FML ASSIGNMENT3

DIVYA CHANDRASEKARAN_811284790

2023-10-15

Summary Report

Problem statement

In the provided code, an analysis is conducted on a dataset containing information about automobile accidents in the United States in 2001. The primary objective is to predict whether an accident will result in injury based on initial reports and associated data.

The code begins by creating a binary "INJURY" variable, representing whether an accident caused an injury or not. Initial predictions are made, and conditional probabilities are computed using Bayes' theorem for different weather and traffic conditions combinations, where the **no injury rate when no further information is available is 20721, and the injury rate for yes is 21462.**

And, the probability rate for INJURY from our dataset is 50.88%, since ~51% of the accidents in our data set resulted in an accident. Therefore, we should predict that an accident will result in injury because it is slightly more likely.

A naive Bayes classifier is also applied to a subset of 24 records to predict injury, and the results are compared with manual calculations. Furthermore, the dataset is divided into a training and validation set, the 2 predictors are identified which are WEATHER_R and TRAF_CON_R, and a naive Bayes classifier is trained on the training set to predict injuries.

The overall error rate on the validation set is computed which is 47.77%

In conclusion, the provided code conducts a comprehensive analysis of automobile accidents with a focus on injury prediction. It involves data preprocessing, the creation of a binary response variable, and the application of both manual and naive Bayes classification methods. The code highlights the importance of partitioning the dataset for model evaluation and identifies relevant predictors for analysis. While the analysis shows promising results, it is essential to remember that the model's performance may vary in practice, and additional variables or advanced modeling techniques may be required for more accurate predictions.

1. **Creating the INJURY Variable:**

The code starts by loading a dataset containing information on automobile accidents.

A binary "INJURY" variable is created, which takes the value "yes" if an accident resulted in an injury (MAX_SEV_IR = 1 or 2) and "no" otherwise.

Creating the "INJURY" variable is a fundamental step in preparing the dataset for injury prediction. It allows us to distinguish accidents that result in injury from those that do not, and the dummy variable called “injury” is created.

2. **Initial Prediction for Accidents:**

Here, we are calculating the initial prediction for accidents when no further information is available. It computes the probability of injury ("INJURY = Yes") based on the proportion of accidents that resulted in injuries in the dataset.

Therefore, a table for injury is created and we calculate the probability for injury. If an accident has just been reported, **the prediction will be no injury because there is less than 50% (49.77%) of an injury**. However, this is a simplistic prediction and does not take into account other factors that might influence injury outcomes.

Injury = NO = 20721

Injury = YES = 21462

And , the probability of injury is 50.88%

3. **Pivot Table Analysis for the First 24 Records:**

Before the pivot table is created, we have converted the variables into categorical type, and created a ”new subset” with the required records.

The code selects the first 24 records from the dataset and focuses on three variables: "INJURY," "WEATHER_R," and "TRAF_CON_R." A pivot table is created to examine how the "INJURY" variable relates to the combinations of "WEATHER_R" and "TRAF_CON_R" for these 24 records. The pivot table analysis provides an initial visual exploration of how weather conditions and traffic conditions may be related to injury outcomes.

4. **Exact Bayes Conditional Probabilities:**

Conditional probabilities of an injury ("INJURY = Yes") are computed for six specific combinations of "WEATHER_R" and "TRAF_CON_R" using Bayes' theorem.

Some combinations have undefined probabilities due to zero denominators, indicating that certain conditions have no occurrences in the dataset. The calculation of Bayes conditional probabilities for specific combinations of predictors provides insights into how the occurrence of injuries depends on weather and traffic conditions. It is crucial to note when probabilities are undefined due to a lack of observations in certain combinations.

5. **Classifying 24 Accidents Using Probabilities:**

The code adds the 6 calculated probabilities to the subset of 24 records.

A new column called "PREDICT_PROB" is created to predict injury based on a cutoff probability of 0.5.

6. **Manual Naive Bayes Conditional Probability:**

The code manually computes the naive Bayes conditional probability of an injury given "WEATHER_R = 1" and "TRAF_CON_R = 1" using the proportions of relevant events.

The manual calculation of a naive Bayes conditional probability demonstrates how the probability of injury can be computed based on conditional events. In this specific case, the probability is computed for the combination of "WEATHER_R = 1" and "TRAF_CON_R = 1."

The answer here was 0.

