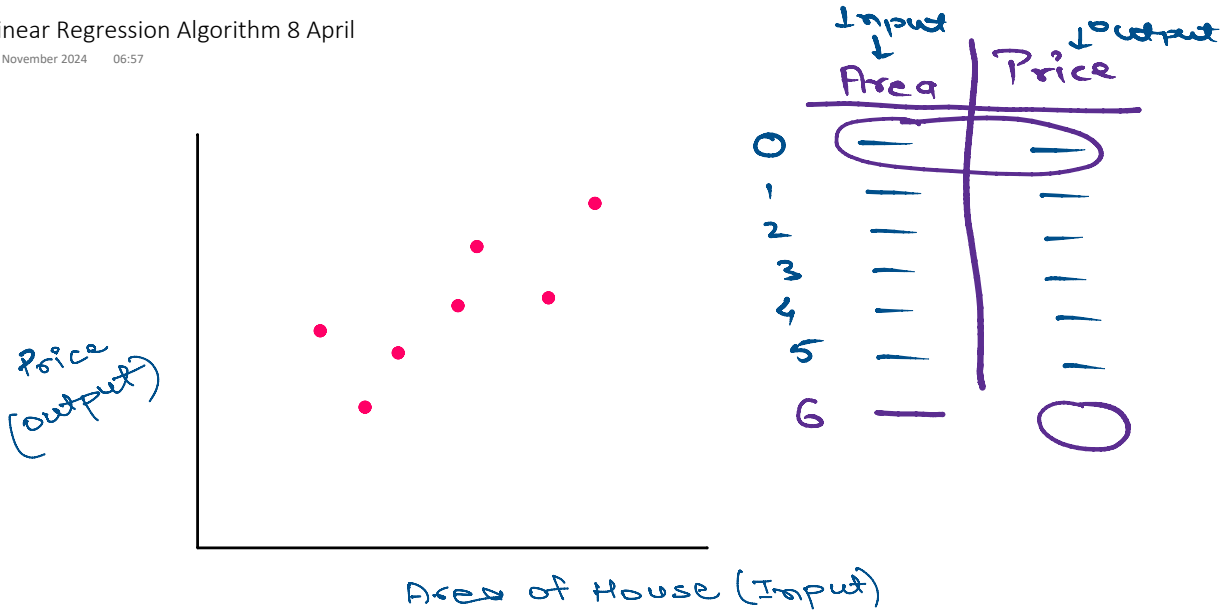
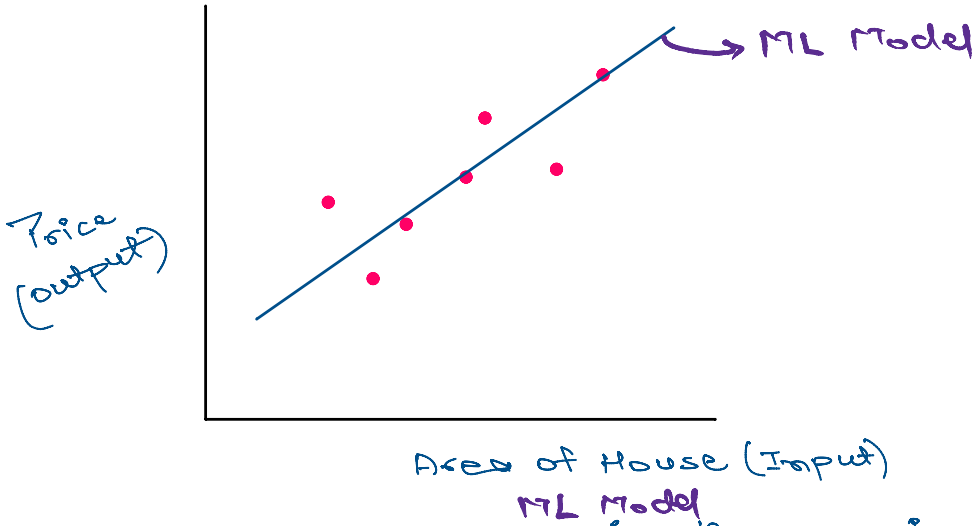
Linear Regression Algorithm 8 April

