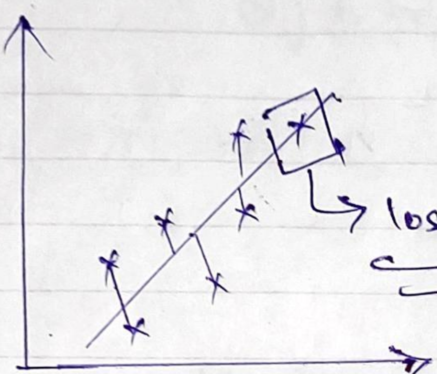
Notes

M.L Class 2

$h_\theta(x) = \theta_0 + \theta_1 x$ → 1 Independent 1 Dependent

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots$ → Multiple independent Dependent



cost function (MSE):

↳ loss function

m → no. of data-points

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

$h_\theta(x)^{(i)}$ → predicted

$y^{(i)}$ → truth point

Convergence Algorithm:

~~Every Obstacl~~ Every observation [loss vs cost function]

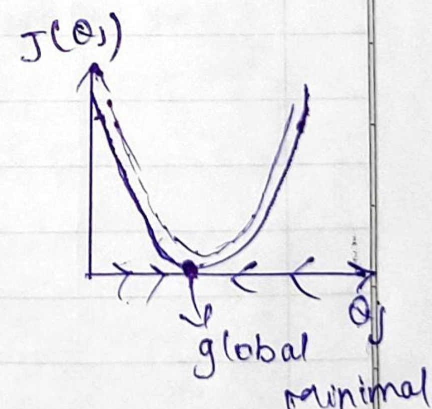
$$\text{loss function} = \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 \Rightarrow \left(\hat{y}_0 - y^{(i)}\right)^2$$

$\hat{y}_0$ → predicted, $y^{(i)}$ → Actual

Convergence Algorithm:

Repeat until convergence

$$\left\{ \quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j) \quad \right\}$$



Expt. No...................... Page. No......................

$$\frac{\partial}{\partial Q_0} J(Q_0, Q_1) = \frac{\partial}{\partial Q_0}\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2\right]$$

$J=0$

$$\therefore h_\theta(x) = Q_0 + Q_1 x$$

$$= \frac{\partial}{\partial Q_0}\left[\frac{1}{m}\sum_{i=1}^{m}\left((Q_0 + Q_1 x)^{(i)} - y^{(i)}\right)^2\right]$$

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

$$= \frac{2}{m}\sum_{i=1}^{m}\left[(Q_0 + Q_1 x) - y^{(i)}\right] * 1$$

$j=1$

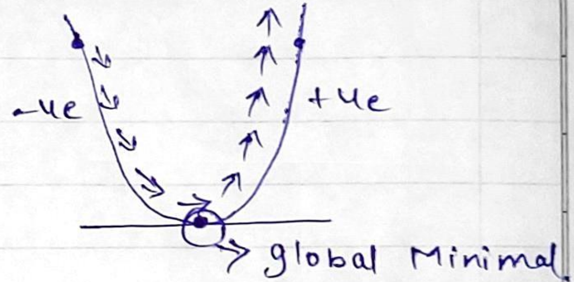
$$= \frac{\partial}{\partial Q_1}\left[\frac{1}{m}\sum_{i=1}^{m}\left((Q_0 + Q_1 x))^{(i)} - y^{(i)}\right)^2\right]$$

$$= \frac{2}{m} \sum_{i=1}^{m} \left[\left((Q_0 + Q_1 x)^{(i)} - y^{(i)} \right) * [x] \right]$$

Repeat until Convergence

$\boxed{\alpha = \text{speed of convergence}}$

$$\left\{ \begin{aligned} Q_0 &:= Q_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_Q(x)^{(i)} - y^{(i)} \right) \\ Q_1 &:= Q_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_Q(x)^{(i)} - y^{(i)} \right) x^{(i)} \end{aligned} \right\}$$



MSE {Mean Squared Error}

$$MSE = \frac{\sum_{i=1}^{n} (y - \hat{y})^2}{n}$$ → Quadratic Equation.

