Machine learning II
Master Data Science
Winter Semester 2023/24

Prof. Tim Downie

Workshop 5
The Naive Bayes Classifier in R

with solutions

The code for this Workshop is given as text in this file, rather than in a script file. You should be able to copy and paste the commands into *RStudio*.

Exercise 1 Naive Bayes Algorithm: Revision

Read through the first part of the website "Naïve Bayes classification in R"[1] by Zhongheng Zhang. This introduces the Naive Bayes Classifier including a medical example followed by fitting a Naive Bayes Model to the Titanic data. Do not run the code in this website; this is covered in the next exercise.

Exercise 2 A simple Example

We will repeat the Titanic example in Zhang using slightly different code given below. The function `naiveBayes()` is part of the R package `e1071`. The obscure package name originates from an internal department code at Vienna University! It includes functions for other ML routines, such as for "support vector machines" from next week. If you are working on your own computer you will probably need to install this package.

```
#install.packages("e1071") #Uncomment if not already installed.
library(e1071)
data(Titanic)
Titanic
```

Note that the data is in the form of a 3-dimensional array that contains frequencies. It is possible to pass frequency data into `naiveBayes()` but it is much easier to analyse the results, when it is in the classic 'data frame' form of a data frame with one row per passenger.

```
#Convert to a data frame and inspect it
Titanic_df<-as.data.frame(Titanic)
View(Titanic_df)
#Titanic_df has one row for each combination of Class,Sex,Age and Survived,
#along with the frequency for this combination
```

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4930525/

```
#Expand the number of rows to correspond to the frequencies
#This repeats each combination equal to the frequency of each combination
repeating_sequence<-rep.int(seq_len(nrow(Titanic_df)), Titanic_df$Freq)


#Create the new dataset by repeating the rows by the frequency
Titanic_df<-Titanic_df[repeating_sequence,]
#We no longer need the frequency column so drop this feature
Titanic_df$Freq<-NULL
```

Fit the Naive Bayes model using the usual R model fitting notation, and obtain the model output.

```
NB_Titanic<-naiveBayes(Survived ~., data=Titanic_df)
NB_Titanic
```

N.B. In the output, the "A-priori probabilities" are the counts for Survived divided by the number of passengers, but the values stored in `NB_Titanic$apriori` are the counts. As we will be using this prior probabilities a lot, store them in an object.

```
TitanicPriors<-prop.table(NB_Titanic$apriori)
```

Note that the conditional probabilities output above are just the observed marginal frequencies from the data. These could be obtained by getting a table of absolute frequencies with `table()` and then calculating the relative row frequencies using `prop.table(,1)`

```
prop.table(table(Titanic_df$Survived))
prop.table(table(Titanic_df$Survived,Titanic_df$Class),1)
prop.table(table(Titanic_df$Survived,Titanic_df$Sex),1)
prop.table(table(Titanic_df$Survived,Titanic_df$Age),1)
```

The posterior probabilities for a second class passenger and a second class child passenger, as shown in Lecture 12 are obtained using:

```
###Given second class passenger
#prior
TitanicPriors
#proportional posterior probability
TitanicPriors*NB_Titanic$tables$Class[,2]
#actual posterior probability (adds up to 1)
prop.table(TitanicPriors*NB_Titanic$tables$Class[,2])

###Given second class passenger and child
TitanicPriors*NB_Titanic$tables$Class[,2]*NB_Titanic$tables$Age[,1]
#actual posterior probability (adds up to 1)
prop.table(TitanicPriors*NB_Titanic$tables$Class[,2]*NB_Titanic$tables$Age[,1])
```

► Obtain the posterior probabilities given a third class adult male, and state the predicted value for such

a passenger.

```
> prop.table(TitanicPriors*NB_Titanic$tables$Class[,3]*
+ NB_Titanic$tables$Age[,2]*NB_Titanic$tables$Sex[,1])
Y
       No       Yes
0.8466171 0.1533829
```

Predicted value is "No" (with a high probability).

The `predict()` function gives the model predictions for `Survived` based on the posterior probabilities given the information on each passenger. The confusion matrix is then printed, with the sensitivity, specificity and the miscalassification rate.

```
NB_Preds<-predict(NB_Titanic,Titanic_df)
#Confusion matrix to check accuracy
confmat<-table(NB_Preds,Titanic_df$Survived)
confmat

#sensitivity
confmat[2,2]/sum(confmat[,2])
#specificity
confmat[1,1]/sum(confmat[,1])
#misclassification rate (1-accuracy)
1-sum(diag(confmat))/sum(confmat)
```

For these data the NB model gives poor sensitivity but the specificity is good.

To obtain a ROC plot and the AUC for this classification we need the predicted posterior probabilities for each passenger. The function `Predict()` for an object of class `naiveBayes` will output these when specifying the argument `type="raw"`. The relevant code is

```
require(pROC)
predprob<-predict(NB_Titanic,Titanic_df,type="raw")[,2]
roc.obj1 <- roc(Titanic_df$Survived,predprob)
ggroc(roc.obj1)
auc(roc.obj1)
```

Exercise 3 The IBM attrition data

The `attrition` data set in the `modeldata` library, contains data on "employee attrition", which means employees leaving the company.

