1- Simple Linear Regression

- this will form a base for various deeplearning algorithms

We know in ML we have 2 types :-

① Supervised ML

② Unsupervised ML

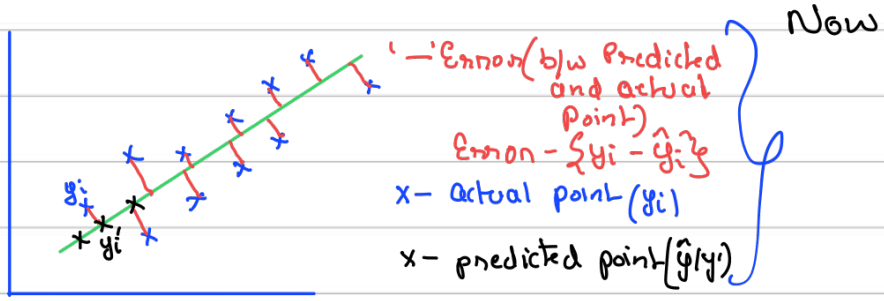
In Supervised ML we usually get 2 types of problem statements:-

Supervised ML → Regression : o/p feature → Continous
→ Classification : o/p feature → Binary or Multi class categories

Q) In a dataset Weight & Height are given. Now create a model to predict height if new weight is provided.

Dataset :

independent feature	dependent feature
Weight	Height



Now,

<u>the main aim of the Simple Linear Regression is to find out the best fit line (—) from which the summation should be minimal.</u>

above line can be represented from eq^n :-

$$\boxed{\hat{y} = mx + c}$$

or,

$$\boxed{\hat{y} = \beta_0 + \beta_1 x}$$

or

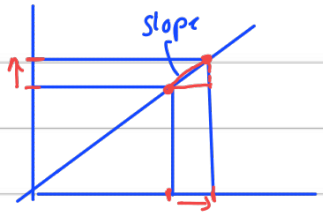
$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x}$$ ★

$h_\theta(x)$ cuz many researchers use this algo. for hypothesis testing.

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x}$$

where, $\theta_0 \rightarrow$ Intercept

$\theta_1 \rightarrow$ Slope or Coefficient



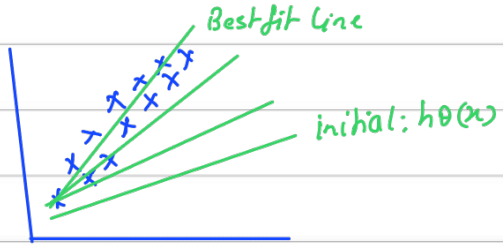
Slope : is basically the amount of moment in y-axis with a unit moment in x-axis.

Intercept: value of y when $x=0$ (i.e where the line intercepts y-axis at $x=0$).

1.1) Optimization Process :-

In optimization process initially θ_0 and θ_1 are initialized with any value and error is calculated and then the values of θ_0 & θ_1 are changed unit error becomes minimum.

visualization:-



Now,

for performing optimization we need to derive our "Cost function"

1.2) Cost Function :-

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x))^2}$$

also called as "Mean Squared Error"

n → no. of datapoints

y_i → actual point

$h_\theta(x)$ → Predicted point

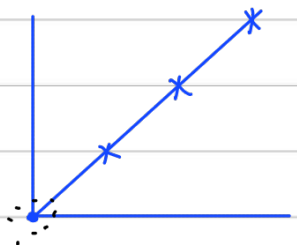
So,

our final aim is to minimize the cost function.

ex:-

x	y
1	1
2	2
3	3

$\Rightarrow$



$\theta_0 = 0 \Rightarrow h_\theta(x) = \theta_1 x_i$

① initialising θ_1 such that $\theta_1 = 1$

$\therefore$, at $x=1$: $h_\theta(x) = 0 + 1(1) \Rightarrow h_\theta(x) = 1$

$x=2$: $h_\theta(x) = 0 + 1(2) \Rightarrow h_\theta(x) = 2$

$x=3$: $h_\theta(x) = 0 + 1(3) \Rightarrow h_\theta(x) = 3$

now,

calculating cost function :- $J(\theta_1, \theta_0) = \frac{1}{n} \Sigma (h_\theta(x) - \hat{y}_i)^2$

