Linear Regression (continued)

* Equation of Best fit line for n independent features:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots\dots + \theta_n x_n$$

where n = No. of independent features.

* cost function V/s loss function.

for mean squared error (MSE)

* cost function is calculated for entire dataset whereas loss function refers to loss or error at individual data point.

* cost function, $J(\theta_0, \theta_1)$

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

$$\boxed{\text{loss function} = \left(h_\theta x^{(i)} - y^{(i)}\right)^2}$$

where $h_\theta(x)$ = predicted value.

$y^{(i)}$ = actual/ Truth value.

m = No. of data points/observations in dataset.

Now lets us calculate partial derivative of cost function $J(\theta_0, \theta_1)$ wrt θ_0 and θ_1.

at $j = 0$

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

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

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

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

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

at $J = 1$

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

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

Replacing $\theta_0 + \theta_1 x = h_\theta(x)$ in equation 1 and 2.

we get new convergence algorithm eqn as below:

Repeat until 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 (rate) of convergence.

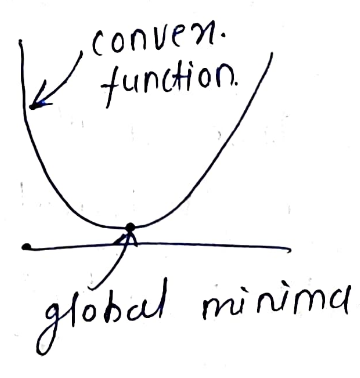
* <u>Cost functions</u>

(1) <u>Mean Square Error (MSE)</u>

$$\boxed{MSE = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y})^2}$$ ← This is a quadratic equation.

where $\hat{y} \equiv$ predicted value. $(\theta_0 + \theta_1 x)$.

The above equation has only one global minima. as shown.

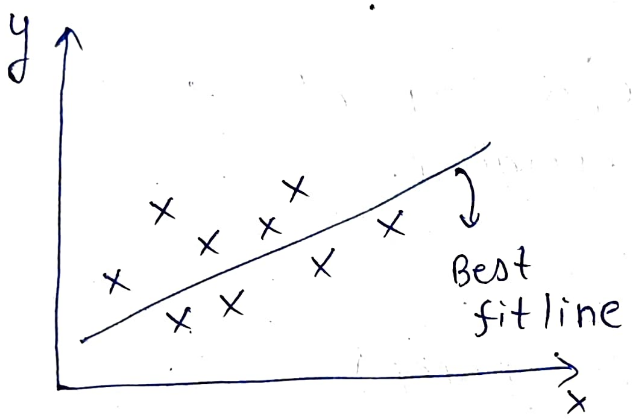


Advantages:

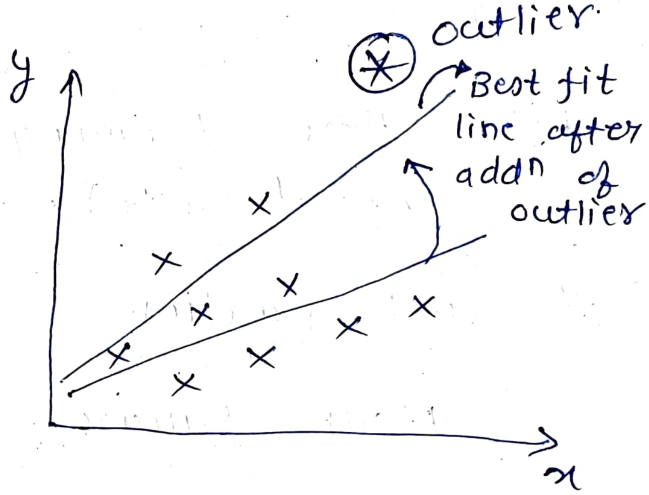
1) The MSE is differentiable.
2) The MSE Equation has only one global minima value.

Disadvantages

1) This equation is not robust to outliers, ie ~~the~~ it cannot handle dataset with outliers.



