Logistic Regression

Logistic Regression is a **supervised learning algorithm** for **binary classification** problems (where the output has two possible classes, e.g., Yes/No, Spam/Not Spam). The goal is to predict the 'Probability' of an instance belonging to a particular class. Used for both Binary Classification and Multi-Class Classification. Despite its name, it is a classification algorithm, not a regression algorithm.

How Does Logistic Regression Work?

Instead of fitting a straight line like Linear Regression, **Logistic Regression** applies the **sigmoid (logistic) function** to map predictions to a probability range between **0 and 1**:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_n X_n)}}$$

Where:

- $P(Y=1|X)$ → Probability that the output belongs to class **1**
- $\beta_0, \beta_1, ..., \beta_n$ → Model parameters (weights)
- $X_1, X_2, ..., X_n$ → Feature variables
- e → Euler's number (~2.718)

📌 **Interpretation:**

- Threshold: It is a value between 0 and 1, that determines the decision boundary for classifying an instance as Positive-class '1' or Negative-class '0'. Generally, the threshold is set to 0.5.
- If the probability is **≥ 0.5**, classify it as **1** (Positive class).
- If the probability is **< 0.5**, classify it as **0** (Negative class).

Sigmoid Function (Logistic Function)

The sigmoid function **squashes values** to the **(0,1) range**, making it useful for probability estimation.

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

Graph Behavior:

- **As $z \to \infty, \sigma(z) \to 1$**
- **As $z \to -\infty, \sigma(z) \to 0$**

✅ Ensures predictions remain in a **probability range**.

Log Loss: Binary Cross-Entropy (BCE) Loss

Binary Cross-Entropy (BCE) Loss, also known as **Log Loss**, is the loss function used for **binary classification problems**. It measures how well a classification model predicts the correct class probabilities.

The formula for BCE Loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} [Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)]$$

Where:

- N = Number of samples
- Y_i = Actual label (0 or 1)
- $\hat{Y}_i$ = Predicted probability for class 1 (output of the **sigmoid function**)
- $\log$ = Natural logarithm

How BCE Loss Works

📌 **Case 1: If the actual label $Y = 1$**

The loss function simplifies to:

$$\mathcal{L} = -\log(\hat{Y})$$

- If $\hat{Y} = 1$ (perfect prediction), loss = **0** (ideal).
- If $\hat{Y} = 0$ (wrong prediction), loss = **∞** (very bad).

📌 **Case 2: If the actual label $Y = 0$**

The loss function simplifies to:

$$\mathcal{L} = -\log(1 - \hat{Y})$$

- If $\hat{Y} = 0$ (perfect prediction), loss = **0**.
- If $\hat{Y} = 1$ (wrong prediction), loss = **∞**.

📌 **Intuition:**

- The loss is **low** when predictions are **confident and correct**.
- The loss is **high** when predictions are **confident but wrong**.

Why Use BCE Loss?

✅ **Handles Probabilities** – Ensures predictions are between 0 and 1.

✅ **Punishes Incorrect Confident Predictions** – If the model predicts **1** when the actual label is **0**, the loss is very high.

✅ **Works Well with Sigmoid Activation** – Commonly used in **logistic regression** and **binary classification neural networks**.
