Logistic Regression

Logistic Regression is designed to predict the probability of a given outcome

- Is popular for building models that discriminate between 2 possible outcomes including classification tasks

Calculating a probability with the sigmoid function

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:

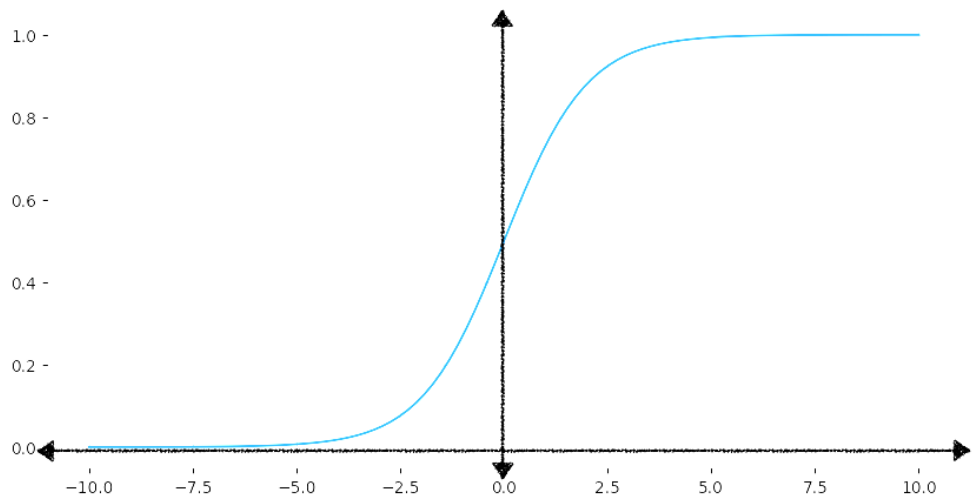
- Applied "as is." For example, if a spam-prediction model takes an email as input and outputs a value of `0.932`, this implies a `93.2%` probability that the email is spam.
- Converted to a **binary category** such as `True` or `False`, `Spam` or `Not Spam`.

Sigmoid Function

A sigmoid function is a function that always represents a probability value between 0 and 1. The equation is represented as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 1 shows the corresponding graph of the sigmoid function.



Loss and Regularisation

Logistic regression models are trained using the same process as **linear regression** models, with two key distinctions:

- Logistic regression models use **Log Loss** as the loss function instead of **squared loss**.
- Applying regularization is critical to prevent **overfitting**.

Log Loss

In the Linear regression module, you used **squared loss** (also called L2 loss) as the **loss function**. Squared loss works well for a linear model where the rate of change of the output values is constant.
Instead, the loss function for logistic regression is
Log Loss. The Log Loss equation returns the logarithm of the magnitude of the change, rather than just the distance from data to prediction.

Regularisation in Logistic Regression

Regularization, a mechanism for penalizing model complexity during training, is extremely important in logistic regression modeling. Without regularization, the asymptotic nature of logistic regression would keep driving loss towards 0 in cases where the model has a large number of features.

Completion