$\Rightarrow$

$$J(\theta_1) = \frac{1}{n} \sum_{i=1}^{3} (y_i - h_\theta(x))^2$$

$$= \frac{1}{3} \times [(1-1)^2 + (2-2)^2 + (3-3)^2]$$

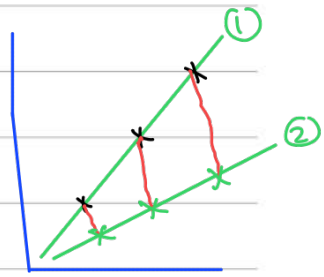
$$J(\theta_1) = 0$$

② <u>lets initialise θ_1 such that $\theta_1 = 0.5$</u> :-

$\Rightarrow$ at $x=1$: $h_\theta(x) = 0 + 0.5(1) = 0.5$

$x=2$: $h_\theta(x) = 1$

$x=3$: $h_\theta(x) = 1.5$



calculating cost function:-

$$J(\theta_1) = \frac{1}{3} \times [(1-0.5)^2 + (2-1)^2 + (3-1.5)^2]$$

$$= \frac{1}{3} \times [(0.5)^2 + (1)^2 + (1.5)^2 + (1.5)^2]$$

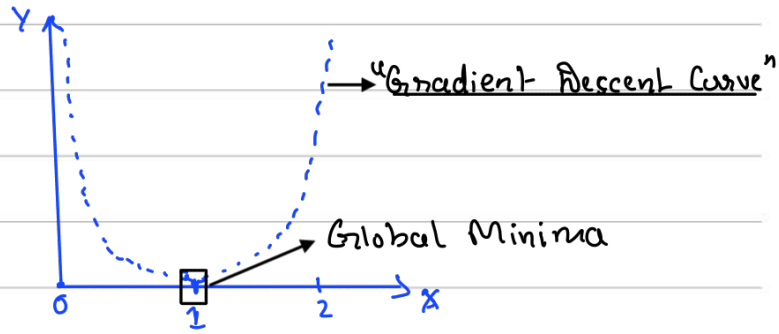
$$= \frac{1}{3} (0.25 + 1 + 2.25)$$

$$= \frac{1}{3} \times 3.5$$

$$= \underline{\underline{1.1\bar{6}}}$$

lly. $J(\theta_1 = 0) = 14/3$

<u>Plotting ($J(\theta_1)$ Vs θ_1)</u>:



Global Minimal: at global minima the cost function is minimum and thus error is also minimum.

☆ But, we can't keep initializing θ_0 and θ_1 randomly. We need a mechanis for initialization.

∴,

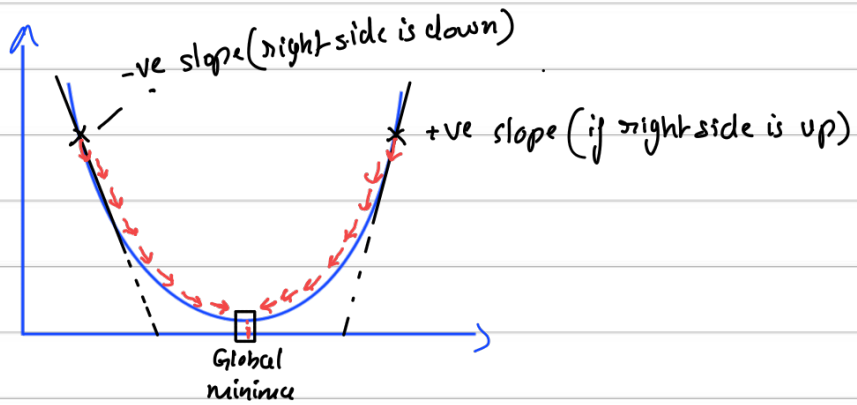
1.37 Convergence Algorithm :-

This algo. is used to optimize the change of θ_1 (θ_0 also if at $x=0 \rightarrow y \neq 0$ i.e intercept is not zero)

$$\boxed{\theta_j : \theta_j - \alpha \frac{\partial J(\theta_j)}{\partial \theta_j}}$$

