Assignment- 3_FML

Spandana Sodadasi

2023-10-13

```
knitr::opts_chunk$set(echo = TRUE, comment = NULL)
```

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?

Answer: The probability of 'No' is 0.4912 and the probability of 'Yes' is 0.5088. Since, the probability of 'Yes' is slightly higher than the probability of 'No', we can predict that if an accident has just been reported, and no further information is available, the prediction should be "INJURY = Yes".

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.

Answer: Selected 24 records and created a pivot table where the "Object1" contingency table reveals the distribution of injuries (Yes/No) with respect to different weather conditions ("WEATHER_R") and traffic conditions ("TRAF_CON_R") and "Object2" contingency table provides insights into how "WEATHER_R" and "TRAF_CON_R" are related. It shows that for each weather condition (1 or 2), there are different traffic conditions (0, 1, or 2) with corresponding counts.

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

Answer: Calculating the exact Bayes conditional probabilities of (Injury=Yes) basing on the two pivot tables created in the previous chunk under the specific weather and traffic conditions.It evaluates the likelihood of injuries occurring under specific conditions of weather and traffic. The output shows that when the weather is categorized as "1" and traffic as "0," there is a 66.67% probability of an injury and in contrast, when weather is "2" and traffic is "0," the probability drops to approximately 18.18%. In certain conditions, such as weather "1" and traffic "1" or "2," the probability is zero, indicating no observed injuries under these circumstances in the provided dataset. On the other hand, when weather is "2" and traffic is "2," the probability is 100%, implying that injuries are guaranteed in this specific scenario. The insights of these conditional probabilities help in identifying the risk factors for accidents.

   (b) Classify the 24 accidents using these probabilities and a cutoff of 0.5.

Answer: The code calculates conditional probabilities of injury in the "accidentsFull24" dataset based on different combinations of weather and traffic conditions. It assigns probabilities of injury for each observation, and then categorizes them into "yes" or "no" based on a threshold probability of 0.5. It provides meaningful insights into the dataset where it includes categorization that helps classify whether each scenario is likely to result in an injury ("yes") or ("no"). These predictions offer valuable insights into the factors influencing injury occurrence in accidents based on weather and traffic conditions.

(c) Compute manually the Naive Bayes conditional probability of an injury given WEATHER_R $= 1$ and TRAF_CON_R $= 1$.

Answer: The answer for the Naive Bayes conditional probability of Injury=yes given WEATHER_R $= 1$ and TRAF_CON_R $= 1$ is "0".

(d) 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.

Answer: Implementing the Naive Bayes classifier on the dataset containing 24 records i.e.,AccidentsFull24 and two predictor variables, "TRAF_CON_R" and "WEATHER_R," to predict the outcome "INJURY." Calculated and added the raw probabilities and corresponding classifications for all 24 records, which can be used for further analysis and understanding the likelihood of injury based on the given predictors.

(i) Compare this to the exact Bayes classification. Are the resulting classifications equivalent?

Answer: After comparing the Exact Bayes and Naive Bayes classification, the result prints out that to be that "The classification is different is for some records" and this is because of the concept called Conditional Independence. The Naive Bayes classifier assumes that predictor variables are independent of each other and the exact Bayes classifier considers all possible dependencies between predictor variables, resulting in potentially more accurate classifications. Therefore, when conditional independence is not met, the two classifiers produce different results, with the exact Bayes classifier offering a more comprehensive analysis of complex relationships in the data. However, the probability of Naive Bayes is not useful from an absolute probability value but its is useful from a ranking view.

(ii) Is the ranking (= ordering) of observations equivalent?

Answer: This code initially involves filtering out the exact Bayes probabilities that are equal to 0 and 1. The purpose of doing this is to align the ranking of Naive Bayes probabilities with those of the exact Bayes probabilities. The need for this alignment stems from the concept of "Laplace Smoothing", The fundamental idea behind Laplace smoothing is to add a small, positive constant (k) to the counts of all categories within a given variable. This ensures that no category in the Naive Bayes has the probability of zero. Therefore, by excluding the exact Bayes probabilities of 0 and 1, we create consistency in the rankings. So, after excluding the rows of 0 and 1 the ranking would now show same for both Exact and Naive Bayes probabilities.

3.(a)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.

Answer: We here are removing the 24th variable as we are using this variable to define the values for 'Injury' in the dataset, so including it wouldn't make sense since the response variable is derived from this specific column. The model eventually exhibits an overall accuracy of around 52.2%, meaning it correctly classifies cases nearly 50% of the time. Specificity and sensitivity of the model are relatively low with 55.58% and 51.63% indicating that there is room for improvement in all aspects of the model's performance.

(b) What is the overall error of the validation set?

