Maths behind Linear Regression

Let us assume a sample dataset with 3 input features (X_1, X_2, X_3) and one output column (Y).
Suppose there are (n) instances. Then the predicted values can be written as:

$$\begin{aligned}\hat{y}_1 &= \beta_0 + \beta_1 X_{11} + \beta_2 X_{12} + \beta_3 X_{13} \\ \hat{y}_2 &= \beta_0 + \beta_1 X_{21} + \beta_2 X_{22} + \beta_3 X_{23} \\ &\vdots \\ \hat{y}_n &= \beta_0 + \beta_1 X_{n1} + \beta_2 X_{n2} + \beta_3 X_{n3}\end{aligned}$$

Matrix form

In matrix form, the above equations become:

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}_{(n \times 1)} = \begin{bmatrix} 1 & X_{11} & X_{12} & X_{13} \\ 1 & X_{21} & X_{22} & X_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & X_{n2} & X_{n3} \end{bmatrix}_{(n \times 4)} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_{(4 \times 1)}$$

We can write the prediction as:

$$\hat{Y} = X\beta$$

where

- $\hat{Y}$ → predicted vector $(n \times 1)$
- X → input/design matrix $(n \times p)$ (includes a column of ones for the intercept)
- β → weight vector $(p \times 1)$

Error Vector

The error (residual) vector is given by:

$$e = \hat{Y} - Y = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \hat{y}_1 - y_1 \\ \hat{y}_2 - y_2 \\ \hat{y}_3 - y_3 \\ \vdots \\ \hat{y}_n - y_n \end{bmatrix}$$

Loss Function

The loss function (Sum of Squared Errors, SSE) is:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Method 1: Direct Formula (Normal Equation)

We can express the loss as:

$$E = e^T e$$

Solving this minimization problem, the optimal weights are obtained by:

$$\beta = (X^T X)^{-1} X^T Y$$

Thus, predictions are given by:

$$\hat{Y} = X\beta$$

Method 2: Gradient Descent

Loss function:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Update rule for each parameter β_j:

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

where η is the learning rate.

Explicitly:

- $\beta_0 = \beta_0 - \eta \cdot \frac{\partial E}{\partial \beta_0}$
- $\beta_1 = \beta_1 - \eta \cdot \frac{\partial E}{\partial \beta_1}$
- $\beta_2 = \beta_2 - \eta \cdot \frac{\partial E}{\partial \beta_2}$
- $\beta_3 = \beta_3 - \eta \cdot \frac{\partial E}{\partial \beta_3}$
- $\ldots$

Final Prediction

In both methods, once we have the optimal β, predictions are given by:

$$\hat{Y} = X\beta$$

Note on Number of Input Features

In this example, we have taken only **three input columns** to show how it works.

The same approach can be applied to **any number of input features**:

- If there are **4 input columns**, the weight vector will be:

$$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$$

- If there are **5 input columns**, the weight vector will be:

$$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$$

- Similarly, for **n input columns**, the weight vector will be:

$$\beta_0, \beta_1, \beta_2, \ldots, \beta_n$$

This shows that linear regression **scales naturally** to any number of features.