Data with out outliers.



* for same dataset when outlier is introduced.

conclusion: Addition of outlier will increase the cost function, but our aim is to reduce cost function to reach global minima.

2) The unit of Dependent feature and Error or residual is different.

Ex: Dependent feature : weight => kg.

Error will be kg^2.

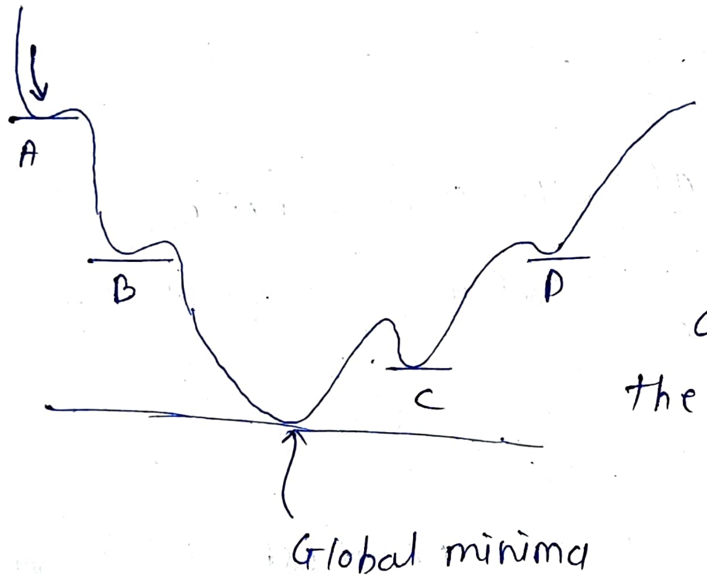
Error = $(\text{True} - \text{predicted})^2 = (100-110)^2 = 100$

Here error is equal to original value.

ie in this case error is penalising cost function.

So MSE is not recommended when data set contains outliers.

Note: Non-convex function



at point A, B, C, D there are local minima.

at local minima => slope = 0

So at this local minima our convergence Algorithm will be stuck for infinite time.

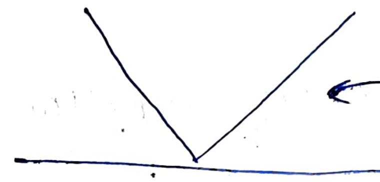
=> conclusion.

Gradient descent convergence algorithm it is best to have a cost function which has convex type graph and single global minima.

② Mean absolute Error (MAE)

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

↳ Graph of this function



← This is not differentiable.

① Advantage:

1) Error not penalising cost function
2) Error unit will be same as that of dependent feature.

2) Disadvantage.

1) optimization is a complex task ie convergence is time consuming
2) It takes more time to reach global minima.
3) Since cost function graph is not differentiable sub gradient method is used to calculate global minima.

③ Root Mean Square Error (RMSE)

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

* Advantages

1) It is differentiable
2) unit of Error and dependent variable is same.

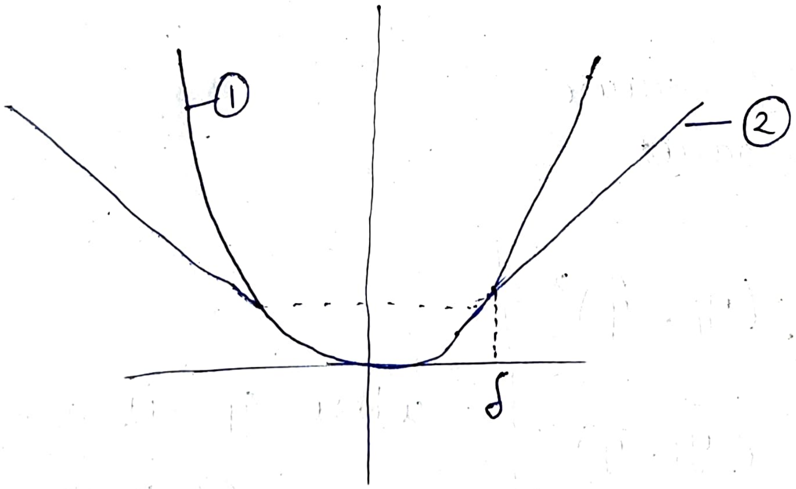
* Disadvantage.

