Machine Learning for Robotics

Shafayetul Islam
Robotics and AI Instructor

Contact Me

✉️ Email: shafatsib@gmail.com 📞 Phone: +880178-0404749 📍 Dhaka, Bangladesh

Google Scholar Github Linkedin Youtube

Class 04-
- Linear Regression

Regression

Regression is a type of supervised learning where the goal is to **predict a continuous numerical value.**

Regression

Regression is a type of supervised learning where the goal is to **predict a continuous numerical value.**

What does “continuous value” mean?

A value that can take **any number within a range**, such as:

- 25.5
- 3.2
- 102.75

Regression

Example

House Price Prediction

- **Input (X)**: house size, number of rooms
- **Output (y)**: house price (for example: 250000)

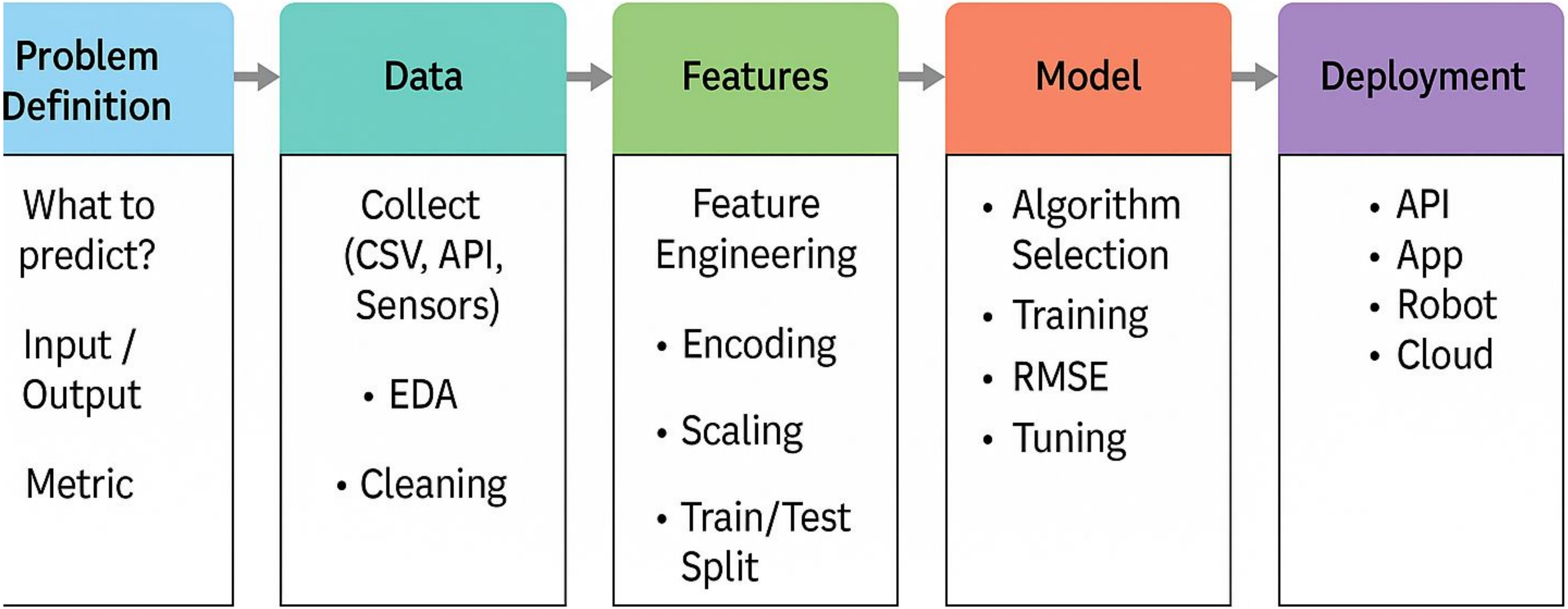
Because the output is a **number**, this is a **regression problem**.

