Practice 0: Naive Bayes

Ignacio Almodóvar, Luis Ángel Rodríguez García

2/9/2022

Introduction

We have chosen a dataset located in kaggle. It contains information about the classification of certain drug types based on different features such as the age, the sex, the blood pressure levels, the cholesterol levels and the sodium-to-potassium ratio.

In our case, the parameter that we want to estimate is θ and its support are formed by six different values: *drugA*, *drugB*, *drugC*, *drugD*, *drugX* and *drugY*. In Bayesian analysis the first thing to do is to calculate the prior of θ, so let's plot the frequency table:

Table 1: Drug Type Table Frequency

Type	Frequency	Probability
drugA	23	0.115
drugB	16	0.08
drugC	16	0.08
drugX	54	0.27
DrugY	91	0.455

In the table above, we can obtain the prior probability using the empirical Bayesian technique: $\pi(\theta = drugA) = \frac{23}{200} = 0.115$, $\pi(\theta = drugB) = \frac{16}{200} = 0.08$, $\pi(\theta = drugC) = \frac{16}{200} = 0.08$, $\pi(\theta = drugX) = \frac{54}{200} = 0.27$ and $\pi(\theta = drugY) = \frac{91}{200} = 0.455$.

If we are not data-driven, we can set the same probability for each type of drug: $\pi(\theta = drugA) = \pi(\theta = drugB) = \pi(\theta = drugC) = \pi(\theta = drugX) = \pi(\theta = drugY) = \frac{1}{5} = 0.2$. This is the case of using an orthodox Bayesian technique.

We have checked that this dataset does not contain missing values in any of their variables and we have split the dataset into train (70% of data) and test (the remaining 30%).

Naive Bayes Classifier

Now that we have split the data into train and set, we are going to train the model using the Naive Bayes classifier.

In Table 2 we can see that the a-priory probabilities obtained from the model are pretty much similar to those ones we got by means of the whole dataset.

Table 2: A-priory probabilities

Type	Probability
drugA	0.13
drugB	0.09
drugC	0.08
drugX	0.26
DrugY	0.45

Qualitative predictors

In table 3 we can see that the qualitative values of the variable sex are not distinguished by any type of drug (A, B, C, X or Y) because the probability in all the cases are close or equal to 0.5.

Table 3: Conditional Probability Sex

	F	M
drugA	0.50	0.50
drugB	0.42	0.58
drugC	0.45	0.55
drugX	0.50	0.50
DrugY	0.56	0.44

In table 4 we can see that the qualitative values of the blood pressure variable are distinguished by some types of drugs accurately. For example:

- Given drug A or B allways will be a high blood pressure
- Given drug C we will get a low blood preasure
- Given drug X we will not get a high pressure, 67% of times we will obtained a normal blood pressure.
- Given drug Y we will get a high blood pressure 43% of times and low blood pressure 32% of times.

Therefore, the variable blood pressure is much more informative than the variable sex.

Table 4: Conditional Probability Blood Pressure

	HIGH	LOW	NORMAL
drugA	1.00	0.00	0.00
drugB	1.00	0.00	0.00
drugC	0.00	1.00	0.00
drugX	0.00	0.33	0.67
DrugY	0.43	0.32	0.25

