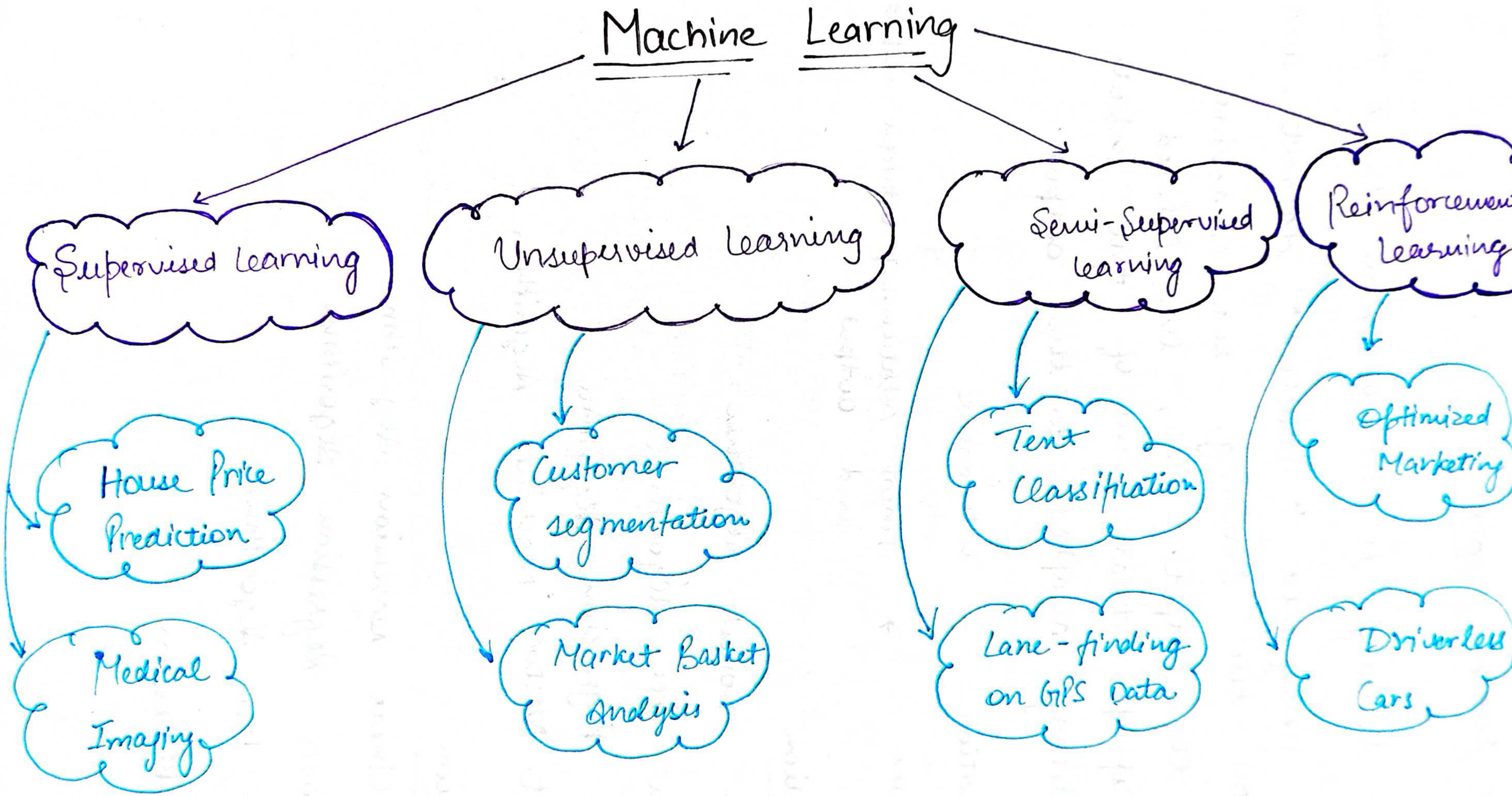

Machine Learning
Supervised Learning
House Price Prediction
Medical Imaging
Unsupervised Learning
Customer segmentation
Market Basket Analysis
Semi-Supervised learning
Text Classification
Lane-finding on GPS Data
Reinforcement Learning
Optimized Marketing
Driverless Cars

I. Supervised Machine Learning

Supervised learning is technique, we train the machine using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output.

- Classification → categorical
- Regression → linear relationship between input and output.

