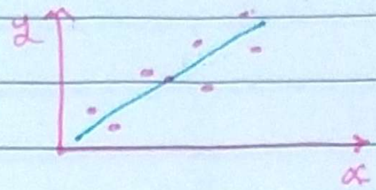
②

1. Linear Regression: → Find the best fit line



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

intercept ↲ ↳ slope

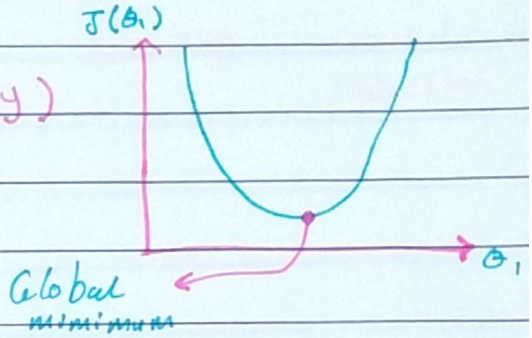
→ The best fit line where the distenation between Predicted Point & the actual Point is minimal

→ cost function :- $\sum_{i=1}^{m} \frac{1}{2m} (h_\theta(x_i) - y_i)^2$

→ Gradiant Descent $\theta_J = \theta_J * \alpha \frac{\partial}{\partial \theta_J} J(\theta_0 \theta_1)$

$m = m -$ learning rate $* \frac{\partial}{\partial m} \rightarrow \frac{2}{m} \sum (\underline{x} * ($ ~~y~~ $y_{Pred} - y)$

$b = b -$ learning rate $* \frac{\partial}{\partial m} \rightarrow \frac{2}{m} \sum (y_{Pred} - y)$



→ Performance matrix :-

R^2 :- $1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$ ($\hat{y}$ → Predicted, $\bar{y}$ → mean)

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

No. samples ← N, P → No. features

* → Regulization: To Prevent over fitting

over fitting: → Model Perform well in train "low bias"
↳ model fail to Perform well in test "high variance"

under fitting: Model acuracy is bad for both test & train
"high bias & high variance"

* → Ridge "L_2 Regulization":

predicted

$$Ridge = (\hat{y}_i - y_i) + \lambda (slope)^2$$

* → lasso "L_1 Regulization":

$$lasso = (\hat{y}_i - y_i) + \lambda |slope|$$ → helps in Feature selection