Logistic Regression

If probability is greater than or Equal to pre-defined Threshold, then TRUE else FALSE. [Generally, Threshold = 0.5]

Logistic Regression can work with continous and discrete data.

→ Sigmoid Function

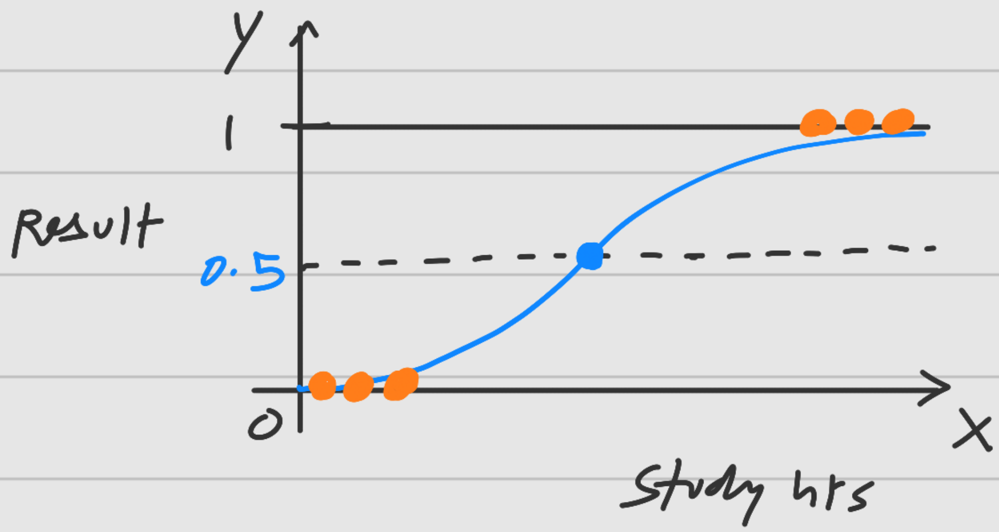
$$\left[y = \frac{1}{1+e^{-(a_0 + a_1 x)}} \right]$$

$y \in (0, 1)$

$a_0 \rightarrow$ intercept

$a_1 \rightarrow$ coefficient

$x \rightarrow$ independent var



Eg

x_0 (bias)	x_1 ✓	y ✓
1	2	0
1	4	0
1	6	1
1	8	1

Soln

we can use np.column_stack

$$X = \begin{array}{c} \begin{matrix} x_0 & x_1 \end{matrix} \\ \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 6 \\ 1 & 8 \end{bmatrix} \end{array} \qquad y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

initialize weights : $\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

learning rate = 0.1

$\{ z = \theta^T x \}$

$$h_\theta(x) = \sigma(z) = \frac{1}{1+e^{-\theta^T x}}$$

initially with $\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, all $z = 0$

$$\sigma(0) = \frac{1}{1+e^{(0)}} = 0.5$$

$$\Rightarrow h = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$ Predictions

→ maximum likelihood → cost f^n

log likelihood:

$$\log L(\theta) = y \log(h) + (1-y)\log(1-h)$$

Taking mean of it

Total cost to minimize:

$$J(\theta) = \frac{-1}{m} \sum \left[y_i \log h_i + (1-y_i)\log(1-h_i) \right]$$

$$y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad h = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$y_1 = 0 \qquad h_1 = 0.5$

$$\text{cost}1 = 0\log(0.5) + (1-0)\log(1-0.5)$$
$$= 0 + \log 0.5$$
$$= -0.693$$

[Base = e]

$y_2 = 0 \qquad h_2 = 0.5$

$$\text{cost}2 = -0.693$$

$y_3 = 1 \qquad h_3 = 0.5$

$$\text{cost}3 = 1\log(0.5) + 0\log(0.5)$$
$$= -0.693$$

$y_4 = 1 \quad h_4 = 0.5$

$\text{cost}4 = 1\log(0.5) + 0\log(0.5)$

$= -0.693$

$$J(\theta) = -\frac{1}{m}\sum\left[y_i \log h_i + (1-y_i)\log h_i\right]$$

$$= -\frac{1}{4}\left[-0.693 \times 4\right] = 0.693$$

→ Gradient Calculation

$$\nabla_\theta J = \frac{1}{m} X^T(h-y)$$

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 6 \\ 1 & 8 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad h = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$h - y = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$X^T(h-y) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 6 & 8 \end{bmatrix}\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

$$\nabla_\theta J = \frac{1}{4}\begin{bmatrix} 0 \\ -4 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

gradient

updating the weight...

$$\theta = \theta - (lr \times gradient)$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left(0.1 \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

→ Repeat for next iteration...

calculate $Z = X \cdot \theta = [0.2, 0.4, 0.6, 0.8]$

compute new $h = \sigma(Z)$

compute new cost

" new gradient

update θ again

Till 1000 iteration (convergence)