Logistic Regression Mathematics

Logistic regression is used for **binary classification**.
The output variable Y can take values 0 or 1.

Sigmoid Function

The predicted probability is modeled using the **sigmoid function**:

$$\hat{Y} = \sigma(Z) = \frac{1}{1 + e^{-Z}}$$

where

$$Z = X\beta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

- X → input matrix (with a column of ones for intercept)
- β → weight vector
- $\hat{Y}$ → predicted probability (between 0 and 1)

Loss Function (Binary Cross-Entropy)

The cost function to minimize is the **log loss / binary cross-entropy**:

$$E(\beta) = -\sum_{i=1}^{n} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]$$

- y_i → actual label (0 or 1)
- $\hat{y}_i$ → predicted probability for instance i

Gradient Descent Update Rule

To find optimal weights, we use **gradient descent**:

$$\beta_j \leftarrow \beta_j - \eta \cdot \frac{\partial E}{\partial \beta_j}$$

where the gradient of the cost function is:

$$\frac{\partial E}{\partial \beta_j} = \sum_{i=1}^{n} (\hat{y}_i - y_i) X_{ij}$$

- η → learning rate
- Repeat the update until convergence

Final Prediction

Once β is optimized:

- Compute predicted probabilities:

$$\hat{Y} = \sigma(X\beta)$$

- Assign classes based on threshold (usually 0.5):

$$\text{Predicted Class} = \begin{cases} 1 & \text{if } \hat{y} \geq 0.5 \\ 0 & \text{if } \hat{y} < 0.5 \end{cases}$$