Logistic Regression

ML/AI @ SSN Coding Club - Meet 4

Scan to ask questions anonymously

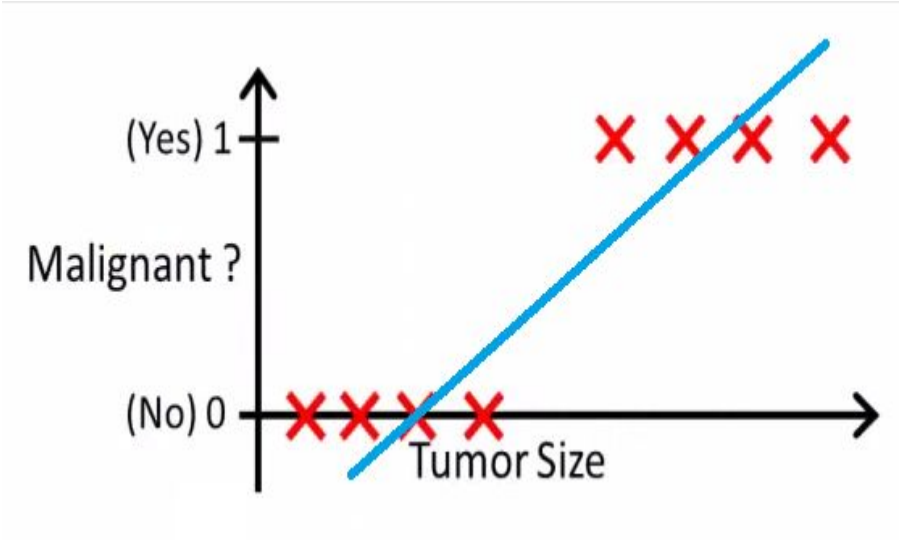
...or just ask away in Teams chat!

Overview

- Supervised learning algorithm to solve classification problems
- Called regression since the underlying technique is similar to linear regression
- Called logistic due to the use of the logit function (sigmoid function) in the algorithm, which we will see soon
- The dependent (target) variable is in the form of discrete classes
- Binary Logistic Regression will be dealt with, today
- Eg. Classifying mail as spam, non-spam

Why not Linear Regression?

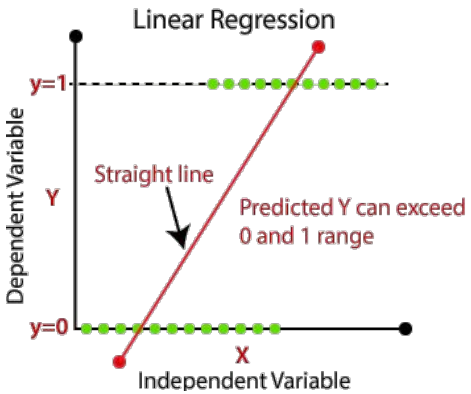
Do you think a linear regression model fits the binary data well?

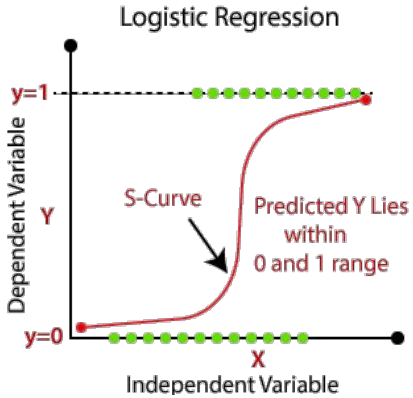


Sigmoid function

It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter “S”. The sigmoid function also called a logistic function.

Y = 1 / (1+e^(-z))

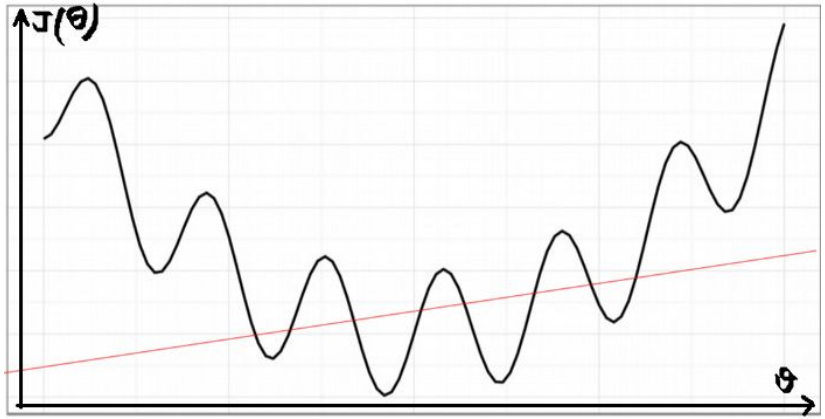




Cost/Loss function

Why we can’t use the same cost function we defined for linear regression?

If we try to use the cost function of the linear regression in ‘Logistic Regression’ then it would be of no use as it would end up being a non-convex function with many local minimums, in which it would be very difficult to minimize the cost value and find the global minimum.



Cost function for logistic regression

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = -\frac{1}{m} \sum \left[y^{(i)} \log(h\theta(x(i))) + \left(1 - y^{(i)}\right) \log(1 - h\theta(x(i))) \right]$$

Gradient Descent

The gradient of the aforementioned cost function comes out to have the same form as that of Linear regression, even after having a complex log loss error function

$$\begin{bmatrix} \frac{\partial J(\theta)}{\partial \theta_0} \\ \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{bmatrix} = \frac{1}{m} x^T (h(x) - y)$$

Predicting the output

- Find the value obtained for the inputs given to the model
- Say, the threshold value = 0.5 to classify each example
- If value >= 0.5, predict class 1
- If value < 0.5, predict class 0