**🧠 What is Multiple Linear Regression?**

**Multiple Linear Regression** is a **Supervised Learning regression technique** that predicts a **continuous target variable** based on **two or more input features**.

“How multiple factors together influence the outcome?”

**⚙️ Formula**

Y=b0+b1X1+b2X2+...+bnXnY = b\_0 + b\_1X\_1 + b\_2X\_2 + ... + b\_nX\_nY=b0​+b1​X1​+b2​X2​+...+bn​Xn​

Where:

* **Y** = Predicted output
* **X₁, X₂, …, Xn** = Input features (independent variables)
* **b₁, b₂, …, bn** = Coefficients (effect of each feature on Y)
* **b₀** = Intercept (Y when all X’s = 0)

**💡 Example**

Predict **house price** based on **size**, **number of bedrooms**, and **age of the house**:

| **Size (sqft)** | **Bedrooms** | **Age (years)** | **Price (₹ lakhs)** |
| --- | --- | --- | --- |
| 1000 | 2 | 5 | 50 |
| 1500 | 3 | 10 | 75 |
| 2000 | 4 | 2 | 100 |
| 2500 | 4 | 15 | 120 |

Here:

* **X₁** = Size
* **X₂** = Bedrooms
* **X₃** = Age
* **Y** = Price

The model learns how **all three factors** together influence the price.

**📊 Visualization Concept**

For **two features**, the prediction is a **plane** instead of a line:

Price (Y)

|

| \*

| \*

| \*

| \*

| \*

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Size (X1)

Bedrooms (X2)

With more than 2 features, it becomes **multi-dimensional**, but the idea is the same: find the best-fit hyperplane.

**⚙️ Python Example**

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

# Example data

X = [

[1000, 2, 5],

[1500, 3, 10],

[2000, 4, 2],

[2500, 4, 15],

[3000, 5, 8]

] # Features: Size, Bedrooms, Age

y = [50, 75, 100, 120, 140] # Price

# Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Predict

y\_pred = model.predict(X\_test)

# Evaluate

print("Predictions:", y\_pred)

print("Mean Squared Error:", mean\_squared\_error(y\_test, y\_pred))

**📏 Evaluation Metrics**

Same as simple linear regression:

* **MSE (Mean Squared Error)**
* **RMSE (Root Mean Squared Error)**
* **R² Score (Coefficient of Determination)**

**🌍 Real-World Applications**

| **Domain** | **Use Case** |
| --- | --- |
| 🏠 Real Estate | Predict house prices using size, location, bedrooms, age |
| 💰 Finance | Predict revenue using ad spend, market trend, season |
| 🏥 Healthcare | Predict patient recovery time using age, BMI, treatment type |
| 🛍️ Retail | Forecast sales using price, promotion, season, competition |
| 🌡️ Weather | Predict temperature using humidity, pressure, wind, season |

**✅ Advantages**

* Captures effect of **multiple variables** simultaneously
* Provides **insights into which features affect the output most**
* Flexible and widely used in real-world problems

**⚠️ Limitations**

* Assumes **linear relationship** between features and output
* Sensitive to **multicollinearity** (features highly correlated with each other)
* Sensitive to **outliers**
* Requires careful **feature scaling and selection** for accuracy

**🧩 Summary**

| **Feature** | **Multiple Linear Regression** |
| --- | --- |
| **Goal** | Predict continuous value using multiple features |
| **Output Type** | Continuous (numbers) |
| **Algorithm Type** | Supervised Learning (Regression) |
| **Formula** | Y=b0+b1X1+...+bnXnY = b\_0 + b\_1X\_1 + ... + b\_nX\_nY=b0​+b1​X1​+...+bn​Xn​ |
| **Metrics** | MSE, RMSE, R² |
| **Applications** | House price prediction, sales forecasting, patient recovery, stock prediction |