Naive Bayes

VAZGEN TADEVOSYAN

May 13, 2019

You are required to submit both Markdown and PDF files. Problem 1 (10 pt.) a. Explain the Naive Bayes algorithm. What are the prior, posterior, class-conditional probabilities? Why do we need to maximize only the numerator term?

Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class.

Suppose following formula.

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

We are trying to get probability that observation is class "c" given given predictor "x". There is interest only in the numerator of that fraction, because the denominator does not depend on C C and the values of the features $ x_{i}$ are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model.

The posterior probability, in the context of a classification problem, can be interpreted as: "What is the probability that a particular object belongs to class i given its observed feature values?" In our case posterior probability is $P(c|x)$

Prior Probability In the context of pattern classification, the prior probabilities are also called class priors, which describe "the general probability of encountering a particular class."

In our case prior probability is $P(c)$

$P(x|c)$ is the likelihood which is the probability of predictor given class class-conditional probability.

b. We have the dataset from figure1. Suppose we are given a test record with the following attribute set: X: (Hone Owner: Yes, Marital Status: Single, Annual Income: 158). Classify the records Based on Naive Bayes. Show all calculations of all class-conditional, prior, posterior probabilities.

```r
library(gtools)
HomeOwner<-c("Yes","No","No","Yes","No","No","Yes","No","No","No","Yes","Yes","Yes","No")
Marital_Status<-c("Single","Married","Single","Married","Divorced","Married","Divorced","Single","Marri
Annul_Income<-c(125,100,70,120,150,60,220,85,75,90,180,200,250,50)
Annul_Income<-quantcut(Annul_Income,3,labels=c("low","medium","high"))




Default<-c("No","No","No","No","Yes","No","No","Yes","No","Yes","Yes","No","No","Yes")
table(Default)
```

```
## Default
##  No Yes
##   9   5
```

```r
addmargins(table(Default,HomeOwner))
```

```
##        HomeOwner
## Default No Yes Sum
##     No   4   5   9
##     Yes  4   1   5
##     Sum  8   6  14
```

```
addmargins(table(Default,Marital_Status))
```

```
##        Marital_Status
## Default Divorced Married Single Sum
##     No         2       5      2   9
##     Yes        2       1      2   5
##     Sum        4       6      4  14
```

```
addmargins(table(Default,Annul_Income))
```

```
##        Annul_Income
## Default low medium high Sum
##     No    3      3    3   9
##     Yes   2      1    2   5
##     Sum   5      4    5  14
```

```
P_defY_Homeyes<-1/5
P_defY_Marit_Sing<-2/5
P_defY_Inc_High<-2/5
P_defY<-5/14
h_yes<-prod(P_defY_Homeyes,P_defY_Marit_Sing,P_defY_Inc_High,P_defY)

P_defN_Homeyes<-5/9
P_defN_Marit_Sing<-2/9
P_defN_Inc_High<-3/9
P_defN<-9/14
h_no<-prod(P_defN_Homeyes,P_defN_Marit_Sing,P_defN_Inc_High,P_defN)
max(h_no,h_yes)
```

```
## [1] 0.02645503
```

```
h_yes/sum(h_no,h_yes)+h_no/sum(h_no,h_yes)## indicates we calculated right.
```

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

The observation's default is predicted as no

c. Bring an example of conditionally independent random variables.

A(I am late for class) and {Levon is late for class} B are conditionally independent given {car crash in Baghramian} C if and only if, given knowledge that crash occurs, knowledge of whether I am late or not provides no information on the likelihood of Levon is being late , and knowledge of whether Levon is late provides no information on the likelihood of me being late.

