Logistic Regression

Classification

Regression v/s Classification

- *In regression, we predict continuous values.*
- *In classification, we predict categories or labels.*

For example:

- *A typical regression problem would be something like, predict price of a house given some parameters.*
- *Whereas, a classification problem would be something like, predict if this tumor is malignant or benign given some parameters.*

Logistic Regression

Definition

- Logistic regression is a classification algorithm that predicts the probability of an event happening.
- It is mainly used for binary classification (e.g., Yes/No , Pass/Fail).

Working

- We predict a probability between 0 and 1.
- Eg) If probability > 0.5, classify as 1 (Yes/True).
  If probability < 0.5, classify as 0 (No/False).

The sigmoid function

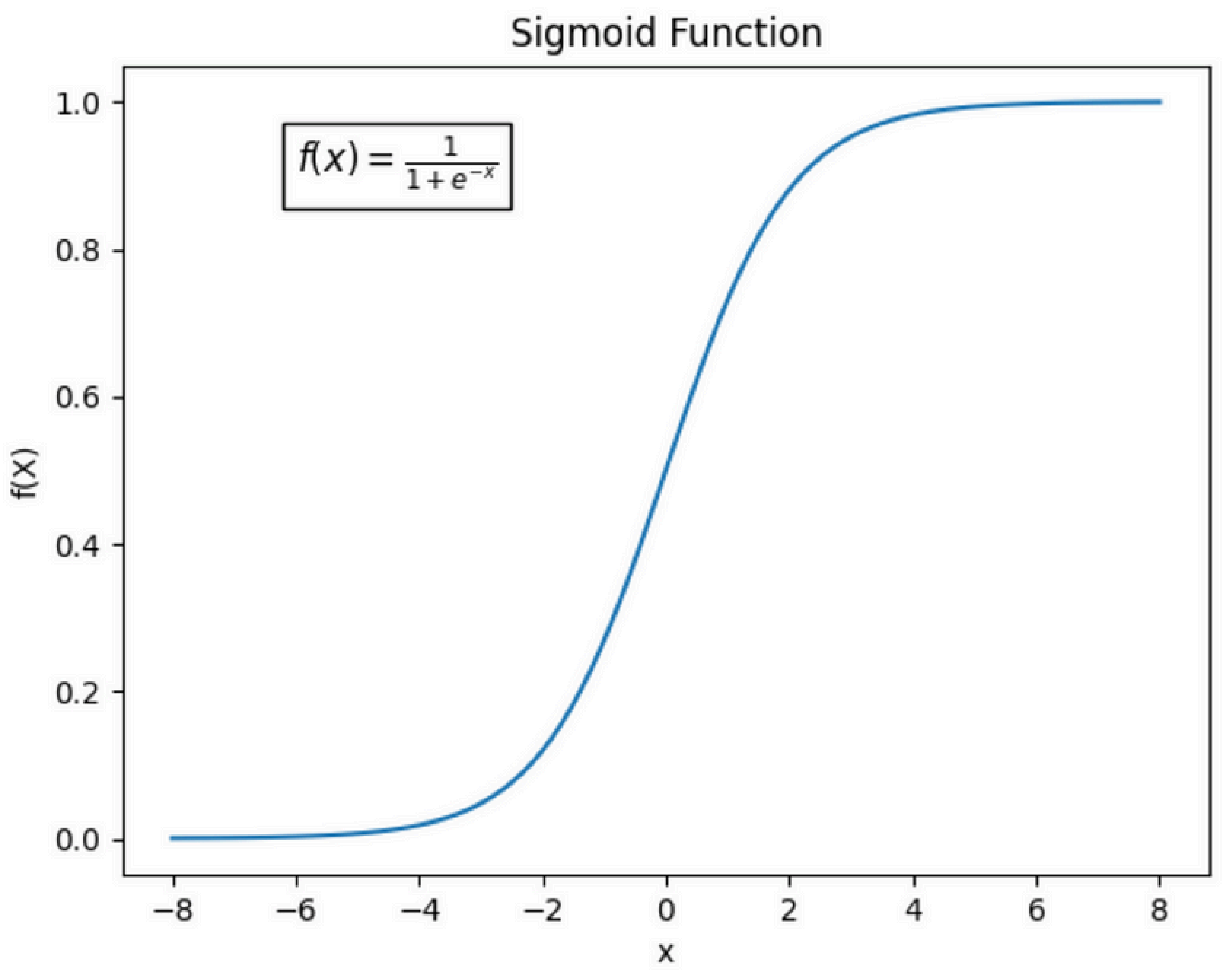
What is it?