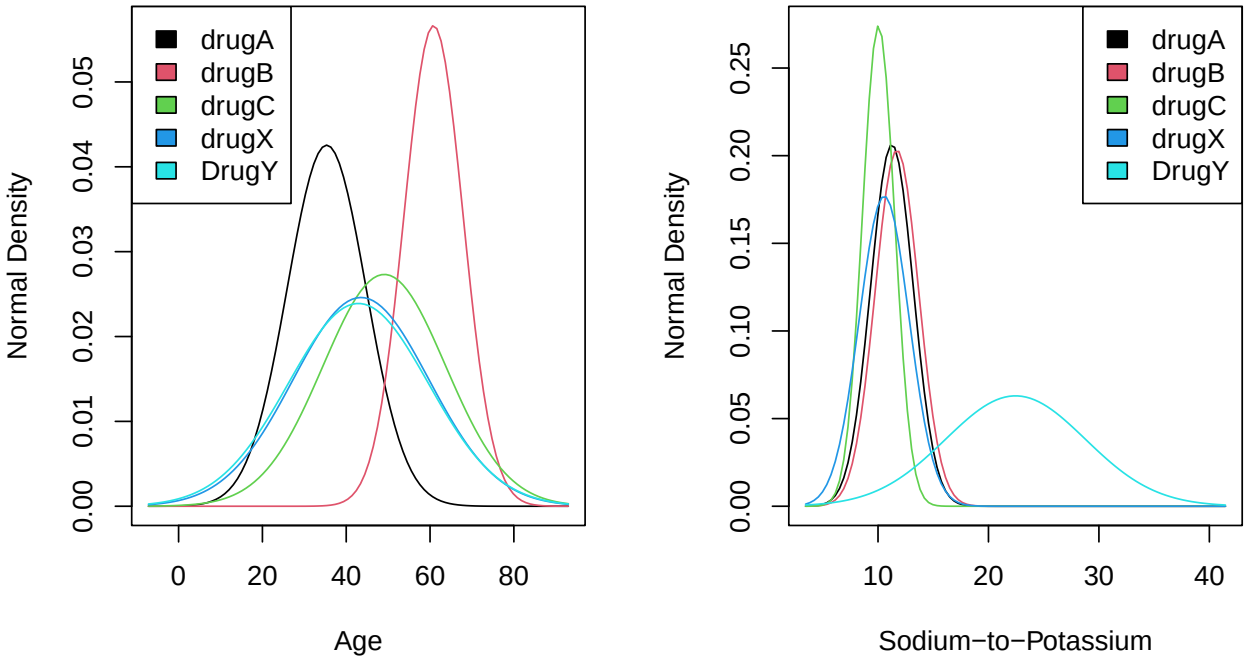
In table 5 we can see that the qualitative value *HIGH* of the cholesterol is distinguished accurately by drug C. Nevertheless, other probabilities are closed to 0.5 so they are not significant. The blood pressure variable is more informative than this one as we can distinguish many more factors given a specific type of drug.

Table 5: Conditional Probability Cholesterol

	HIGH	NORMAL
drugA	0.44	0.56
drugB	0.42	0.58
drugC	1.00	0.00
drugX	0.31	0.69
DrugY	0.49	0.51

Quantitative predictors

Let's look at the following plots which represent for each type of drug the likelihood depending on the values of the covariates *Age* and *Na_to_K*. Notice that we are using the train set as we want to know how the Naive Bayes model predicts.



As we can see above in the first plot, for people at the age of 65 the more likely drug associated is the B one meanwhile for people around 33 years old the more likely drug associated is the A one. In addition, drug C is the most used one for people at the age of 50. Lastly, it looks like drugs X and Y have the same likelihood to use in patients based on the quantitative variable *Age*.

In the second plot, we can see clearly that for a value around 22.5 of sodium-to-potassium we get in all the cases the drug Y. Therefore, this variable is so informative for classifying the drug Y. Moreover, we can see that for levels of Na-to-K of almost 10, the most likely drugs is the C one. At last, for a value around 12.5 we can see that the likelihood of getting some drug A, B, or X is the same.

Prediction

Now that we have trained the model, we can predict the values using the test set. Let's observed what happens for example with rows 1 and 7:

Table 6: Predictions: Rows 1 and 7

	drugA	**drugB**	**drugC**	**drugX**	**DrugY**
row1	0	0	0.44	0.37	0.19
row7	0	0	0.00	0.97	0.02

We can see in Table 6 that the prediction for the first observation on the test set is 0.44 likelihood to be associated with drug C, 0.37 with drug X and 0.19 with drug Y. For the seventh observation, we can see that the prediction classifies the data as drug X 97% of the times and as drug Y for the remaining percentage of times.

In the Table 7 it is visualize the confusion matrix associated to the prediction of the test set. We can see how well our model classifies our test data, there are only 3 wrong classifications:

- Two predictions classifying drug X when it is drug C
- One prediction classifying drug Y when it is C

Table 7: Confussion matrix

	drugA	**drugB**	**drugC**	**drugX**	**DrugY**
drugA	5	0	0	0	0
drugB	0	4	0	0	0
drugC	0	0	2	0	0
drugX	0	0	2	18	0
DrugY	0	0	1	0	28