$*\ (a-b)^2 = \boxed{a^2 - 2ab + b^2}$ ⇒ linear Algebra

$$\boxed{ax^2 + by + c = 0}$$

$\hat{y} = Q_0 + Q_1 x$ → Predicted value

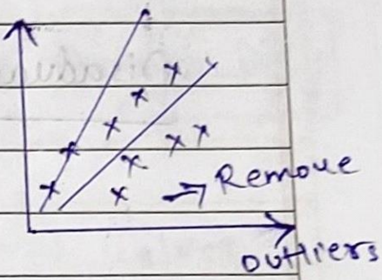
Advantage

cost function

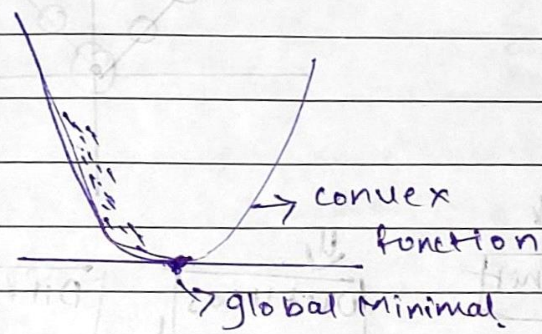
↑↑↑

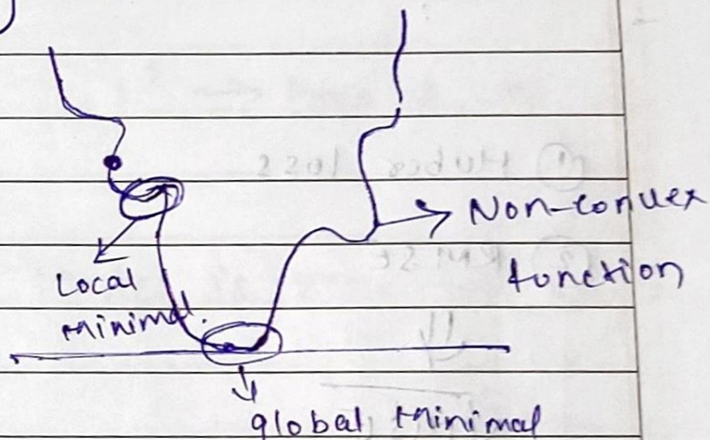
(1) This equation is differentiable
(2) This equation also has one global Minimal.

Disadvantage



(1) This is not robuct to Outliers.
(2) penalizing the Error changing the unit





x	y	
Exp	salary → Dependent feature	
—	—	
—	—	
—	—	

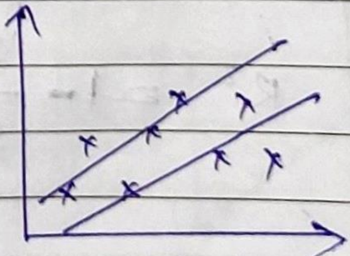
$$MSE = \sum_{i=1}^{n} \frac{(y - \hat{y})^2}{n}$$

⇓

cost fun

(2) Mean Absolute Error (MAE)

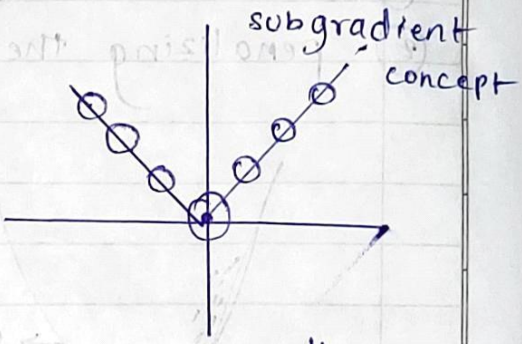
$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y - \hat{y}|$$



MAE

Advantage ① Robust to Outliers
② It will also be in the same unit.