Classification

(i) Random Forest Algorithm
(ii) Decision Tree Algorithm
(iii) Logistics Regression Algorithm
(iv) Support Vector Machine Algorithm.

Regression

(i) Simple linear regression Algorithm
(ii) Multivariate Regression Algorithm
(iii) Decision Tree Algorithm
(iv) Lasso Regression.

2. Unsupervised Machine learning

Unsupervised learning is different from the Supervised learning technique, as its name suggests, there is no need for supervision. It means, in unsupervised machine learning the machine is trained using the unlabled dataset, and the machine predicts the output without any supervision.

- Clustering :- use when we want to find the inherent groups from the data.

It is a way to group the objects into a cluster such that objects with the most similarities remain in one group and have fewer or no similarities with the object groups. eg:- purchasing behaviour

→ k-mean clustering algorithm

→ Mean-shift algorithm

→ DBSCAN Algorithm

→ Principal Component Analysis

→ Independent Component Analysis.

2) <u>Association</u> :-

Association rule learning is an unsupervised technique, which find interesting relationships among variable within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variable accordingly so that it can generate maximum profit. eg:- Market Basket analysis Basket anaylsis, webusage mining, continous production etc.

Simple Linear Regression

Linear Regression

Dataset:-

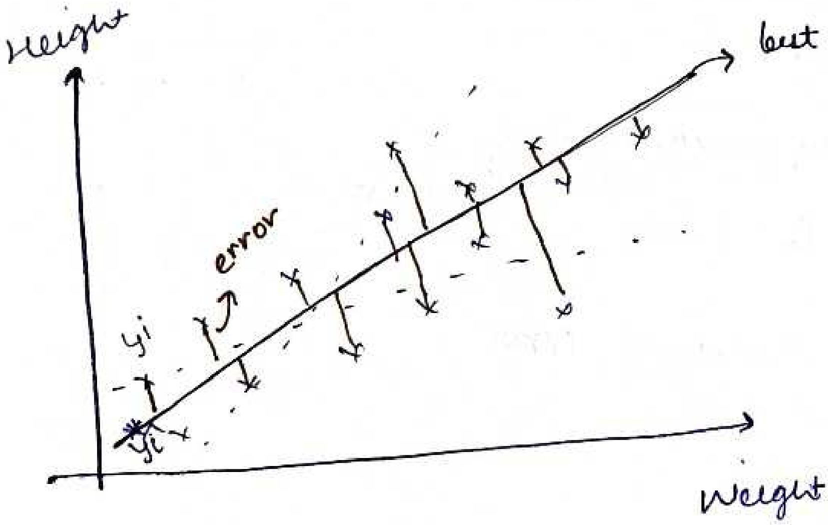
Weight (Independent)	Height (dependent)
74	170
80	180
75	175.5
—	—
—	—
—	—
—	—

