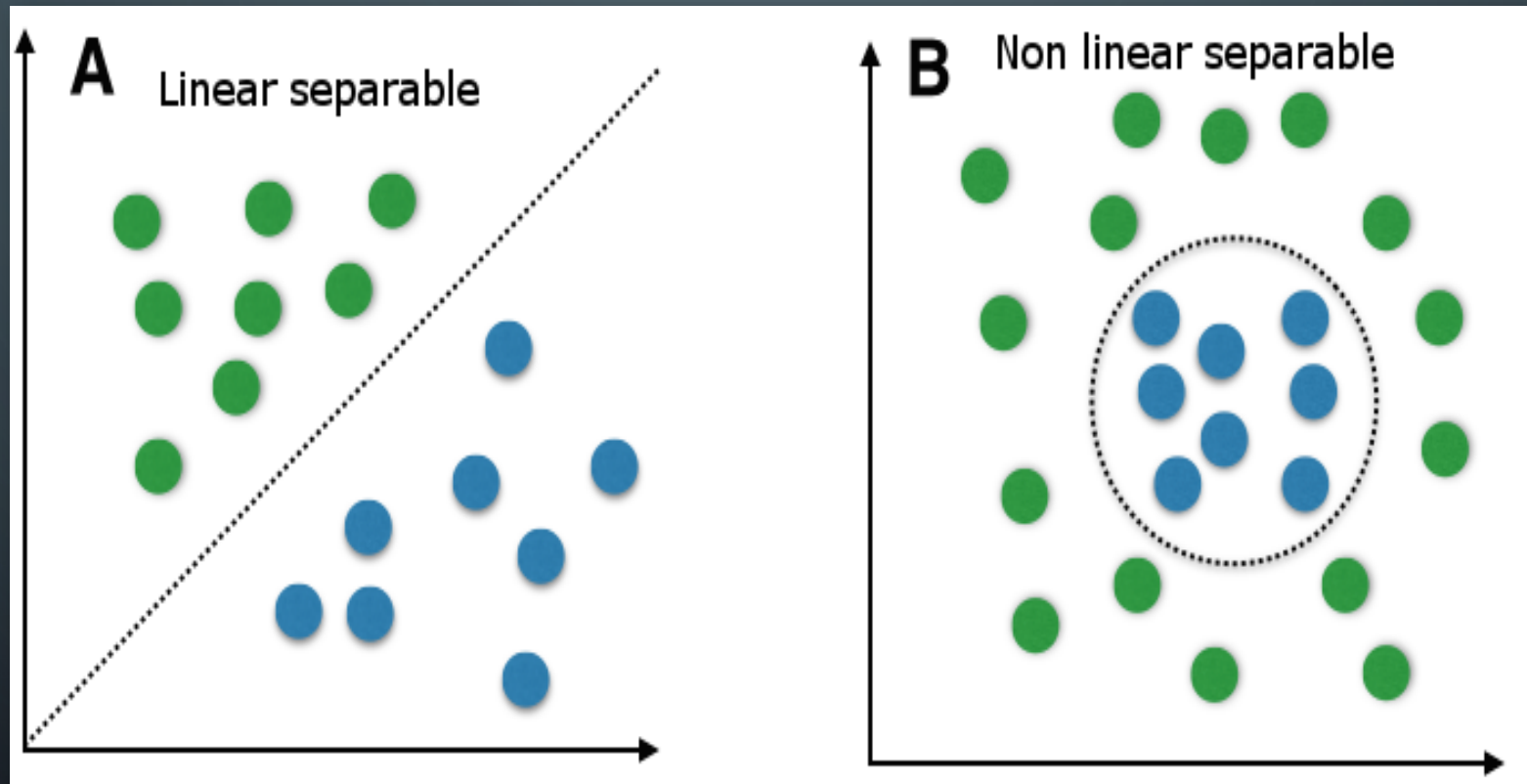




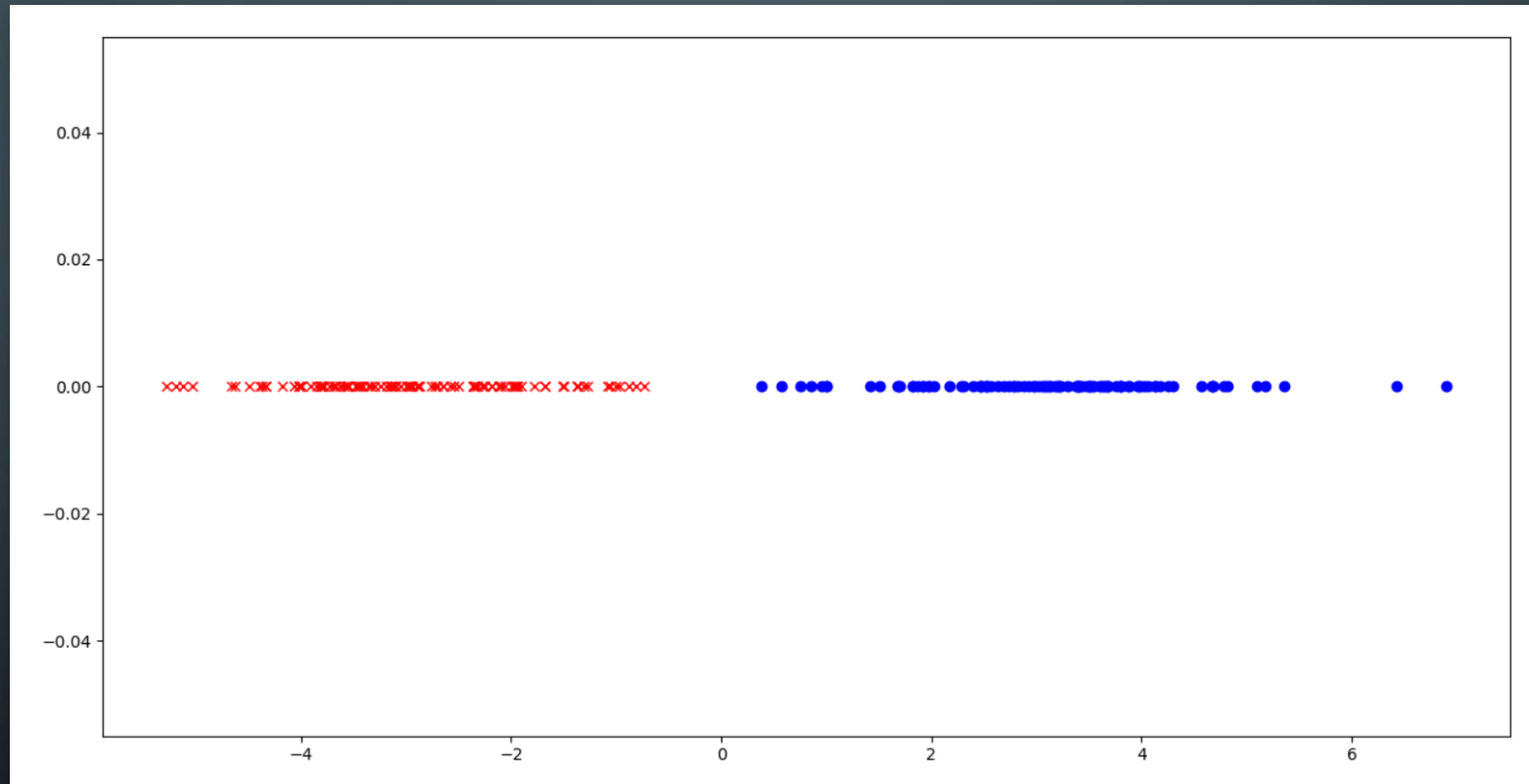
LOGISTIC REGRESSION

MOHAMMAD GHODDOSI

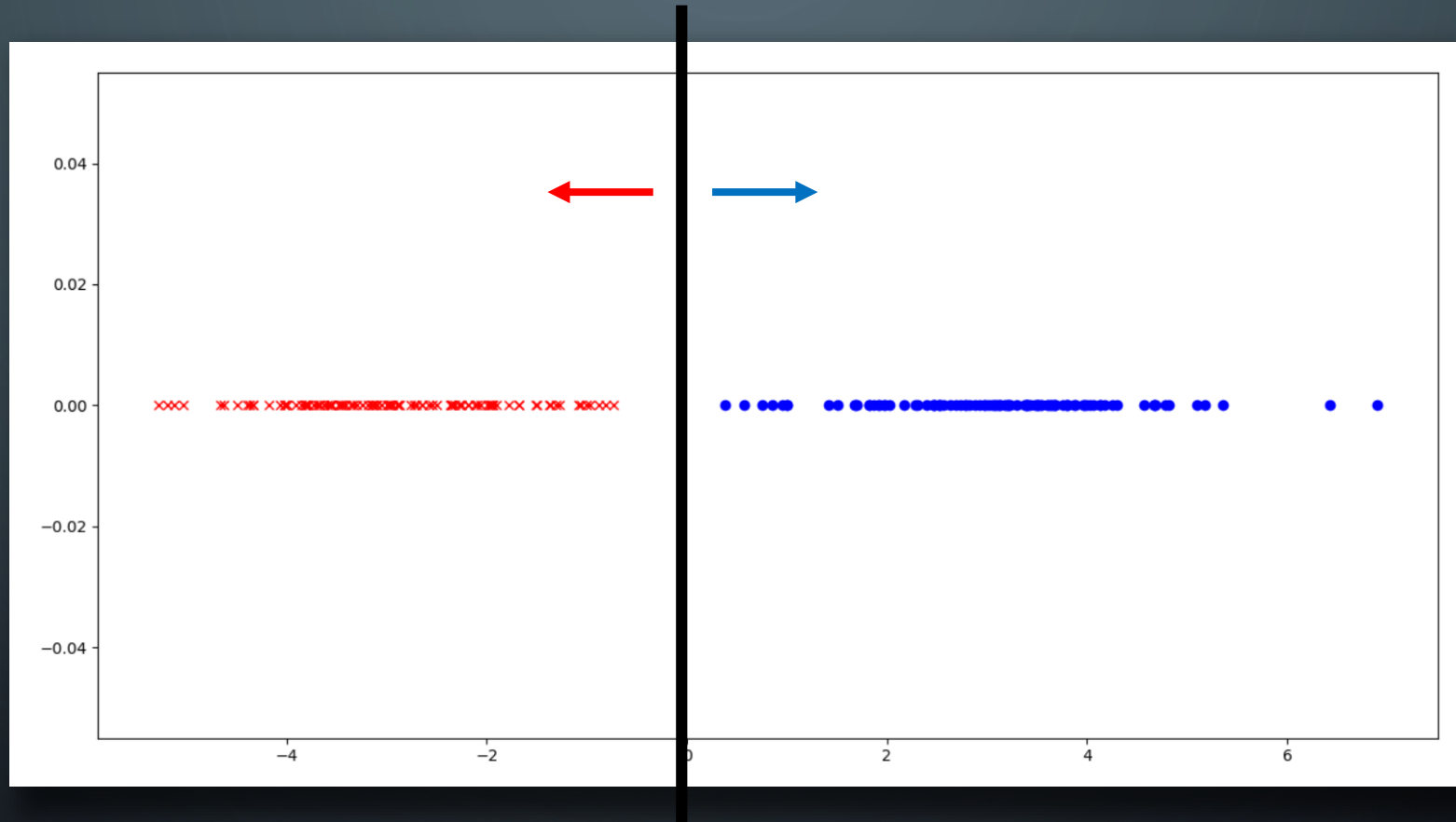
