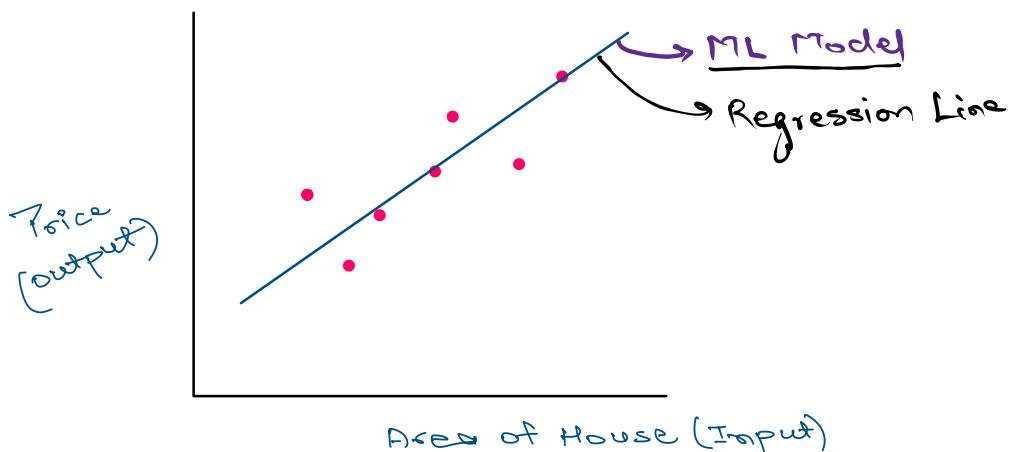
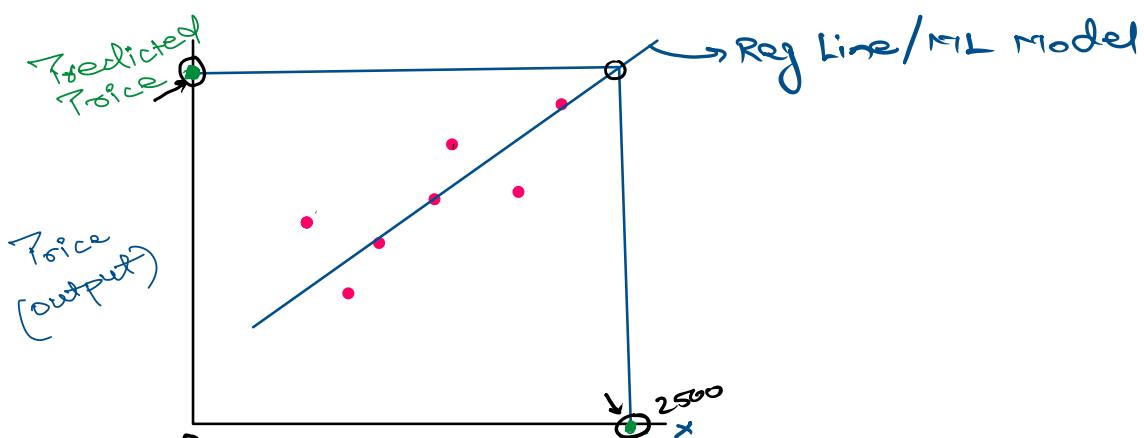


In Linear Regression Algorithms, we need to pass a line through the data such that it is closest to all the datapoints.

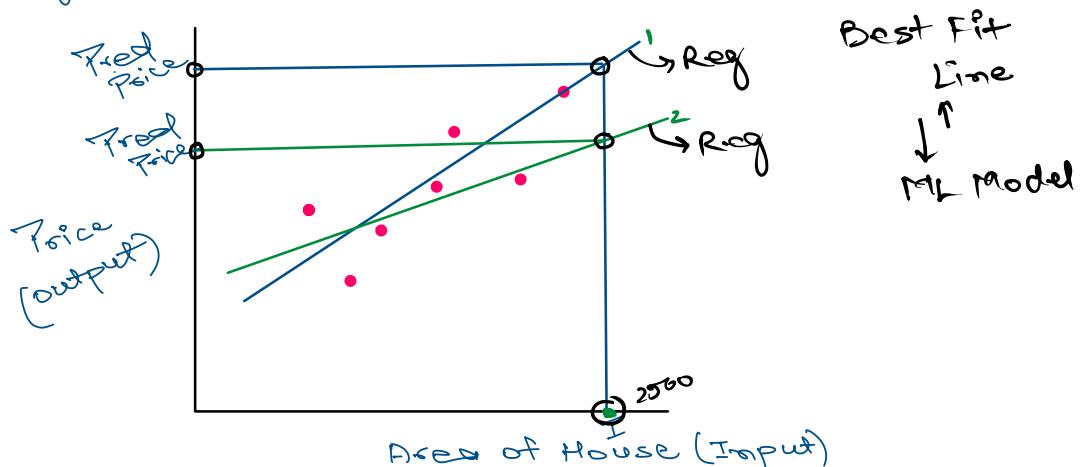


Once we have the regression line passing through the data, we can use the line to make predictions.