Train dataset ↓

Weight → Model → Prediction Height

↓

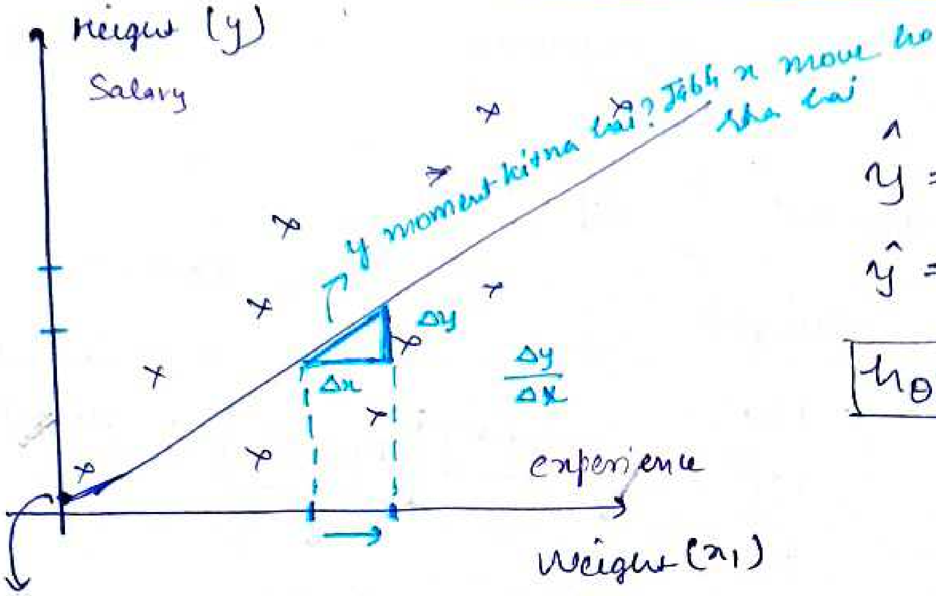
Acc ↑↑



best fit line → prediction

Σ erro ↓↓ minimal

Aim:- The main Aim of simple linear Regression is to find best fit line in such a way when we do the ~~sum~~ summisution (addtion) of actual points and the predicted point or ~~sum~~ addition of the all point it should be minimal



$\hat{y} = mx + c$

$\hat{y} = \beta_0 + \beta_1 x_1$

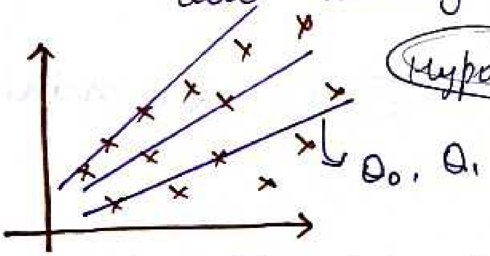
$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

θ_0 = Intercept

θ_1 = slope or coefficient

Intercept :- when value of x is 0 then there line join the point is called Intercept.

eg:- where 0 experience person have also salary.



Hypothesis Testing

* which line has minimal error

$\theta_0, \theta_1 \to$ Change

Error ↓↓

Cost function [Error]

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (\underset{\text{Actual}}{y_i} - \underset{\text{Predicted}}{h_\theta(x)_i})^2 \quad \text{[mean squared error]}$$

$h_\theta(x)_i = \hat{y}_i$

n = no. of data points

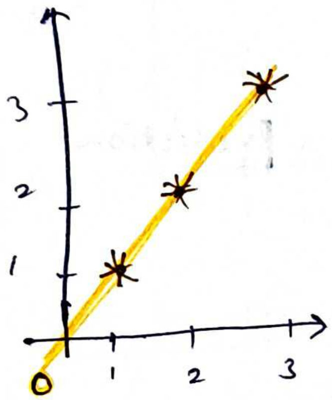
y_i = Actual value

$h_\theta(x)_i$ = Predicted value

Final aim : [In order to get best fit line]

$$\text{Minimize} \quad J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=0}^{n} (y_i - h_\theta(x)_i)^2$$