7. **Running Naive Bayes Classifier on 24 Records:**

The code employs the naive Bayes classifier on the same 24 records using the 'naiveBayes' function from the 'e1071' package.

The model's predictions are compared to the manually calculated results, and a confusion matrix is generated to assess the model's performance, **where the accuracy is about 52 percent.**

And, the resulting classification is equivalent since the accuracy values are closely similar.

8. **Partitioning the Dataset:**

The dataset is divided into a training set (60%) and a validation set (40%) using random sampling. Partitioning the dataset into training and validation sets is essential for evaluating the model's generalization to unseen data. This step prepares the data for more rigorous testing.

9. **Running Naive Bayes Classifier on the Complete Training Set:**

A naive Bayes classifier is trained on the complete training set, using relevant predictors that are assumed to be available when no accident-specific information is known.

The confusion matrix generated from this step will be valuable for evaluating the model's performance **where the accuracy is 52.2 percent**.

10. **Overall Error on Validation Set:**

Lastly, the overall error rate is computed for the validation set by applying the trained naive Bayes model to the validation data and comparing the predictions to the true values of "INJURY." The computation of the overall error rate on the validation set assesses how well the trained model performs on new, unseen data, **and the error rate 47.7 percent.**

CONCLUSION

From a business standpoint, prioritizing safety in accident prediction is important, especially when minimal information is available. Assuming the worst-case scenario and

predicting the injury, continuously monitoring and analyzing data, utilizing pivot tables and data analysis to identify risk factors, and assessing conditional probabilities through Bayes analysis is very useful in real life for assessing how many accidents lead to injury and due to what factors. Developing real-time reporting systems, collaborating with authorities, and ensuring regulatory compliance can make this model better and more plausible for the real world. Maintain a feedback loop for ongoing model improvement and prioritize user-friendly interfaces for reporting and response. Safety remains the primary goal in all decision-making.

```
#QUESTION1
##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.”
#Load the accidentsFull.csv and install/load any required packages. Create
and insert a dummy variable called “INJURY” in the data.

library(readr)
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(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)
accidentsFull<- read_csv("accidentsFull.csv")

## Rows: 42183 Columns: 24

## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
## dbl (24): HOUR_I_R, ALCHL_I, ALIGN_I, STRATUM_R, WRK_ZONE, WKDY_I_R,
INT_HWY...
##
```

```
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
View(accidentsFull)
accidentsFull$INJURY <- ifelse(accidentsFull$MAX_SEV_IR>0, "yes", "no")
```

```
for (i in 1:dim(accidentsFull)[2]) {
  if (is.character(accidentsFull[, i])) {
    accidentsFull[, i] <- as.factor(accidentsFull[, i])
  }
}
head(accidentsFull, n=24)
```

```
## # A tibble: 24 × 25
##    HOUR_I_R ALCHL_I ALIGN_I STRATUM_R WRK_ZONE WKDY_I_R INT_HWY LGTCON_I_R
##       <dbl>   <dbl>   <dbl>     <dbl>    <dbl>    <dbl>   <dbl>      <dbl>
##  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
## # ℹ 14 more rows
## # ℹ 17 more variables: MANCOL_I_R <dbl>, PED_ACC_R <dbl>, RELJCT_I_R
<dbl>,
##   REL_RWY_R <dbl>, PROFIL_I_R <dbl>, SPD_LIM <dbl>, SUR_COND <dbl>,
##   TRAF_CON_R <dbl>, TRAF_WAY <dbl>, VEH_INVL <dbl>, WEATHER_R <dbl>,
##   INJURY_CRASH <dbl>, NO_INJ_I <dbl>, PRPTYDMG_CRASH <dbl>, FATALITIES
<dbl>,
##   MAX_SEV_IR <dbl>, INJURY <chr>
```

```
#QUESTION2
#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?

#CREATING A TABLE BASED ON INJURY.
injury.table <- table(accidentsFull$INJURY)
show(injury.table)
```

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

#cALUCATING THE PROBABILITY OF THE INJURY
injury.probablilty =
scales::percent(injury.table["yes"]/(injury.table["yes"]+injury.table["no"]),
0.01)
injury.probablilty

##      yes
## "50.88%"

##Since ~51% of the accidents in our data set resulted in an accident, we
should predict that an accident will result in injury because it is slightly
more likely.
```

```
#QUESTION3
#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.