1) This equation is not robust to outliers.

④ Huber loss.

$$\text{Huber loss} = \begin{cases} \frac{1}{2}(y-\hat{y})^2 \quad -① \quad \text{for } y_i - \hat{y} \le \delta \\ \delta |(y-\hat{y})| - \frac{1}{2}\delta^2 \quad -② \quad \text{for } y_i \hat{y} > \delta \end{cases}$$

where δ = point at which the st line and parabola meet.



when Error $\le \delta$ => MSE is used.

Error $> \delta$ => MAE is used.

It has advantages of both MSE and MAE.

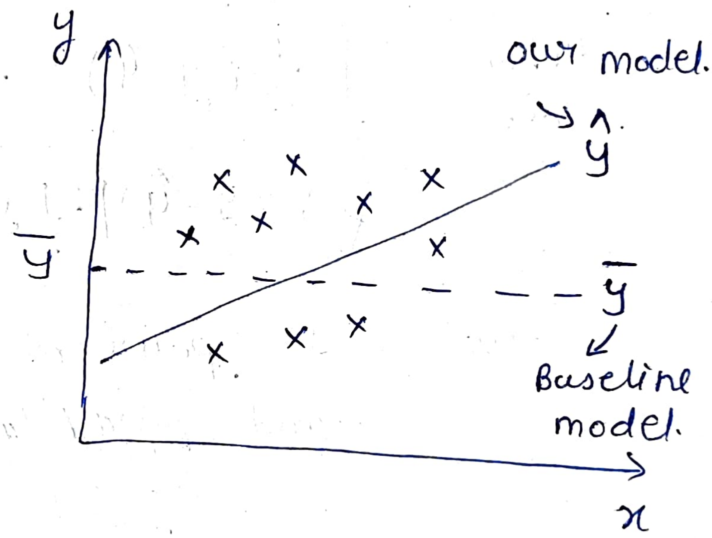
Performance matrix

It is used to evaluate the performance or quality of model.

① R squared Score

R squared score enables us to compare our model ($\hat{y}$) with a constant baseline model ($\bar{y}$) ie Average or mean to determine the performance of model.

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



where, SS_{Res} = sum of Square of Residuals or Error.

SS_{Total} = Sum of Square of averages.

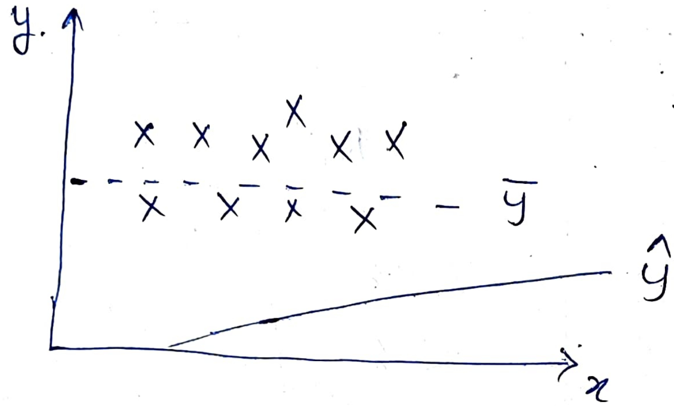
$$\boxed{R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

where $\bar{y}$ is average or mean.

R^2 lies b/w 0 to 1

Ex: if $R^2 = 0.85$ this means model is 85% accurate.

if R^2 value is very low this means our models performance is very bad. in this case our base line model performs better than predicted model.



Here performance of base model will be better than our predicted model.

② <u>Adjusted R square Score</u>

It is an improved version of R^2 score. As No. of independent features increases or an very low correlated Independent feature with dependent feature is taken into consideration, then the Result of R^2 can be misleading.