CLASSIFICATION



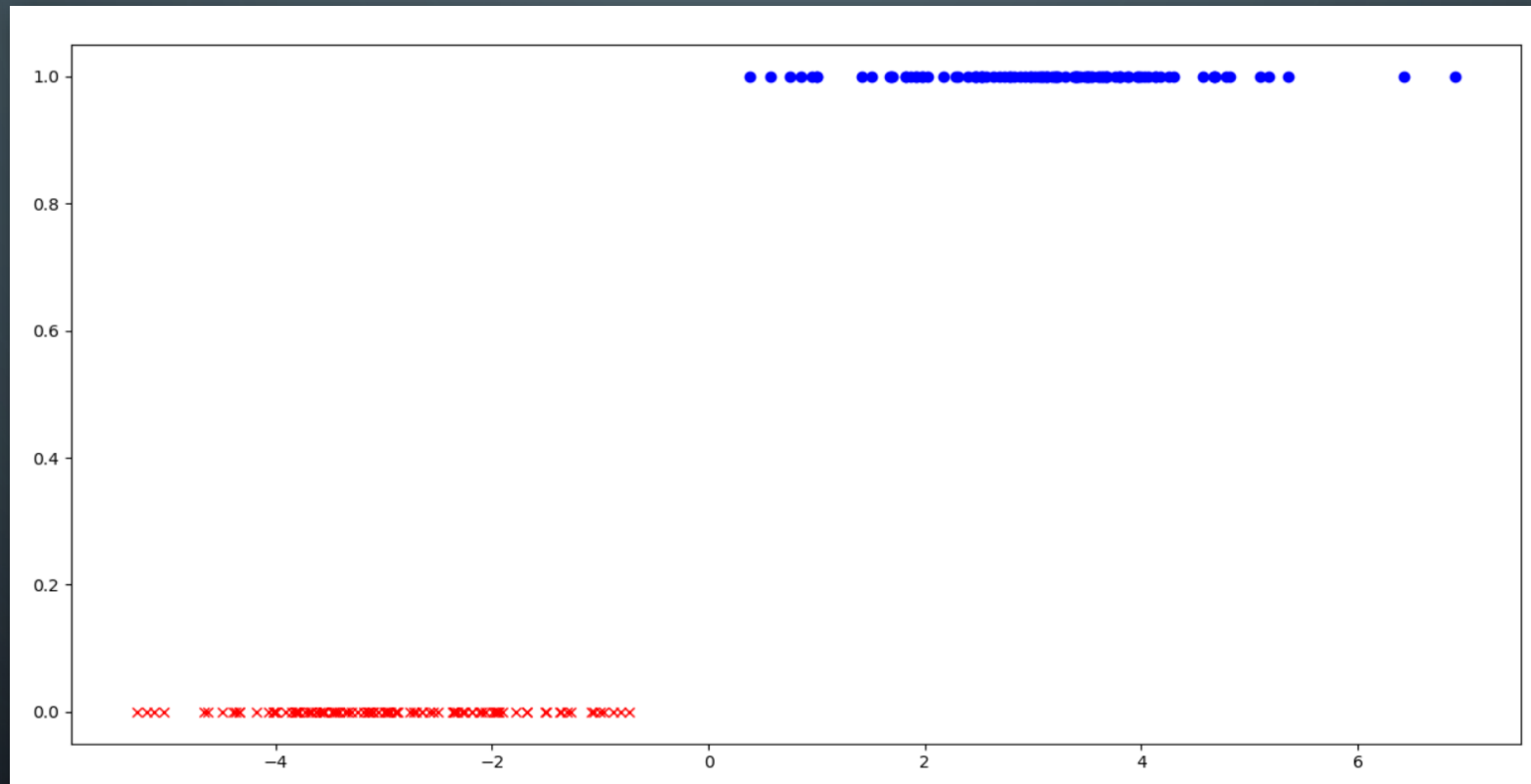
1D – CLASSIFICATION



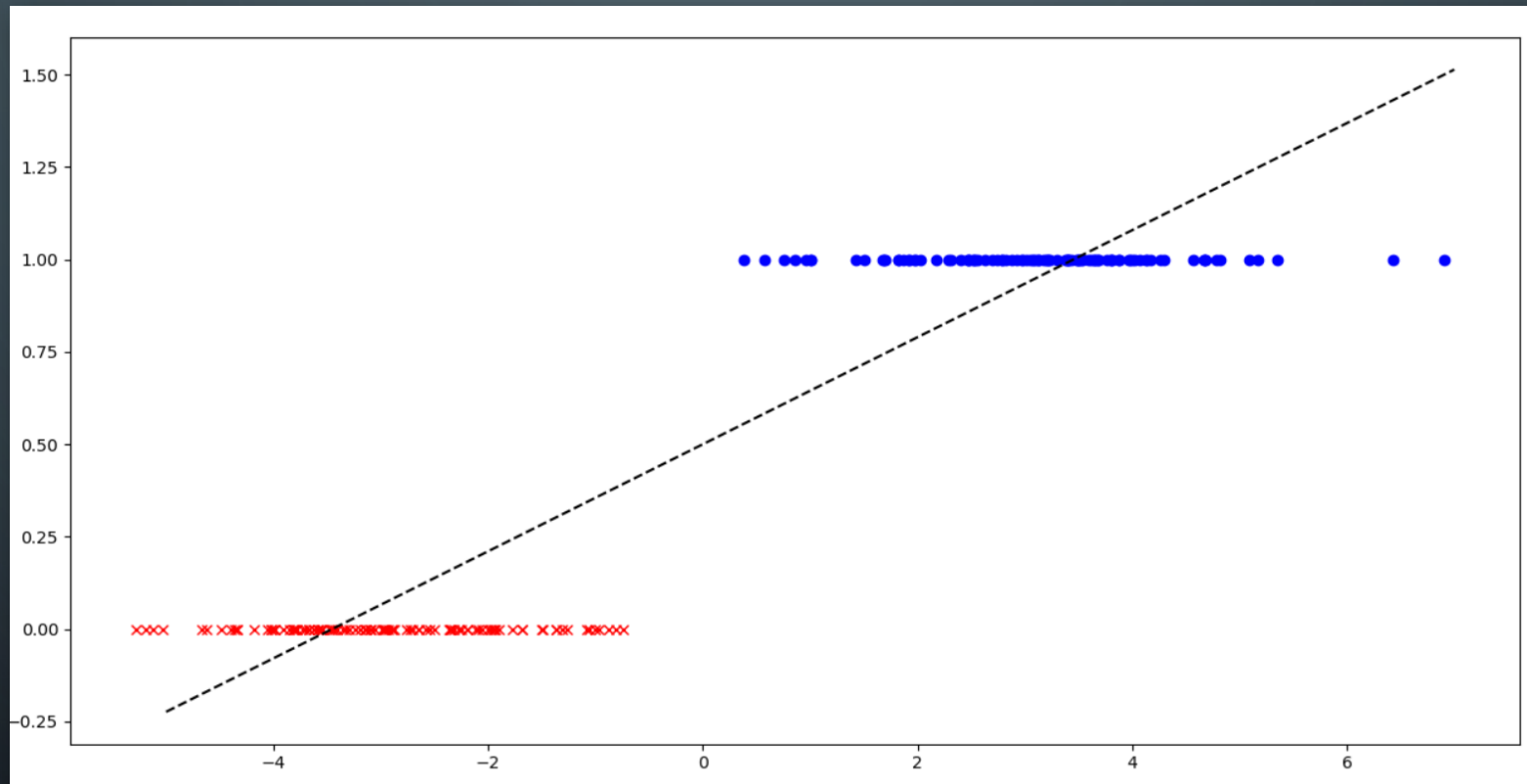
1D - CLASSIFICATION