#CONVERTING THE VARIABLES TO CATEGORICAL TYPE
# IDENTIFYING THE TARGET VARIABLE COLUMN INDEX (ASSUMING IT'S THE LAST
COLUMN)
target_col_index <- dim(accidentsFull)[2]

#CONVERTING ALL COLUMNS EXCEPT THE TARGRT VARIABLE TO FACTORS
accidentsFull[, 1:(target_col_index - 1)] <- lapply(accidentsFull[,
1:(target_col_index - 1)], as.factor)
print(target_col_index)

## [1] 25

#create a new subset with only the required records
new.df <- accidentsFull[1:24, c('INJURY','WEATHER_R','TRAF_CON_R')]
new.df

## # A tibble: 24 × 3
##    INJURY WEATHER_R TRAF_CON_R
##    <chr>  <fct>     <fct>
##  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
## # ℹ 14 more rows
```

```
dt1 <- ftable(new.df)
dt2 <- ftable(new.df [,-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
```

```
#CREATING A PIVOT TABLE THAT EXAMINES INJURY AS A FUCTION OF THE TWO PREDICTORS FOR THESE 12 RECORDS, AND USING ALL THREE VARAIBLES IN THE PIVOT TABLE AS ROWS/COLUMNS.

rpivotTable::rpivotTable(new.df)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.
```


Table
INJURY
WEATH
Count
Totals
24

```r
#QUESTION4
#COMPUTING THE BAYES CONDITIONAL PROBABLITIES OF AN INJURY (INJURY = Yes)
GIVEN THE SIX POSSIBILE COMBINATIONS OF THE PREDITCTORS.

# Injury = yes

probability1 = dt1[3,1] / dt2[1,1] # Injury, Weather=1 and Traf=0
probability2 = dt1[4,1] / dt2[2,1] # Injury, Weather=2, Traf=0
probability3 = dt1[3,2] / dt2[1,2] # Injury, W=1, T=1
probability4 = dt1[4,2] / dt2[2,2] # I, W=2,T=1
probability5 = dt1[3,3] / dt2[1,3] # I, W=1,T=2
probability6 = dt1[4,3]/ dt2[2,3] #I,W=2,T=2
print(c(probability1,probability2,probability3,probability4,probability5,prob
ability6))
```

```
## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000
```

```r
# Injury = no

n1 = dt1[1,1] / dt2[1,1] # Weather=1 and Traf=0
n2 = dt1[2,1] / dt2[2,1] # Weather=2, Traf=0
n3 = dt1[1,2] / dt2[1,2] # W=1, T=1
n4 = dt1[2,2] / dt2[2,2] # W=2,T=1
n5 = dt1[1,3] / dt2[1,3] # W=1,T=2
n6 = dt1[2,3] / dt2[2,3] # W=2,T=2
print(c(n1,n2,n3,n4,n5,n6))
```

```
## [1] 0.3333333 0.8181818 1.0000000 1.0000000 1.0000000 0.0000000
```

```r
#In the above 12 observations there is no observation with (Injury=yes,
WEATHER_R = 2, TRAF_CON_R =2). The conditional probability here is undefined,
since the denominator is zero. (NOTE)
```

```r
#QUESTION5
#CLASSIFYING THE 24 ACCIDENTS USING THESES PROBABLITIES AND CUTOFF OF 0.5

#ADDING PROBABILITY RESULTS TO THE SUBSET
prob.inj <- rep(0,24)

for (i in 1:24) {
  print(c(new.df$WEATHER_R[i],new.df$TRAF_CON_R[i]))
    if (new.df$WEATHER_R[i] == "1") {
      if (new.df$TRAF_CON_R[i]=="0"){
        prob.inj[i] = probability1
      }
      else if (new.df$TRAF_CON_R[i]=="1") {
        prob.inj[i] = probability3
      }
      else if (new.df$TRAF_CON_R[i]=="2") {
```

```
        prob.inj[i] = probability5
      }
    }
    else {
      if (new.df$TRAF_CON_R[i]=="0"){
        prob.inj[i] = probability2
      }
      else if (new.df$TRAF_CON_R[i]=="1") {
        prob.inj[i] = probability4
      }
      else if (new.df$TRAF_CON_R[i]=="2") {
        prob.inj[i] = probability6
      }
    }
  }
```

```
## [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
```

```
new.df$prob.inj <- prob.inj

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

#QUESTION6
#COMPUTING MANUALLY THE NAIVE BAYES CONDITIONAL PROBABILITY OF AN INJURY
GIVEN THE WEATHER_R =1 AND TRAF_CON_R =1.