Regression

Common regression types

- **Linear Regression**
- **Multiple Linear Regression**
- **Polynomial Regression**

ML Pipeline

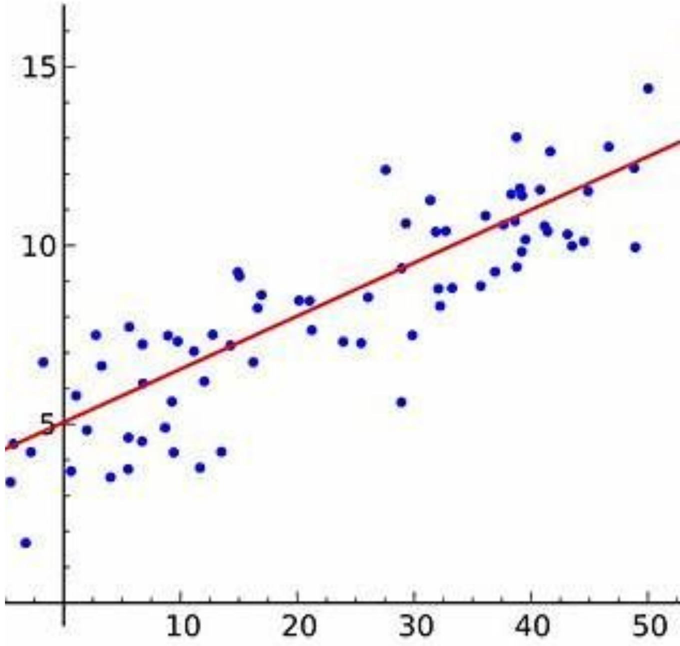

Problem Definition
What to predict?
Input / Output
Metric
Data
Collect (CSV, API, Sensors)
• EDA
• Cleaning
Features
Feature Engineering
• Encoding
• Scaling
• Train/Test Split
Model
• Algorithm Selection
• Training
• RMSE
• Tuning
Deployment
• API
• App
• Robot
• Cloud

Linear Regression

Simple Linear Regression is a **supervised learning regression** method where:

- there is **one input feature (x)**
- and **one output target (y)**
- the relationship is modeled using a **straight line**

study_hours	math_marks
1.0	35
1.2	37
1.5	40
1.7	38
2.0	42



Linear Regression

Mathematical model

The equation of simple linear regression is:

$$\mathbf{y=mx+b}$$

Where:

- **x** → input feature
- **y** → predicted output
- **m** → slope (how fast y changes when x increases)
- **b** → intercept (value of y when x = 0)

Linear Regression

- **Step 01: Define Goal**

Linear Regression

- **Step 02: Data Analysis**

Load Dataset	Visualize the Dataset	Data Preprocessing
● Read csv file: **pandas**	● Draw Scatter Plot to find whether data is in linear: **matplotlib**	● Check Missing Values : pandas

Linear Regression

- **Step 03: Feature Engineering**

Feature Engineering	Train / Test Split
• Scaling (Standardization): sklearn	• Separate features (X) and labels (y) : pandas • Split data to train, test: sklearn

Linear Regression

- **Step 04: Model Training**

• Model selection ○ Linear Regression	• Model training

Linear Regression

- **Step 05: Model Evaluation**

- Prediction on test data
- Error: MAE, MSE, RMSE
- Performance: R2, Adjusted R2
- New Prediction

Error

1️⃣ MAE — Mean Absolute Error

Formula

$$MAE = \frac{1}{n}\sum |y - \hat{y}|$$

Meaning

- Average of **absolute differences**
- Treats **all errors equally**

Pros	Cons
• Easy to understand • Robust to outliers (less sensitive) • Error is in **same unit as target**	• Does not penalize large errors strongly • Less useful when large mistakes are critical

Error

2️⃣ MSE — Mean Squared Error

Formula

$$MSE = \frac{1}{n}\sum(y - \hat{y})^2$$

Meaning

- Squares the error before averaging
- Large errors get much higher penalty

Pros	Cons
• Strongly penalizes large errors • Smooth and mathematically convenient • Used in model training	• Very sensitive to outliers • Unit becomes squared, hard to interpret

Error

3️⃣ RMSE — Root Mean Squared Error

Formula

$$RMSE = \sqrt{MSE}$$

Meaning

- Square root of MSE
- Balances MAE and MSE behavior

Pros	Cons
• Penalizes large errors • Error is in same unit as target • Easy to compare with MAE	• Still sensitive to outliers • More complex than MAE

Error

Metric	Penalizes large error	Outlier sensitivity	Unit
MAE	Low	Low	Same as target
MSE	Very high	Very high	Squared
RMSE	High	High	Same as target

Performance Metrics

1️⃣ R Square (R^2)

Formula

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Where:

- **SS res** → prediction error
- **SS tot** → total variation in data

Meaning

- R Squared tells us how much of the variance in the target is explained by the model.
- How good the model fits the data.