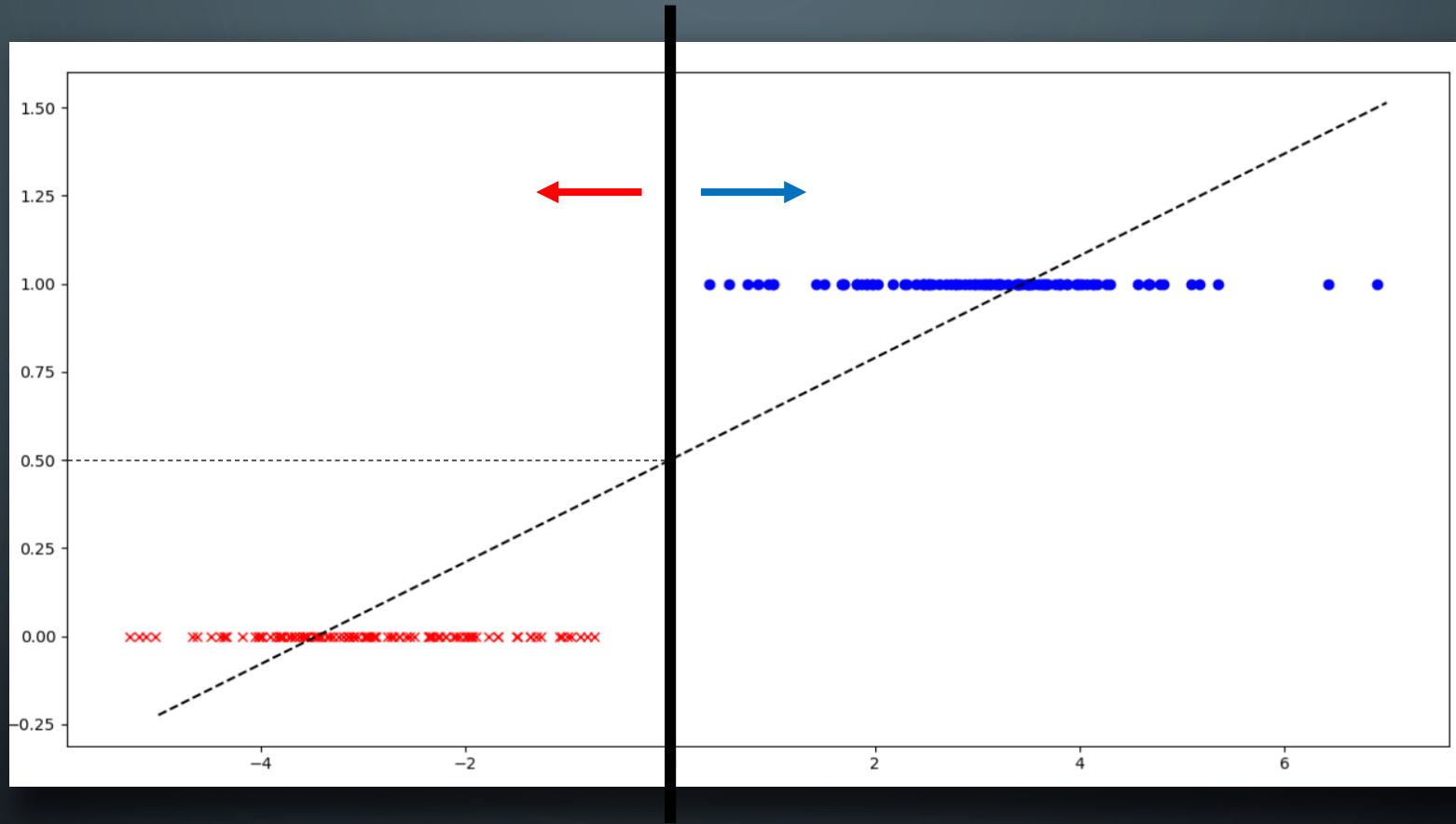
1D – CLASSIFICATION AS 2D REGRESSION



1D – CLASSIFICATION AS 2D REGRESSION



1D – CLASSIFICATION AS 2D REGRESSION

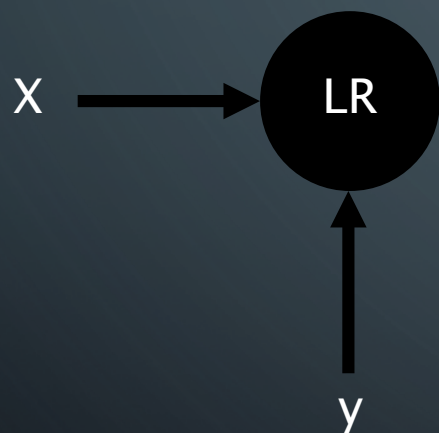


PROBLEMS

- It doesn't work fine everywhere
 - One noisy point far away from data
- Error is not categorical
- What if we need probability

