① Linear Regression:

(i) Simple Linear Regression:-

Simple Linear Regression is a model for predicting the value of one dependent variable based on one independent variables.

$$Y = mx + c \qquad m \to \text{slop}$$
$$Y = b_0 + b_1 * x_1 \qquad c \to \text{intercept}$$

take a two continuous point

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

take a one point, to find c

$$y = mx + c$$

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

(ii) multiple Linear Regression:-

multiple linear regression is a model for predicting the value of one dependent variable based on two or more independent variables.

$$y = b_0 + b_1 * x_1 + b_2 * x_2 \dots + b_n * x_n$$

$b_1 \to$ slop

$b_0 \to$ intercept

$x_1, x_2 .. x_n \to$ features.

Advantages of linear Regression:-

① Very simple to implement

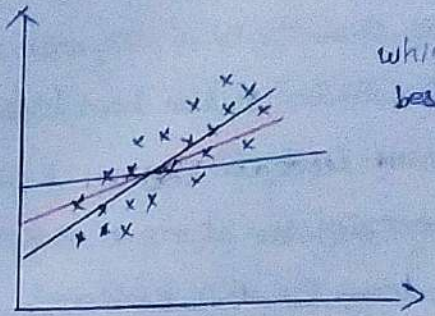
② Performs well on data with linear relationship.

Disadvantages of linear ~~relationship~~:

① Not suitable for data having non-linear relationship

② underfitting issue

③ sensitive to outliers.



Here Loss function comes to play, to find best fit line,

* Loss function:-

(i) Loss function measures how far an estimated value is from it's true value.
↳ predicted value

(ii) It's helpful to determine which model performs better & which parameter are better.

{ Very low Loss function it's called as a "Best fit" }

$$\text{Loss} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$n \to$ total number of data points.

$y_i \to$ Actual value

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

How loss function works??

Step1:-

It will Randomly assigned 'm' & 'c' {Slope & Intercept}

Step2:-

Calculate loss function:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Step3:-

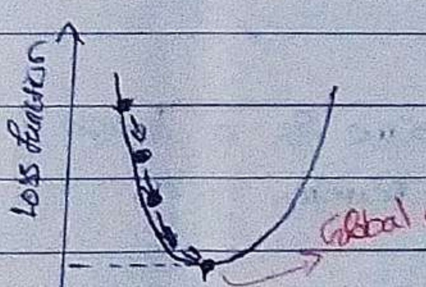
finding low loss function.

* model optimization:-

Optimization refers to determining best parameters for a model, Such that the loss function of the model decreases as a result of which the model can predict more accurately.

* Gradient Decent :-

How to find best parameters? here "Gradient decent" comes to play.



* Gradient Decent is an optimization algorithm used for minimizing the loss function in various machine learning algorithms. It is used for updating the parameters of the learning model

$$m = m - L D_m$$

$$c = c - L D_c$$

$m \rightarrow$ slope

$c \rightarrow$ intercept

$L \rightarrow$ Learning Rate

$D_m \rightarrow$ Partial Derivative of loss function with respect to m

$D_c \rightarrow$ Partial Derivative of loss function with respect to c

How it's work?

$$D_m = \frac{\partial(\text{Cost function})}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - (mx + c))$$

$$\boxed{D_m = \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - Y_{pred})}$$

$$D_c = \frac{\partial(\text{Cost Funtion})}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - y_{pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=1}^{n}(y_i - (mx_i + c))^2\right) \quad \because (a+b)^2 \text{ \& } (a-b)^2$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=1}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=1}^{n}(y_i - (mx + c))$$

$$D_c \; \boxed{= \frac{-2}{n}\sum_{i=1}^{n}(y_i - y_{pred})}$$