Answer: Overall error is nothing but comprehensive measurement of inaccuracies in a model. It is basically the sum of false positives and false negatives divided by total number of cases or it can also be calculated by using the formula '1-Accuracy'. Hence, the overall error of the validation set is 0.4753 or 47.53% .

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 24 records. Use all three variables in the pivot table as rows/columns. (a)Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors. (b)Classify the 24 accidents using these probabilities and a cutoff of 0.5. (c)Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1. (d)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. (i) Compare this to the exact Bayes classification. Are the resulting classifications equivalent? (ii) 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%). (a)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. (b)What is the overall error of the validation set?

Data Input and cleaning:-

Loading the required libraries.

```
library(caret, warn.conflicts = FALSE)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```
library(e1071, warn.conflicts = FALSE)
library(dplyr, warn.conflicts = FALSE)
library(tinytex)
```

Importing and reading the dataset.

```
library(readr)
AccidentsFull.df <- read.csv("C:/Users/spand/Downloads/accidentsFull.csv")
dim(AccidentsFull.df)
```

```
[1] 42183    24
```

Creating a dummy variable called 'INJURY' and converting all the variables to factors.

```
AccidentsFull.df$INJURY <- ifelse(AccidentsFull.df$MAX_SEV_IR %in% c(1, 2),
                                  "Yes", "No")
# Converting all variables to factor
for (i in c(1:dim(AccidentsFull.df)[2])){
  AccidentsFull.df[,i] <- as.factor(AccidentsFull.df[,i])
}
head(AccidentsFull.df)
```

```
  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
  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
  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
  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
```

All the variables are now converted into factors.

Questions:

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?

```
Injury_table <- table(AccidentsFull.df$INJURY)
Injury_table
```

```
   No   Yes
20721 21462
```

```
Probability_No <- Injury_table["No"] / sum(Injury_table)
Probability_Yes <- Injury_table["Yes"] / sum(Injury_table)
cat("Probability of 'No': ", round(Probability_No,digits=4), "\n")
```

Probability of 'No': 0.4912

```
cat("Probability of 'Yes': ", round(Probability_Yes, digits = 4), "\n")
```

Probability of 'Yes': 0.5088

The probability of 'No' is 0.4912 and the probability of 'Yes' is 0.5088. Since, the probability of 'Yes' is slightly higher than the probability of 'No', we can predict that if an accident has just been reported, and no further information is available, the prediction should be "INJURY = Yes".

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.

```
AccidentsFull24 <-
  AccidentsFull.df[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
#head(AccidentsFull)
Object1 <- ftable(AccidentsFull24)
Object2 <- ftable(AccidentsFull24[,-1])
Object1
```

```
               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
```

```
Object2
```

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

Selected 24 records and created a pivot table where the "Object1" contingency table reveals the distribution of injuries (Yes/No) with respect to different weather conditions ("WEATHER_R") and traffic conditions ("TRAF_CON_R") and "Object2" contingency table provides insights into how "WEATHER_R" and "TRAF_CON_R" are related. It shows that for each weather condition (1 or 2), there are different traffic conditions (0, 1, or 2) with corresponding counts.

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

```
P1 = Object1[3,1] / Object2[1,1] # Injury=yes, Weather=1 & Traffic=0
P2 = Object1[4,1] / Object2[2,1] # Injury=yes, Weather=2 & Traffic=0
P3 = Object1[3,2] / Object2[1,2] # Injury=yes, Weather=1 & Traffic=1
P4 = Object1[4,2] / Object2[2,2] # Injury=yes, Weather=2 & Traffic=1
P5 = Object1[3,3] / Object2[1,3] # Injury=yes, Weather=1 & Traffic=2
P6 = Object1[4,3] / Object2[2,3] # Injury=yes, Weather=2 & Traffic=2
print(c(P1))
```

```
[1] 0.6666667
```

```
print(c(P2))
```

```
[1] 0.1818182
```

```
print(c(P3))
```

```
[1] 0
```

```
print(c(P4))
```

```
[1] 0
```

```
print(c(P5))
```

```
[1] 0
```

```
print(c(P6))
```

```
[1] 1
```

Calculating the exact Bayes conditional probabilities of (Injury=Yes) basing on the two pivot tables created in the previous chunk under the specific weather and traffic conditions.It evaluates the likelihood of injuries occurring under specific conditions of weather and traffic. The output shows that when the weather is categorized as "1" and traffic as "0," there is a 66.67% probability of an injury and in contrast, when weather is "2" and traffic is "0," the probability drops to approximately 18.18%. In certain conditions, such as weather "1" and traffic "1" or "2," the probability is zero, indicating no observed injuries under these circumstances in the provided dataset. On the other hand, when weather is "2" and traffic is "2," the probability is 100%, implying that injuries are guaranteed in this specific scenario. The insights of these conditional probabilities help in identifying the risk factors for accidents.