To overcome this, adjusted R^2 score will be used which will always show lower value than R^2. It shows lower value as compared to R^2 score because it adjusts the value of increasing predictors and only shows improvement if there is real improvement.

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

where $N \equiv$ No. of data points

$P \equiv$ No. of Independent feature.

Example

No. of Independent feature. (P)	R^2	R_a^2
1	65%	63%
2	75%	73%
3	88%	86%.
4	90%	84%

addition of 1 low correlation Independent feature

* overfitting and underfitting (Bias and variance).

for example say we have 1000 datapoints in our dataset.

1000 data points

used for model training → 700 Training dataset

300 Test dataset ← used for model testing

$\boxed{700}$ ← Training dataset.

To train model → $\boxed{500}$

$\boxed{200}$ ← for hyper-parameter tuning the model.

Note: for train data accuracy bias is used.
for test data accuracy Variance is used.

Model 1

Train	very good accuracy	low bias
Test	very good accuracy	low Variance

← Generalised model. (good model).

Model 2

Train	very bad accuracy	High bias
Test	very bad accuracy	High variance

← very bad model.

Model 3.

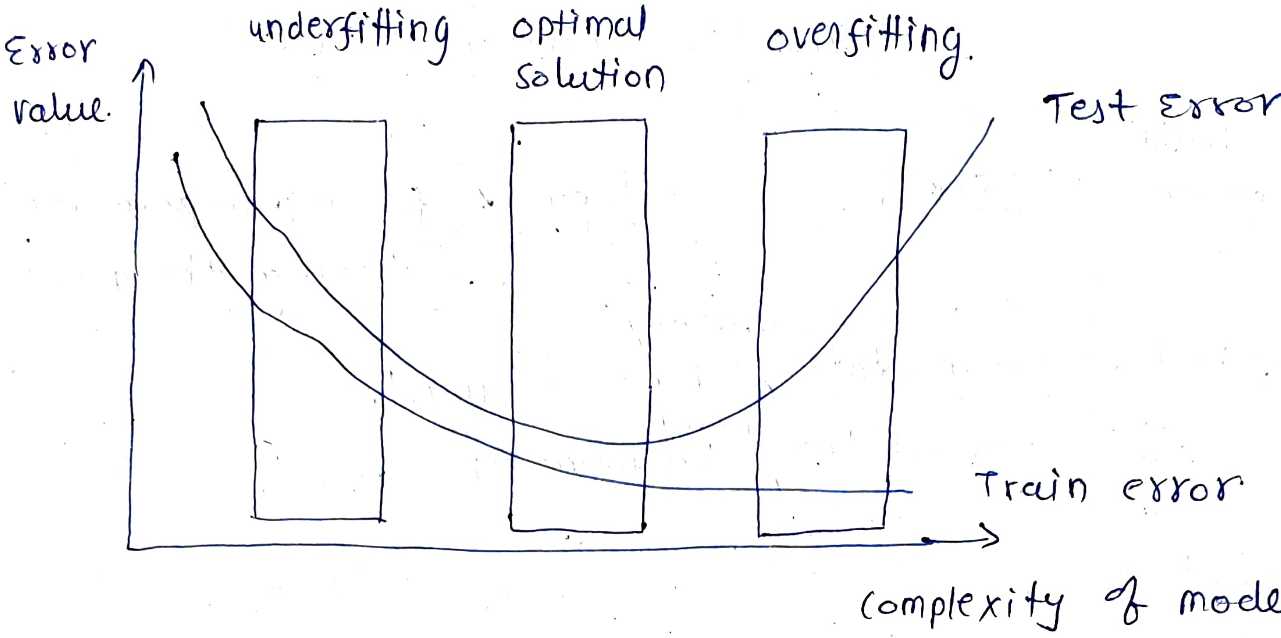
Train	very good accuracy	low bias
Test	bad accuracy	High varience