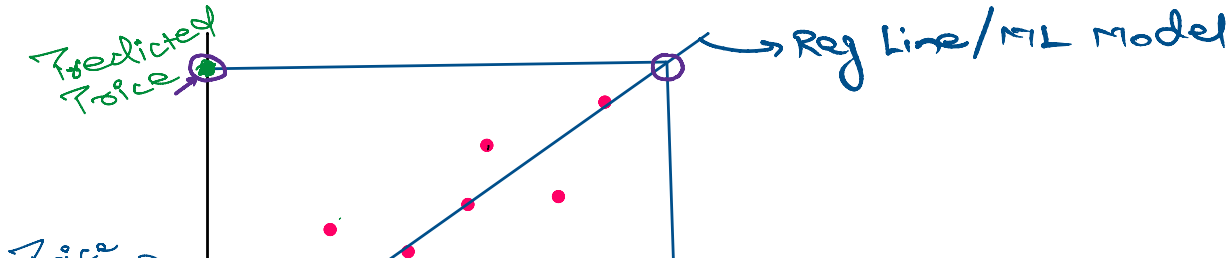
21 November 2024 06:57

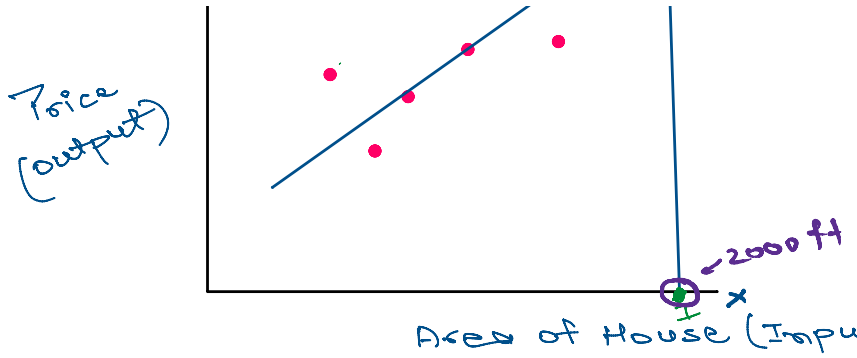


In Linear Regression Algorithm, we need to pass a line through the data such that it is closest to all the datapoints.

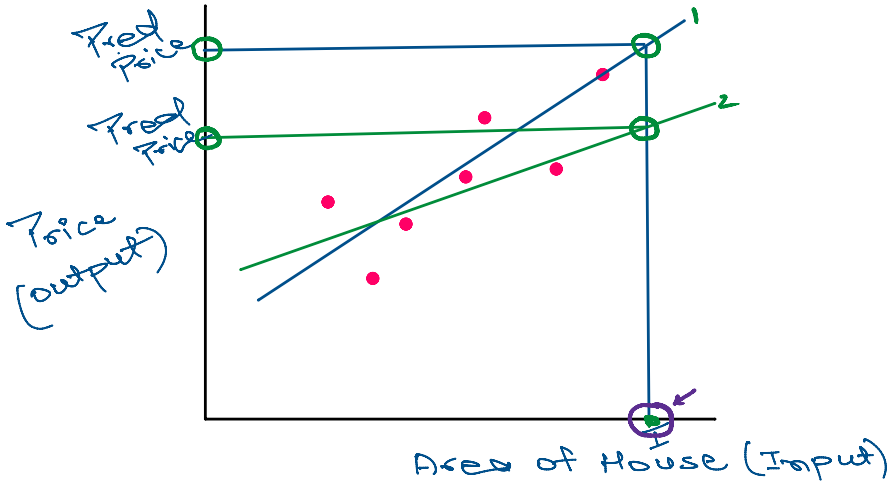


Once we have the regression line passing through the data, we can use the line to make predictions.





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.