Pros	Cons
• Easy to understand • Good for comparing models with same features • Popular and widely used	• Always increases when you add more features • Cannot detect overfitting • Misleading for multiple features

Performance Metrics

2️⃣ Adjusted R Square (Adjusted R^2)

Formula

$$Adjusted\ R^2 = 1 - (1 - R^2)\,\frac{n-1}{n-p-1}$$

Where:

- **n** → number of samples
- **p** → number of features

Meaning

- Adjusted R Square fixes the main problem of R^2.
- It penalizes unnecessary features.
- Only increases if the new feature actually improves the model.

Pros	Cons
• Detects overfitting • Best for multiple linear regression • More reliable than R^2	• Slightly harder to interpret • Can decrease when weak features are added

Performance Metrics

Metric	Penalizes extra features	Best use case
R Square	No	Simple models
Adjusted R Square	Yes	Multiple features

Performance Analysis for LR

Metric	Penalizes large error	Outlier sensitivity	Unit
MAE	Low	Low	Same as target
MSE	Very high	Very high	Squared
RMSE	High	High	Same as target

Metric	Penalizes extra features	Best use case
R Square	No	Simple models
Adjusted R Square	Yes	Multiple features

Multiple Linear Regression

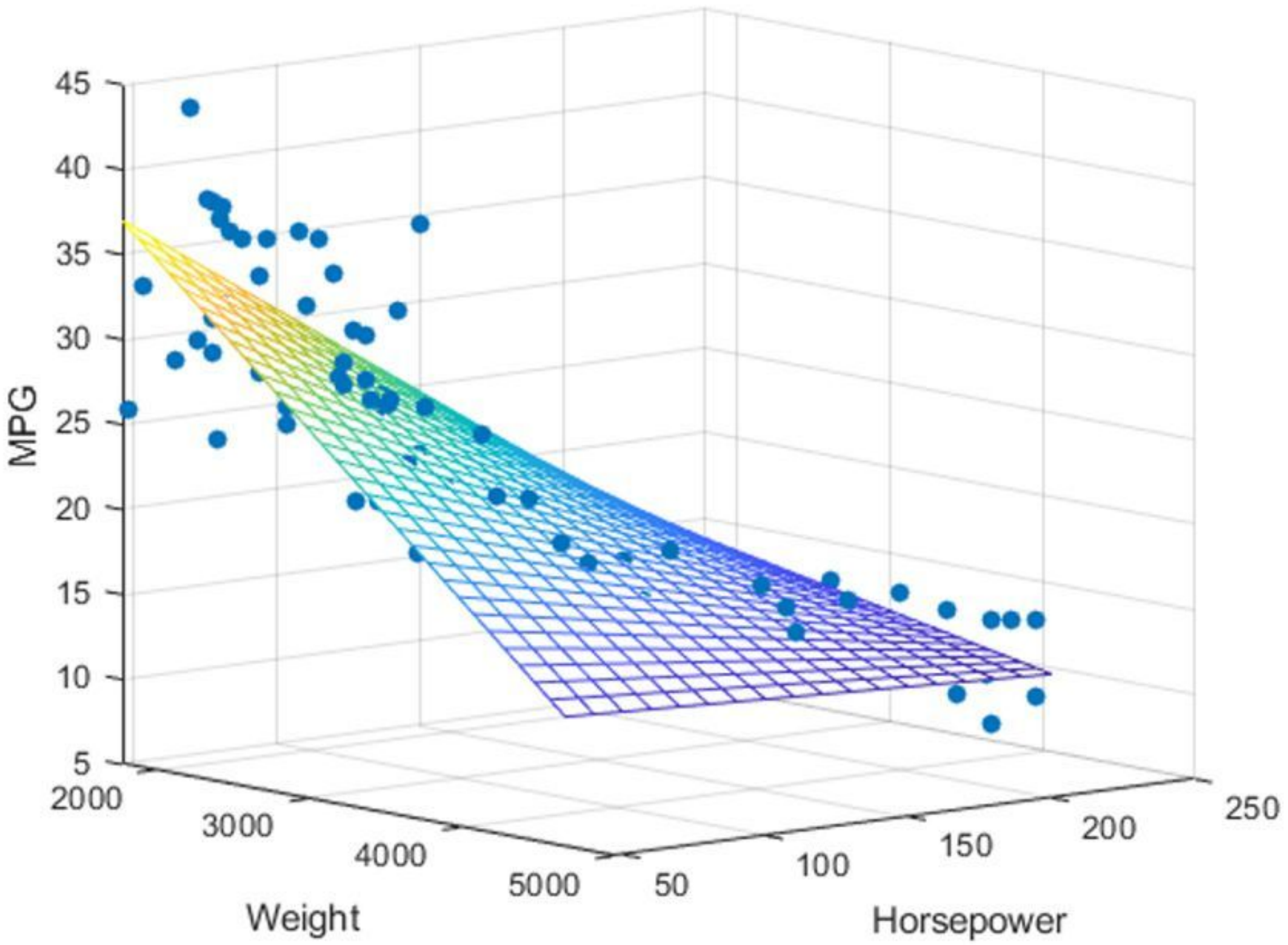
Multiple Linear Regression is a supervised learning regression technique used to predict one continuous target value using **two or more input features.**