- Its a mathematical function that converts any number into a value between 0 and 1

Mathematical denotion:

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

Where: $z=wX+b$



Cost function

- *We cant simply use MSE for classification because its cost curve is not convex in this case*
- *Hence, we formulate the following loss function*

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

Simplified form:

$$\boxed{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)}$$

Hence, we write cost function as:

loss

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \underbrace{-y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)}$$

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left[\underbrace{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)}\right]$$

Gradient Decent

Just like during linear regression, we now use gradient decent to minimize the cost function

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{\mathrm{w}}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{\mathrm{w}}, b)$$

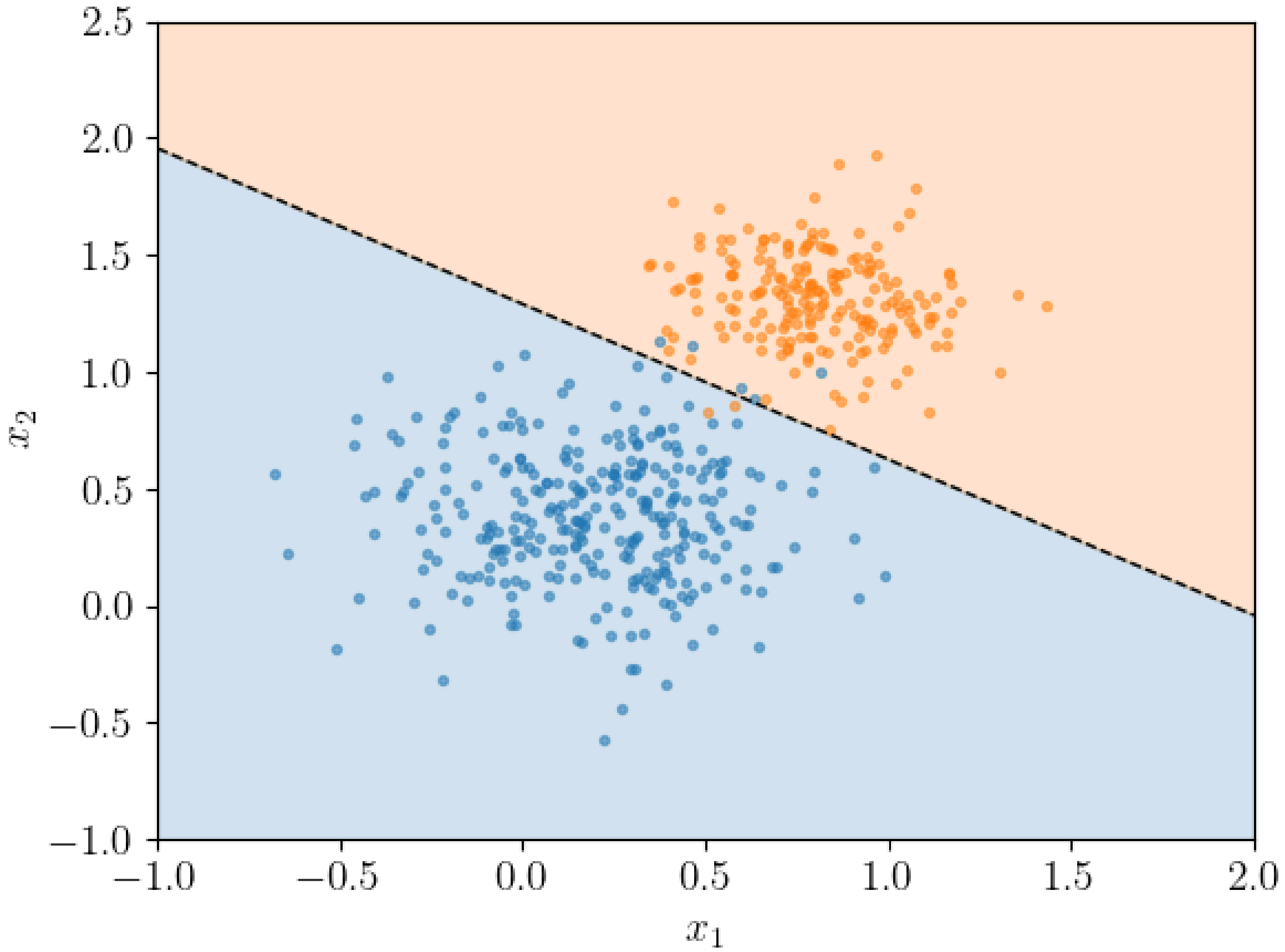
The partial derivatives look like this:

$$\frac{\partial}{\partial w_j} J(\vec{\mathrm{w}}, b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

$$\frac{\partial}{\partial b} J(\vec{\mathrm{w}}, b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}^{(i)}\right) - y^{(i)}\right)$$

Decision boundary

Lets revisit the sigmoid curve:



Here, to classify, we set a threshold

- *If* $\boldsymbol{\sigma}(z) \geq 0.5 \rightarrow$ *predict class 1.*
- *If* $\boldsymbol{\sigma}(z) < 0.5 \rightarrow$ *predict class 0.*

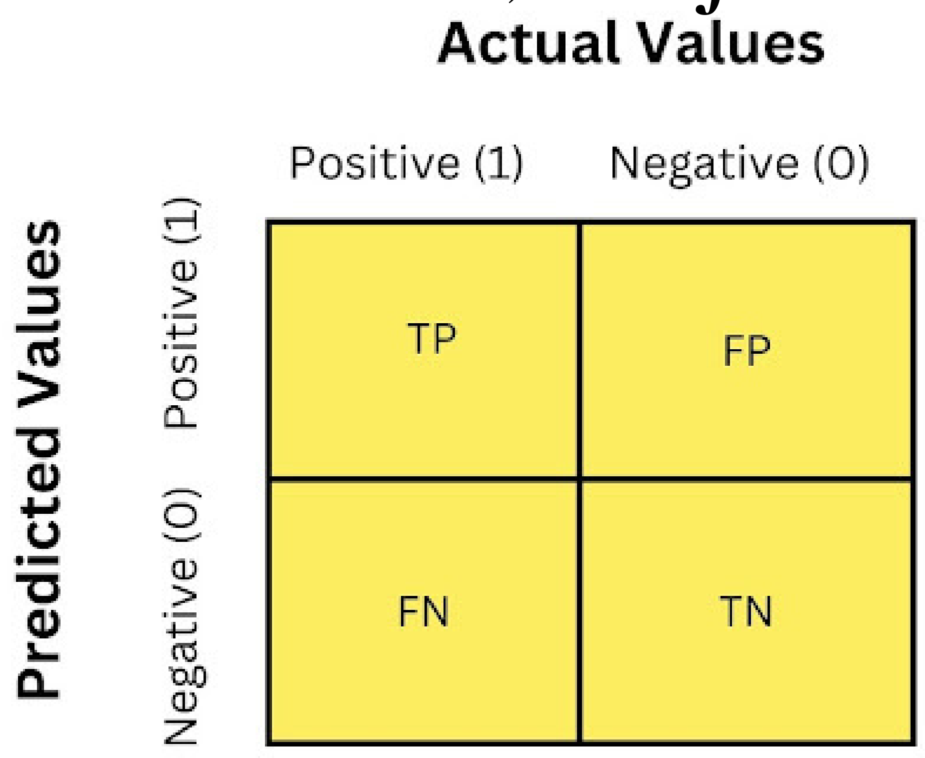
So, we have defined the decision boundary at: $wX+b=0$

Performance Metrics

Now that we've trained our model, we need to see how well it has performed.

Accuracy is not a good measure for classification as it can be misleading.
Consider this case, if a dataset has 98% +ve results and only 2% -ve results. A model that only predicts y=1 will have 98% accuracy!

How to solve this? Enter, Confusion Matrix:



From this , we can calculate Precision and Recall:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Performance Metrics

It is often convenient to combine precision and recall into a single metric called the F1 score. It's just harmonic mean of Precision and Recall

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \cdot \frac{Precision * Recall}{Precision + Recall}$$

$$\Rightarrow F1\ score = 2 \cdot \frac{Precision * Recall}{Precision + Recall}$$

Precision Recall Tradeoff: Generally as you increase threshold, your precision increases and recall decreases and vice versa. Based on what you want to do, you may want higher precision or higher recall.