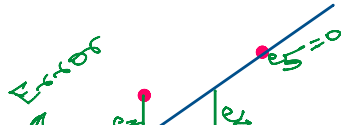
So which line to use then for final prediction?

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?

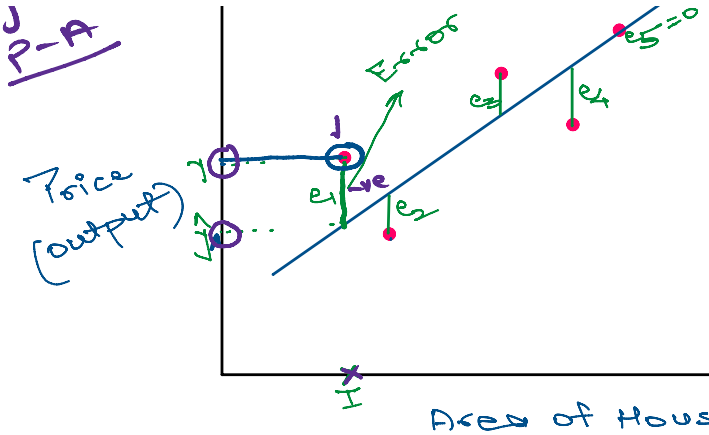
$\hat{y} - y$

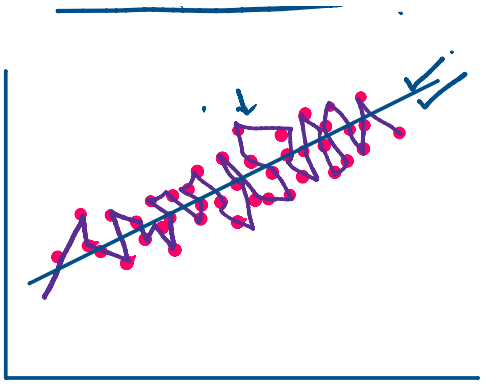
P - A



Total Error

J
P-A





Here,

y = Actual Price

$\hat{y}$ = Predicted Price for the given input(I)

e_i = Error in prediction.

Error = Predicted − Actual

$e = \hat{y} - y$

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

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.

Total Error = $e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$

↓

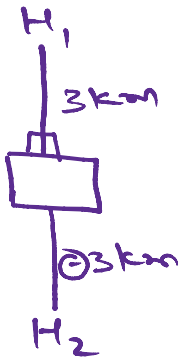
We call it sum squared Error (SSE).

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.

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

where SSE can be written as,

$\hat{y}$ (J)



Mod

$e_1 = -3$

$|e_1| = |-3|$

$= 3$

$(-3)^2 = -3 \times -3$

$= +9$

5 million

5 mil error.

