Statistical Learning, Homework #1

Veronica Vinciotti, Marco Chierici

Released: 22/03/2023. Due: 02/04/2023

This homework deals with classification methods. You should submit via Moodle a PDF file of the report, rendered directly from an RMarkdown source file (using `output: pdf_document`; see Guidelines) and not converted from any other output format.

You should write your report like a mini scientific paper. In particular, you should: introduce the analysis, discuss/justify the choices that you make, provide comments on the results that you obtain and draw some conclusions.

Please note that the **maximum allowed number of pages is 8**.

Guidelines

- Show only what is relevant to the analysis and the results that you obtain:
  - avoid displaying many lines of "boilerplate" code or output messages;
  - focus on highlighting the main parts of your code and output;
  - visualize/summarize the results with a selection of informative tables and figures.
- Note that by default RMarkdown will repeat (echo) the R code of a code chunk in the final output.
- To display only the output of a code chunk without echoing the corresponding code, use the option `echo=FALSE` in the code chunk.

~~~
```{r, echo=FALSE}
my_function <- function(a, b) {
    # ...
    # ...
}

glm.fit <- glm(Output ~ ., data=dataf, family=binomial)
# ...
```
~~~

- To add a plot, just insert the plotting code in a code chunk: then, RMarkdown will output the resulting plot below the corresponding chunk. You can use code chunk options `fig.width` and `fig.height` to customize the size of plots, `fig.align` to change the horizontal alignment of a plot, and `fig.cap` to add a caption.

~~~
```{r, fig.width=10, fig.height=4, fig.cap="Figure 1: your caption here.", fig.align="center"}
...
plot(...)
...
```
~~~

- Use the following options in your RMarkdown header:

```
---
title: <title>
date: <date>
author: <name and ID>
output:
  pdf_document:
    latex_engine: xelatex
---
```

- Include the following code chunk at the beginning of your RMarkdown file, and adjust `width.cutoff` to avoid that long lines of code go beyond the margins in the output file (typical values are 50, 60, 70, 80):

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(warning=FALSE,
                      message=FALSE,
                      tidy.opts=list(width.cutoff = 80),
                      tidy = TRUE)
```
~~~

Markdown cheat sheet

Markdown is a combination of regular text (like this), combined with tags that change the way the text is formatted. Here are the most common formatting tags:

- To make headers, use one or more # (pound) symbols at the beginning of the line: the number of # dictates the level of the header;
- To make text *italic*, wrap the text between *;
- To make text **bold**, wrap the text between **;
- To make numbered lists, just use a number followed by a dot (`1.`) at the beginning of each line:
    1. First element
    2. Second element
    3. Third element
- To make unordered lists, use a dash (-) or an asterisk (*) followed by a space at the beginning of each line
    - item
    - item
    - item

Additional resources

- Ten simple (empirical) rules for writing science, Cody J. Weinberger, James A. Evans, and Stefano Allesina, PLOS Computational Biology 11(4): e1004205, 2015.
- RMarkdown intro and reference guide

Assignment

The data set for this homework is available at breastfeed.Rdata. The data come from a study conducted at a UK hospital, investigating the possible factors affecting the decision of pregnant women to breastfeed their babies. The outcome of the study could aid in targeting breastfeeding promotions towards women with a lower probability of choosing it.

For the study, 135 expectant mothers were asked what kind of feeding method they would use for their coming baby. The responses were classified into two categories (variable **breast** in the dataset): the first category (coded 1) includes the cases "breastfeeding", "try to breastfeed" and "mixed breast- and bottle-feeding", while the second category (coded 0) corresponds to "exclusive bottle-feeding". The possible factors, that are available in the data, are the advancement of the pregnancy (**pregnancy**), how the mothers were fed as babies (**howfed**), how the mother's friend fed their babies (**howfedfr**), if they have a partner (**partner**), their age (**age**), the age at which they left full-time education (**educat**), their ethnic group (**ethnic**) and if they have ever smoked (**smokebf**) or if they have stopped smoking (**smokenow**). All of the factors are two-level factors. The first listed level of each factor is used as the reference (and coded with 0).

In your report, you should:

- Explore the data: what is the nature of the response variable (`breast`)? Are there potential issues with any of the predictors? Do you need pre-processing or can you proceed with the data as it is?
- Split the data into (reproducible) training and test sets. Given the class imbalance, you could aim for sets that have the same imbalance with respect to the outcome variable. In order to do this, you could either perform the splitting manually on each class, or use dedicated functions (for example, `caret::createDataPartition(labels, p=train_size)`, with `train_size` a number between 0 and 1 representing the percentage of data you would like to use for training.
- Fit the following GLM model:

$$\begin{aligned}\text{logit}(\text{E}(\text{breast})) = \beta_0 &+ \beta_1\text{pregnancy} + \beta_2\text{howfed} + \beta_3\text{howfedfr}\\ &+ \beta_4\text{partner} + \beta_5\text{age} + \beta_6\text{educat} + \beta_7\text{ethnic} + \beta_8\text{smokenow} + \beta_9\text{smokebf}\end{aligned}$$

  Discuss the `summary` and the interpretation of the model in the context of the study.
- Fit a k-nn classifier, by performing a careful selection of the tuning parameter k.
- Fit a Naïve Bayes classifier.
- Evaluate the performance of the methods and compare the results.

Hints

In R, the Naïve Bayes (NB) classifier is included in the package `e1071`. The syntax is `naiveBayes(formula, data)` for model fitting, and the usual `predict(fit, newdata)` for predicting new data. A fitted `naiveBayes` object stores the conditional probabilities for each feature, together with the *a priori* probabilities. To compute the posterior probabilities, you call `predict()` with the argument `type="raw"`.