Apendix

Code

```
knitr::opts_chunk$set(echo = TRUE)
if (!require(dplyr)) install.packages("dplyr")
  library(dplyr)
if (!require(scales)) install.packages("scales")
  library(scales)
if (!require(kableExtra)) install.packages("kableExtra")
  library(kableExtra)
if (!require(magrittr)) install.packages("magrittr")
  library(magrittr)
if (!require(e1071)) install.packages("e1071")
  library(e1071)
if (!require(caret)) install.packages("caret")
  library(caret)

drugs <- read.csv("drug200.csv", header = T)
#Target variable is the drug type A,B,C,X,Y
freq.table <- table(drugs$Drug)
freq.df <- cbind("Type"=names(freq.table),
                 "Frequency"=as.vector(freq.table),
                 "Probability"=as.vector(freq.table)/sum(as.vector(freq.table)))
knitr::kable(freq.df, caption = "Drug Type Table Frequency") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

# Function to check NA
missingValues=function(data){
  count=0
  a=cbind(lapply(lapply(data, is.na), sum))
  for(i in 1:ncol(data)){
    if(a[i]!=0){
      cat("There are", a[i], "missing values in column ", i,"\n" )
      count=count+1
    }
  }
  if(count==0){
    cat("There are no missing values in this dataset")
  }
}

missingValues(drugs)

# Split into train and test
set.seed(17)
index=nrow(drugs)
trainSet=(1:index)%in%sample(index,floor(index*0.7))
testSet=!trainSet

#build the model
model <- naiveBayes(Drug ~ ., data=drugs, subset=trainSet)
```

```
apriory.prob <- as.vector(model[1]$apriori)/sum(as.vector(model[1]$apriori))
apriory.prob <- round(apriory.prob, digits=2)
apriory.df <- cbind("Type"=names(model[1]$apriori),
                "Probability"=apriory.prob)
knitr::kable(apriory.df, digits = 2, caption = "A-priory probabilities") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

knitr::kable(model[2]$tables$Sex, digits = 2,
             caption = "Conditional Probability Sex") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

knitr::kable(model[2]$tables$BP, digits = 2,
             caption = "Conditional Probability Blood Pressure") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

knitr::kable(model[2]$tables$Cholesterol, digits = 2,
             caption = "Conditional Probability Cholesterol") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

quantitative.vars <- drugs[trainSet,] %>% dplyr::select(where(is.numeric))

mms=apply(quantitative.vars,
          MARGIN=2,
          function(x) unlist(by(x,drugs[trainSet,]$Drug,mean)))
sds=apply(quantitative.vars,
          MARGIN=2,
          function(x) unlist(by(x,drugs[trainSet,]$Drug,sd)))

par(mfrow=c(1,2))
colnames.var <- colnames(mms)
colnames.var[2] <- "Sodium-to-Potassium"
for(i in 1:2){
  rrx=range(c(mms[,i]+3*sds[,i], mms[,i]-3*sds[,i]))
  rry=c(0,max(dnorm(mms[,i], mms[,i], sds[,i]+0.0001)))
  plot(0,1, type="n", xlab=colnames.var[i],
       ylab="Normal Density", xlim=rrx, ylim=rry)
  ss=seq(rrx[1],rrx[2],length.out = 100)
  for(j in 1:5) points(ss,dnorm(ss,mms[j,i],sds[j,i]),type="l",col=j)
  if(i==1)
    legend(x="topleft",
           legend=levels(factor(drugs[trainSet,]$Drug)),
           fill = c(1:5))
  else
    legend(x="topright",
           legend=levels(factor(drugs[trainSet,]$Drug)),
           fill = c(1:5))
}

header = names(predict(model,drugs[testSet,],type="raw")[1,])
```

```
row1 = round(predict(model,drugs[testSet,],type="raw")[1,], digits=2)
row7 = round(predict(model,drugs[testSet,],type="raw")[7,], digits=2)
predicts.df <- rbind(row1, row7)
colnames(predicts.df) <- header

knitr::kable(predicts.df, digits = 3,
             caption = "Predictions: Rows 1 and 7") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

#Maximum a posteriori
drug_prediction=predict(model,drugs[testSet,])
#Now we see the drug prediction for each variable

#Validate the model
confussion.matrix <- table(drug_prediction, drugs$Drug[testSet])
knitr::kable(confussion.matrix, caption = "Confussion matrix") %>%
  column_spec(1, bold=TRUE) %>% row_spec(0, bold=TRUE) %>%
  kable_styling(latex_options = "HOLD_position")
```