Simple Linear Regression

It is a supervised machine learning algorithm.

As the name suggests it is a regression algorithm that solves to find numerical value.

Linear Regression → Simple LR

Linear Regression → Multi LR

Linear Regression → Polynomial LR

Cgpa	Package
6.66	3.01
7.20	8.20

Simple LR – 1 input

Multiple LR – More than 1 i/p

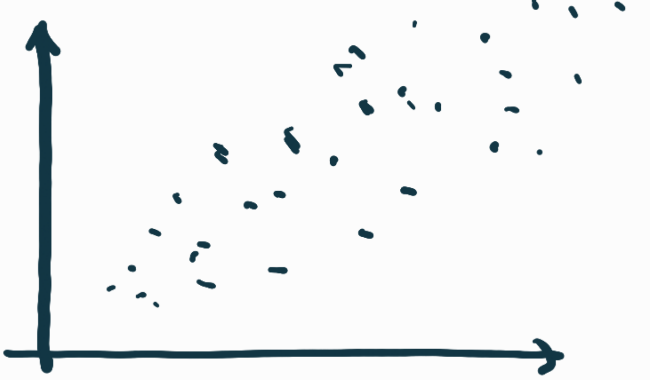
polynomial – when data is not in linear fashion

Geometric Intuition of Simple LR

Ex.

CGPA	package
7.1	3.5
4.7	1.2
8.9	4.2
8.1	3.9

plot this data



sort of linear

In real world scenarios, the data is not completely linear.

"Stochastic error" → some factors that are not mathemati quantifiable.

Eg. Luck, etc.

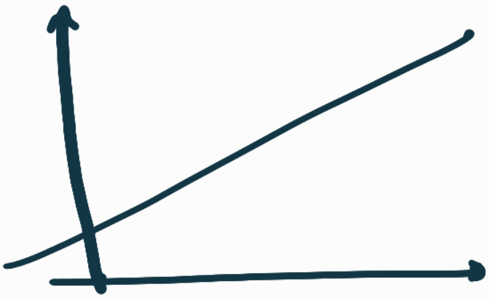
$$\begin{cases} m \rightarrow \text{slope} \\ b \rightarrow y \text{ intercept} \end{cases}$$

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

* We can use this line to determine package.

  but our data is not perfectly linear

* We will draw a best fit line that will try to go as closely as possible with all the data points.



$y = mx + b$

package = $m \times$ cgpa $+ b$

(slope) $m \rightarrow$ weightage

(intercept) $b \rightarrow$ offset

$y = mx + b$

↑

(m, b)

Closed-form soln. (Mathematical form.)

* OLS
* Difficult to use in higher dimn.
* Scikit learn

Non-closed soln. (Approximation)

* Gradient Descent
* Easier to use in higher dimn
* SGD Regr

OLS

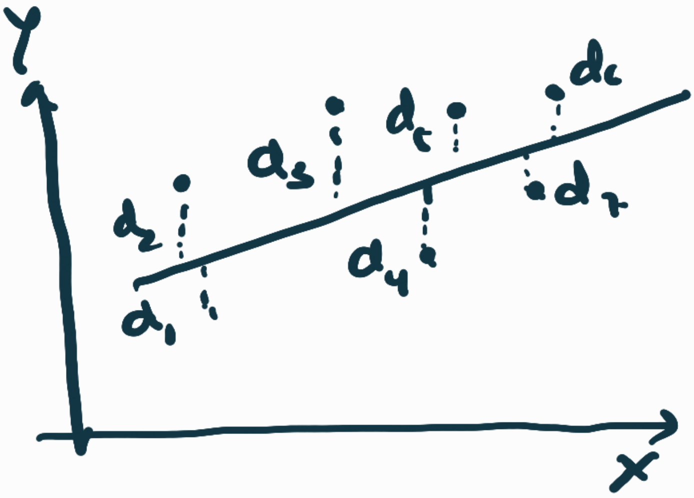
$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$b = \bar{y} - m\bar{x}$$

$\bar{x}$ – mean x

$\bar{y}$ – mean y

Derivation



Here, we use $d^2 \rightarrow$ to make sure +ve and -ve error are both counted.

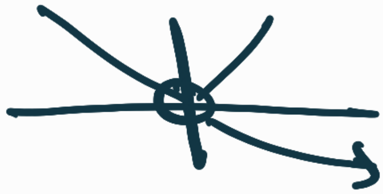
$$E = d_1 + d_2 + d_3 + \dots + d_n$$

$$E = d_1^2 + d_2^2 + d_3^2 + \dots + d_n^2$$

Why are we not using $|d|$:

1) d^2 helps to penalize the outliers.