CLASSIFICATION AS REGRESSION

Training

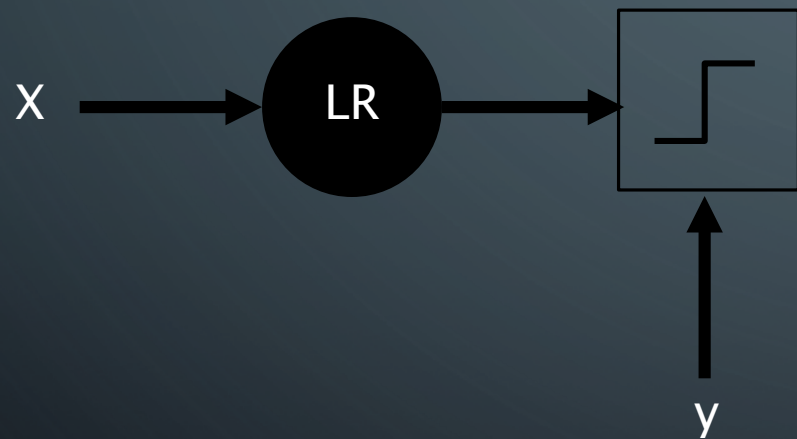


Inference



CLASSIFICATION

Training

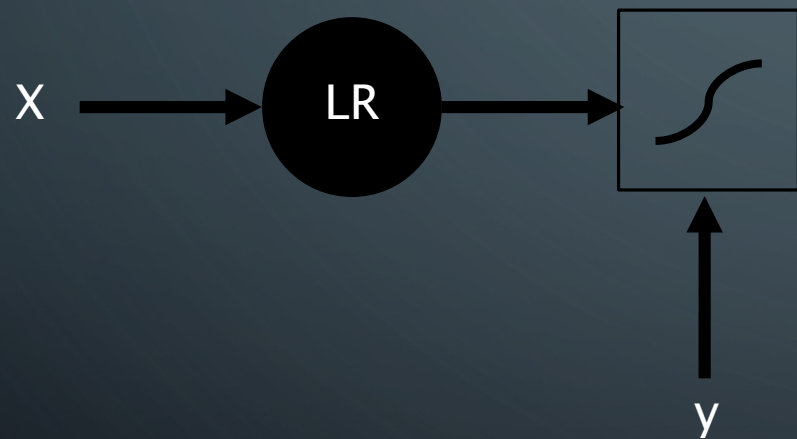


Inference



CLASSIFICATION

Training



Inference

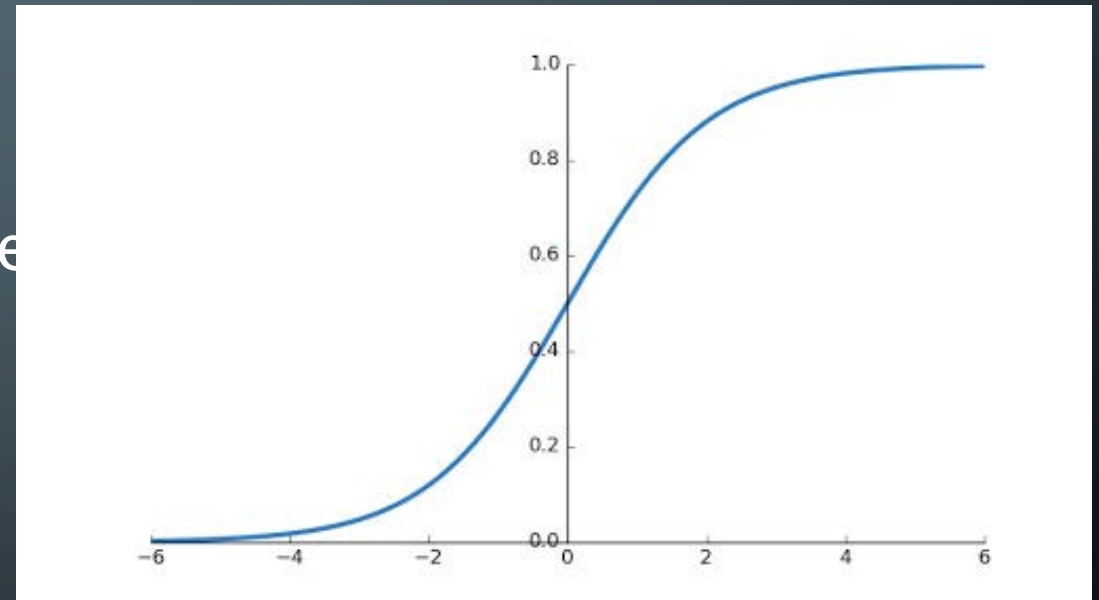


SIGMOID FUNCTION

- Output is a probability
- Output is between 0 and 1
- It has nice and smooth derivative

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

$$\sigma'(z) = \sigma(z) * (1 - \sigma(z))$$



