**Linear Regression:**

Linear Regression is a **supervised** learning algorithm used to predict a continuous numeric value based on one or more input features.

It assumes there is a linear relationship between the input variables (X) and the output variable (Y).

* Easy to Understand
* Works well when data follows a linear trend
* Assumes linearity (fails if the relationship is non linear)
* Performance drops if there is multicollinearity
* Use this when u have to predict continuous outcomes e.g, prices, scores, sales
* When data shows a linear relationship

**Simple Linear Regression:**

* It has one input column and only one output column
* Models a straight line relationship between X and Y
* **Use Case Example:** Predicting marks based on hours studied.

**Multiple Linear Regression:**

* It has two or more input columns (independent variables) and only one output column (dependent variable)
* Models a straight line relationship in multi dimensional space
* **Use Case Example:** Predicting house prices based on area, number of bedrooms, location rating, and age of the house

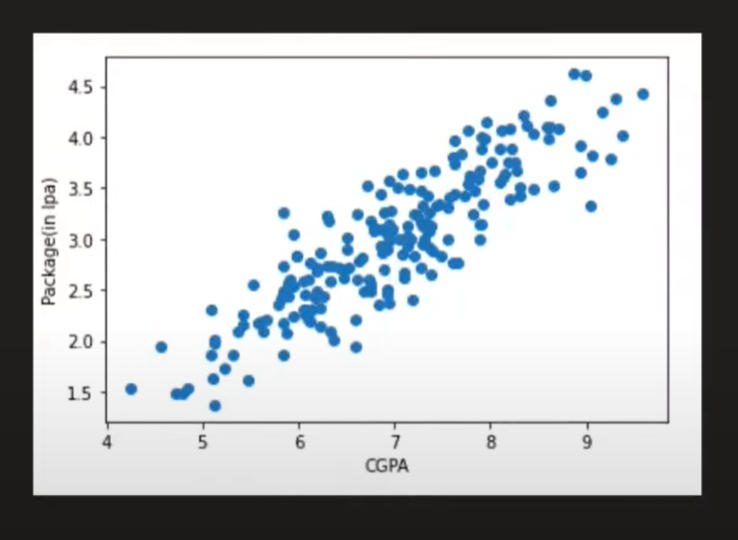
**Polynomial Linear Regression:**

* It has one input column, but the model includes higher degree powers of that input (X², X³, …)
* Models a curved (non linear) relationship between X and Y while still being linear in terms of coefficients
* **Use Case Example:** Predicting car price depreciation over years, where the price drops non linearly with time

**Linear Regression:**

In linear regression, if we plot a graph of two columns, it must be linear.

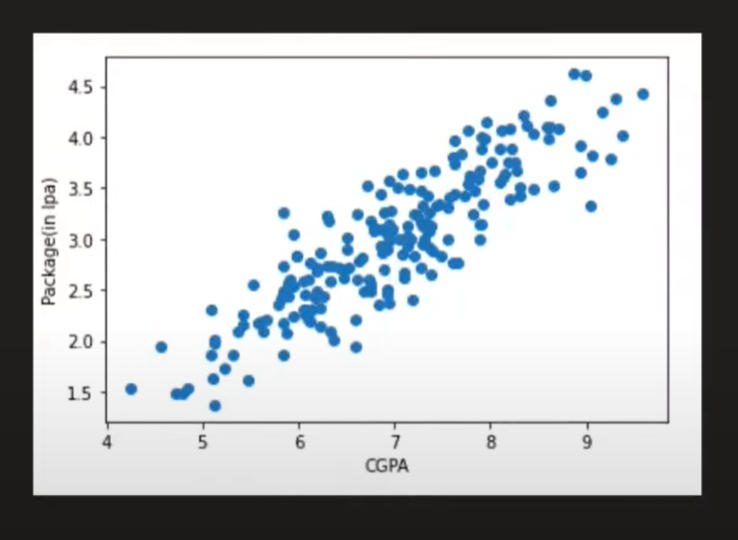
But in case of real world data, the graph may or may not be linear due to real world factor.

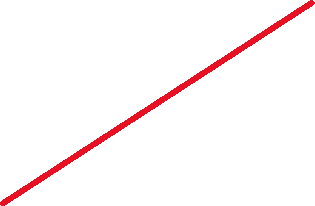


This is a graph of CGPA vs Salary. As it’s a real world example. The graph isn’t linear. The factors could be anything.

This is called stochastic errors, due to which the data is sort of linear but not entirely linear.

To solve this problem easily, we draw a **best fit line.** It isa linear line from the origin which aims to minimize the distance between the line and all the data points





This is a best fit line, it does the minimum mistake in the calculations.

**We use the following eq in this calculation:**

***y = mx+c***

*where m is slope and c is y-intercept*

**In 3D, we use:**

***y = mx1+nx2+b***

OR

***y = β0+β1x1+ β2x2***

**For n-dim:**

***y = β0+β1x1+ β2x2*** +…… ***βnxn***

*where β is called weight, tells us the weightage of each column.*

In multiple linear regression, we formulate the value of coefficients using the formula:

*Β = (XTX)-1 \*XTY*

*where X = x\_train, Y = y\_train*

**Regression Metrics:**

A screenshot of a computer

AI-generated content may be incorrect.