2) |d| graph is non differentiable at one point



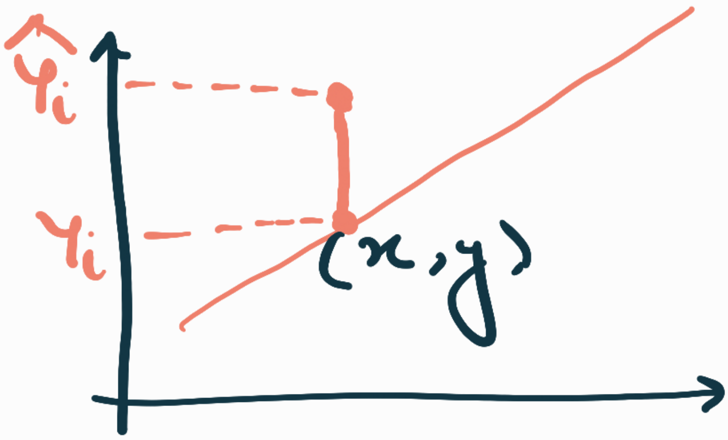
→ non-diff^n

$$\boxed{E = \sum_{i=1}^{n} d_i^2}$$ ← Error function (J)

minimize the error

$d_i = (y_i - \hat{y}_i)$

error for one pt.



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

↳ not m & b in the eq^n

$\hat{y}_i = mx + b$

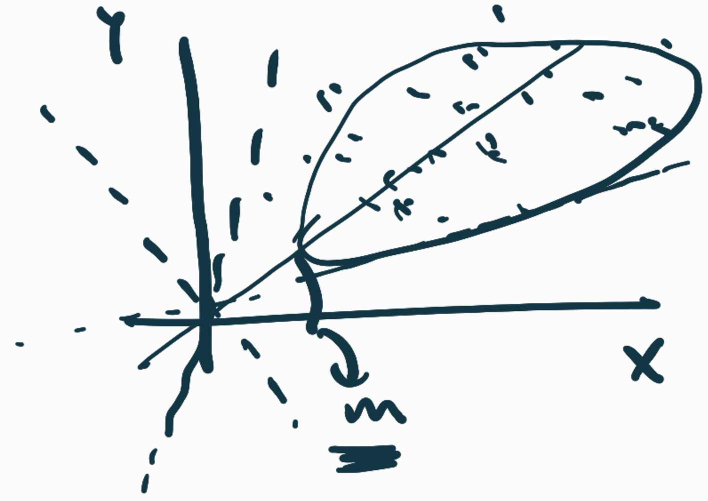
$$\boxed{E(m,b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2}$$

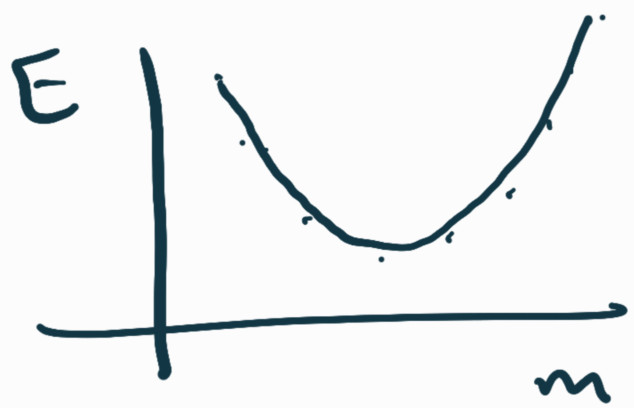
$$\boxed{E(m,b) = \sum_{i=1} (y_i - mx_i - b)}$$

Now, we need to find that value of $E(m,b)$ where E is minimum.

So, lets consider b as 0.