LOGISTIC REGRESSION (OUTPUT)

$$z = \sum_{i=1}^N w_i x_i$$

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

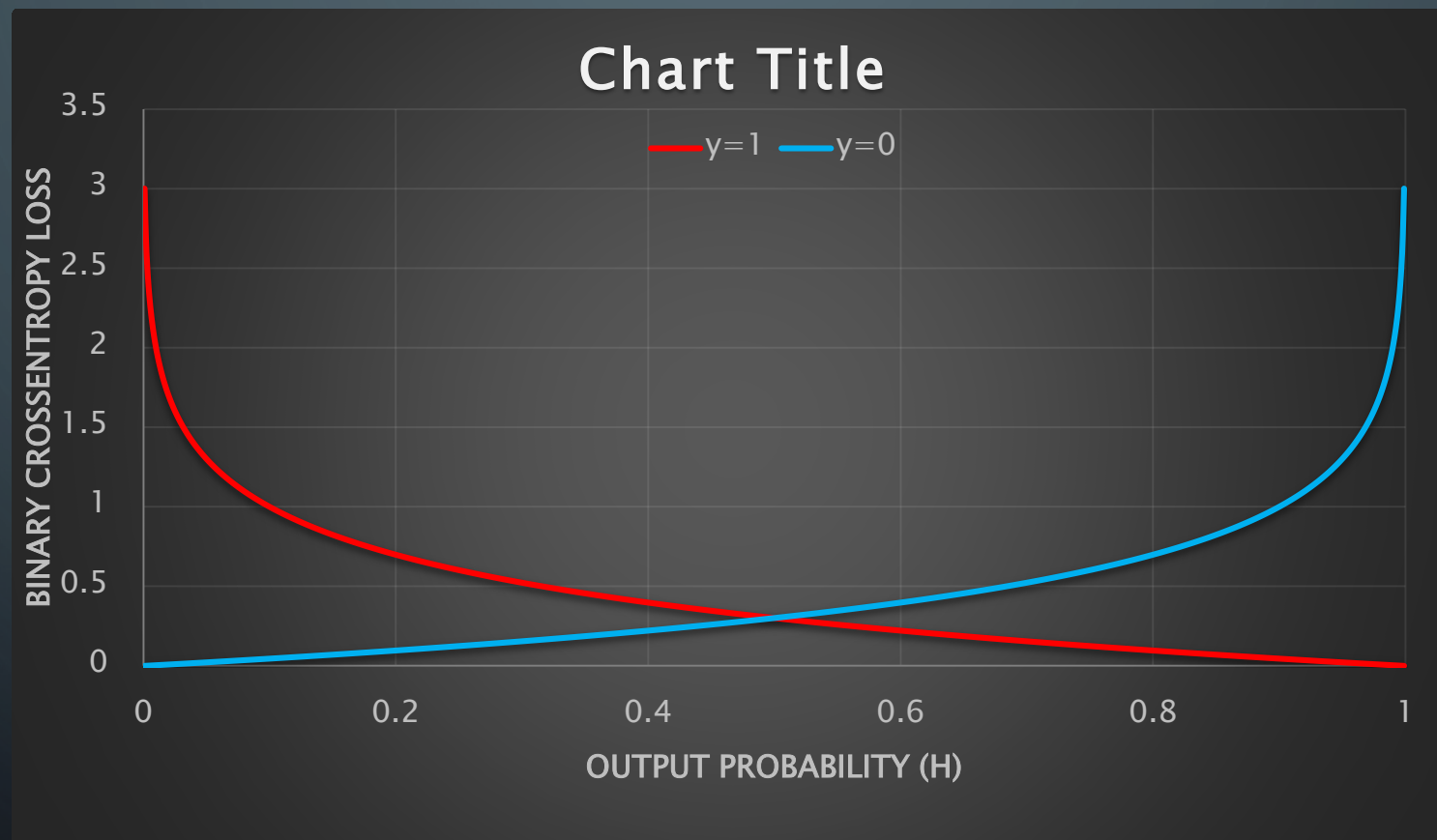
LOGISTIC REGRESSION (COST)

$$J = \frac{1}{N} \sum_{i=1}^N -y^{(i)} * \log(h^{(i)}) - (1 - y^{(i)}) * \log(1 - h^{(i)})$$

$$loss = -y * \log(h) - (1 - y) * \log(1 - h)$$

$$loss = \begin{cases} -\log(h) & y = 1 \\ -\log(1 - h) & y = 0 \end{cases}$$

LOGISTIC REGRESSION (LOSS)