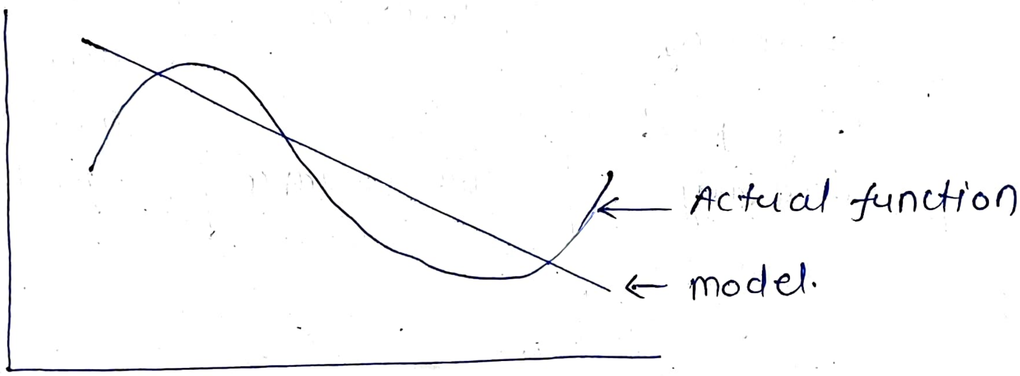
← overfitting.

model 4

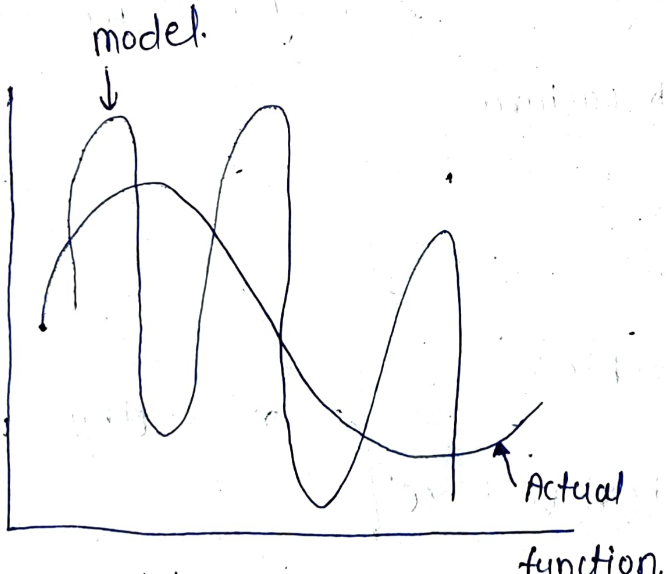
Train	low accuracy	High bias
Test	Low/high accuracy	low/High varience

← underfitting

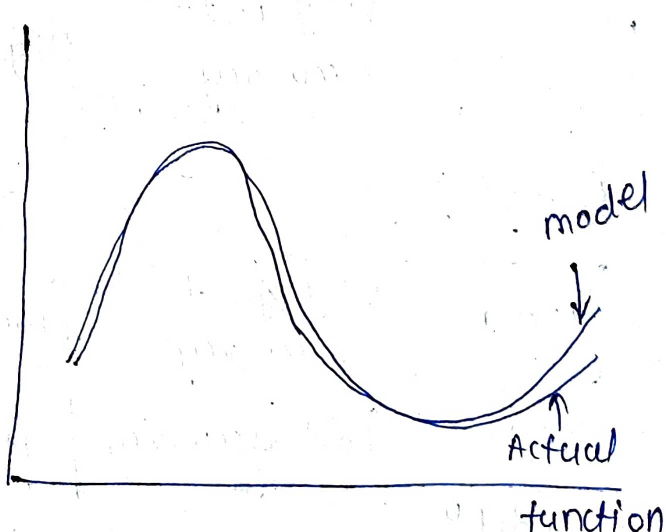




model is too simple
ie underfitting.



model is too complex.
ie overfitting



Good model.