Disadvantage :- ① Convergence usually takes more time. optimization is a complex task
② Time consuming.



① Huber loss
② RMSE

$\Downarrow$

$\sqrt{RMSE}$

$\Downarrow$ $\boxed{\text{Unit}}$ $\Downarrow$ $\boxed{\text{Outliers}}$ $\Downarrow$ $\boxed{\text{Differentiable}}$

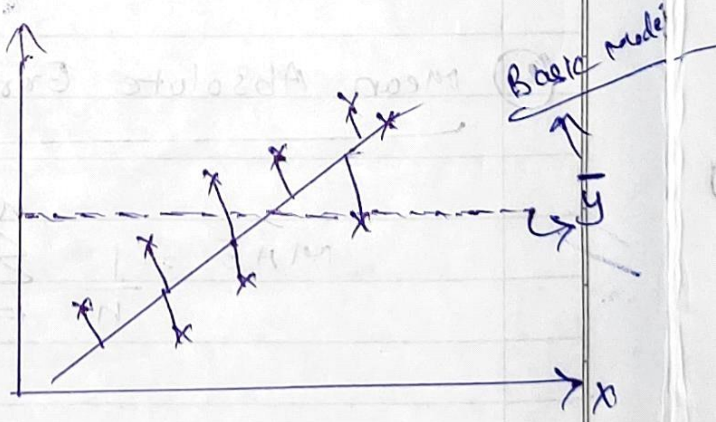
loss functions :- MAE, RMSE, Huber loss

$\boxed{\text{Performance Metrics.}}$

① R squared ② Adjusted R squared

① R squared :-

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



SSRes = Sum of Square Residuals

SSTotal = Sum of Square Average.

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

$\boxed{\bar{y} = \text{Average of } y.}$

$= 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$ → low values. / → high values

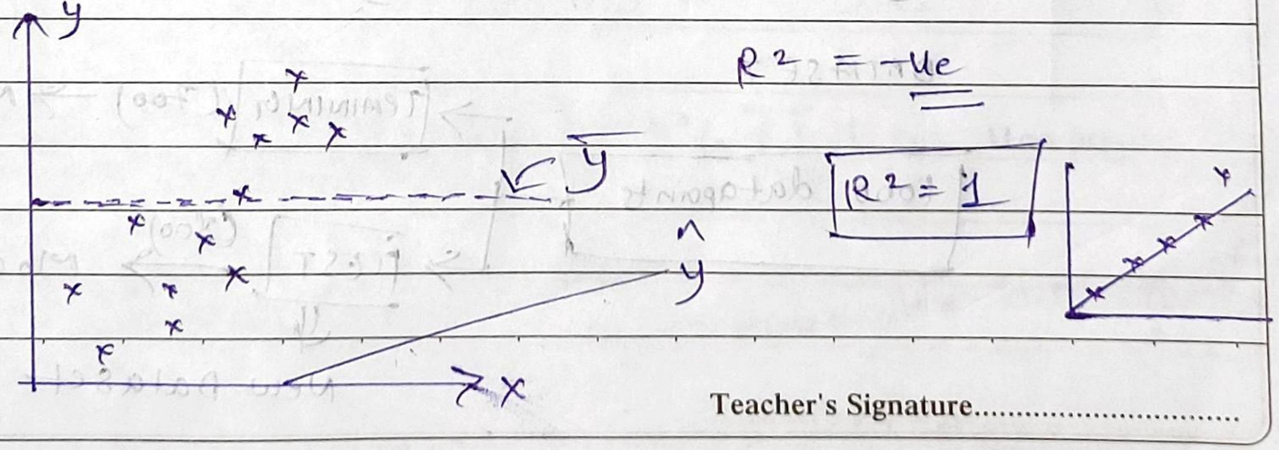
$\boxed{R^2 \text{ squared, be } -ve}$ ⇒ Model is very bad.

