Machine Learning & DATA SCIENCE
DAY-2

AGENDA

1. Revision of DAY 1
2. COST FUNCTION
3. LOSS FUNCTION
4. PERFORMANE METRIX
5. UNDER FITTING & OVER FITTING

* Linear Regression Algorithum
→ Simple Linear Regression

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

→ Multi Linear Regression
Equation for Best fit line

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

n = no. of indipendent features

COST FUNC VS LOSS FUNC

Cost function is calculated for entire dataset, where loss function handle to loss or error at individual Data Point.

COST FUNCTION

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^i - (y)^i\right)^2$$

$h_\theta x$ ⇒ Predicted value

y = Actual value

m = No. of Data Point

LOSS FUNCTION

$$= \left(h_\theta(x)^i - (y)^i\right)^2$$

Let's Calculate Partial derivative of cost func'
$J(\theta_0, \theta_1)$.

At $J = 0$

$$\frac{\partial}{\partial \theta_0}\left(J(\theta_0, \theta_1)\right) = \frac{\partial}{\partial \theta_0}\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta x^{(i)} - y^{(i)}\right)^2\right]$$

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

$$= \frac{\partial}{\partial \theta_0}\left[\frac{1}{m}\sum_{i=1}^{m}\left\{(\theta_0+\theta_1 x)^i - y^i\right\}^2\right]$$

$$= \frac{2}{m}\sum_{i=1}^{m}\left[(\theta_0+\theta_1 x)^i - y^i\right] \times \{1\}$$

$$\frac{\partial}{\partial \theta_0} J(\theta_1) = \frac{2}{m}\sum_{i=1}^{m}\left[(\theta_0+\theta_1 x)^i - y^i\right] \quad -①$$

at $J = 1$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_1}\left[\frac{1}{m}\sum_{i=1}^{m}\left\{(\theta_0+\theta_1 x)^i - y^i\right\}^2\right]$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{2}{m}\sum_{i=1}^{m}\left[(\theta_0+\theta_1 x)^i - y^i\right] \times \{x\} \quad -②$$

Replacing $\theta_0 + \theta_1 x = h_\theta(x)$ in Eqn ① & ②

Repeat unit convergence

$$\theta_0 = \theta_0 - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^i - y^i \right) \right]$$

$$\theta_1 = \theta_1 - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^i - y^i \right) \times x \right]$$

Note :-> here Learning Rate α Controls the speed of convergence

COST FUNCTION

1. Mean Square error (MSE)

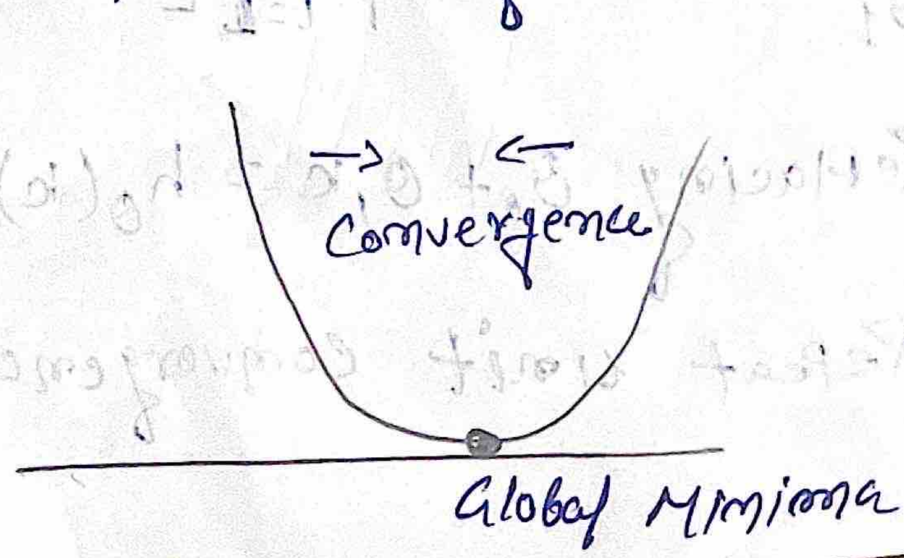
⇒ MSE Measured the average of Square of the error that is; the Average Square difference between the estimated value & the Actual value.