Load the `modeldata` and `rsample` packages, installing them if necessary. Make the dataset visible using `data(attrition)`. Read the very short help page for the data set `attrition`.

Data preprocessing There are two numeric variables which are coded as numbers but are really factor variables. Convert these variables, then define a training and test data set using the function `initial_split`, `training` and `testing`, which are part of the package `rsample`.

```
attrition$JobLevel<-as.factor(attrition$JobLevel)
attrition$StockoptionLevel<-as.factor(attrition$StockOptionLevel)
set.seed(1)
split <- initial_split(attrition, prop = .7, strata = "Attrition")
train <- training(split)
test  <- testing(split)
prop.table(table(train$Attrition))
prop.table(table(test$Attrition))
```

Notice that the training and test datasets have been stratified so that both contain the same proportion of Yes to No.

► Use Exercise 2 as a guide to fit a naive Bayes Model to the test data dependent on the following subset of variables:

`Attrition~Age+DailyRate+DistanceFromHome+HourlyRate+MonthlyIncome+MonthlyRate`

► Obtain the predictions for this model, using the test data. It is clear that not enough elements are predicted to be Yes. Repeat the model with all of the available variables. The sensitivity increases to 59% but the specificity and accuracy are worse.

Obtain an ROC plot for the two NB prediction models and the AUC values. Hint: to put two ROC curves on one diagram use `ggroc(list(roc.obj1,roc.obj2))`

Which model is better based on ROC and AUC? Notice that choosing the model with the highest accuracy will chose the other model.[2]

Exercise 4 Comparison with LDA

The MASS library contains the function `lda()` to fit a linear discriminant analysis. Use this to fit a similar model to the attrition data including the ROC and AUC. Compare the results with the Naive Bayes results.

Hints

The `lda()` function gives a warning that there is collinearity in the predictor variables. This means that that two or more variables are linearly dependent. This is a problem for the parameter estimates but has no effect on the predicted values.

[2] A more comprehensive NB analysis of these data, using a different style of R-code can be found at the website: `http://uc-r.github.io/naive_bayes`.

The default output from `predict()` gives a matrix of probabilities and the predicted classes. To access just the predicted classes for the classification matrix use:
`predict()$class` To access just the predicted probabilities for the ROC curve use:
`predict()$posterior[,2]`

Written exercises to do at home

Exercise 5 Conditional independence
Use the elementary definitions in probability theory of stochastic independence and conditional probability, to solve the following exercises.

(a) Show that

$$P(X_1|X_2) = P(X_1)$$

is equivalent to

$$P(X_1 \cap X_2) = P(X_1)P(X_2)$$

and

$$P(X_2|X_1) = P(X_2)$$

(b) Show that

$$P(X_1|Y, X_2) = P(X_1|Y)$$

is equivalent to

$$P(X_1 \cap X_2|Y) = P(X_1|Y)P(X_2|Y)$$

and

$$P(X_2|Y, X_1) = P(X_2|Y)$$

Exercise 6 Conditional independence
A box contains two coins: a regular fair coin M_1 with Heads H and Tails T, and one trick two-headed coin M_2, i.e. $P(H|M_2) = 1$. A coin is chosen at random 50-50 from the box and this coin is tossed twice.

We define the following events:

- $A =$ First coin toss results in an H
- $B =$ Second coin toss results in an H

Calculate

(a) $P(A|M_1)$, $\frac{1}{2}$

(b) $P(B|M_1)$, $\frac{1}{2}$

(c) $P(A \cap B|M_1)$. You may assume that *if* we know that we are using a fair coin that the result of the second toss is independent from the first. $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

(d) $P(A|M_2)$, 1

(e) $P(B|M_2)$, 1

(f) $P(A \cap B|M_2)$. 1

(g) Use the law of total probability: $P(A) = P(A|M_1)P(M_1) + P(A|M_2)P(M_2)$ to calculate $P(A)$. $\frac{1}{2} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{3}{4}$

(h) Calculate $P(B)$ similarly, $\frac{1}{2} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{3}{4}$ and

(i) Use the law of total probability: $P(A \cap B) = P(A \cap B|M_1)P(M_1) + P(A \cap B|M_2)P(M_2)$ to calculate $P(A \cap B)$. $\frac{1}{4} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{5}{8}$

Confirm that A and B are NOT independent, but they are conditionally independent given which coin M_1 or M_2 is used.

$P(A) \cdot P(B) = \frac{9}{16}$,
$P(A \cap B) = \frac{5}{8} \Rightarrow P(A) \cdot P(B) \neq P(A \cap B)$, *A and B are not independent.*
$P(A|M_1) \cdot P(B|M_1) = \frac{1}{4}$, $P(A \cap B|M_1) = \frac{1}{4} \Rightarrow P(A|M_1) \cdot P(B|M_1) = P(A \cap B|M_1)$.

$P(A|M_2) \cdot P(B|M_2) = 1$, $P(A \cap B|M_2) = 1 \Rightarrow P(A|M_2) \cdot P(B|M_2) = P(A \cap B|M_2)$.

A and B are independent given M_1 or M_2