### Area of House (Input)

But there can be many such lines passing through the data and each line would give different prediction. Let's see an example using two such lines.

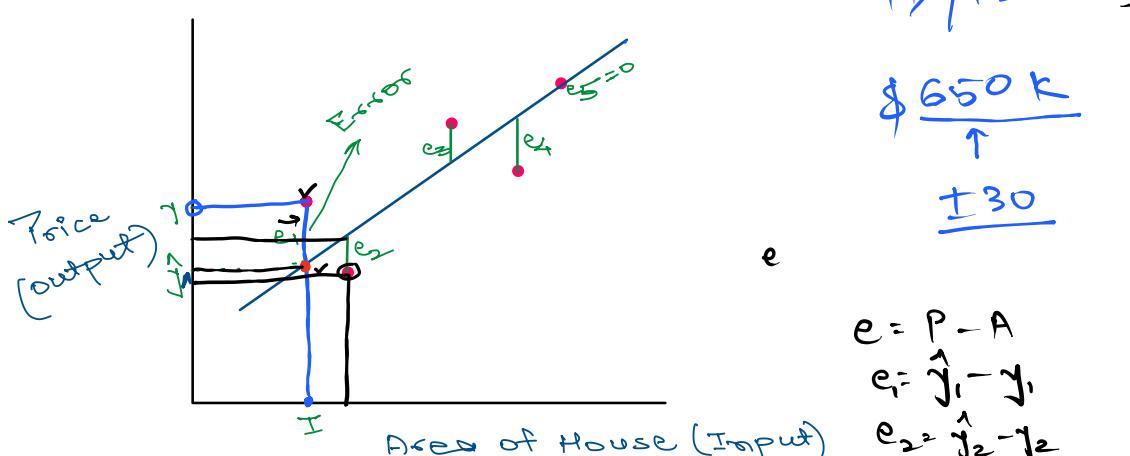


As we can observe above, each line gives different predicted output for the same input.

So which line to use then for final prediction?

The regression line we use is the one which is closest to the datapoint since that is the line which has learned the relationship b/w input & output with least errors. This line is called 'Best Fit Line'.

Now, what do we mean by an error?



Here,

$\hat{y}$  = Actual Price

$\hat{y}$  → no. 1st Price for the given input(I)

$$y = \text{predicted value}$$

$e_i$  = Errors in prediction.

$$\text{Error} = \text{Predicted} - \text{Actual}$$

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

TE

To get the total errors made by the line, we add all the errors:

$$\text{Total errors} = e_1 + e_2 + e_3 + e_4 + e_5$$

But here some errors would be positive and some negative and might cancel out each other, so to avoid this issue, we square the errors before adding them.

$$\text{Total Errors} = e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$$



We call it sum squared Error (SSE).

But there is one small problem. If there is a large training data, then SSE will be a very large value and can't be saved in the memory. So we take average of SSE.

$$\frac{1}{n} \times \underline{\text{SSE}}$$

where SSE can be written as,

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

Hence,

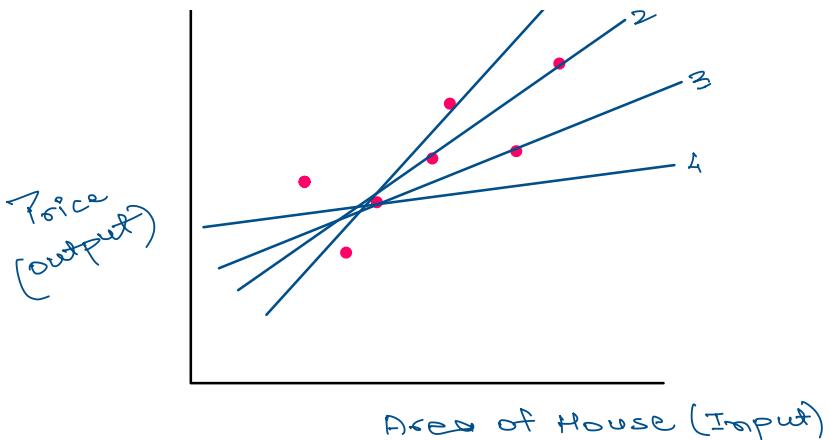
$$\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \rightarrow \text{Mean Square Errors.}$$