LINEAR REGRESSION

$$h = \sum_{i=1}^N w_i x_i$$

VS

LOGISTIC REGRESSION

$$z = \sum_{i=1}^N w_i x_i \quad h = \frac{1}{1 + e^{-z}}$$

LINEAR REGRESSION

$$h = \sum_{i=1}^N w_i x_i$$

$$J = \frac{1}{2N} \sum_{i=1}^N (y^{(i)} - h^{(i)})^2$$

VS

LOGISTIC REGRESSION

$$z = \sum_{i=1}^N w_i x_i \quad h = \frac{1}{1 + e^{-z}}$$

$$J = \frac{1}{N} \sum_{i=1}^N -y^{(i)} * \log(h^{(i)}) - (1 - y^{(i)}) * \log(1 - h^{(i)})$$

LINEAR REGRESSION

VS

LOGISTIC REGRESSION

$$h = \sum_{i=1}^N w_i x_i$$

$$J = \frac{1}{2N} \sum_{i=1}^N (y^{(i)} - h^{(i)})^2$$

$$\frac{\partial J}{\partial w_j} = \frac{1}{N} \sum_{i=1}^N (h^{(i)} - y^{(i)}) x_j^{(i)}$$

$$z = \sum_{i=1}^N w_i x_i \quad h = \frac{1}{1 + e^{-z}}$$

$$J = \frac{1}{N} \sum_{i=1}^N -y^{(i)} * \log(h^{(i)}) - (1 - y^{(i)}) * \log(1 - h^{(i)})$$

$$\frac{\partial J}{\partial w_j} = \frac{1}{N} \sum_{i=1}^N (h^{(i)} - y^{(i)}) x_j^{(i)}$$

DERIVATIVES

$$\frac{\partial z^{(i)}}{\partial w_j} = x_j^{(i)}$$

$$\frac{\partial h^{(i)}}{\partial z^{(i)}} = h^{(i)} * (1 - h^{(i)})$$

$$\begin{aligned} \frac{\partial J}{\partial h^{(i)}} &= \frac{-y^{(i)}}{h^{(i)}} + \frac{1 - y^{(i)}}{1 - h^{(i)}} \\ &= \frac{h^{(i)} - y^{(i)}}{h^{(i)} * (1 - h^{(i)})} \end{aligned}$$

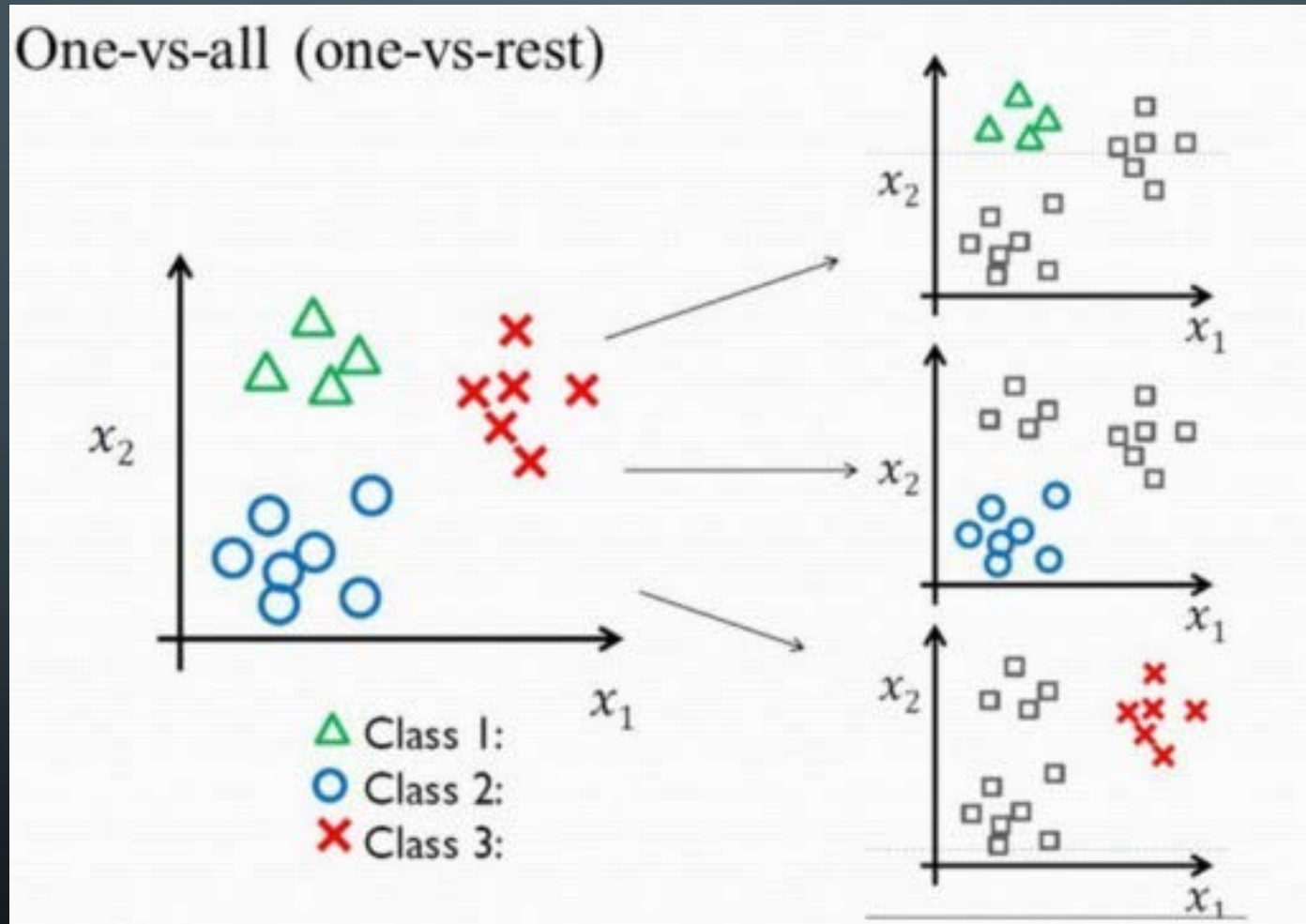
$$\frac{\partial J}{\partial w_j} = \frac{1}{N} \sum_{i=1}^N \frac{\partial J}{\partial h^{(i)}} * \frac{\partial h^{(i)}}{\partial z^{(i)}} * \frac{\partial z^{(i)}}{\partial w_j}$$