* changing the value θ_0, and θ_1 for minize and get best fit line



$$h_\theta(x) = \overset{\overset{0}{\downarrow}}{\theta_0} + \theta_1 x$$

Lets consider $\theta_0 = 0$

$$h_\theta(x) = \theta_1 x_1$$

Let $\theta_1 = 1$

Dataset

x	y	$h_\theta(x)$
1	1	
2	2	
3	3	

$x = 1 = h_\theta(x) = 1(1)$
$= 1$

$x = 2 = h_\theta(x) = 1(2)$
$= 2$

$x = 3 \quad h_\theta(x) = 1(3)$
$= 3$

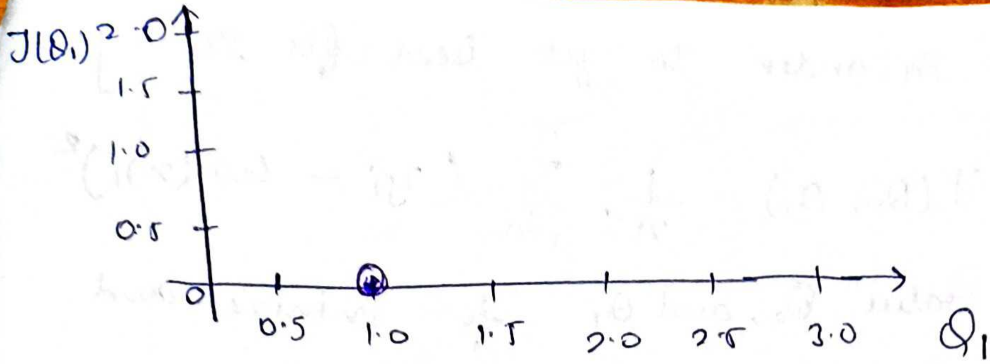
Predicted

Cost function

$$J(\theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2$$

$$= \frac{1}{3} [(1-1)^2 + (2-2)^2 + (3-3)^2]$$

$$= 0$$



$J(1) = 0$

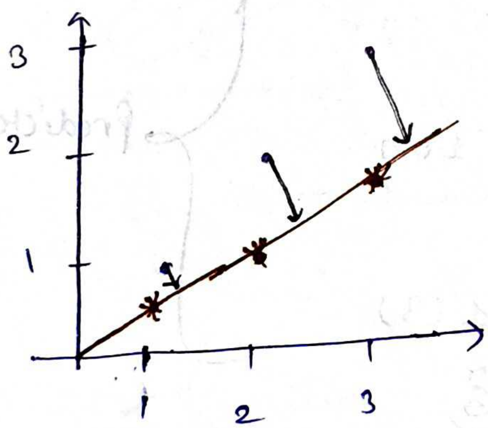
If let $\theta_1 = 0.5$ $\quad x=1$

$h\theta(x) = 0.5$

$h\theta(x) = 1.0$ $\quad x=2$

$h\theta(x) = 1.5$ $\quad x=3$

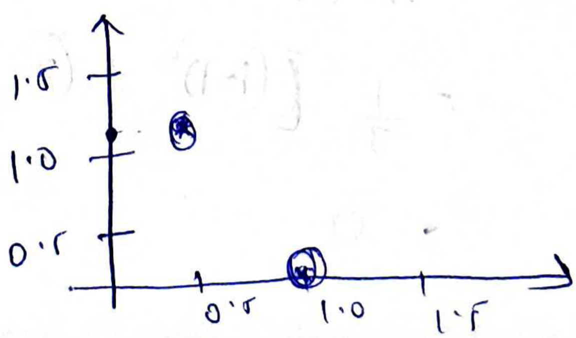
} Prediction



Cost fn

$$J(\theta_1) = \frac{1}{3}\left[(1-0.5)^2 + (2-1)^2 + (3-1.5)^2\right]$$

$$= 1.16$$

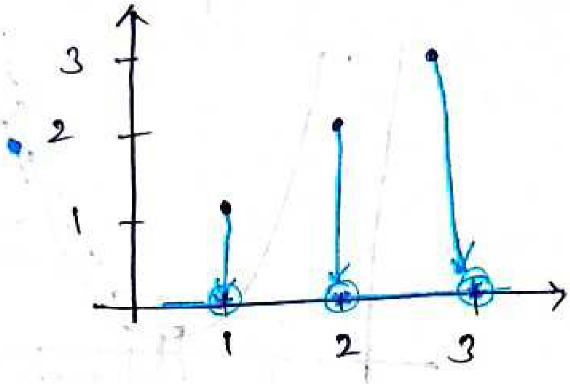


let $\theta_1 = 0$

$h_\theta(x) = 0$ $x = 1$

$h_\theta(x) = 0$ $x = 2$ } Prediction