10^6

where SSE can be written as,

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

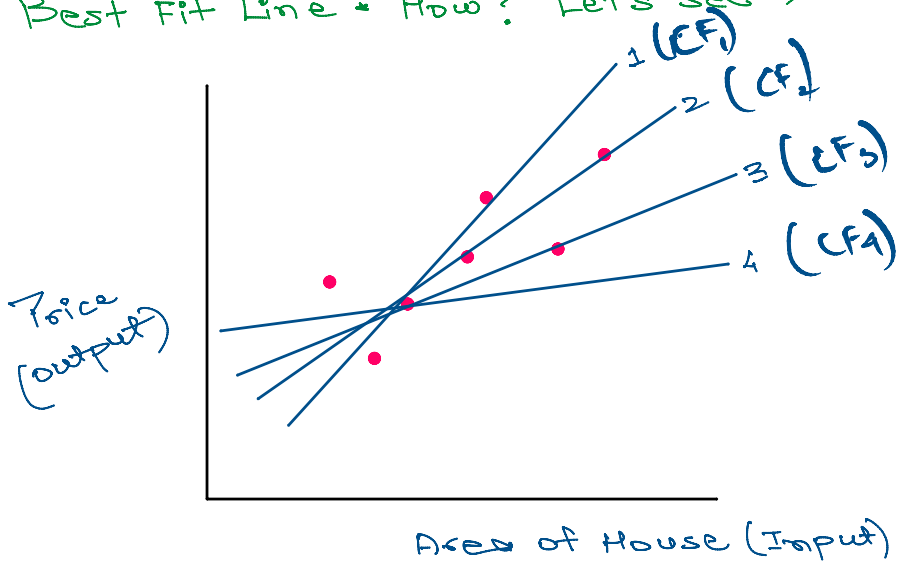
Hence,

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

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 →

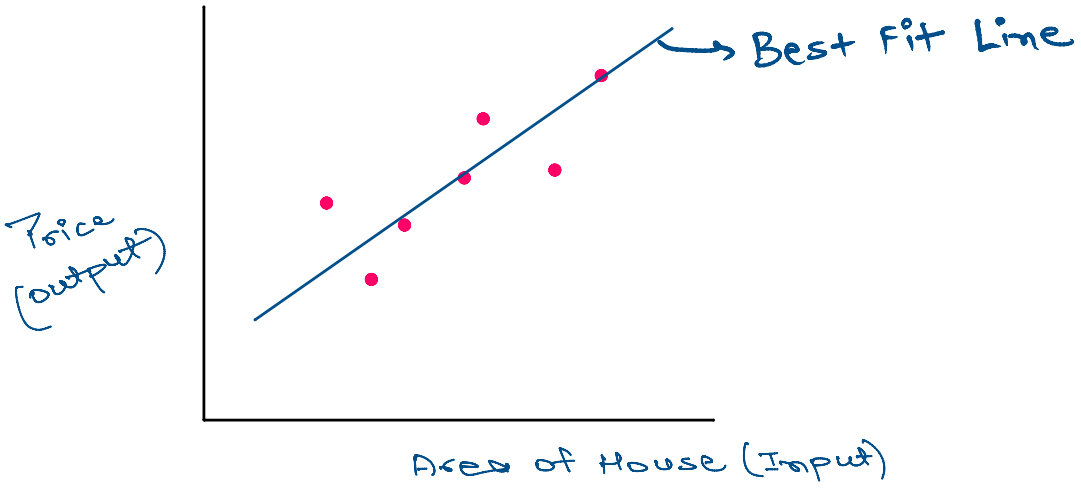


To find the Best Fit Line, the algorithm draws many lines passing through the datapoints and then calculate "Cost Function for each line" which is nothing but the total error made by each line. Then it compares the Cost Functions →

CF(line1) vs CF(line2) vs CF(line3) vs .....

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

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 →

$$y = mx + c$$

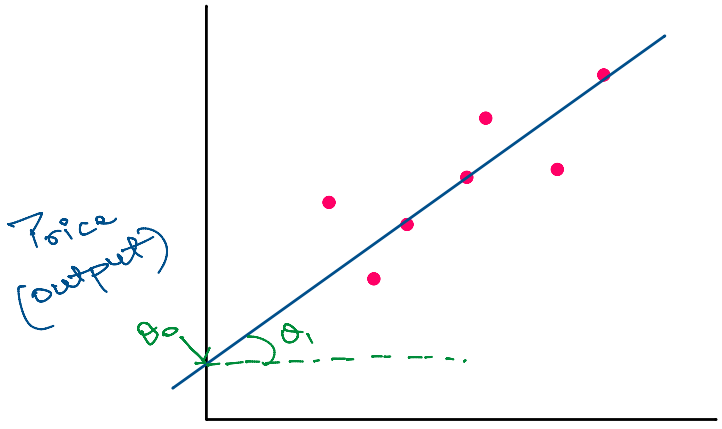
that we write as:

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

and in final form it will look like →

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

where the value of θ_0 and θ_1 will be taken from the best fit line and x is the future input.