(b) Classify the 24 accidents using these probabilities and a cutoff of 0.5.

```
Probability_Injury <- rep(0,24)
for (i in 1:24) {
  (c(AccidentsFull24$WEATHER_R[i],AccidentsFull24$TRAF_CON_R[i]))
    if (AccidentsFull24$WEATHER_R[i] == "1") {
      if (AccidentsFull24$TRAF_CON_R[i]=="0"){
        Probability_Injury[i] = P1
```

```r
    }
    else if (AccidentsFull24$TRAF_CON_R[i]=="1") {
      Probability_Injury[i] = P3
    }
    else if (AccidentsFull24$TRAF_CON_R[i]=="2") {
      Probability_Injury[i] = P5
    }
  }
  else {
    if (AccidentsFull24$TRAF_CON_R[i]=="0"){
      Probability_Injury[i] = P2
    }
    else if (AccidentsFull24$TRAF_CON_R[i]=="1") {
      Probability_Injury[i] = P4
    }
    else if (AccidentsFull24$TRAF_CON_R[i]=="2") {
      Probability_Injury[i] = P6
    }
  }
 }
}
AccidentsFull24$Probability_Injury <- Probability_Injury
AccidentsFull24$Pred_Probability <- ifelse(AccidentsFull24$Probability_Injury > 0.5,
                                    "Yes", "No")

AccidentsFull24
```

	INJURY	WEATHER_R	TRAF_CON_R	Probability_Injury	Pred_Probability
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

The code calculates conditional probabilities of injury in the "accidentsFull24" dataset based on different combinations of weather and traffic conditions. It assigns probabilities of injury for each observation, and

then categorizes them into "yes" or "no" based on a threshold probability of 0.5. It provides meaningful insights into the dataset where it includes categorization that helps classify whether each scenario is likely to result in an injury ("yes") or ("no"). These predictions offer valuable insights into the factors influencing injury occurrence in accidents based on weather and traffic conditions.

(c) Compute manually the Naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

```
nb_Numerator <- P3
nb_Denominator <- sum(P3, Object1[1,2] / Object2[1,2])
nb_probability <- nb_Numerator / nb_Denominator
nb_probability
```

```
[1] 0
```

The answer for the Naive Bayes conditional probability of Injury=yes given WEATHER_R = 1 and TRAF_CON_R = 1 is "0".

(d) 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.

```
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = AccidentsFull24)
nbt_raw <- predict(nb, newdata = AccidentsFull24,type = "raw")
AccidentsFull24$NBpred_Injury <- nbt_raw[,2]
nbt_class <- predict(nb, newdata = AccidentsFull24,type = "class")
AccidentsFull24$NBpred_Probability <-nbt_class
head(AccidentsFull24)
```

	INJURY	WEATHER_R	TRAF_CON_R	Probability_Injury	Pred_Probability	NBpred_Injury
1	Yes	1	0	0.6666667	Yes	0.571428571
2	No	2	0	0.1818182	No	0.250000000
3	No	2	1	0.0000000	No	0.002244949
4	No	1	1	0.0000000	No	0.008919722
5	No	1	0	0.6666667	Yes	0.571428571
6	Yes	2	0	0.1818182	No	0.250000000

	NBpred_Probability
1	Yes
2	No
3	No
4	No
5	Yes
6	No

Implementing the Naive Bayes classifier on the dataset containing 24 records i.e.,AccidentsFull24 and two predictor variables, "TRAF_CON_R" and "WEATHER_R," to predict the outcome "INJURY." Calculated and added the raw probabilities and corresponding classifications for all 24 records, which can be used for further analysis and understanding the likelihood of injury based on the given predictors.

(i) Compare this to the exact Bayes classification. Are the resulting classifications equivalent?

```
Classification_Comparison <- all(AccidentsFull24$Pred_Probability ==
                                 AccidentsFull24$NBpred_Probability)
if (Classification_Comparison) {
  cat("The classification is the same for all records.\n")
} else {
  cat("The classification is different for some records.\n")
}
```

```
The classification is different for some records.
```

After comparing the Exact Bayes and Naive Bayes classification, the result prints out that to be that "The classification is different is for some records" and this is because of the concept called Conditional Independence. The Naive Bayes classifier assumes that predictor variables are independent of each other and the exact Bayes classifier considers all possible dependencies between predictor variables, resulting in potentially more accurate classifications. Therefore, when conditional independence is not met, the two classifiers produce different results, with the exact Bayes classifier offering a more comprehensive analysis of complex relationships in the data. However, the probability of Naive Bayes is not useful from an absolute probability value but its is useful from a ranking view.

