## 6      no         2          0   0.1818182             No 0.250000000
## 7      no         2          0   0.1818182             No 0.250000000
## 8      no         2          0   0.1818182             No 0.250000000
## 9      no         2          0   0.1818182             No 0.250000000
## 10     no         2          0   0.1818182             No 0.250000000
## 11     no         2          0   0.1818182             No 0.250000000
## 12     no         2          0   0.1818182             No 0.250000000
## 13     no         2          0   0.1818182             No 0.250000000
## 14    yes         2          0   0.1818182             No 0.250000000
## 15    yes         1          0   0.6666667            Yes 0.571428571
## 16     no         1          0   0.6666667            Yes 0.571428571
## 17    yes         1          0   0.6666667            Yes 0.571428571
## 18    yes         1          0   0.6666667            Yes 0.571428571
## 19     no         1          0   0.6666667           Yes 0.571428571
## 20    yes         1          0   0.6666667           Yes 0.571428571
## 21    yes         1          0   0.6666667           Yes 0.571428571
## 22    yes         1          0   0.6666667           Yes 0.571428571
## 23     no         1          0   0.6666667           Yes 0.571428571
## 24    yes         2          2   1.0000000           Yes 0.333333333
```

---

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?

Here, we are trying 2 methods to apply the naive Bayes Classifier.

Method 1: Done with only a subset of the dimensions that are associated with the target variable

```
# Install and load necessary packages
install.packages(c("naivebayes", "caret"))
```

```
## Warning: package 'caret' is in use and will not be installed
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'
```

```
## Warning: package 'naivebayes' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(caret)

# Set seed for reproducibility
set.seed(22)

# Splitting the data into training and validation datasets
train.index <- sample(c(1:dim(accidents)[1]), dim(accidents)[1]*0.6)
train.df <- accidents[train.index,]
valid.df <- accidents[-train.index,]
```

```r
# Specifying the variables for the model
vars <- c("INJURY", "HOUR_I_R", "ALIGN_I", "WRK_ZONE", "WKDY_I_R",
          "INT_HWY", "LGTCON_I_R", "PROFIL_I_R", "SPD_LIM", "SUR_COND",
          "TRAF_CON_R", "TRAF_WAY", "WEATHER_R")

# Training the Naive Bayes classifier
nbTotal <- naive_bayes(INJURY ~ ., data = train.df[,vars])

# Generate and display the confusion matrix for training data
confusion_matrix_result <- confusionMatrix(train.df$INJURY, predict(nbTotal, train.df[, vars]), positive
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```r
print(confusion_matrix_result)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5574 6972
##        yes 4829 7934
##
##                Accuracy : 0.5337
##                  95% CI : (0.5276, 0.5399)
##     No Information Rate : 0.589
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.066
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5323
##             Specificity : 0.5358
##          Pos Pred Value : 0.6216
##          Neg Pred Value : 0.4443
##              Prevalence : 0.5890
##          Detection Rate : 0.3135
##    Detection Prevalence : 0.5043
##       Balanced Accuracy : 0.5340
##
##        'Positive' Class : yes
##
```

Method 2: Done with all of the dimensions that are associated with the target variable except MAX___SEV_IR and SPD_LIM

```r
library(dplyr)

# Removing the MAX_SEV_IR column
accidents <- accidents %>% select(-MAX_SEV_IR)
accidents <- accidents %>% select(-SPD_LIM)

