(https://google.com/racialequity?authuser=0)

Logistic Regression: Calculating a Probability

ıted Time: 10 minutes

Many problems require a probability estimate as output. Logistic regression is an extremely efficient mechanism for calculating probabilities. Practically speaking, you can use the returned probability in either of the following two ways:

- "As is"
- Converted to a binary category.

Let's consider how we might use the probability "as is." Suppose we create a logistic regression model to predict the probability that a dog will bark during the middle of the night. We'll call that probability:

$$p(bark|night)$$

If the logistic regression model predicts a `p(bark | night)` of 0.05, then over a year, the dog's owners should be startled awake approximately 18 times:

$$\begin{aligned} startled &= p(bark|night) \cdot nights \\ &= 0.05 \cdot 365 \\ &= 18 \end{aligned}$$

In many cases, you'll map the logistic regression output into the solution to a binary classification problem, in which the goal is to correctly predict one of two possible labels (e.g., "spam" or "not spam"). A later module
(https://developers.google.com/machine-learning/crash-course/classification/video-lecture?authuser=0)
focuses on that.

You might be wondering how a logistic regression model can ensure output that always falls between 0 and 1. As it happens, a **sigmoid function**, defined as follows, produces output having those same characteristics:

$$y = \frac{1}{1 + e^{-z}}$$

The sigmoid function yields the following plot:

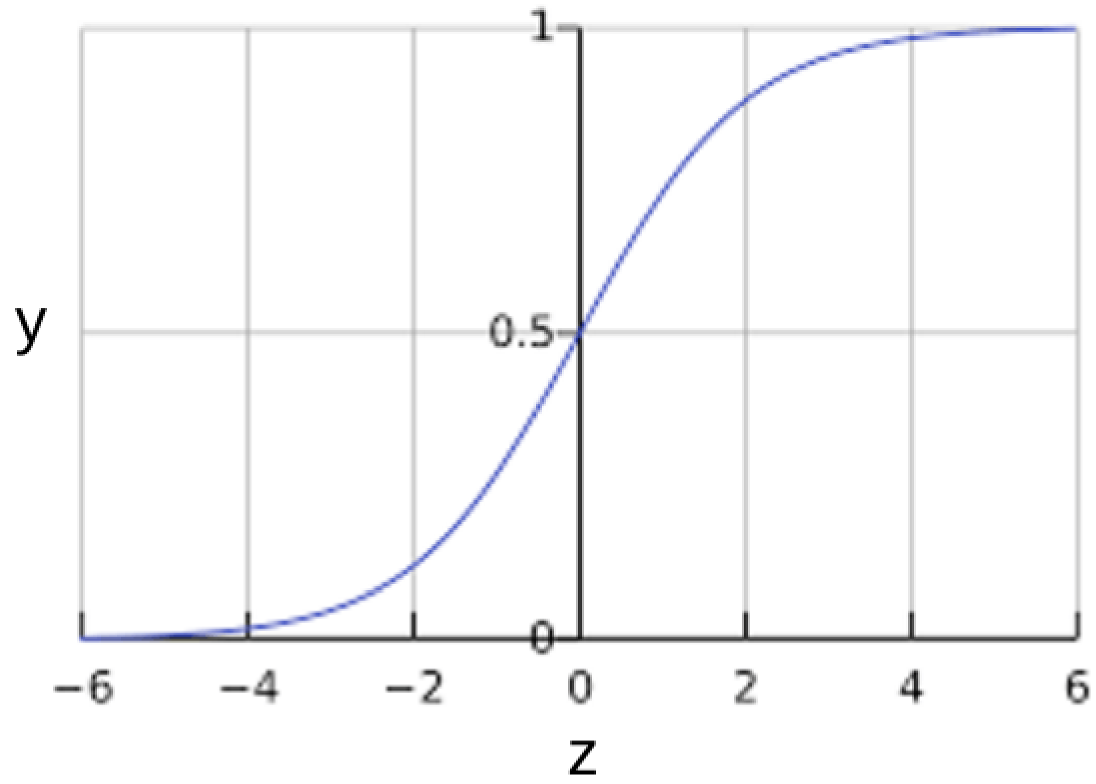

Figure 1: Sigmoid function.

If z represents the output of the linear layer of a model trained with logistic regression, then sigmoid(z) will yield a value (a probability) between 0 and 1. In mathematical terms:

$$y' = \frac{1}{1 + e^{-(z)}}$$

where:

- y' is the output of the logistic regression model for a particular example.
- $z = b + w_1x_1 + w_2x_2 + \ldots + w_Nx_N$
    - The *w* values are the model's learned weights, and *b* is the bias.
    - The *x* values are the feature values for a particular example.