and finally,

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

This cost function is used to find the "Best Fit Line". How? Let's see →

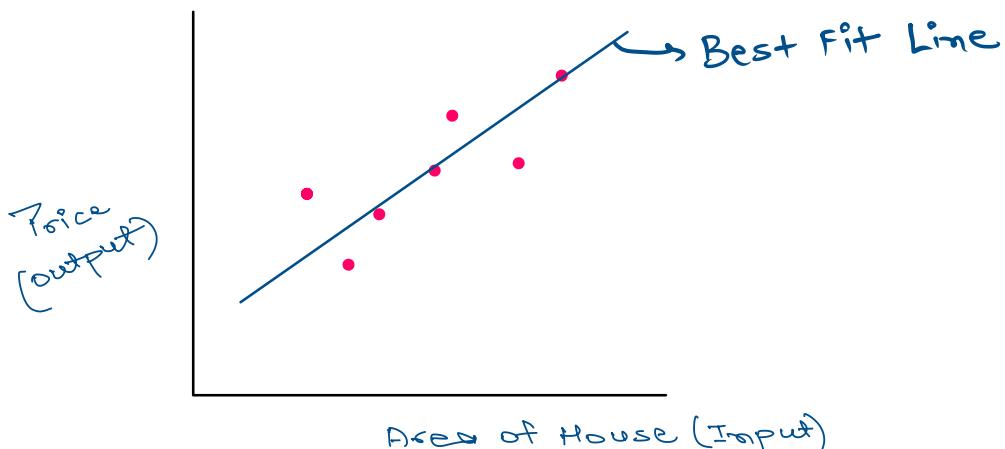
/ 1



To find the Best Fit Line, the algorithm draws many lines passing through the datapoints and then calculate "Cost Function for each line" which is nothing but the total error made by each line. Then it compares the Cost Functions →

$$CF(\text{line1}) \text{ vs } CF(\text{line2}) \text{ vs } CF(\text{line3}) \text{ vs } \dots$$

Now, whichever line has least cost function will be our "Best Fit Line".



This line can be represented as an equation which is in this form →

$$y = mx + c$$

that we write as :

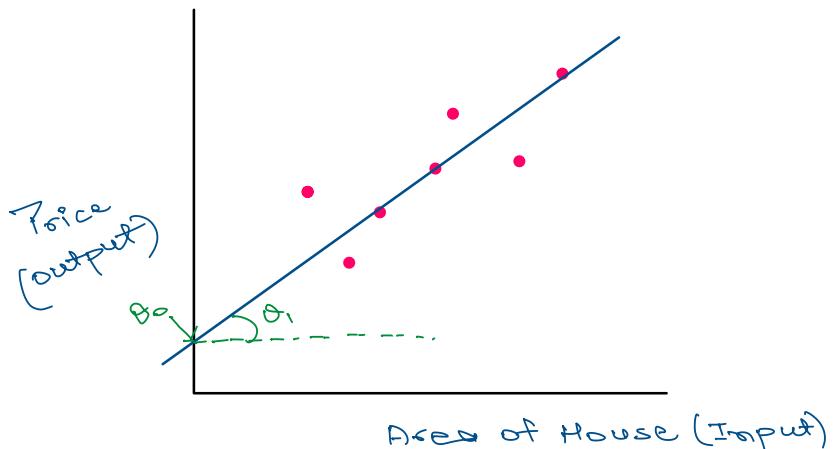
$$\hat{y} = \theta_0 + \theta_1 x$$

and in final form it will look like →

$$\text{Predicted } (\hat{y}) = 1.8 + 0.9x \quad (\text{suppose})$$

Price  $\rightarrow$   $y = \theta_0 + \theta_1 x$

where the value of  $\theta_0$  and  $\theta_1$  will be taken from the best fit line and  $x$  is the future input.



In this process, the algorithm has to try 100s of combination of  $\theta_0 + \theta_1$  to find the ones that minimize the cost function.