$$\Rightarrow E(m) = \sum_{i=1}^{n} (y_i - \underline{m} x_i)^2$$

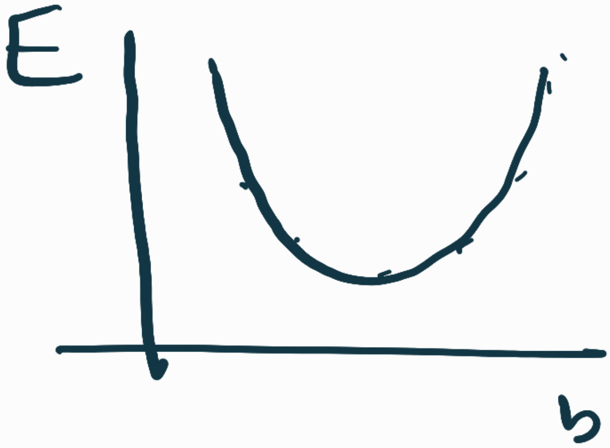


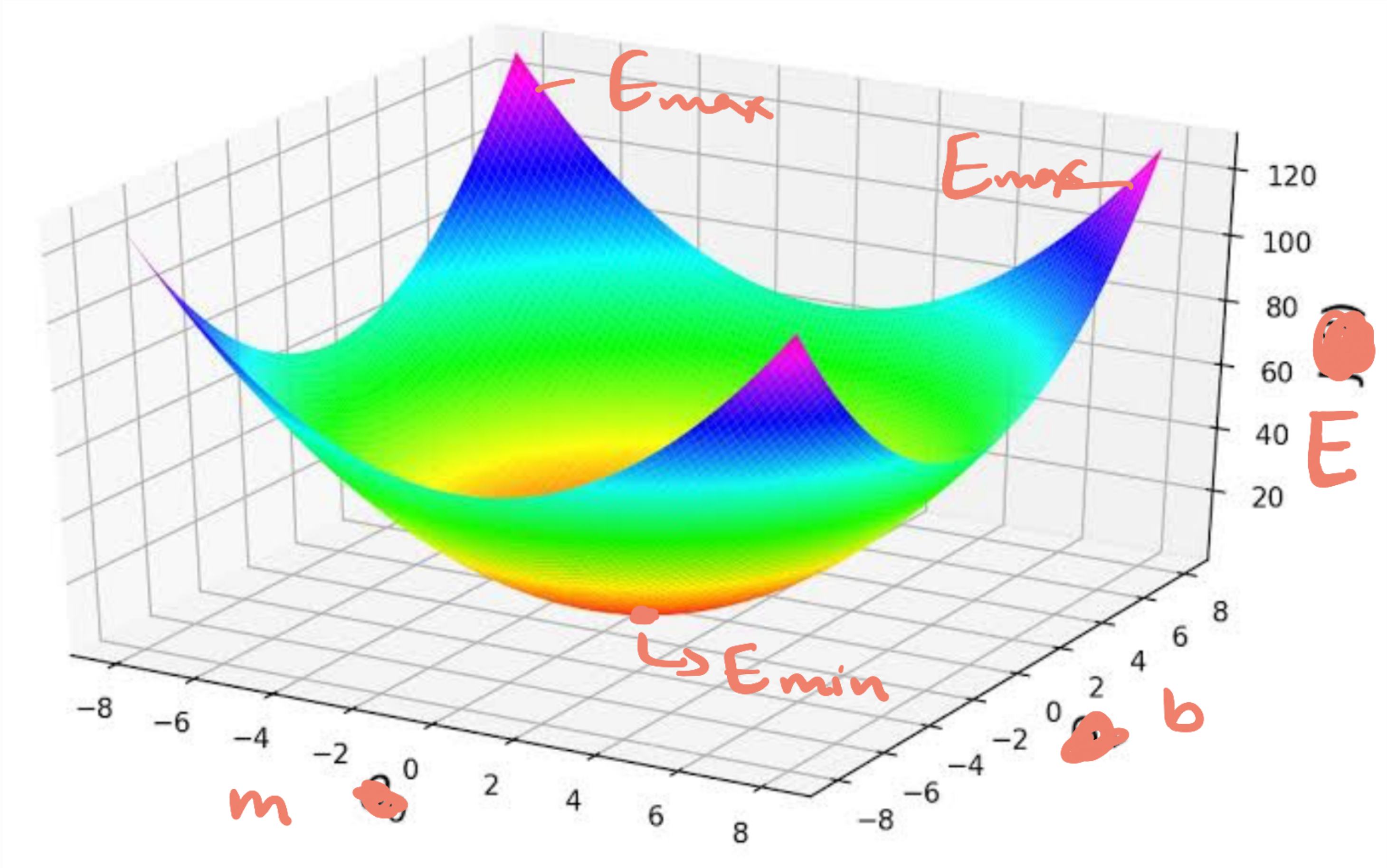


when m changes such that it moves towards

Now, lets consider m as const.

$$E(b) = \sum_{i=1}^{n} (y_i - x_i - b)^2$$





maxima , minima

$f(x)$

$$\frac{d\,f(x)}{dx} = 0$$

$E(m,b)$

$$\frac{\partial E}{\partial m} = 0 \,, \quad \frac{\partial E}{\partial b} = 0$$

$$\frac{\partial E}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^{n} (y_i - m x_i - b)^2 = 0$$

$$\Rightarrow \sum \frac{\partial}{\partial b} (y_i - m x_i - b)^2 = 0$$

$$\Rightarrow -\sum 2 (\overset{0}{y_i} - \overset{0}{m x_i} - \overset{-1}{b}) = 0$$

$$\Rightarrow \sum (y_i - m x_i - b) = 0$$

$$\Rightarrow \sum y_i - \sum m x_i - \sum b = 0$$

$$\Rightarrow \frac{\sum y_i}{n} - \frac{\sum m x_i}{n} - \frac{\sum b}{n} = 0$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\bar{y} - m \bar{x} - \frac{\cancel{n} b}{\cancel{n}} = 0$$

$$\bar{y} - m \bar{x} = b$$

$$\boxed{b = \bar{y} - m \bar{x}}$$

$$\frac{\partial E}{\partial m} = \sum (y_i - mx_i - \bar{y} + m\bar{x})^2 = 0$$

$$\Rightarrow \sum -2(y_i - mx_i - \bar{y} + m\bar{x})(x_i - \bar{x}) = 0$$

$$\Rightarrow \sum [(y_i - \bar{y}) - m(x_i - \bar{x}) - m(x_i - \bar{x})] = 0$$

$$\Rightarrow \sum (y_i - \bar{y})(x_i - \bar{x}) = m \sum (x_i - \bar{x})^2$$

$$m = \frac{\sum_{i=1}^{n} (x - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})}$$

<u>Important</u>

$$\boxed{m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}}$$

$$\boxed{b = y - mx}$$

$\bar{x}$ – mean x

$\bar{y}$ – mean y