* Hare Krishna *

Linear Regression Algorithm:-

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

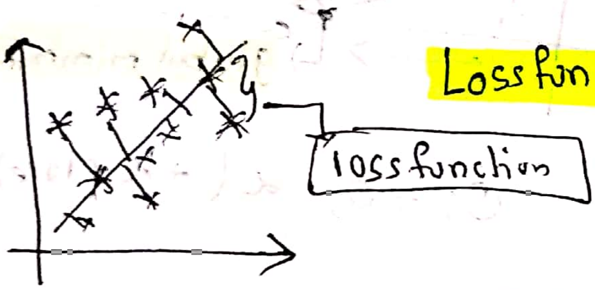
Convergence Algorithm:

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

$\boxed{\text{MSE}} \leftarrow$

↳ cost function

Every observation ← [Loss function vs cost function]



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

$$\hookrightarrow \left(\hat{y}_{(i)} - y_{(i)}\right)^2$$

Predicted $\quad$ Actual

loss function is about every Single Point different.

derivative:-

$$\frac{\partial}{\partial x}(x)^2 = 2x$$

$$\frac{\partial}{\partial x}(x)^n = n x^{n-1}\,(1)$$

$$\frac{\partial}{\partial x}(x+1)^2 = 2\cancel{x}(x+1) \times (1+0)$$
$$= 2(x+1)$$

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

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$j = 0$

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

$$= \frac{2}{m}\sum_{i=1}^{m}\left[(\theta_0 + \theta_1 x) - y^{(i)}\right] * (1 + 0)$$

$j = 1$

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

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

Repeat until Convergence:

{

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

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

}

Cost function:

① MSE :- [Mean Squared Error]

② MAE :- Mean Absolute error

③ RMSE :- Root Mean Square error

① MSE:- (Mean Squared Error]

$$MSE = \sum_{i=1}^{n}\underbrace{\frac{(y-\hat{y})^2}{n}}_{\text{quadratic equation}} \qquad \hat{y} = \theta_0 + \theta_1 x$$

↳ predicted value

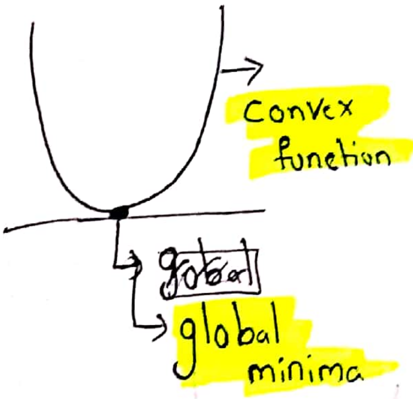
Equation of straight line = $y = mx + c$

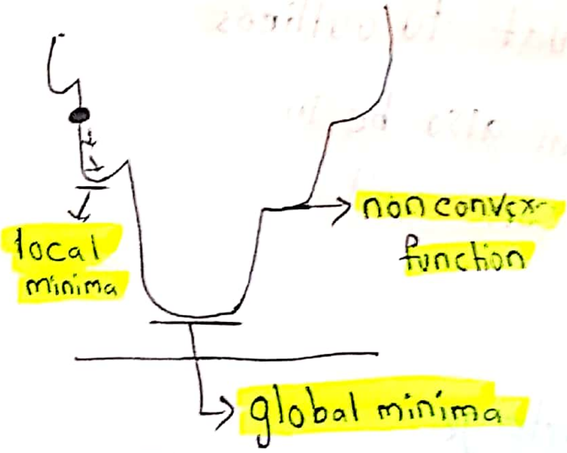
$$\boxed{ax^2+by+c=0} \rightarrow \text{quadratic equation}$$

MSE (Advantages):

Penalizing ⇒ increasing

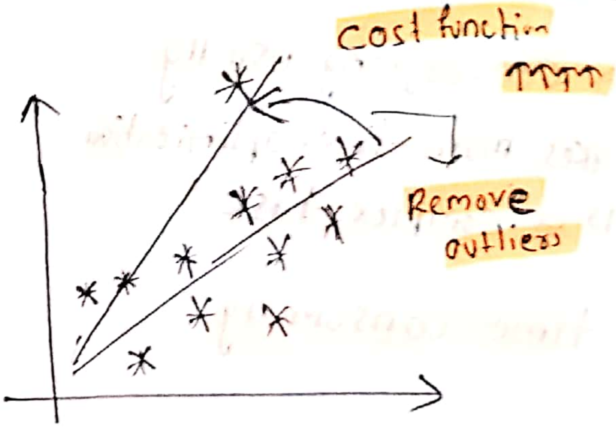
- this Equation is differentiable
- This Equation also has one global minima





