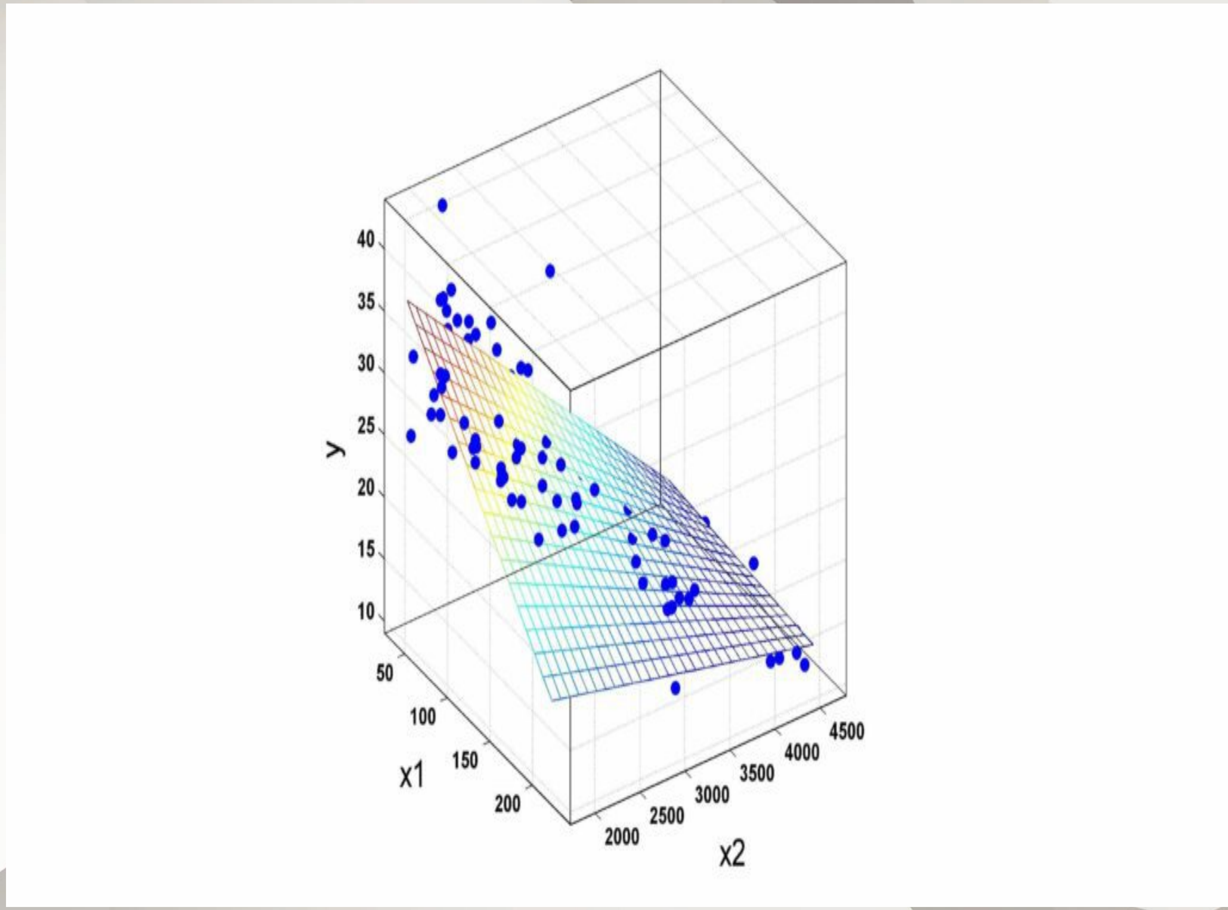


MULTIPLE LINEAR REGRESSION I

Unit- III

Multiple linear Regression:

- It’s a type of regression model that predicts a **dependent variable** (target) using **two or more independent variables** (features).
- **Example:**
  Predicting house price using square footage, number of bedrooms, and distance to a school.