### Multiple Linear Regression $\rightarrow$

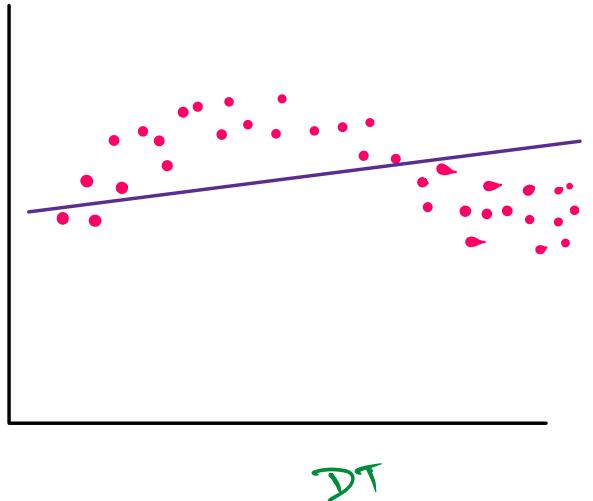
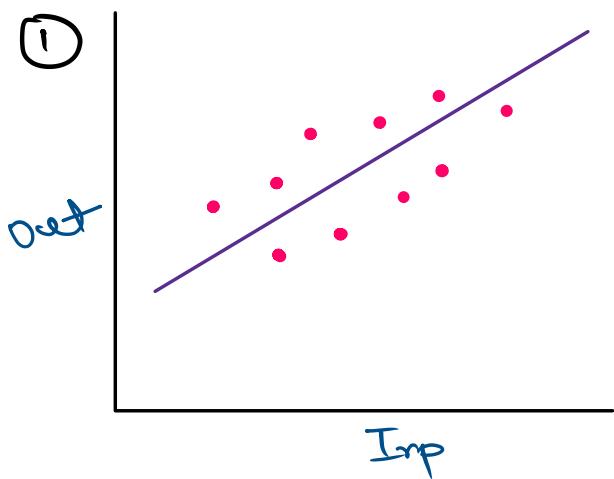
$x_1$ No of bedrooms	$x_2$ Age	$x_3$ No of floors	$x_4$ Area of House	Output col ↓ Price

$$\hat{y} = m_1 x_1 + m_2 x_2 + m_3 x_3 + m_4 x_4 + c$$

$$= 1.1 \times \text{No of Bedrooms} + (-1.26) \times \text{Age of House} + 1.8 \times \text{No of floors} +$$

$$2.1 \times \text{Area of House} + 1.19$$

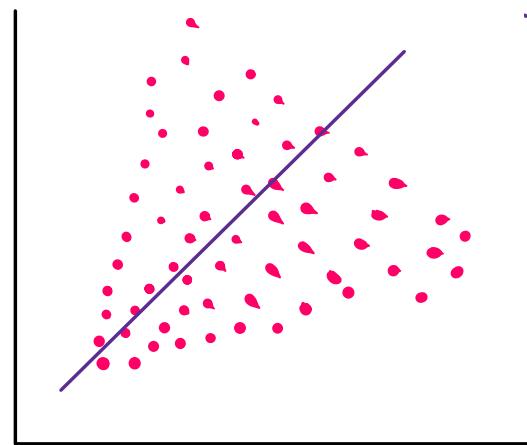
## Assumptions →



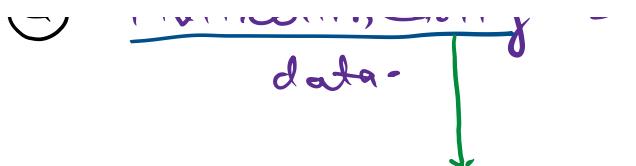
②  $e_1 + e_2 + e_3 + e_4 + \dots$

↓  
zero or close zero

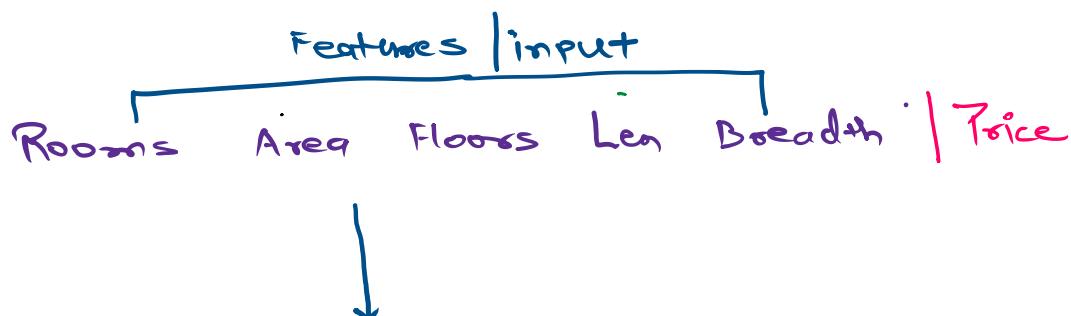
Then you can use Linear Regression on this data.