Multiple Linear Regression

Multiple Linear Regression is a supervised learning regression technique used to predict one continuous target value using **two or more input features.**

lin_acc_x	lin_acc_y	ang_vel_z	robot_linear_velocity
0.10	0.01	0.01	0.43
0.13	0.06	0.02	0.44
0.13	0.03	0.03	0.37
0.14	NaN	0.02	0.48
0.17	0.04	0.01	0.49

Multiple Linear Regression



Multiple Linear Regression

Mathematical model

The equation of Multiple linear regression is:

$$y = b0 + b1X1 + b2X2 + \cdots + bnXn$$

Where:

- **y** → predicted output (target)
- **b0** → intercept (bias term)
- **X1, X2, …, Xn** → input features
- **b1, b2, …, bn** → coefficients (weights of each feature)

Multiple Linear Regression

Mathematical model : Example

Predict house price

- **x1** → house size
- **x2** → number of rooms
- **x3** → distance from city
- **y** →price

price= b0 + b1(size) + b2(rooms) + b3(distance)

Data Distribution

Data distribution describes **how values in a dataset are spread or arranged**.

It tells us:

- which values occur **most frequently**
- how **spread out** the data is
- whether the data is **symmetric or asymmetric**

Data Distribution

Normal Distribution

A **normal distribution** is a **symmetric, bell-shaped** distribution.

Key characteristics

- Most values are near the **center (mean)**
- Left and right sides are **mirror images**
- Mean ≈ Median ≈ Mode