# Checking the structure to ensure the column has been removed
str(accidents)
```

```
## 'data.frame':    42183 obs. of  23 variables:
##  $ HOUR_I_R      : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 1 ...
##  $ ALCHL_I       : Factor w/ 2 levels "1","2": 2 2 2 2 1 2 2 2 2 2 ...
##  $ ALIGN_I       : Factor w/ 2 levels "1","2": 2 1 1 1 1 1 1 1 1 1 ...
##  $ STRATUM_R     : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 1 2 2 1 ...
##  $ WRK_ZONE      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ WKDY_I_R      : Factor w/ 2 levels "0","1": 2 2 2 1 2 2 2 2 2 1 ...
##  $ INT_HWY       : Factor w/ 3 levels "0","1","9": 1 2 1 1 1 1 2 1 1 1 ...
##  $ LGTCON_I_R    : Factor w/ 3 levels "1","2","3": 3 3 3 3 3 3 3 3 3 3 ...
##  $ MANCOL_I_R    : Factor w/ 3 levels "0","1","2": 1 3 3 3 3 1 1 1 1 1 ...
##  $ PED_ACC_R     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ RELJCT_I_R    : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 1 1 2 2 ...
##  $ REL_RWY_R     : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 1 1 1 1 ...
##  $ PROFIL_I_R    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ SUR_COND      : Factor w/ 5 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ TRAF_CON_R    : Factor w/ 3 levels "0","1","2": 1 1 2 2 1 1 1 1 1 1 ...
##  $ TRAF_WAY      : Factor w/ 3 levels "1","2","3": 3 3 2 2 2 2 2 1 1 1 ...
##  $ VEH_INVL      : Factor w/ 11 levels "1","2","3","4",..: 1 2 2 2 3 1 1 1 1 1 ...
##  $ WEATHER_R     : Factor w/ 2 levels "1","2": 1 2 2 1 1 2 2 1 2 2 ...
##  $ INJURY_CRASH  : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 1 1 ...
##  $ NO_INJ_I      : Factor w/ 15 levels "0","1","2","3",..: 2 1 1 1 1 2 1 2 1 1 ...
##  $ PRPTYDMG_CRASH: Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 2 2 ...
##  $ FATALITIES    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ INJURY        : Factor w/ 2 levels "no","yes": 2 1 1 1 1 2 1 2 1 1 ...
```

```r
#Splitting the data into 60% Training and 40% Validation set
install.packages("caTools")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: unable to access index for repository https://cran.stat.ucla.edu/src/contrib:
##   cannot open URL 'https://cran.stat.ucla.edu/src/contrib/PACKAGES'

## Warning: package 'caTools' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository https://cran.stat.ucla.edu/bin/windows/contrib/4.3:
##   cannot open URL 'https://cran.stat.ucla.edu/bin/windows/contrib/4.3/PACKAGES'
```

```r
library(caTools)
set.seed(1)
split <- sample.split(accidents, SplitRatio = 0.6)
```

```r
training_set <- subset(accidents, split == TRUE)
validation_set <- subset(accidents, split == FALSE)


# 1. Install and load necessary packages
install.packages("naivebayes")
```

```
## Warning: package 'naivebayes' is in use and will not be installed
```

```r
library(naivebayes)

# 2. Train a Naive Bayes classifier
# Assuming all columns except 'INJURY' are predictors and 'INJURY' is the response variable
model <- naive_bayes(INJURY ~ ., data = training_set)
```

```
## Warning: naive_bayes(): Feature VEH_INVL - zero probabilities are present.
## Consider Laplace smoothing.

## Warning: naive_bayes(): Feature INJURY_CRASH - zero probabilities are present.
## Consider Laplace smoothing.

## Warning: naive_bayes(): Feature NO_INJ_I - zero probabilities are present.
## Consider Laplace smoothing.

## Warning: naive_bayes(): Feature PRPTYDMG_CRASH - zero probabilities are
## present. Consider Laplace smoothing.

## Warning: naive_bayes(): Feature FATALITIES - zero probabilities are present.
## Consider Laplace smoothing.
```

```r
# 3. Predict using the validation set
predictions <- predict(model, validation_set)
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```r
# 4. Produce the confusion matrix
confusion_mtx <- table(Predicted = predictions, Actual = validation_set$INJURY)
print(confusion_mtx)
```

```
##          Actual
## Predicted   no  yes
##       no  9030    0
##       yes    0 9310
```

```r
# 5. Calculate the overall error
overall_error <- 1 - sum(diag(confusion_mtx)) / sum(confusion_mtx)
print(overall_error)
```

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