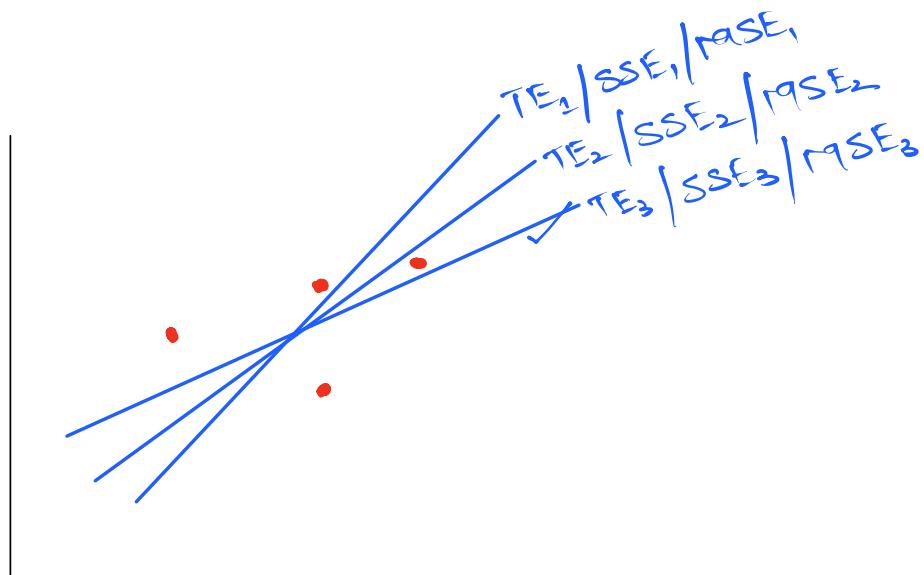
④ 'Multicollinearity' should not be present in the



When input cols are highly correlated to each other that is called as multicollinearity.



VIF (variance Inflation Factor) helps in identifying and removing multicollinearity.