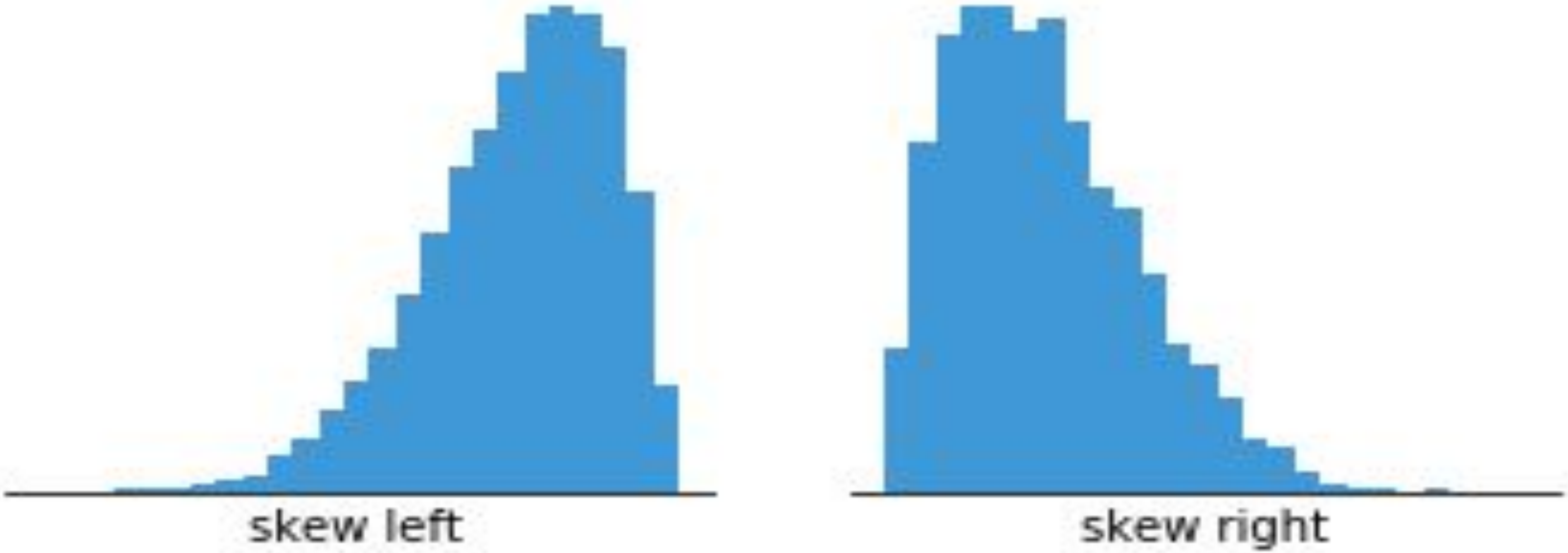
Intuition

👉 Data is evenly spread around the average.

Data Distribution

Skewed Distribution

A **skewed distribution** is **not symmetric**.
One side has a **longer tail** than the other.

