[DS-06] Logistic regression

Miguel-Angel Canela
Associate Professor, IESE Business School

Classification

Classification is Machine Learning most frequent job. A classification model allocates instances to two or more prespecified groups or classes. This course only covers **binary classification**, in which there are only two classes, which are typically called **positive** and **negative**. The labels positive/negative should be assigned so that they favor your intuition. If you leave this to the computer, it may call positive what you call negative.

In the business context, two classics are:

- **Churn modeling**. A telephone company classifies its customers as either churners or non-churners (see the example). The class has two values, "churn" and "no churn".
- **Credit scoring**. A bank classifies credit applications as either "good" or "bad".

Let me assume, to simplify, a case of binary classification, and that the classes to be predicted are coded with a dummy (1/0). Although it is not equally evident in the various methods and implementations, a binary classification method produces a **predictive score** for every instance. The score is a number in the 0-1 range, which is later transformed into a **predicted class**, based on a **cutoff** value. The instances whose scores exceed the cutoff are classified as positive and the rest as negative.

The simplest approach would be to set 0.5 as the cutoff. Nevertheless, it can be replaced by another value with a better performance. In a business application, the choice of the cutoff may be based on a cost/benefit analysis. Specialized software can find the **optimal cutoff** for a user-specified cost matrix.

Classification methods

Classification models can be obtained with various methods. They differ in the way in which they calculate the scores.

- **Logistic regression**. The score is calculated by means of a (nonlinear) regression equation. This method is discussed in this note.
- **Neural networks**. The regression equation of the above paragraph is replaced by a neural network, which is a model which combines several equations.
- **Decision trees**. The scores are derived from a tree. We enter the tree by the **root** and proceed along the branches until arriving to the **leaves**. At each **node**, the branching is based on the value of one variable. The same score is assigned to all the instances of the same leaf. The score is equal to the positive rate in that leaf.
- **Random forests**. There are many methods based on combining several decision trees. The random forest models are very popular, owing to their performance on big data sets.

How to evaluate a classification model

The evaluation of a classification model is usually based on a **confusion matrix**, obtained by crosstabulation of the actual class and the predicted class. Although there is not a universal consensus, in the confusion matrix the predicted class usually comes in the rows and the actual class in the columns.

Table 1 is an example of a confusion matrix of a churn model. The four cells of the table are referred to as **true positive** (TP = 114), **false positive** (FN = 91), **false negative** (FN = 369) and **true negative** (TN = 2,759), respectively.

TABLE 1. Confusion matrix

	Actual positive	Actual negative
Predicted positive	114	91
Predicted negative	369	2,759

The proportion of instances classified in the right way, frequently called the **accuracy**, would be

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{114 + 2759}{114 + 91 + 369 + 2759} = 86.2\%.$$

The proportion of right classification is not always the main criterion in the evaluation. Other measures which can be prioritized, depending on the application, are:

- The **true positive rate**, or proportion of right classification among the actual positives,

$$\text{TP rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{114}{114 + 369} = 23.6\%.$$

- The **false positive rate**, or proportion of wrong classification among the actual negatives,

$$\text{FP rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{91}{91 + 2759} = 3.2\%.$$

Not everybody agrees on this terminology. I use these terms as in Witten *et al* (2011). In a good model, the TP rate should be high and the FP rate low. Nevertheless, the relative importance given to these statistics depends on the actual application.

Logistic regression

Let me suppose that the two classes to be predicted are coded with a dummy (positive = 1, negative = 0) and that there is a collection of numeric variables $X_1, \ldots, X_k$ available for prediction (some of them may come from coding categorical variables). The logistic regression equation is

$$p = F\big(a + b_1\, X_1 + \cdots + b_k\, X_k\big).$$

Here, p is the predictive score, and F is the **logistic function**,

$$F(x) = \frac{1}{1 + \exp(-x)}\,.$$

Inverting the logistic function, the equation can be rewritten as

$$\log\left(\frac{p}{1-p}\right) = a + b_1\, X_1 + \cdots + b_k\, X_k.$$

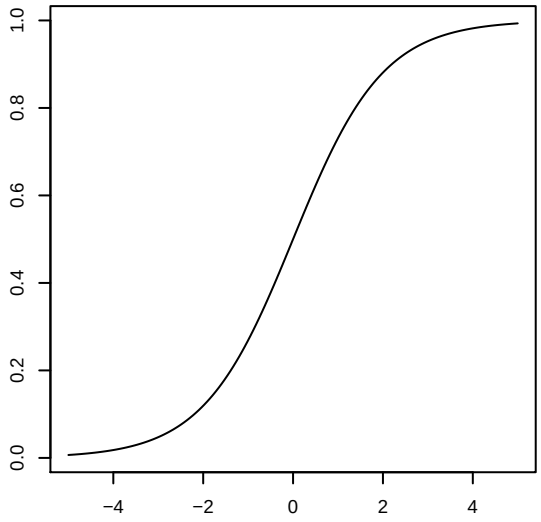

Figure 1. Logistic function

The graph of the logistic function has an inverted S shape, as shown in Figure 1. As given by the logistic function, the scores fall within the unit interval ($0 < p < 1$). So, the score can be regarded as the probability, or propensity, of an instance to have positive class. In logistic regression, this is obvious to the user.

A word of caution about the predictions of a logistic regression equation. First, since the equation involves a transformation, statistical software can offer us a choice of the scale of the predicted values. More specifically, in logistic regression, we can predict either p or $\log[p/(1-p)]$ (this happens in R). Second, sometimes, as in the Python package `scikit-learn`, the user can choose between predicting the score or ask directly for the predicted class (1/0).

References

1. MJ Crawley (2012), *The R Book*, Wiley. Free access at `ftp.tuebingen.mpg.de/pub/kyb/bresciani`.
2. F Provost & T Fawcett (2013), *Data Science for Business — What You Need to Know About Data Mining and Data-Analytic Thinking*, O'Reilly.
3. J VanderPlas (2017), *Python Data Science Handbook*, O'Reilly.
4. IH Witten, E Frank & MA Hall (2011), *Data Mining — Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.