SE (disadvantage)

- this Equation is not robuest to outliers
- Penalizing the error changing the size



$$\boxed{(y-y^{\wedge})^2}$$

Exp → independent

Salary → dependent (lakhs) (INR)

$(\text{Salary} - \text{predicted Salary})^2$

lakhs ↑ Salary, lakhs ↑ predicted Salary

Error

$(\text{lakhs})^2$ ↑ unit changing

$$MSE \Rightarrow \frac{\sum_{i=1}^{n} (y-y^{\wedge})^2}{n}$$

↳ cost function should ↓↓↓↓

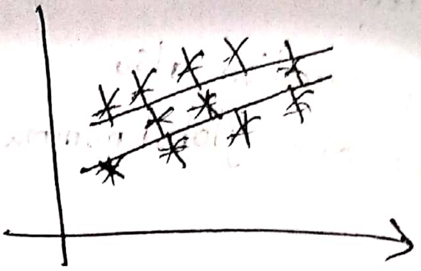
$(y-y^{\wedge})^2 \Rightarrow$ lakhs

Error ↳ squared ↳ error

increasing ← (Penalized)

② Mean Absolute Error :- (MAE)

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



Advantage:

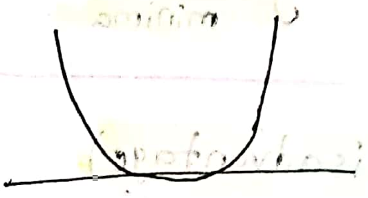
⇒ Robuat to outliers

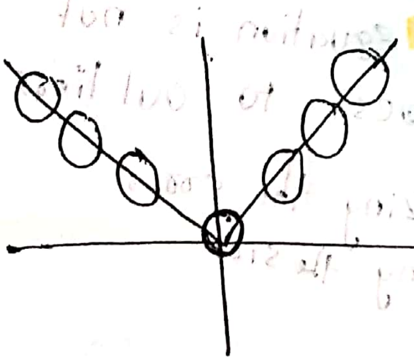
⇒ It will also be in the same unit.

Disadvantage:

⇒ Converging usually takes more time. optimization is a complex task

⇒ time consuming.





Sub gradient concept

we cannot find the derivation when it is '0', we can find derivating by using sub gradiant

Sub-gradiant means

you have to take a region of a part and divide that region and find

Peoformance matoices:

- R Squared
- Adjusted R Squared

Loss function:

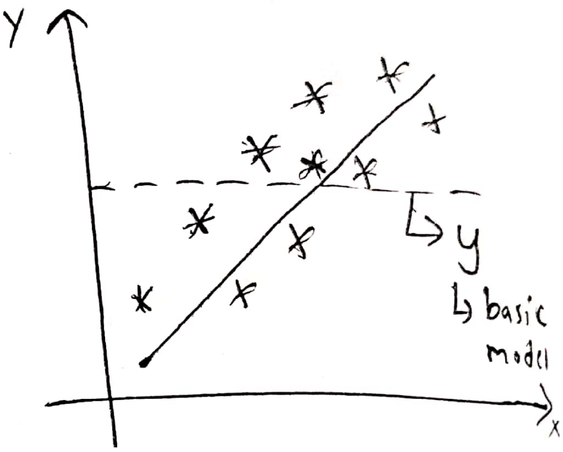
- MAE
- MSE
- RMSE
- Huber loss

R-Squared:

$$R\text{-squared} = \frac{1 - SS_{Res}}{1 - SS_{total}}$$

SS_{Res} = Sum of square residuals

SS_{total} = Sum of Square Average



$$= 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \rightarrow \text{low Valve}}{\sum_{i=n}^{n} (y_i - \bar{y})^2 \rightarrow \text{high Valve}}$$

$\bar{y}$ = Aveage of y

R-Squared $= 1 - \left\{ \frac{\text{Small number}}{\text{Bigger number}} \right\}$ → Small number