, $\alpha \rightarrow$ Learning rate (decides speed of convergence)

$\frac{\partial J(\theta_j)}{\partial \theta_j}$: slope at a point in $J\theta_j$ vs θ_j graph



① Suppose 1st pt at +ve slope

$\theta_{new} = \theta_1 - \alpha\,(+ve)$

$\theta_{new} = \theta_{old} - \alpha\,(+ve)$

$\Rightarrow \boxed{\theta_{new} < \theta_{old}}$

② Suppose 2nd point at -ve slope

$\theta_{1new} = \theta_{old} - \alpha\,(-ve)$

$\theta_{new} = \theta_{old} + \alpha\,(+ve)$

$\Rightarrow \boxed{\theta_{new} > \theta_{old}}$

1.4 Conclusion :-

Cost function :- used to calculate mean sq error

Gradient Descent :- $J(\theta_1)$ vs θ_1 graph to find Global minima

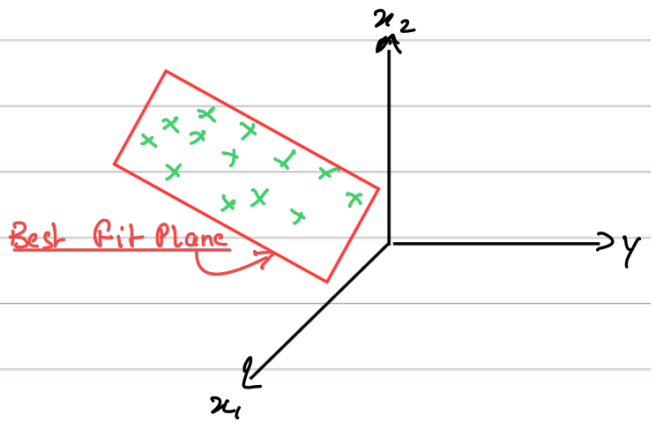
Convergence algo :- Repeate convergence unit reach global local

2- Multiple Linear Regression

- For multiple linear regression we must have more than one Independent features.

Ex:-

x_1	x_2	Y
No. of Rooms	Size of Rooms	Price



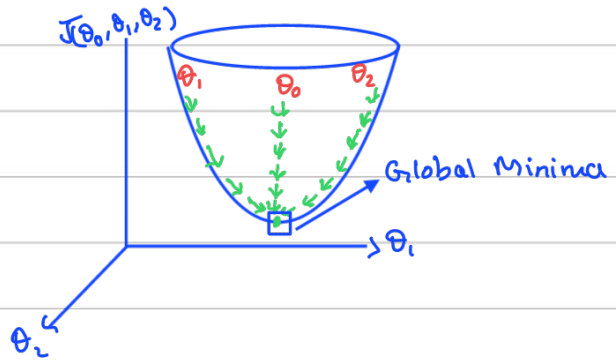
So equation for best fit plane becomes:

$$\boxed{\begin{array}{l} h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ \quad , \ \theta_1 \ \& \ \theta_2 : \text{Slope} \\ \quad \theta_0 : \text{Intercept} \end{array}}$$

Gradient Descent:-

Instead of a 2D parabolla in this case the gradient descent will look like a 3D parabolla.

ex:-



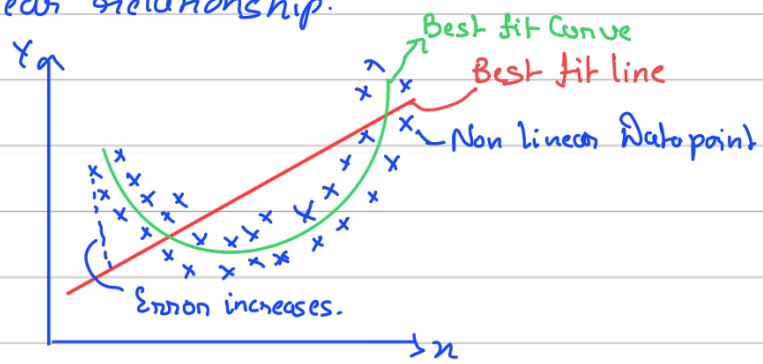
Generic Equation for Multiple Regression:-