$\boxed{< 1}$

$R^2 = 1 - \left\{ \frac{\text{small number}}{\text{Bigger number}} \right\}$

0.85 %
⇓
85 % Accurate

performance of Model that you have created.



* Adjusted R Squared

size of house | city location | No. of bedrooms | Gender | Price

{ Overfitting And underfitting }

R^2 65% ✓ Adj R^2 = 63% p = 1

R^2 75% ✓ Adj R^2 = 73% p = 2

R^2 88% ✓

R^2 90%

} ⟹ No

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

N = No. of data points

p = No. of Independent feature

Overfitting And Underfitting (Bias And Variance)

DATASET

1000 datapoints → TRAINING (700) → Model training

1000 datapoints → TEST (300) → Model Test

TEST ⇓ New Dataset

700

Training DATASET

500 → TRAIN → training Model

200 → VALIDATION → Hyper parameter tuning. (improved model)

Example ÷

Exp	Age
—	—
—	—
—	—
—	—

Model → Train DATA → Very Good Accuracy (90%) [Low Bias]

TEST DATA → Very Good Accuracy (85%) [Low Bias]

↓

Generalized model

TRAIN Very Godd Accuracy [90%] [Low Bias]

TEST Bad Accuracy [50%] [High Variance]

⇓ Overfitting

⇓ Hyper parameter tuning.

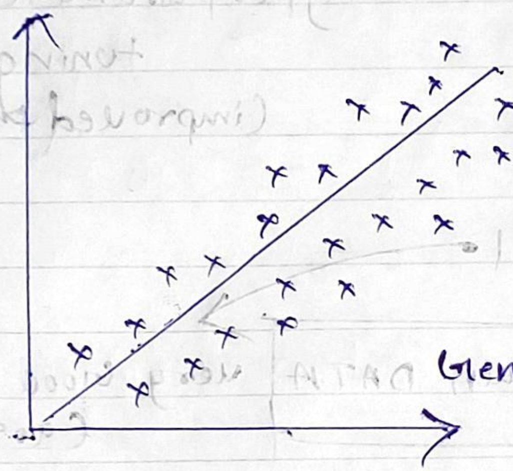
Model Accuracy is low [Low Bias]

Model Accuracy is low/High [low or High variance]

⇓

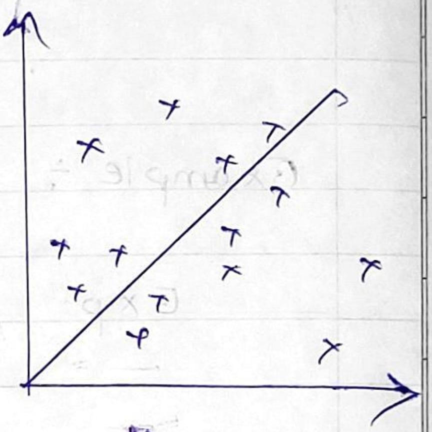
Model unfitting

Training DATA



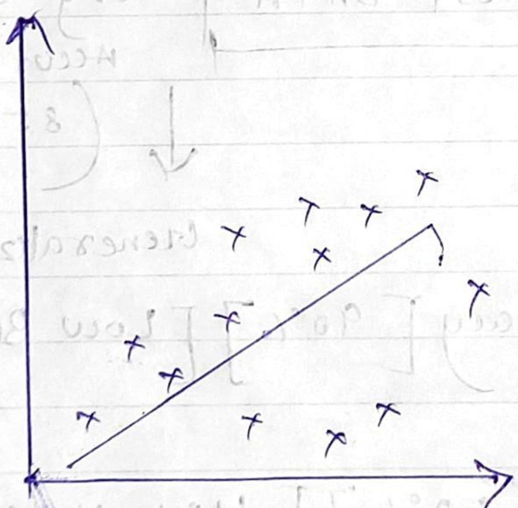
Best fit

Generalized model



⇓

overfitting



⇒ underfitting