= 0.85

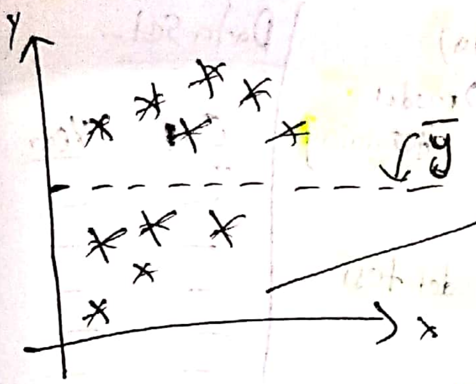
↳ 85% accurate

= 0.75 →

Pdf

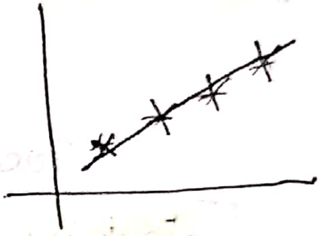
Performance of the model that You have created:

R^2 squared {-ve}



$R^2 = -ve$

$\boxed{R^2 = 1}$



$\bar{y}$ = average of y Point

⇒ Adjusted R-squared:

{over filling & underfilling}

Size of house

city location

No. of bedroom

Gender

$\boxed{\text{Price}}$

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

N = NO of data Point

P = NO of Independent feature

R^2 63%

R^2 75%

R^2 88%

R^2 90% ⇒ NO

We use adjusted R-squared.

Adj R^2	= 63%	P = 1
Adj R^2	= 73%	P = 2
Adj R^2	= 88%	P = 3
⋮	⋮	⋮
Adj R^2	85%	

Over fitting & under fitting:

(Bias and Variance)

Dataset

1000 data points

→ Training (700)

↳ model ~~tr~~ training

→ Test (300)

↳ model test

↓ New Data set

Data set:

Exp	Year
—	—
—	—
—	—
—	—
—	—
—	—

700

Training dataset

500 → train ⇓ train the model

200 → Validation ⇓ hyperparameter ~~tra~~ tunning model

model

TRain Data | Very good Accuracy (90%) [~~Lo~~ Bias]

TEST Data | very good Accuray (85%) [Variance]

↳ Generalized model

Traning data | Very good accuracy [90%] [low Bias]

TEST Data | Bad Accuracy (50%) [High Variance]

⇓

Over fitting } to fix it we need from hyper Parameter tunning

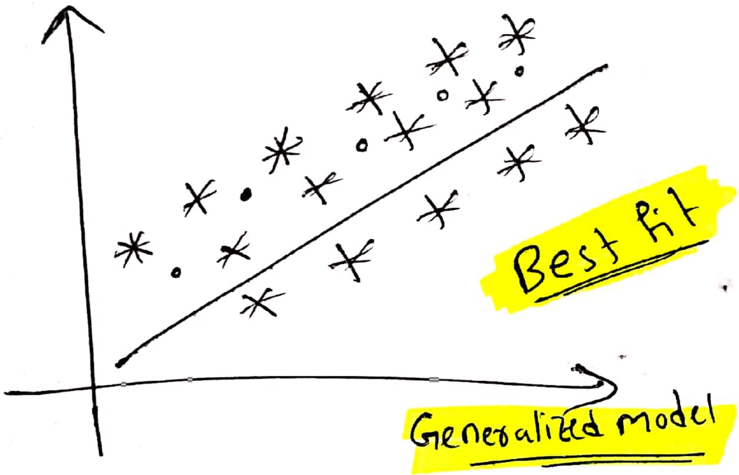
Model accuracy is low [High bias]

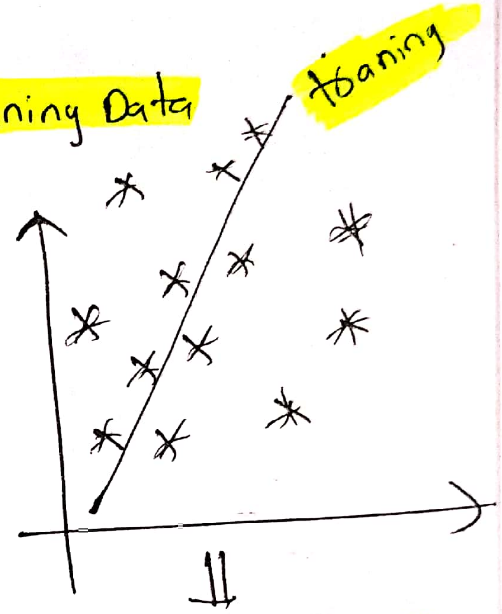
Model Accuracy is low/high [low or high Variance]

⇓

model is under fitting

* → training Data





⇓

over fitting