#The Naive Bayes conditional probability is computed using the Naive Bayes
formula as follows:
#P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1) = (P(INJURY = Yes |
WEATHER_R = 1) * P(INJURY = Yes | TRAF_CON_R = 1) * P(INJURY = Yes)) /
(P(WEATHER_R = 1) * P(TRAF_CON_R = 1))

p_manual_injury_given_weather_1_traf_con_1 <- probability3
cat("Manual Naive Bayes Conditional Probability (Injury = Yes | Weather_R =
1, TRAF_CON_R = 1):", p_manual_injury_given_weather_1_traf_con_1)
```

```
## Manual Naive Bayes Conditional Probability (Injury = Yes | Weather_R = 1,
TRAF_CON_R = 1): 0
```

```
#QUESTION7
#RUNNING A NAIVE BAYES CLASSIFIER ON THE 24 RECORDS AND TWO PREDICTORS.
#NOW,WE HAVE TO CHECK THE MODEL OUTPUT TO OBTAIN PROBABILITIES AND
CLASSIFCATIONS FOR ALL 24 RECORDS.
##AND THEN, WE ARE COMPARING TO BAYES CLASSIFCATION TO SEE IF THE RESULTING
CLASSIFICATIONS ARE EQUIVALENT OR NOT.

library(e1071)
nb<-naiveBayes(INJURY ~ ., data = new.df)
nbt <- predict(nb, newdata = new.df,type = "raw")
new.df$nbpred.prob <- nbt[,2] # Transfer the "Yes" nb prediction

library(caret)
nb2 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = new.df, 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

## 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 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 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

## 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 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
```

```r
predict(nb2, newdata = new.df[,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
no
## Levels: no yes
```

```r
predict(nb2, newdata = new.df[,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
no
## Levels: no yes

## AND TO CHECK IF THE RANKING (= ordering) OBSERVATIONS EQUIVALENT
#LOADIND THE PACKAGES AND RUNNING NAIVE BAYES CLASSIFIER

library(klaR)

## Loading required package: MASS

##
## Attaching package: 'MASS'

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

library(naivebayes)

## naivebayes 0.9.7 loaded

accidentsFull$INJURY <- ifelse(accidentsFull$MAX_SEV_IR %in% c(1,2), "Yes",
"No")


#QUESTION8
#Splitting the data into training (60%) and validation (40%)

set.seed(123)
trainIndex <- createDataPartition(accidentsFull$INJURY, p = 0.6, list =
FALSE)
train_data <- accidentsFull[trainIndex, ]
val_data <- accidentsFull[-trainIndex, ]

#Creating a naive bayes model with the relavant predictors
nb <- naive_bayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = train_data)


#Predicting on the validation set
val_pred <-predict(nb, newdata = val_data)

## 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
#Converting val_pred into a character vector
val_pred <- as.character(val_pred)

#Converting val_data$Injury to a character vector
val_data$INJURY <- as.character(val_data$INJURY)

#Creating a factor with matching levels
val_pred <- factor(val_pred, levels = c("No", "Yes"))
val_data$INJURY <- factor(val_data$INJURY, levels = c("No", "Yes"))

#Creating a confusion matrix
confusion_matrix <- confusionMatrix(val_pred, val_data$INJURY)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1294 1064
##        Yes 6994 7520
##
##                Accuracy : 0.5224
##                  95% CI : (0.5148, 0.53)
##     No Information Rate : 0.5088
##     P-Value [Acc > NIR] : 0.0002039
##
##                   Kappa : 0.0326
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.1561
##             Specificity : 0.8760
##          Pos Pred Value : 0.5488
##          Neg Pred Value : 0.5181
##              Prevalence : 0.4912
##          Detection Rate : 0.0767
##    Detection Prevalence : 0.1398
##       Balanced Accuracy : 0.5161
##
##        'Positive' Class : No
##
```

```r
#QUESTION9
#OVERALL ERROR OF THE VALIDATION SET
overall_error <- 1 - confusion_matrix$overall["Accuracy"]
cat("overall error of the validation set:", overall_error, "\n")
```

```
## overall error of the validation set: 0.477596
```