Area of House (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 →

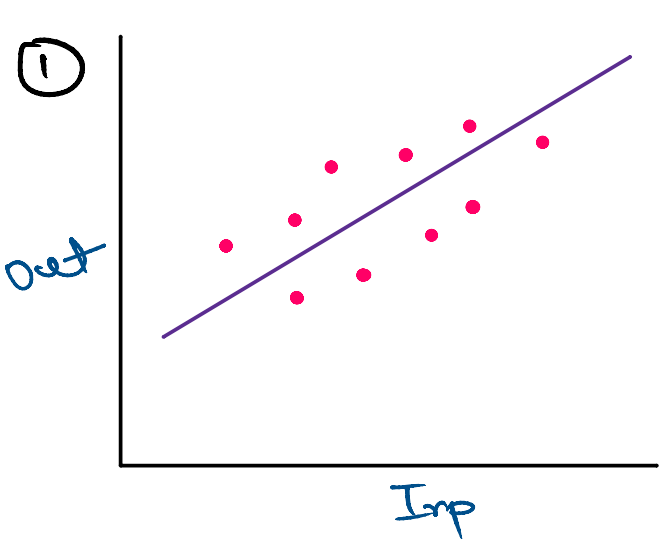
x_1 No of bedrooms	x_2 Age	x_3 No of floors	x_4 Area of House	Output col ↓ Price

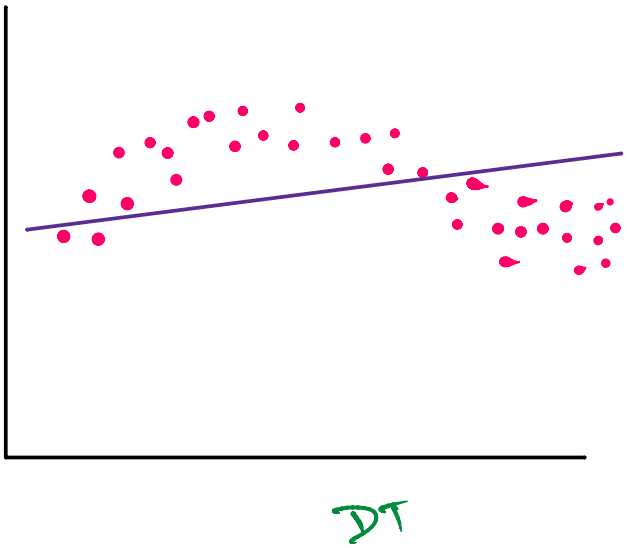
$$\hat{y} = m_1 x_1 + m_2 x_2 + m_3 x_3 + m_4 x_4 + C$$

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

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

Assumptions →

① 

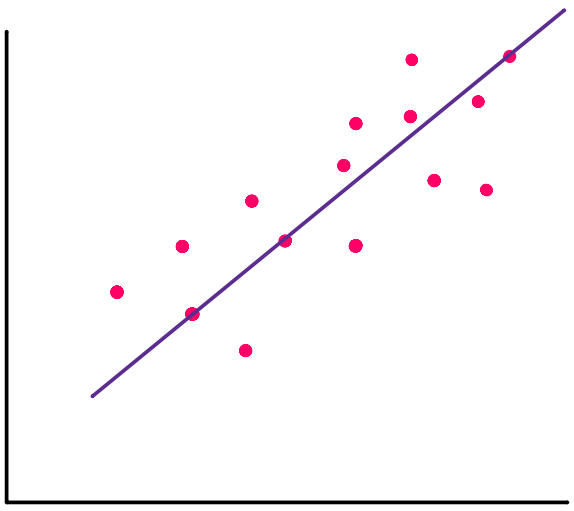


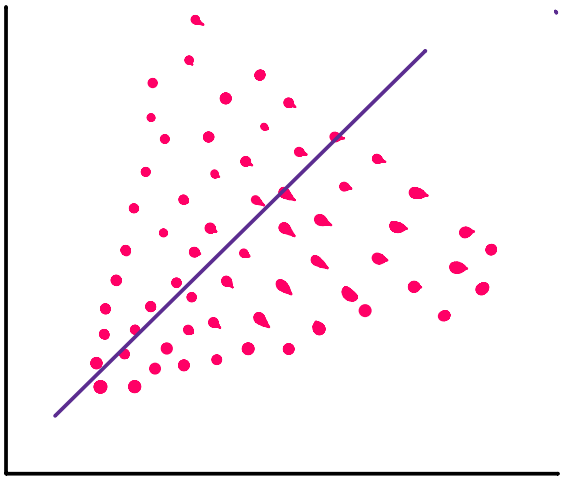
② $e_1 + e_2 + e_3 + e_4 + \cdots\cdots$