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots\cdots + \theta_n x_n}$$

3- Polynomial Regression

this type of regression is used when our data is having a non-linear relationship.

eg:-



Now,

① Polynomial degree = 0 : $h_\theta(x) = \theta_0 * 1 = \theta_0 x_1^0$ (straight- line)

② Polynomial degree = 1: $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1$ (Simple Linear Regg.)

③ Polynomial degree = 2 : $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_2^2$

④ Polynomial degree = 3: $h_\theta(x) = \theta_0 x_0^0 + \theta_1 x_1^1 + \theta_2 x_2^2 + \theta_3 x_3^3$

⋮

Simple Polynomial Regression

✱ $\boxed{\text{Polynomial degree } n : h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_2^2 + \cdots + \theta_n x_1^n}$

also,

for multiple polynomial regression we will have multiple independent features like

$\underbrace{x_1 \quad x_2}_{\text{indipendent feature}}$ $\qquad$ $\underbrace{y}_{\text{dependent feature}}$

Such that,

eq^n for a dataset of polynomial degree = 2 we have

✱ $\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2}$

"Multi polynomial Regression"

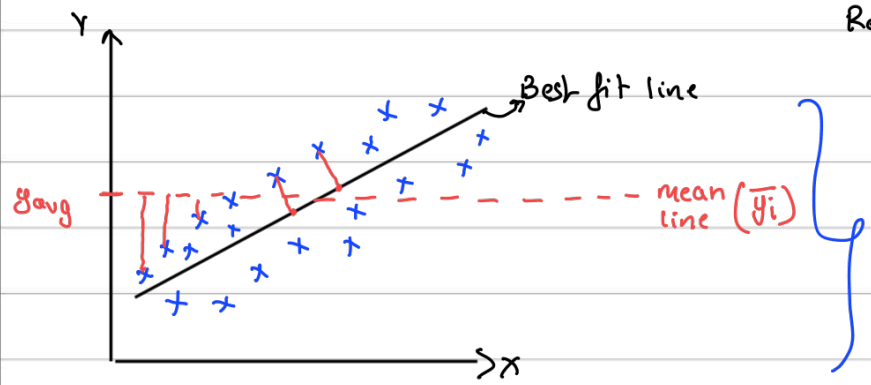
4- Performance Matrices Used in Linear Regression

4.1- R squared
4.2- Adjusted R squared

4.1) R-Squared Performance matrix :-

$$\boxed{R_{squared} = 1 - \frac{SS_{Res}}{SS_{Total}}}$$

SS_{Res} - Sum of Square Residuals (Error)
SS_{Total} - Sum of Squares Total
Residual - $y_i - \hat{y}_i$



SS_{Total} = Sum of Sq total

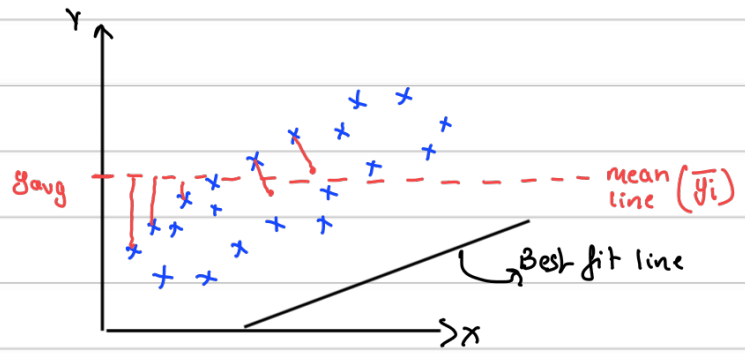
$$\Rightarrow \boxed{R_{squared} = 1 - \frac{\Sigma (y_i - \hat{y}_i)^2}{\Sigma (y_i - \bar{y}_i)^2}}$$

, Rsquared ↑ accuracy ↑es.

this R squared term is saying that the best fit line should be less than the avg of y_i.