$h_\theta(x) = 0$ $x = 3$

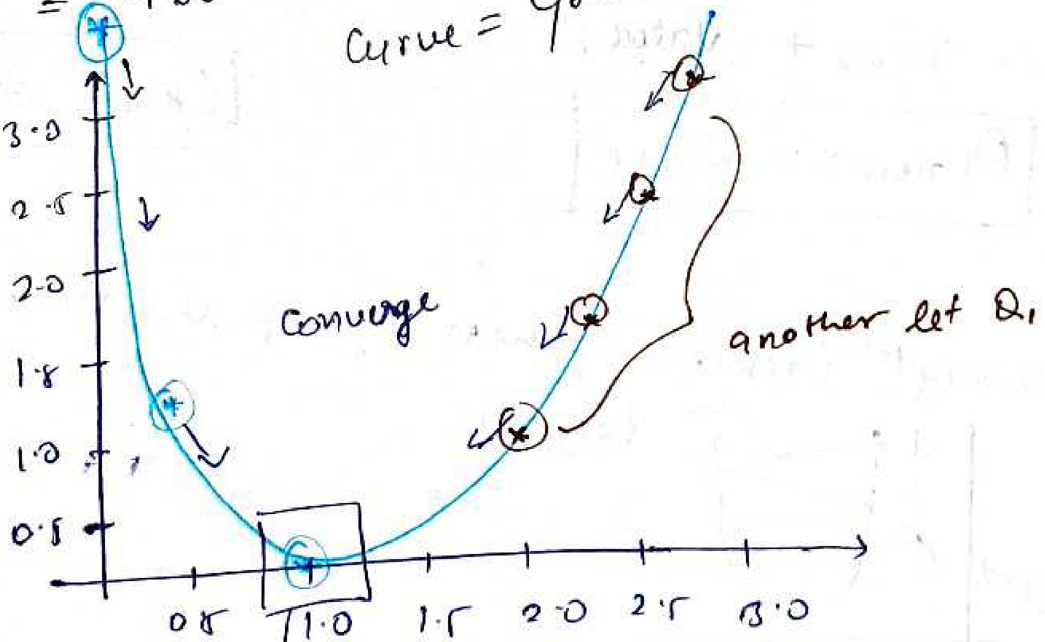


cost function

$$J(\theta_1) = \frac{1}{3}\left[(1-0)^2 + (2-0)^2 + (3-0)^2\right]$$

$= 4.66$

Curve = Gradient descents



global minimal

Interview question

* **Convergence Algorithm** :- {Optimise the change of θ_0 θ_1 to Global Minimum}

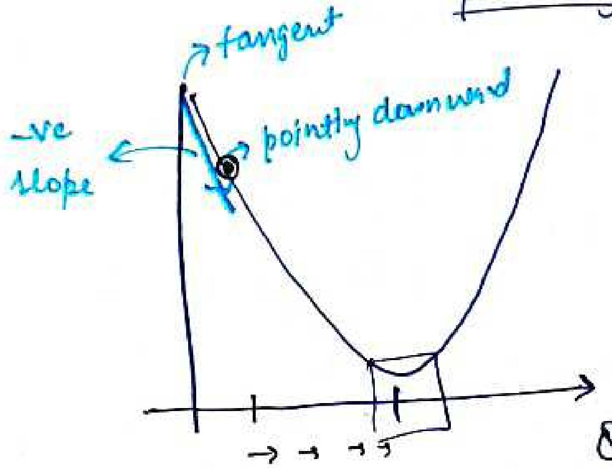
Repeat until Convergence

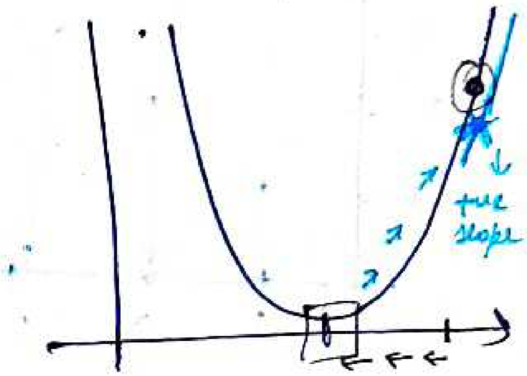
{

$$\theta_j : \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_j)}$$ ⇒ Derivative ⇓ slope of a point

$\boxed{\text{usually Learning rate} = 0.01}$

$\boxed{\text{Learning rate}}$





$\theta_1 = \theta_1 - \alpha(-ve)$

$\theta_1 new = \theta_1 old + (value)$

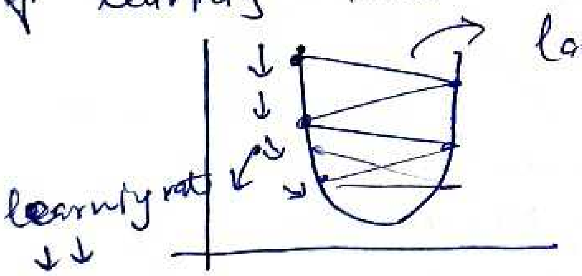
$\boxed{\theta_1 new >> \theta_1 old}$

$\theta_1 new = \theta_1 old - \alpha(+ve)$

$\theta_1 new = \theta_1 old - (value)$

$\boxed{\theta_1 new << \theta_1 old}$

if learning rate ↑↑ increase then jump the value



if learning rate ↓↓ is small then converge the value is slow.

