Linear Regression

- Mathematical Model

Linear Regression tries to learn a straight line:

$$\boxed{\hat{y} = mX + c}$$

The model tries to find best m and c.

Where:
- X = input feature
- $\hat{y}$ = predicted output
- m = slope (weight)
- c = intercept (bias)

- Loss Function – MSE (Mean Squared Error)

We measure how wrong predictions are wrong:

$$\boxed{Loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Goal: Minimize this loss

Where:
- y_i = actual value
- $\hat{y}_i$ = predicted value
- N = number of data points.

- Gradient Descent

We update parameters by moving in the direction of negative gradient (because gradients show direction of increasing error)

· Update rules

$$\boxed{m = m - \alpha \frac{\partial Loss}{\partial m}} \quad \boxed{c = c - \alpha \frac{\partial Loss}{\partial c}}$$

Here:
- α = learning rate (small number like 0.01)
- Partial derivatives tell us how change in m or c changes the error.

- Partial derivatives Calculations

1. Derivative w.r.t. m

$$\boxed{\frac{\partial Loss}{\partial m} = \frac{-2}{N}\sum X_i(y_i - \hat{y}_i)}$$

2. Derivative w.r.t. c

$$\boxed{\frac{\partial Loss}{\partial c} = \frac{-2}{N}\sum (y_i - \hat{y}_i)}$$

We plug these into update rules.

- Full update Step Inside loop

$$\boxed{m := m - \alpha\left(-\frac{2}{N}\sum x_i\,(y_i - (m x_i + c))\right)}$$

$$\boxed{c := c - \alpha\left(-\frac{2}{N}\sum (y_i - (m x_i + c))\right)}$$

Repeat for many epochs → Loss decreases → best fit line appears.

- Intuition

Math Part	Meanings
Loss	How bad our line is
Derivatives	In which direction the error increases
Gradient Descent	Move downhill → reduce error
Iterations	Slowly reach the best values.

"Linear Regression learns the best straight line by minimizing squared error using gradient descent".