Gradient Descent is used in ML to **optimize model parameters** (like weights) by **minimizing the loss function**, helping the model learn and make accurate predictions.

A **loss function** measures how far the model's predictions are from the actual values — it tells the model **how wrong** it is.
👉 The goal of training is to **minimize this loss** using algorithms like Gradient Descent.

Gradient Descent helps by **adjusting model parameters step by step** in the direction that **reduces the loss** most quickly — eventually reaching the point where the loss is minimal.

Gradient Descent (GD) – Detailed Explanation:

1. **Purpose:**
   Gradient Descent is an optimization algorithm used to minimize a **loss (cost) function** in machine learning models by adjusting parameters (like weights).
2. **Idea:**
   Think of the loss function as a curved surface (like a hill). The goal is to find the **lowest point** (minimum loss). GD helps move "downhill" step by step toward that point.
3. **Mathematical formula:**

$$\theta = \theta - \alpha \times \frac{\partial J(\theta)}{\partial \theta}$$

where,

- **θ** = parameter (like weight),
- **α** = learning rate (step size),
- **J(θ)** = loss function,
- **∂J/∂θ** = gradient (slope).

Algorithm: Gradient Descent

1. **Start**
2. Initialize parameter (x) (or weights) randomly.
3. Choose a learning rate (\alpha).
4. Repeat until convergence:
   a. Compute gradient (g = \frac{df(x)}{dx}).
   b. Update parameter: (x = x - \alpha \times g).
   c. Check if change in (x) is less than threshold → stop.
5. **Output** the value of (x) where the function has minimum value.
6. **End**