$$\boxed{MSE = \frac{1}{m} \sum_{i=1}^{m} (y - \hat{y})^2}$$

This is Quadratic equation

$\hat{y}$ = Predicted value $(\theta_0 + \theta_1 x)$

The Above eqn has only one Global Minima



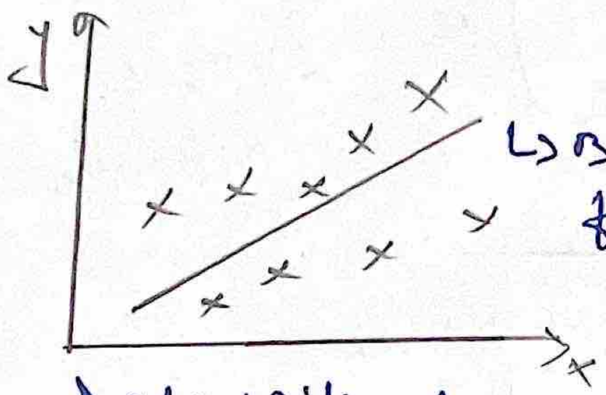
Advantages

→ The MSE is differentiable.

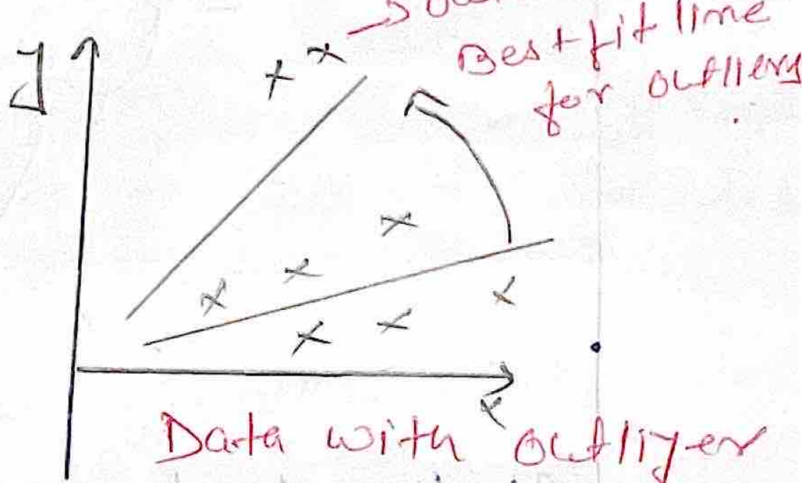
→ The MSE eqn has only one Global Minima

Disadvantages

→ This equation is not robust to outliers, i.e it doesnot handle dataset outliers.



Data without outlier



Data with outlier

Conclusion

Addition of outlier will increase the cost function, but our main aim to reduce the cost function to reach the Global Minima.

- The unit of dependent feature & error or residual is Different.