↓

zero or close zero

Then you can use Linear Regression on this data.

③ 



④ 'Multicollinearity' should not be present in the data.

data.

↓

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

Features | input

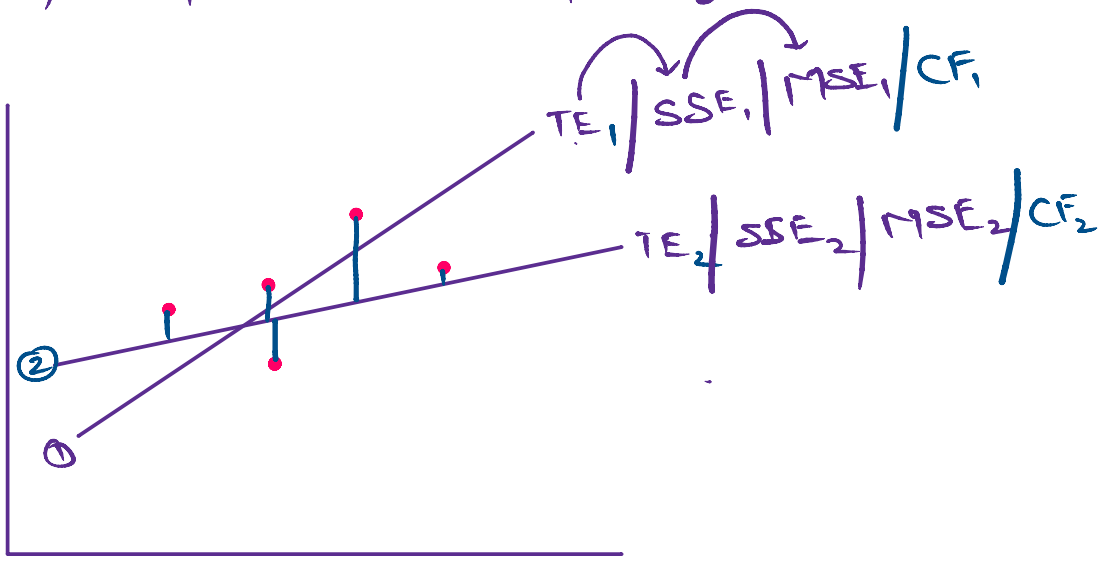
Rooms	Area	Floors	Len	Breadth	Price

↓

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

Total Error = $e_1 + e_2 + e_3 + e_4 + e_5$

SSE(TE) = $e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$



2 lines

TE_1 vs TE_2

↓

least

↓

This line which has least total error is our Best Fit Line.

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

TE

↓

SSE

↓

MSE

$$MSE = \frac{1}{n} \times SSE$$

↓

Cost Function

$$CF = \frac{MSE}{2} \quad \text{or} \quad \frac{1}{2n} \times SSE$$

Gradient Descent

CF_1 vs CF_2 vs CF_3 ------ CF_n

↓

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

$$= (\hat{y}_1 - y_1) + (\hat{y}_2 - y_2) + (\hat{y}_3 - y_3) + \dots$$

$$= \sum_{i=1}^{5} (\hat{y}_i - y_i)$$

$$SSE = \sum_{i=1}^{5} (\hat{y}_i - y_i)^2$$

↓

$$MSE = \frac{1}{n} \times \sum_{i=1}^{5} (\hat{y}_i - y_i)^2$$

↓

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

$\sum_{i=1}^{n}$