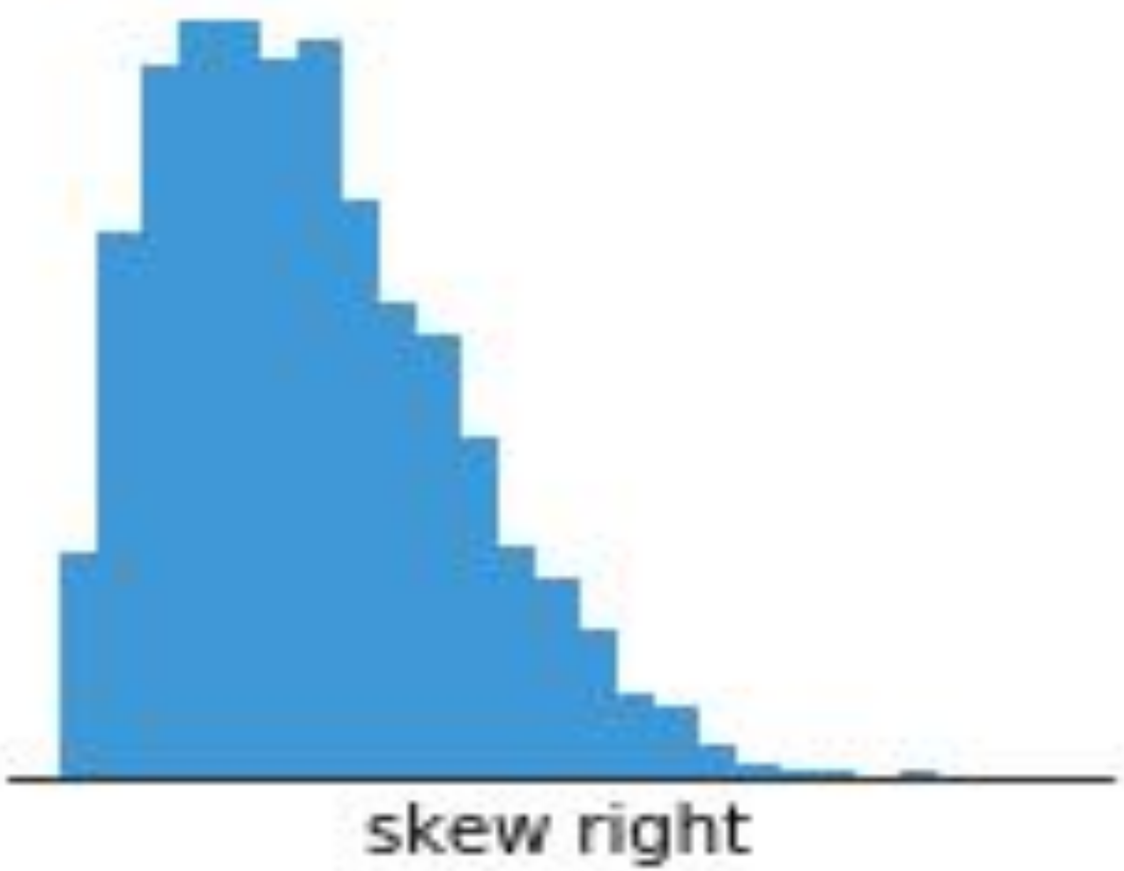


Data Distribution

Skewed Distribution

1. Right-skewed (Positive skew)

- Tail is **long on the right side**
- Most values are **small**
- Mean > Median

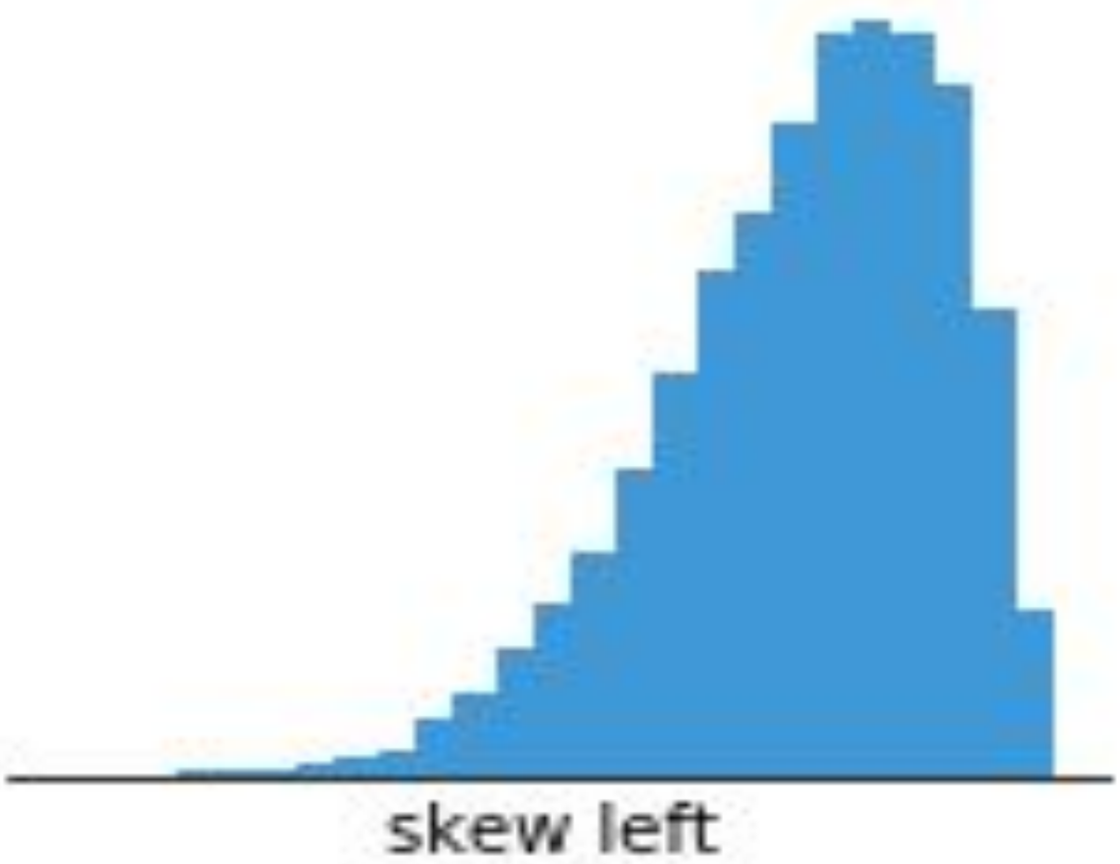


Data Distribution

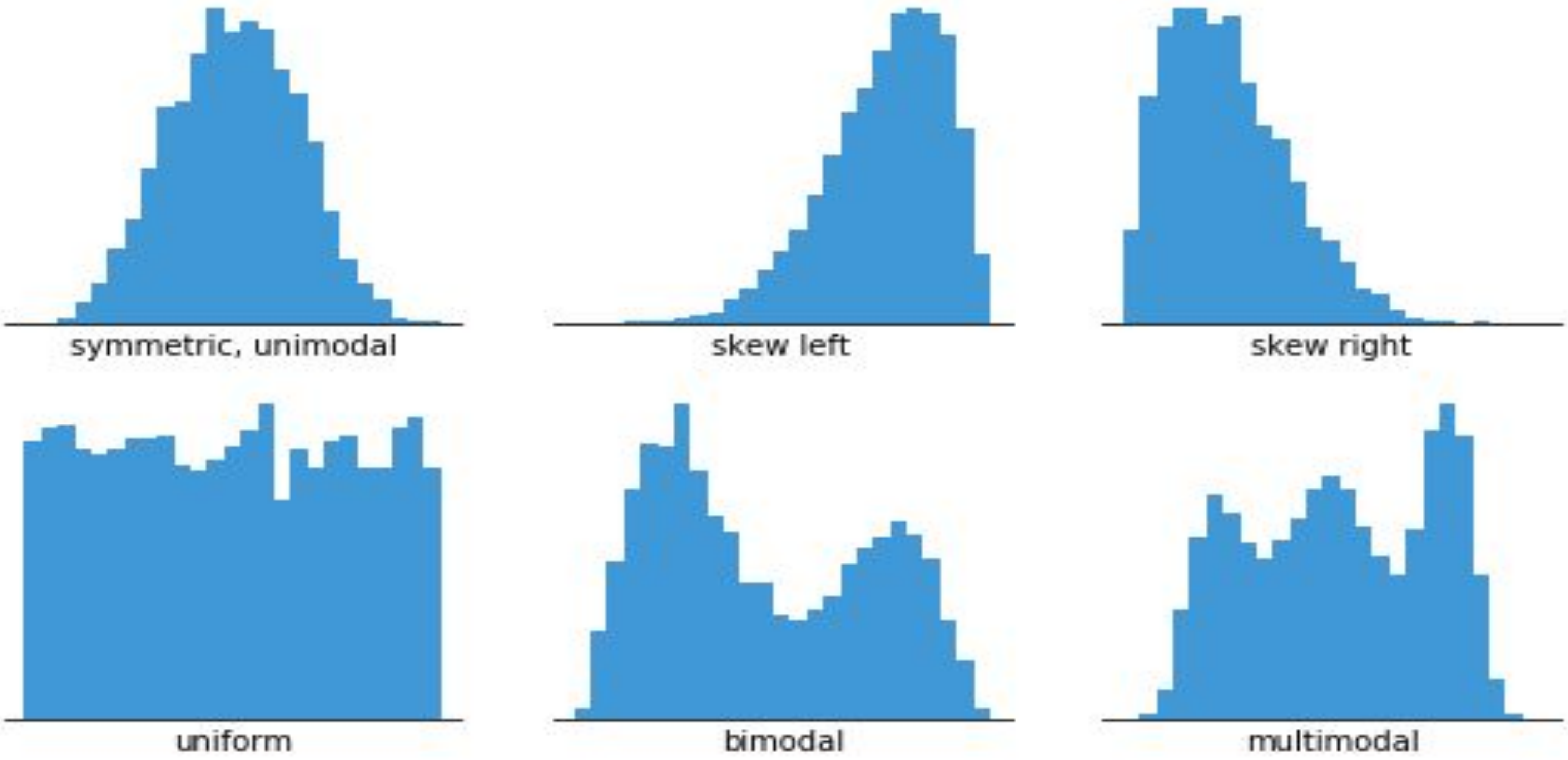
Skewed Distribution

2. Left-skewed (Negative skew)

- Tail is **long on the left side**
- Most values are **large**
- Mean < Median



Data Distribution



ANY QUESTIONS ?

THANK YOU

Contact Me

✉️ Email: shafatsib@gmail.com 📞 Phone: +880178-0404749 📍 Dhaka, Bangladesh

Google Scholar Github Linkedin Youtube