Eg:- Dependent feature – lakh

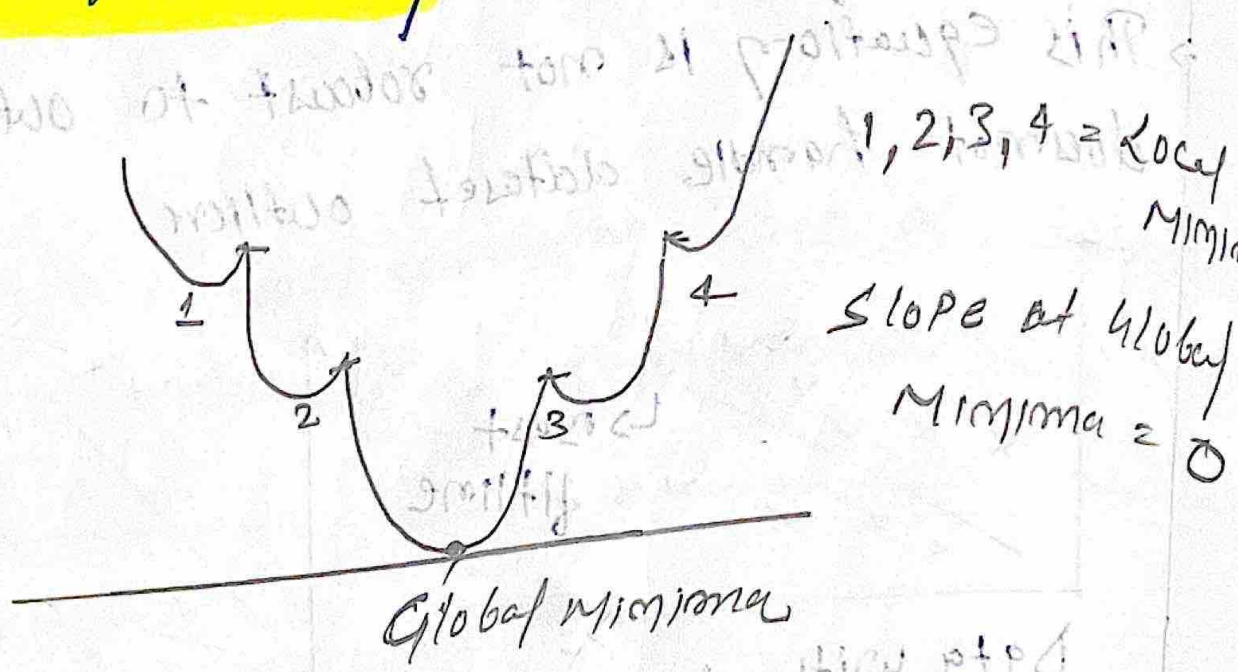
Independent feature – $(lakh)^2$

ERROR = True – Predicted $= (100-110)^2 = 100$

Result is equal to original value

* MSE is not recommended when Dataset contains outliers

NON-CONVEX FUNCTION



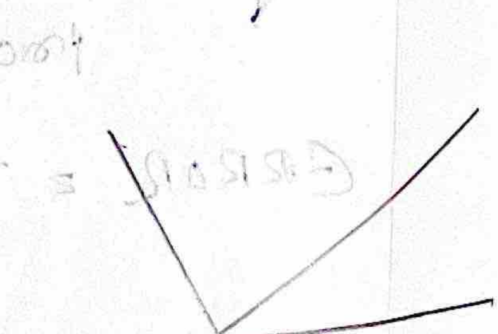
At the Local Minima, convergence algorithm stuck for infinite time

* Gradient Descent convergence Algorithm is Best to have cost function which has convex type Graph & Single Global Minima

2) Mean Absolute Error

Mean absolute error is a model evaluation matrix used with regression Model

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



This is not Differentiable

Advantages

* Error not penalising cost function
* Error unit will be same as that of dependent feature.

Disadvantages

* Optimization is complex task i.e. Convergence is time consuming.
* It takes more time to reach Global Minima
* Since cost function graph is not differentiable Sub Gradient method is used to calculate Global minima

3. Root mean Square Error

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (y-\hat{y})^2}$$

Advantages :-

* It is differentiable.
* Unit of error & dependent variable is Sam

Disadvantages

This eqn is not robust to outliers.

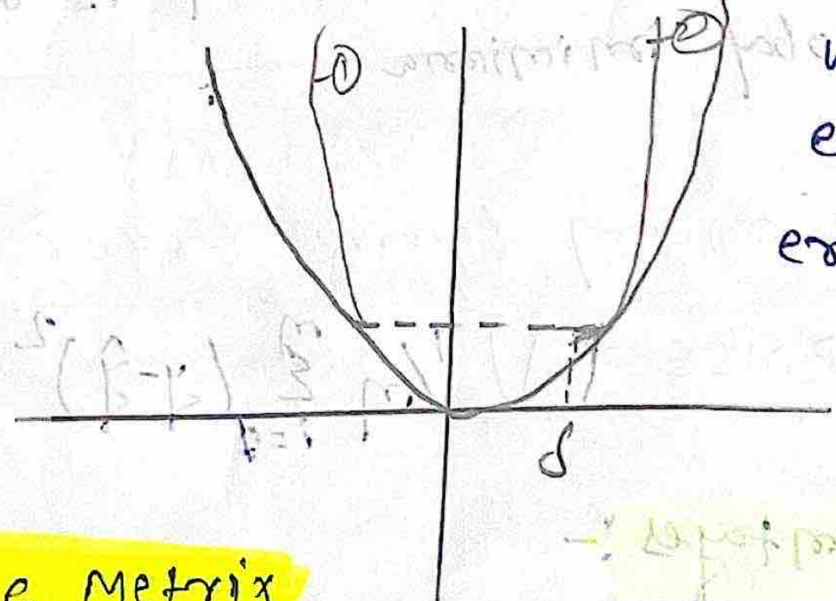
4) Huber loss

⇒ function used in robust Regression, that is less sensitive to outliers in data than squared error loss.

$$\text{Huber loss} = \frac{1}{2}(y-\hat{y})^2 \quad \text{—①} \quad y_i-\hat{y} \leq \delta$$

$$\delta\,|(y-\hat{y})| - \frac{1}{2}\delta^2 \quad \text{—②} \quad y_i-\hat{y} > \delta$$

δ - Point at which the st line & Parabola meets.