✅ **Equation :**

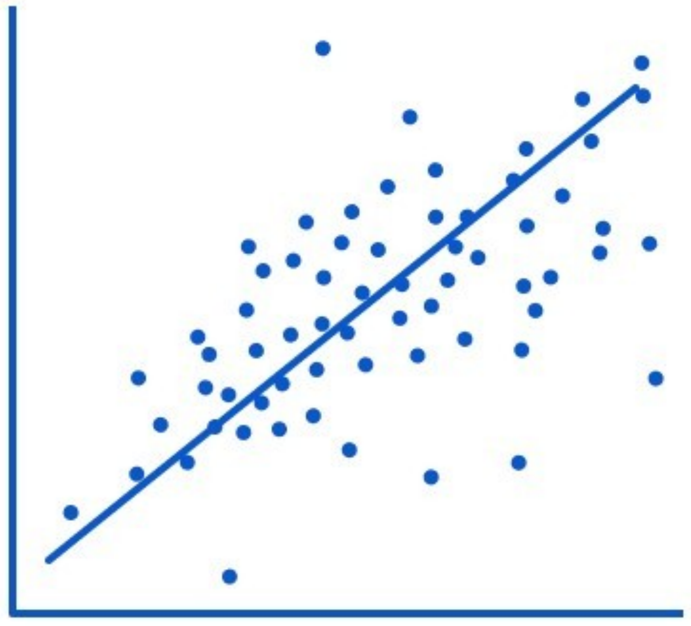
The mathematical formula looks like this

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n$$

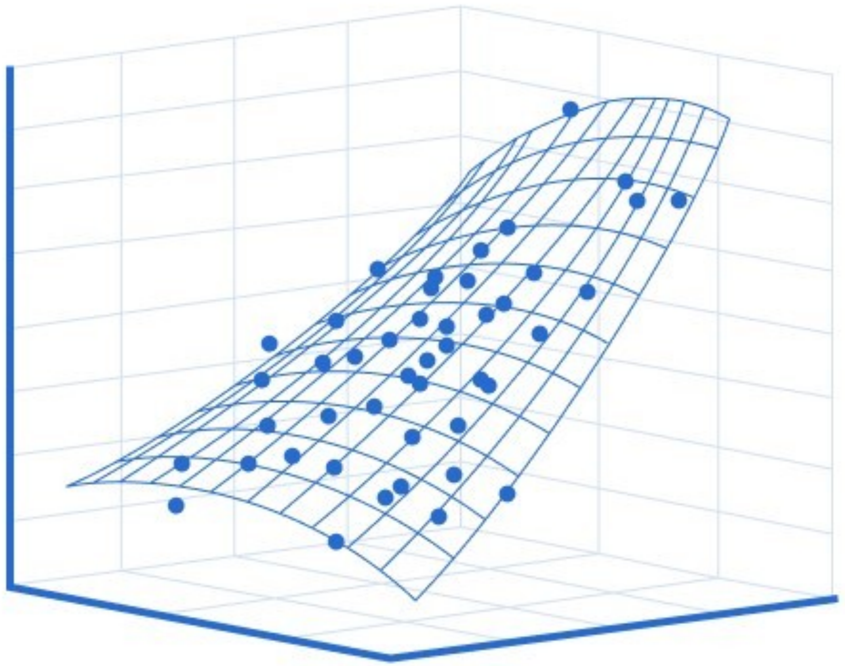
Where:

- y = predicted value (target variable)
- β_0 = intercept (constant term)
- $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ = coefficients for each independent variable
- $x_1, x_2, x_3, \ldots, x_n$ = independent variables (features)

Simple Linear Regression



Multiple Linear Regression



How it works:

1 Data Collection

You gather data with multiple **features** and a **target** variable. Example:

Size (sq ft)	Bedrooms	Distance to School (km)	Price (target)
1500	3	2	300000
1800	4	1	350000
1200	2	5	200000

2 Model Assumption

The model assumes:

- There’s a **linear relationship** between features and target.
- Features are **independent** from each other (no strong correlation).

3 Training the Model

- The model calculates **coefficients** (β values) that best fit the data.
- This is done using techniques like **Ordinary Least Squares (OLS)** which minimizes the difference between actual and predicted values.
- In simple terms, the model **finds the line/plane/hyperplane that fits the data best in higher dimensions**.

4 Prediction

Once trained, the model can predict new values by plugging in new feature values into the equation.