Problem 2 (10 pt.) Data set (Well Switching in Bangladesh) relevant for this problem can be found in the R package "carData".

```
library(carData)
data("Wells")
head(Wells)
```

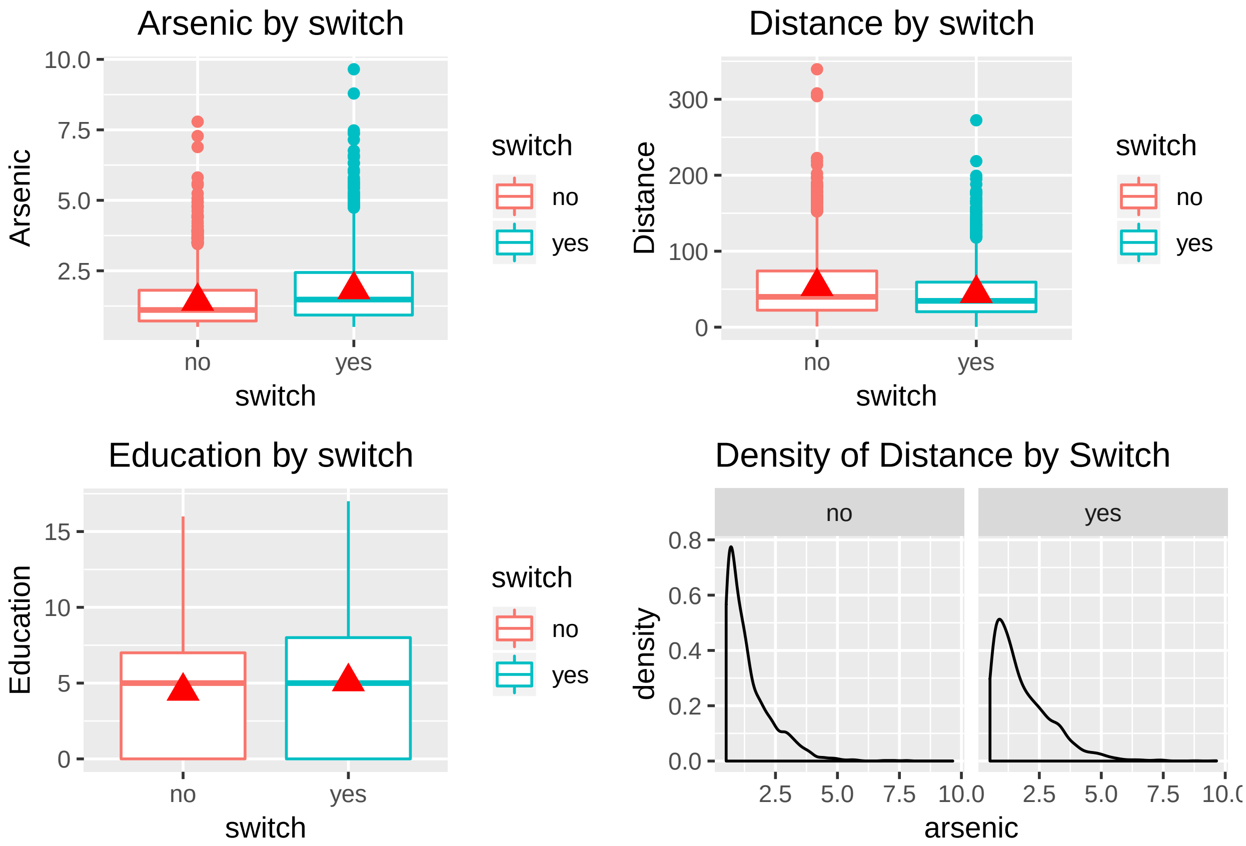
```
##   switch arsenic distance education association
## 1    yes    2.36   16.826         0          no
## 2    yes    0.71   47.322         0          no
## 3     no    2.07   20.967        10          no
## 4    yes    1.15   21.486        12          no
## 5    yes    1.10   40.874        14         yes
## 6    yes    3.90   69.518          9          yes
```