When error $\leq \delta$, use MSE
error $> \delta$, use MAE

Performance Metrix

⇒ It is used to evaluate the Performance or Quality of Model

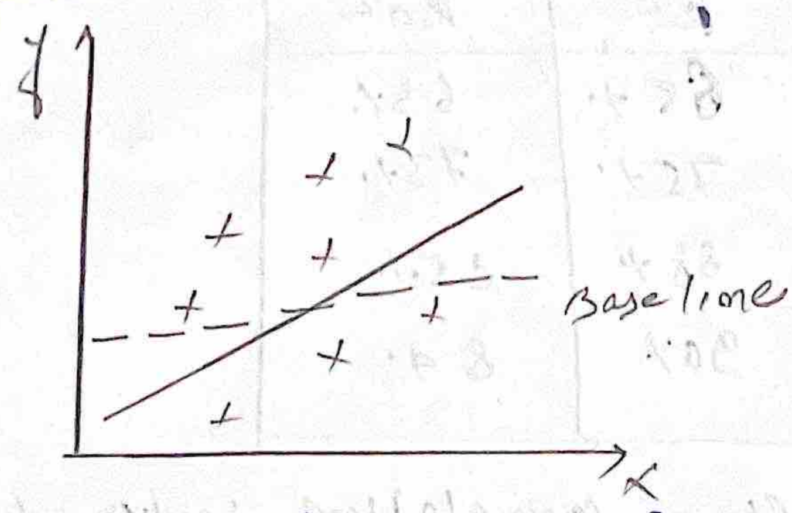
1. R Square Score

→ It enables us to compare our Model $\hat{y}$ with a Constant Base Line ($\bar{y}$) i.e average or mean to determine the Performance of the ~~Matrix~~. Model

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}}$$

SS_{Res} = Sum of Square error or residuals

SS_{Total} = Sum of Square of Average



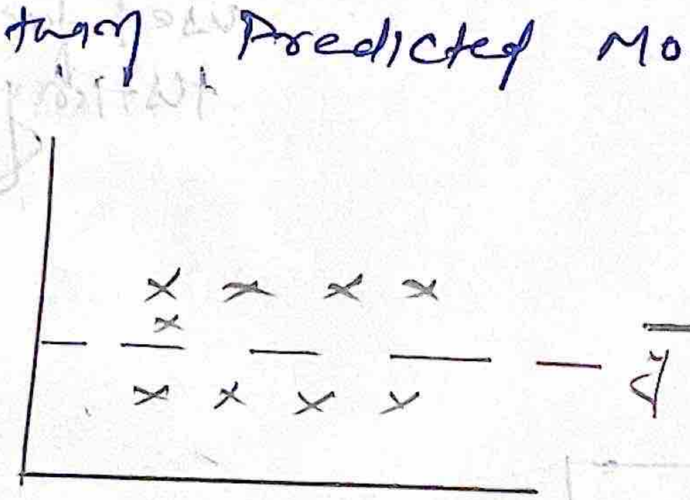
$$R^2 = 1 - \frac{\sum_{i=1}^{m} (y_i - \hat{y})^2}{\sum_{i=1}^{m} (y_i - \bar{y})^2}$$

$\bar{y}$ = Average or mean

R^2 always lies b/w 0 & 1

Eg - $R^2 = 0.85$, Model is Accurate

If R^2 is very low, this means our Model Performance is very low or Bad. In this case the Baseline Model Perform better than Predicted Model.



Base Model Performance will be better than Predicted model

(2) ADJUSTED R Square Score

It is Modified version of R-square, & it is Adjusted for the no. of independent variable in the Model & it will alway less than R^2

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

Eg.

No. of independent feature	R^2	Ra^2
1	65%	63%
2	75%	73%
3	88%	85%
4	90%	84%

Addition of 1 show correlation independent feat.

Overfitting & underfitting (Bias & variance)

Eg:

1000 DATA Points

- 700 Training Dataset — used for Model training
- 300 Test Dataset — used for Model testing

700

- 500 — To train Model
- 200 — Hyper Parameter tuning the Model

Note :- For Train Data Accuracy — Bias
For test Data Accuracy — variance

Model1-

Train	Very Good Accuracy	Low Bias
Test	Very Good Accuracy	Low variance

Generalised Model (Good Model)