Example:

$$\text{Price} = 50000 + 120 \cdot (\text{size}) + 20000 \cdot (\text{bedrooms}) - 5000 \cdot (\text{distance})$$

For a house with:

- Size = 1600 sq ft
- Bedrooms = 3
- Distance = 2 km

$$\text{Price} = 50000 + 120 \cdot 1600 + 20000 \cdot 3 - 5000 \cdot 2$$

$$= 50000 + 192000 + 60000 - 10000 = 292000$$

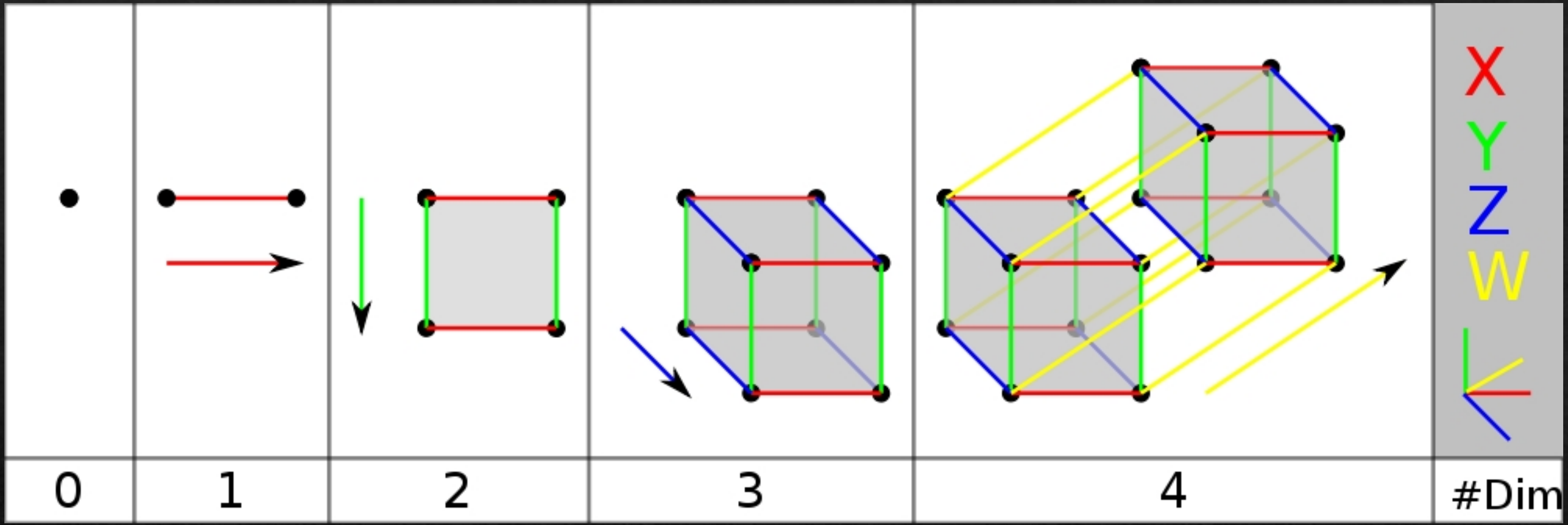
Evaluation

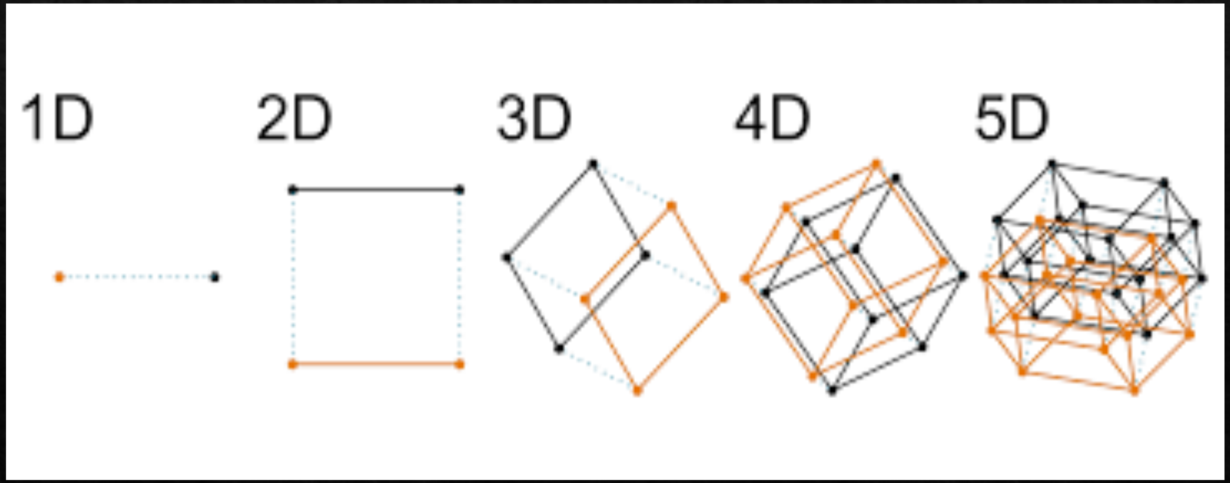
You check how well the model performs using metrics like:

- **R^2 (R-squared)** — how much variance in the target is explained by features
- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**

Number of Dimensions = Number of Features (Independent Variables)

- If you have **1 independent variable**, the regression is a **line** (2D space).
- If you have **2 independent variables**, the regression is a **plane** (3D space).
- If you have **3 independent variables**, the regression fits into a **4D space**.
- And so on…


X
Y
Z
W
0
1
2
3
4
#Dim


1D
2D
3D
4D
5D

What Happens in Higher Dimensions?

In math, the regression finds a **hyperplane** (a generalized plane) that best fits the data in high-dimensional space.

- 2 features: 3D plane
- 3 features: 4D hyperplane
- 10 features: 11D hyperplane

🧠 **Key Takeaway:**

Multiple linear regression can fit into any dimensional space depending on how many features you have — it is not limited to 3D.

How Do We Fit the Plane?

1 Ordinary Least Squares (OLS)

This is the most common method.

- It calculates the plane (coefficients) by **minimizing the sum of squared residuals**.
- A residual is the difference between the actual y value and the predicted y value.

The cost function (which we minimize) is:

$$\text{Cost} = \sum (y_i - \hat{y}_i)^2$$

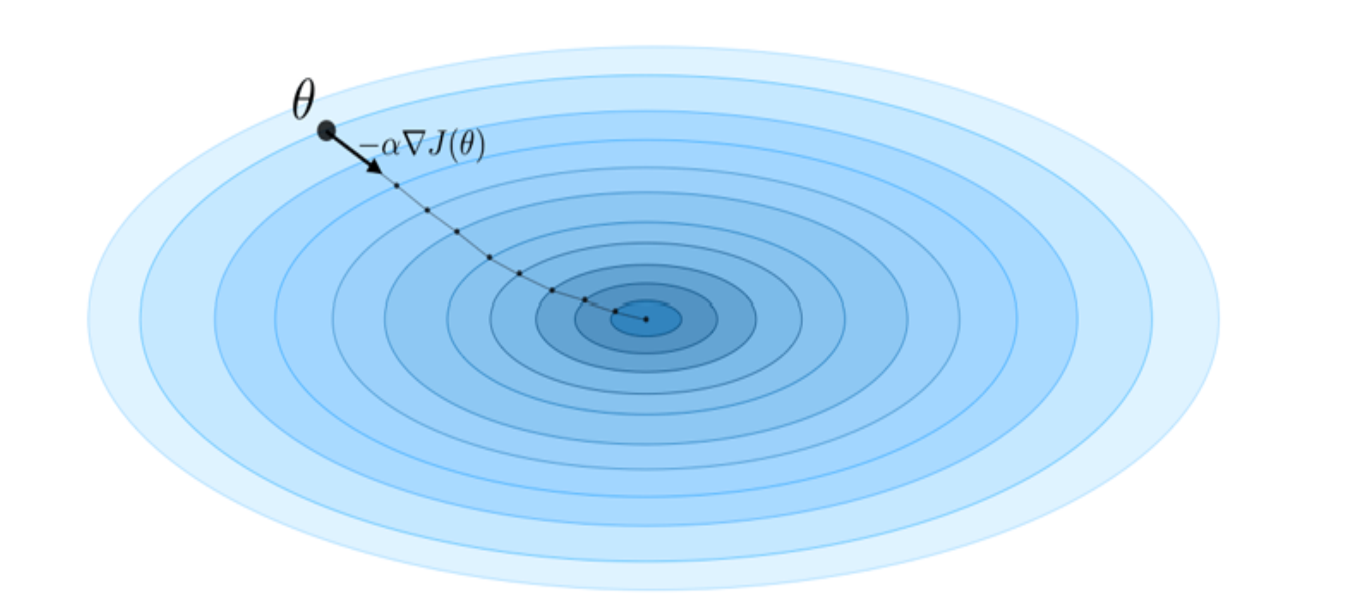
where:

- y_i = actual value
- $\hat{y}_i$ = predicted value using the plane equation

OLS uses **matrix algebra** to directly calculate the optimal β coefficients.