Q) Can this Rsquared value be -ve?

Soln Yes it is possible only if the best fit line is farce than the avg line eg:-



$R_{sq} = 1$ means Overfitting Problem.

② Adjusted R squared:-

Suppose we have a feature dataset:-

Size of house (x_1)	No. of bedrooms (x_2)	Price (Y)

for the above dataset the Rsquared value comes out to be

$$R_{sq} = 0.85 = 85\%$$

Now,

we add another feature lets say 'Price' To the dataset and Price being highly related to the Price of the house will increase the R_{sq} value such that:

$$R_{sq} = 90\%$$

Now,

Lets add another feature 'Gender' to the dataset but a gender is not that impactfull in determining the price of the house ∴ R_{sq} value will increase but not that much;

$$R_{sq} = 91\%$$

So we can observe

✱ "Addition of any feature will increase the Rsquared value irrespective of that feature being important or irrelivant for determination of the feature.
This this won't happen in Adjusted Rsquared"

✱ $$\boxed{\text{Adjusted Rsquared} = 1 - \frac{(1-R^2)(N-1)}{N-P-1}}$$

, N = no. of data points
P = no. of Independent features

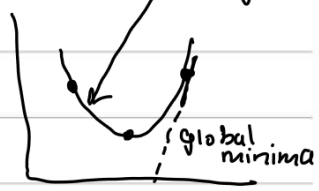
5- Important Cost Functions:(MSE, MAE, RMSE)

5.1- Mean Squared Error
5.2- Mean Absolute Error
5.3- Root Mean Squared Error

5.1) Mean Squared Error (MSE) :-

$$\boxed{MSE = J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

→ quadratic eq^n allows us to form parabola (Gradient descent) and find global minima also c/a "Convex function"



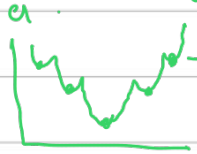
Advantage of using MSE:-

① It is differentiable. As it is a parabola curve differentiability is possible at any point

② It has one local and one global minima as it is a quadratic equation

for non quadratic eq^n we will have multiple local minima & 1 global minima.

ex:



– non convex function

Disadvantage of MSE

① Not robust to outliers

due to outliers and squaring of error the actual line will shift.

② It is not in the same unit

as the o/p is in Price (Rs) but the error being squared the o/p will be squares i.e (Rs^2)

Now to remove the above two disadvantages we use:-

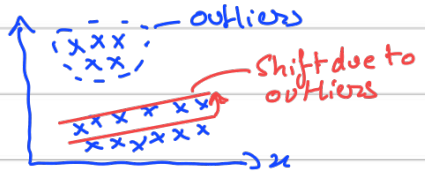
5.2 Mean Absolute Error:-

$$\boxed{MAE = \frac{1}{n}\sum_{i=1}^{n} |y - \hat{y}|}$$

there is no squaring of error in this due to which

Advantage:-

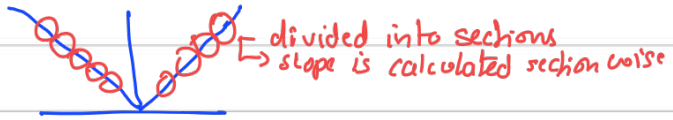
① It is robust to outliers :-



② It will be in the same unit : cuz no squaring.

Disadvantage:-

① Optimization is complex:-



② Convergence is time consuming.

5.3 Root Mean Squared Error:- (RMSE)

it is basically the root of MSE.

$$\boxed{RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}$$

Advantage:-

Same unit

also differential

Disadvantage

not robust to outliers

Note:- In case of "Linear Regression"

for performance check:- R^2 & Adjested R^2

for cost function :- MSE, MAE, RMSE.