$TE_1$  vs  $TE_2$  vs  $TE_3$

modulus

$$|-10| = 10$$

$$TE = e_1 + e_2 + e_3 + e_4 + e_5$$

↓

$$(-10)^2 = 100$$

1 million

$$\text{SSE} = [e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2]$$

↓  
Sum of squared errors

↓

↓  
 $10^7$

11487652197328

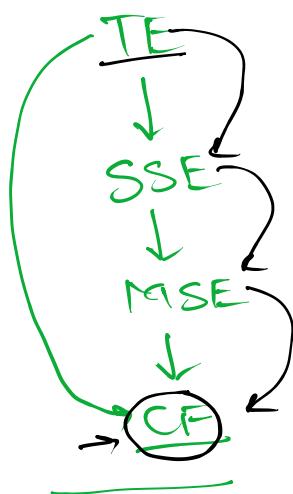
$$MSE = \frac{SSE}{n}$$

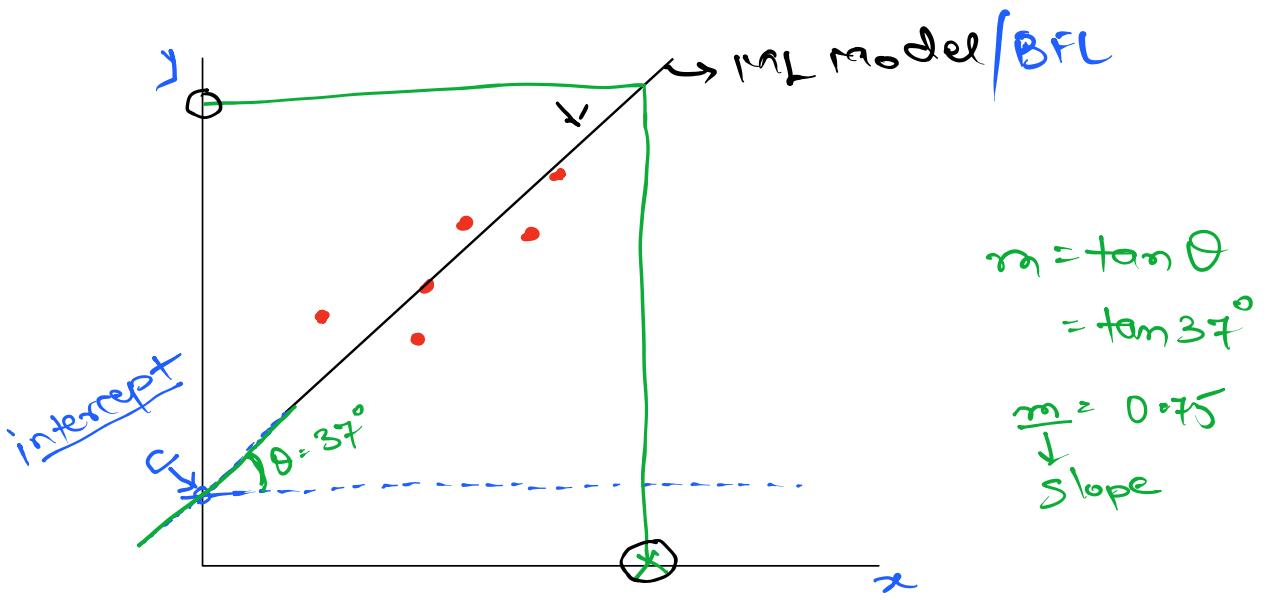
Mean  
squared  
Error

$$CF = \frac{MSE}{2} \quad \text{or} \quad \frac{SSE}{2n}$$

Cost function

↓  
↓  
⇒ Gradient Descent



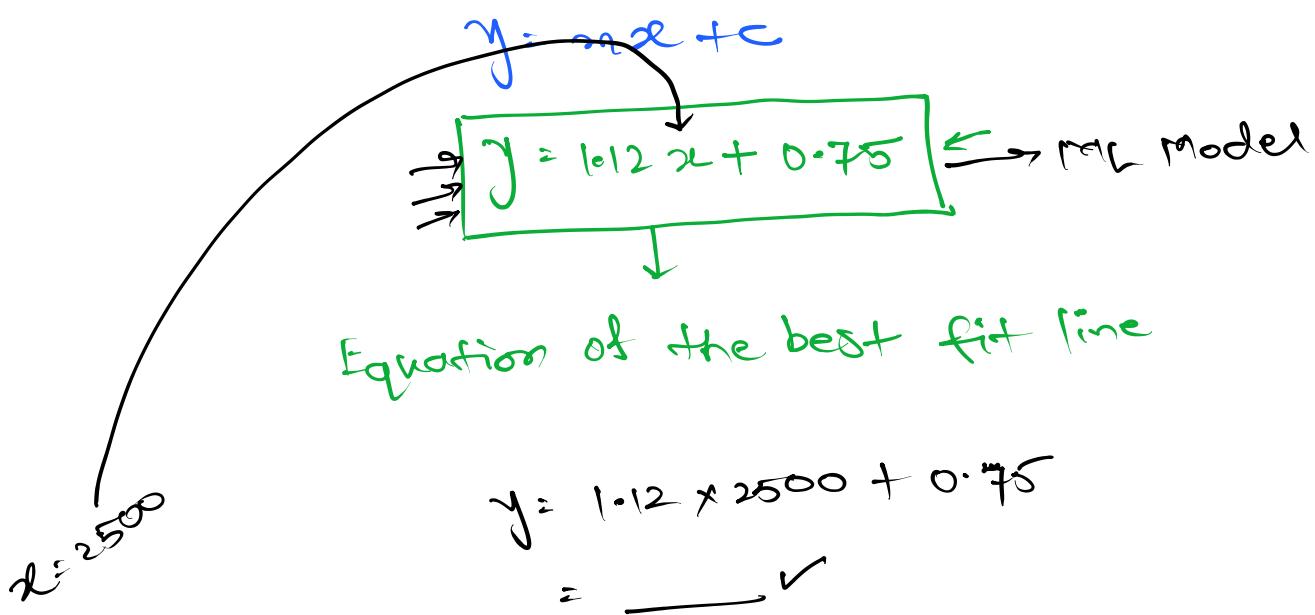


Equation of the best fit line

$$y = mx + c$$

$y = \text{Price (Pred)}$

$x = \text{Area}$



?

1  
 2  
 3  
 ...  
 999  
 1000  
1001 → Price