2 Gradient Descent (Alternative)

- For very large datasets, we might use **Gradient Descent**, an iterative algorithm that gradually adjusts the coefficients to minimize the cost.
- This is less common for basic multiple linear regression, but useful when data is massive.



1 Least Squares Estimation - multiple regression.

Let $\mathbf{y} = \{y_1, \cdots, y_n\}'$ be a $n \times 1$ vector of dependent variable observations. Let $\boldsymbol{\beta} = \{\beta_0, \beta_1\}'$ be the 2×1 vector of regression parameters, and $\boldsymbol{\epsilon} = \{\epsilon_1, \cdots, \epsilon_n\}'$ be the $n \times 1$ vector of additive errors. We construct the so-called *design matrix* X (dimension $n \times 2$) as follows:

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ 1 & x_n \end{pmatrix}$$

We can now write the simple linear regression model in two ways:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \; i = 1, \cdots, n, \tag{1}$$

or equivalently

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2}$$

The matrix formulation easily generalizes to multiple linear regression, involving predictor variables $x_1, \cdots, x_{p-1}$. We construct the $n \times p$ design matrix X:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & . & x_{1,p-1} \\ 1 & x_{21} & x_{22} & . & x_{2,p-1} \\ . & . & . & . & . \\ 1 & x_{n1} & x_{n2} & . & x_{n,p-1} \end{pmatrix}$$

The multiple regression can be written as

$$y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i, \;\; i = 1, \cdots, n, \tag{3}$$

or equivalently

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{4}$$

where $\boldsymbol{\beta} = \{\beta_0, \beta_1, \cdots, \beta_{p-1}\}'$.

We use Least-Squares to fit a regression line to the data $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where $\mathbf{x}_i = \{x_{i,1}, \cdots, x_{i,p-1}\}$. That is, we find the regression coefficient estimates $\widehat{\boldsymbol{\beta}}$ that minimizes the criterion

$$Q(\boldsymbol{\beta}) = (\mathbf{y} - X\boldsymbol{\beta})'(y - X\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\beta})^2.$$

Taking derivatives with respect to $\boldsymbol{\beta}$, and setting these to 0, we obtain the *normal equations*:

$$\frac{dQ}{d\boldsymbol{\beta}} = -2X'(\mathbf{y} - X\boldsymbol{\beta}) = \mathbf{0} \Rightarrow$$

$$(X'X)\boldsymbol{\beta} = X'\mathbf{y} \tag{5}$$

To solve for $\boldsymbol{\beta}$ we apply the inverse of $X'X$ to both sides of equation (5) and obtain:

$$\widehat{\boldsymbol{\beta}} = (X'X)^{-1}X'\mathbf{y} \tag{6}$$

$$\underset{n\times 1}{Y} = \underset{n\times p}{X}\underset{p\times 1}{\beta} + \underset{n\times 1}{\varepsilon}$$

No.

Date

p = No. of parameters

n = Sample size.

$\hat{y} \to$ predicted value.

$$e = y - \hat{y}$$

minimizing $\sum_{i=1}^{n} e_i^2 = \min e'e$

$$= \min (y-\hat{y})'(y-\hat{y})$$

$$= \min (y-Xb)'(y-Xb)$$

$$\frac{\partial}{\partial b}(y-Xb)'(y-Xb) = 0$$

$$\frac{\partial}{\partial b}(y-b'X')(y-Xb) = 0$$

$$\frac{\partial}{\partial b}(y'y - y'Xb - b'X'y + b'X'Xb)$$

$$\frac{\partial}{\partial b}(y'y - 2b'X'y + b'X'Xb) = 0$$

$$-2X'y + 2X'Xb = 0$$

No.

Date

$$2x'xb = 2x'y$$

$$x'xb = x'y$$

multiplying $(x'x)^{-1}$ both the sides

$$(x'x)^{-1}(x'x)b = (x'x)^{-1}x'y$$

$$b = (x'x)^{-1}x'y$$

Least Square Estimator

$$\boxed{b = (x'x)^{-1}x'y}$$