Note that *z* is also referred to as the *log-odds* because the inverse of the sigmoid states that z can be defined as the log of the probability of the "1" label (e.g., "dog barks") divided by the probability of the "0" label (e.g., "dog doesn't bark"):

$$z = log(\frac{y}{1-y})$$

Here is the sigmoid function with ML labels:

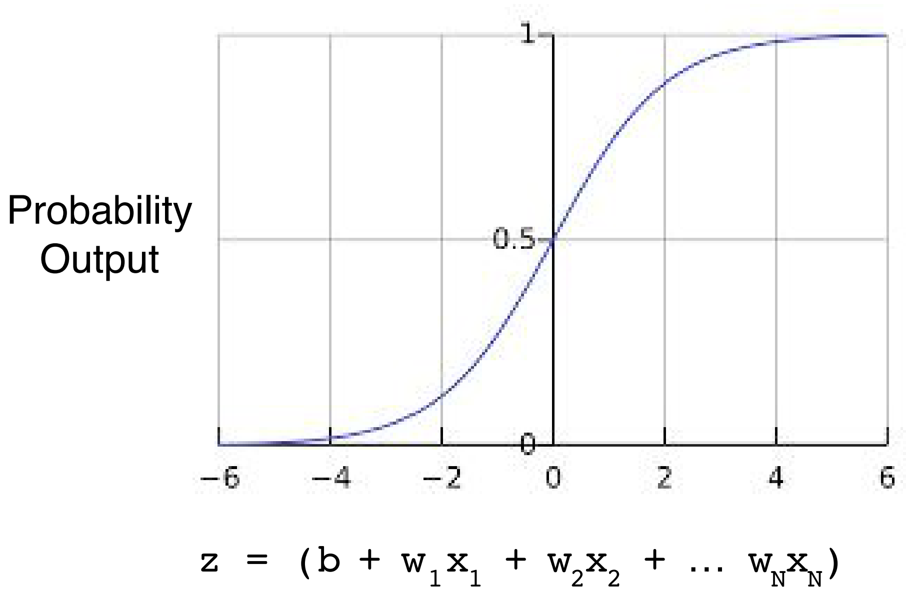

Figure 2: Logistic regression output.

\+ **Click the plus icon to see a sample logistic regression inference calculation.**

Suppose we had a logistic regression model with three features that learned the following bias and weights:

- b = 1
- $w_1 = 2$
- $w_2 = -1$
- $w_3 = 5$

Further suppose the following feature values for a given example:

- $x_1 = 0$
- $x_2 = 10$
- $x_3 = 2$

Therefore, the log-odds:

$$b + w_1x_1 + w_2x_2 + w_3x_3$$

will be:

$$(1) + (2)(0) + (-1)(10) + (5)(2) = 1$$

Consequently, the logistic regression prediction for this particular example will be 0.731:

$$y' = \frac{1}{1 + e^{-(1)}} = 0.731$$

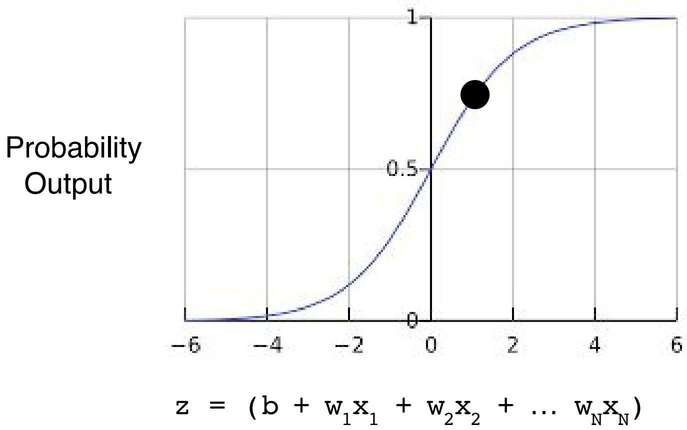

Figure 3: 73.1% probability.

rms

- ary classification ://developers.google.com/machine-ıg/glossary?authuser=0#binary_classification)
- ımoid function ://developers.google.com/machine-ıg/glossary?authuser=0#sigmoid_function)
- logistic regression (https://developers.google.com/machine-learning/glossary?authuser=0#logistic_regression)

Help Center (https://support.google.com/machinelearningeducation?authuser=0)

Previous
← Video Lecture (https://developers.google.com/machine-learning/crash-course/logistic-regression/video-lecture?authuser=0)

Next