$$\frac{\partial J}{\partial w_j} = \frac{1}{N} \sum_{i=1}^N (h^{(i)} - y^{(i)}) * x_j^{(i)}$$

The Jupyter logo is centered in the image. It consists of two orange, curved, crescent-like shapes that form a circle around the word "jupyter". The word "jupyter" is written in a white, lowercase, sans-serif font. There are four white circles of varying sizes positioned around the logo: one at the top left, one at the top right, one at the bottom left, and one at the bottom right. The background is a dark blue gradient. In the corners, there are faint, light blue circuit-like patterns with lines and small circles.

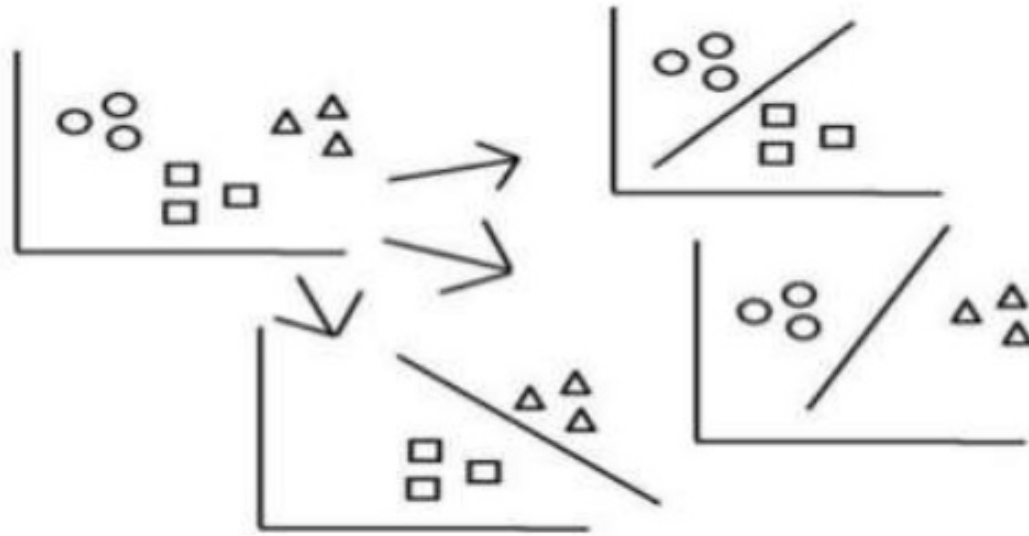
jupyter

MULTICLASS CLASSIFICATION(OVR)



MULTICLASS CLASSIFICATION(OVO)

One-vs-One (OVO)



BINARY LOGISTIC REGRESSION AS MATRIX MULTIPLICATION

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} = \sigma \left(\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_f^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_f^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_f^{(N)} \end{bmatrix} \cdot [w_1 \quad w_2 \quad \dots \quad w_M \quad w_f]^T \right)$$

$$Y = \sigma(X \cdot W^T)$$

BINARY LOGISTIC REGRESSION AS MATRIX MULTIPLICATION

$$\begin{bmatrix} y_1^{(1)} & y_2^{(1)} & \dots & y_c^{(1)} \\ y_1^{(2)} & y_2^{(2)} & \dots & y_c^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{(N)} & y_2^{(N)} & \dots & y_c^{(N)} \end{bmatrix} = \sigma \left(\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_f^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_f^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_f^{(N)} \end{bmatrix} \cdot \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,f} \\ w_{2,1} & w_{2,2} & \dots & w_{2,f} \\ \vdots & \vdots & \ddots & \vdots \\ w_{c,1} & w_{c,2} & \dots & w_{c,f} \end{bmatrix}^T \right)$$

$$Y = \sigma(X.W^T)$$

The Jupyter logo is centered in the image. It consists of two orange, curved, crescent-like shapes that form a circle around the word "jupyter". The word "jupyter" is written in a white, lowercase, sans-serif font. There are four white circles of varying sizes positioned around the logo: one at the top left, one at the top right, one at the bottom left, and one at the bottom right. The background is a dark blue gradient. In the corners, there are faint, light blue circuit-like patterns with lines and small circles.

jupyter