a. Choose the predictors and the target variable. Make appropriate visualization (boxplot and density plots) for understanding the relationship between them. Comment on it.

```
library(ggplot2)

library(gridExtra)
g1 = ggplot(data = Wells, aes(y = arsenic, x = switch,color = switch))+geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=4)+
  labs(title ="Arsenic by switch " ,color =  "switch",y='Arsenic',x="switch")+
  theme(plot.title = element_text(hjust = 0.5))
g2 = ggplot(data = Wells, aes(y = distance, x = switch,color = switch))+geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=4)+
  labs(title ="Distance by switch " ,color =  "switch",y='Distance',x="switch")+
  theme(plot.title = element_text(hjust = 0.5))
g3 = ggplot(data = Wells, aes(y = education, x = switch,color = switch))+geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=4)+
  labs(title ="Education by switch " ,color =  "switch",y='Education',x="switch")+
  theme(plot.title = element_text(hjust = 0.5))
g4=ggplot(Wells,aes(arsenic))+geom_density()+facet_grid(.~switch)+ggtitle("Density of Distance by Switcl

grid.arrange(g1, g2, g3,g4,nrow=2)
```



```
colnames(Wells)
```

```
## [1] "switch"      "arsenic"     "distance"    "education"   "association"
```

It is seen from the plot that numeric variables are not significantly different by switch variable however Arsenic and Education tend to be higher for an switched household than unswitched. From the Density plot it is seen that switch yes is wider than switch no. Switch yes has more variance than switch no.

b. Calculate prior probabilities using the naiveBayes() function.

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:gtools':
##
##     permutations
```

```
library(caret)
```

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

```
set.seed(1)
index<- createDataPartition(Wells$switch, p=0.8, list=F)

train<-Wells[index,]
test<-Wells[-index,]

model<-naiveBayes(switch~., data=train, laplace = 1)
names(model)
```

```
## [1] "apriori" "tables"  "levels"  "call"
```

```
model$apriori
```

```
## Y
##   no  yes
## 1027 1390
```

```
p_yes<-model$apriori[2]/sum(model$apriori)
p_no<-model$apriori[1]/sum(model$apriori)
print(paste("Prior probabity for switch yes is",p_yes))
```

```
## [1] "Prior probabity for switch yes is 0.575093090608192"
```

c. Describe and interpret the table from your output of your model (for one categorical, one numeric variables)

```
model$tables$association
```

```
##      association
## Y            no       yes
##   no  0.5558795 0.4441205
##   yes 0.5797414 0.4202586
```

Probability of switch_yes given association no is 0.5797414

```
model$tables$distance
```

```
##      distance
## Y          [,1]     [,2]
##   no  54.10339 42.91848
##   yes 44.34009 34.14521
```

For numeric variable we should calulate z-score. Mean and Standard Deviation is given per group in table. For example,suppose new observation come it's distance is 40. Probability of that it switched yes should be calculated as follows We should get z score and then get probability but in R we easily use pnorm, ouput will be the same.

```
pnorm(40,mean = 44.84026,sd = 34.42759)
```

```
## [1] 0.4440959
```

d. Make a prediction using the Naive Bayes Algorithm. Can the prior probabilities influence on the final result? Check the goodness of prediction on test data.

```
predicted<-predict(model,test,type = "raw")
pred_test<-ifelse(predicted[,2]>.5,"yes","no")
pred_test<-as.factor(pred_test)
confusionMatrix(pred_test,test$switch,positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no   68  56
##        yes 188 291
##
##                Accuracy : 0.5954
##                  95% CI : (0.555, 0.6348)
##     No Information Rate : 0.5755
##     P-Value [Acc > NIR] : 0.1718
##
##                   Kappa : 0.1118
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8386
##             Specificity : 0.2656
##          Pos Pred Value : 0.6075
##          Neg Pred Value : 0.5484
##              Prevalence : 0.5755
##          Detection Rate : 0.4826
##    Detection Prevalence : 0.7944
##       Balanced Accuracy : 0.5521
##
##        'Positive' Class : yes
##
```

Overall model accuracy is 0.6368 , which is very good if we take the portion of classes into account (43% 57%).Sensitiviy score is 0.85 which is better than our benchmark for positive case (0.57) We are interested in predicting yes so we should pay attention to sensitivity which measures the proportion of actual positives that are correctly identified as such.