(ii) Is the ranking (= ordering) of observations equivalent?

```
Bayesrank <-AccidentsFull24 %>%
  select(Probability_Injury, NBpred_Injury) %>%
  filter(!Probability_Injury == 0) %>%
  filter(!Probability_Injury == 1) %>%
  arrange(Probability_Injury)
Ranking1<-AccidentsFull24$Bayesrank

Naiverank <-AccidentsFull24 %>%
  select(Probability_Injury, NBpred_Injury) %>%
  filter(!Probability_Injury == 0) %>%
  filter(!Probability_Injury == 1) %>%
  arrange(NBpred_Injury)
Ranking2 <- AccidentsFull24$Naiverank

Rank_order <- all(Bayesrank == Naiverank)
if (Rank_order) {
  cat("The rank orders are the same for all records.\n")
} else {
  cat("The rank orders are different for some records.\n")
}
```

```
The rank orders are the same for all records.
```

This code initially involves filtering out the exact Bayes probabilities that are equal to 0 and 1. The purpose of doing this is to align the ranking of Naive Bayes probabilities with those of the exact Bayes probabilities. The need for this alignment stems from the concept of "Laplace Smoothing", The fundamental idea behind Laplace smoothing is to add a small, positive constant (k) to the counts of all categories within a given variable. This ensures that no category in the Naive Bayes has the probability of zero. Therefore, by excluding the exact Bayes probabilities of 0 and 1, we create consistency in the rankings. So, after excluding the rows of 0 and 1 the ranking would now show same for both Exact and Naive Bayes probabilities.

3.(a)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.

Splitting the data into 60% training and 40% validation.

```
set.seed(1)
train.index <- sample(row.names(AccidentsFull.df),0.6*dim(AccidentsFull.df)[1])
valid.index <- setdiff(row.names(AccidentsFull.df),train.index)
train.df <- AccidentsFull.df[train.index,]
valid.df <- AccidentsFull.df[valid.index,]
```

Normalizing the data and running the naive Bayes classifier on the complete training set with the relevant predictors and demonstrating the confusion matrix.

```
# Remove the 24th column
train.norm.df <- train.df[,-24]
valid.norm.df <- valid.df[,-24]

nb_model <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = train.norm.df)
train_pred <- predict(nb_model, new=train.norm.df)
conf_matrix1 <- confusionMatrix(train.df$INJURY, train_pred, positive = "Yes")
conf_matrix1
```

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No   2012 10490
       Yes  1608 11199

              Accuracy : 0.522
                95% CI : (0.5158, 0.5282)
   No Information Rate : 0.857
   P-Value [Acc > NIR] : 1

                 Kappa : 0.0357

 Mcnemar's Test P-Value : <2e-16

           Sensitivity : 0.5163
           Specificity : 0.5558
        Pos Pred Value : 0.8744
        Neg Pred Value : 0.1609
            Prevalence : 0.8570
        Detection Rate : 0.4425
  Detection Prevalence : 0.5060
     Balanced Accuracy : 0.5361

      'Positive' Class : Yes
```

We here are removing the 24th variable as we are using this variable to define the values for 'Injury' in the dataset, so including it wouldn't make sense since the response variable is derived from this specific column.

The model eventually exhibits an overall accuracy of around 52.2%, meaning it correctly classifies cases nearly 50% of the time. Specificity and sensitivity of the model are relatively low with 55.58% and 51.63% indicating that there is room for improvement in all aspects of the model's performance.

(b) What is the overall error of the validation set?

```
nb_model <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R, data = valid.norm.df)
valid_pred <- predict(nb_model, new=valid.norm.df)
conf_matrix2 <- confusionMatrix(valid.df$INJURY,  valid_pred, positive = "Yes")
conf_matrix2
```

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1079 7140
       Yes  880 7775

               Accuracy : 0.5247
                 95% CI : (0.5171, 0.5323)
    No Information Rate : 0.8839
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0302

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5213
            Specificity : 0.5508
         Pos Pred Value : 0.8983
         Neg Pred Value : 0.1313
             Prevalence : 0.8839
         Detection Rate : 0.4608
   Detection Prevalence : 0.5129
      Balanced Accuracy : 0.5360

       'Positive' Class : Yes
```

```
Overall_error=1-.5247
Overall_error
```

```
[1] 0.4753
```

Overall error is nothing but comprehensive measurement of inaccuracies in a model. It is basically the sum of false positives and false negatives divided by total number of cases or it can also be calculated by using the formula '1-Accuracy'. Hence, the overall error of the validation set is 0.4753 or 47.53% .