Model 2

Train	very Good Accuracy	Low Bias
Test	very Bad Accuracy	High variance

very Bad Model

Model 3

Train	Very Good Accuracy	Low Bias
Test	Bad Accuracy	High variance

Overfitting

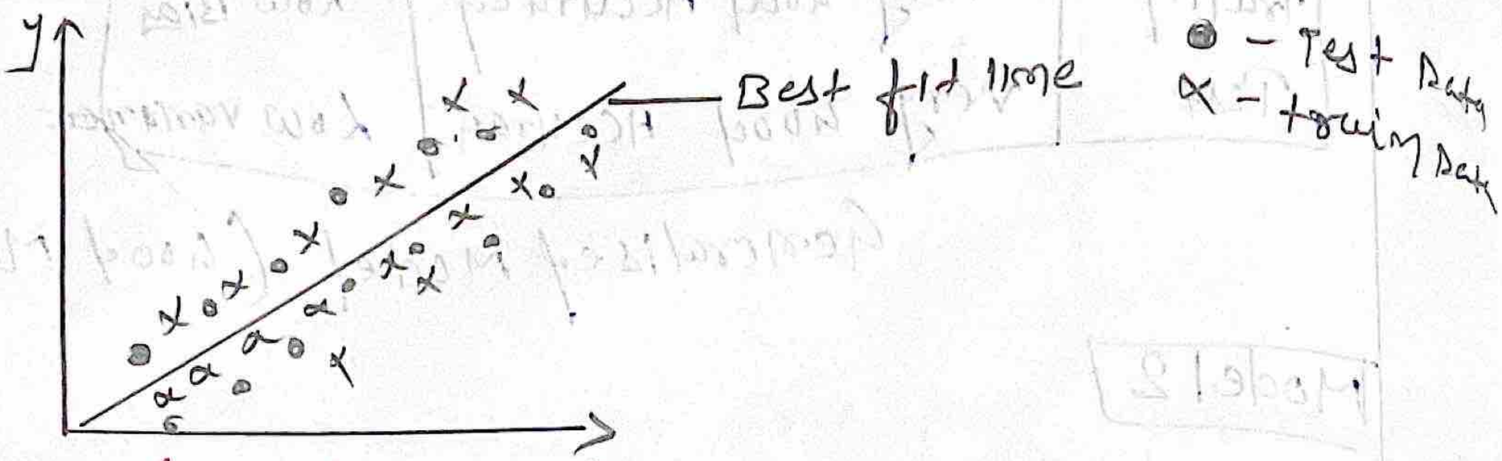
→ For such type of Models, where training Data is very Good & test Data is Bad, this Scenario is called overfitting.

Model 4

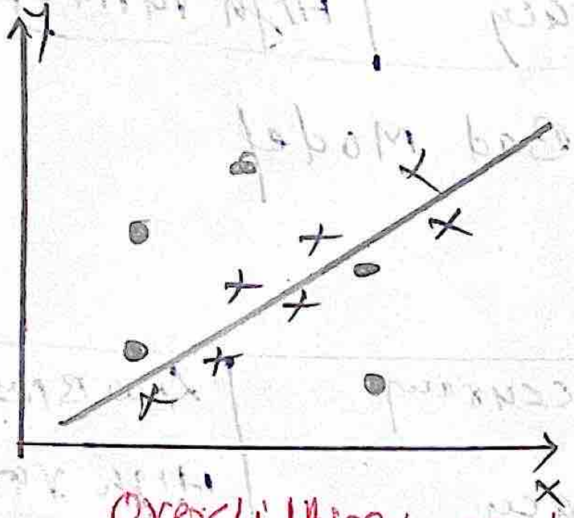
Train	Low Accuracy	High Bias
Test	Low Accuracy	Low/High variance

UNDERFITTING

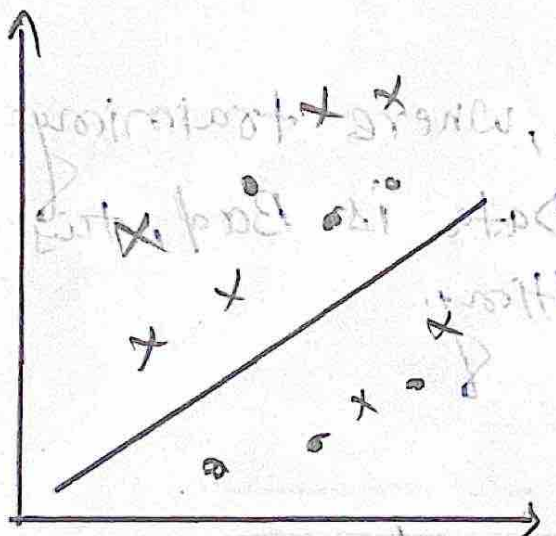
Graphical Representation of Models.

① 

Generalized Model

② 

Overfitting Model

③ 

Underfitting Model