```
library(pROC)
```

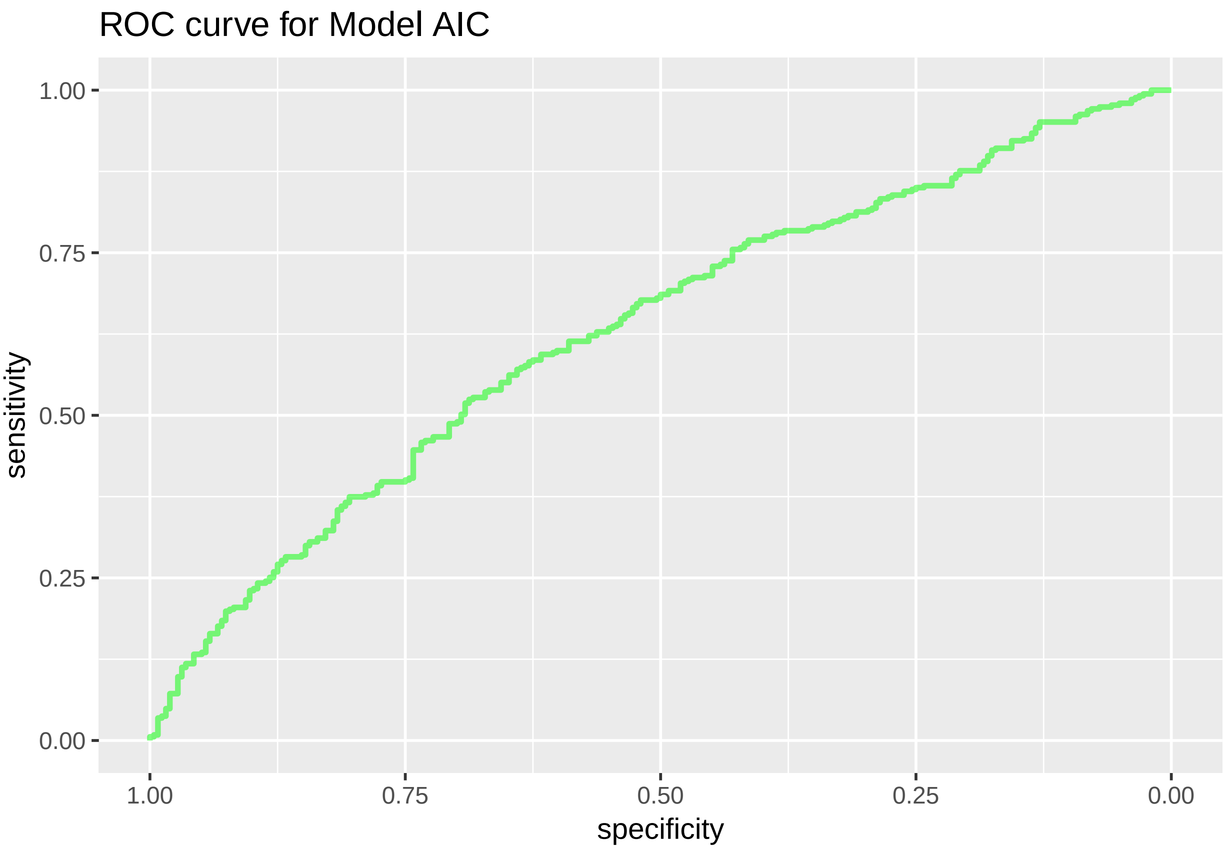
```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
rrr1<-roc(test$switch,predicted[,2])
rrr1
```

```
##
## Call:
## roc.default(response = test$switch, predictor = predicted[, 2])
##
## Data: predicted[, 2] in 256 controls (test$switch no) < 347 cases (test$switch yes).
## Area under the curve: 0.6303
```

```
g2<-ggroc(rrr1,alpha = 0.5, colour = "green", linetype = 1, size = 1)+ggtitle("ROC curve for Model AIC")
g2
```



Area under the curve is 0.6758

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

Yes Prior probability can influence the final result as it is part of the equation.

1