$\boxed{\text{Learning rate} \Rightarrow \text{Speed of Conversion}}$

Conclusion:- Repeat the step until convergence

{

$$\theta_j : \theta_i - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2$$

$\Downarrow$

mean squared error

Multiple Linear Regression

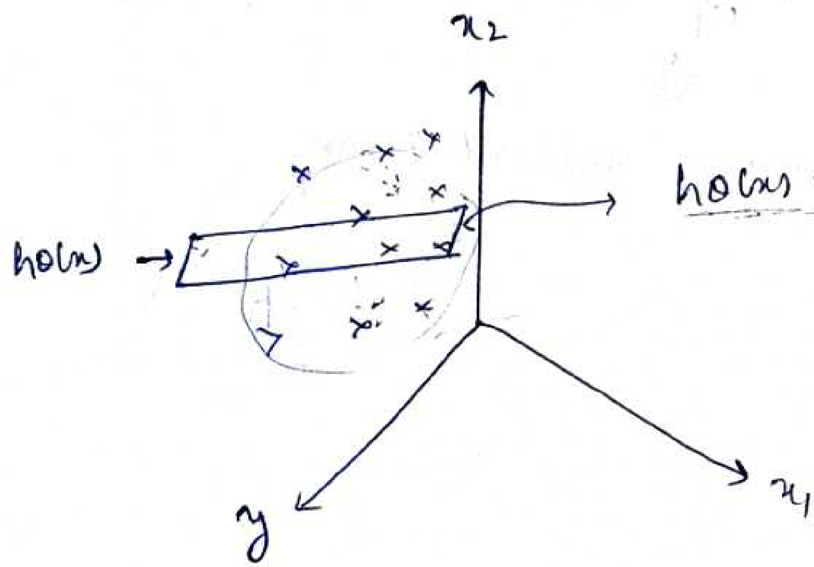
House Pricing Dataset

Independent

~~House Pricing Dataset~~

Dependent

No. of Rooms x_1 | Size of rooms x_2 | Price y



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

$\theta_1, \theta_2 \Rightarrow$ slope of cofficient

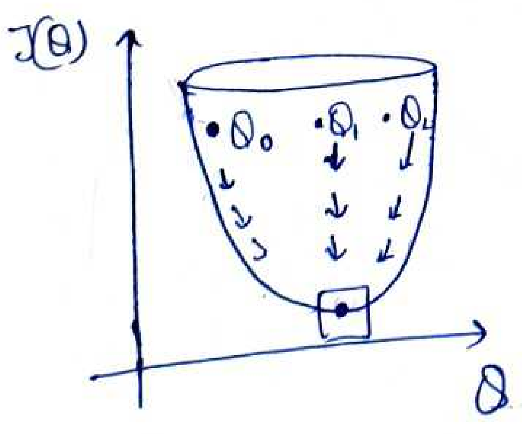
$\theta_0 \Rightarrow$ Interupt

Generic Equation Multiple regression :-

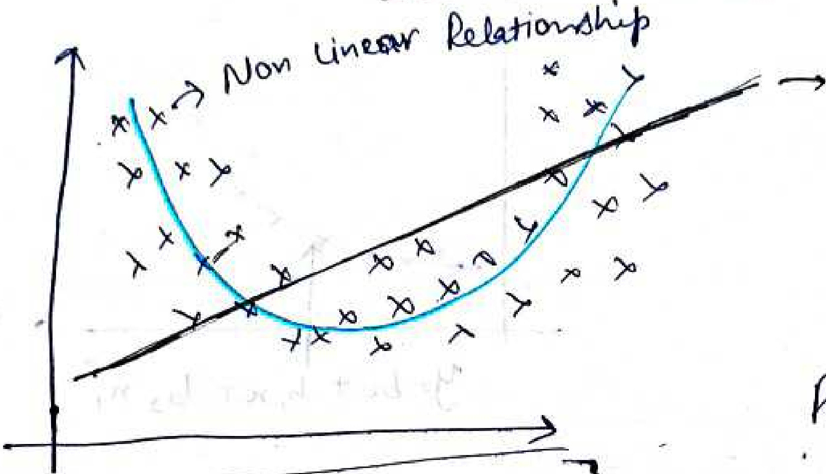
$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n}$$

Sample linear regression

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



Polynomial Regression



Simple Linear Regression

$h_\theta(x) = \theta_0 + \theta_1 x_1$

error ↑ ↑

Polynomial Regression
- Simple Polynomial Reg
- Multi " "

Polynomial Regression
⇓
Polynomial degrees

Simple polynomial Regression ⇓

Polynomial degree = 0 $\left\{ \begin{array}{l} h_\theta(x) = \theta_0 \times 1 \\ = \theta_0 x_1^0 \end{array} \right.$

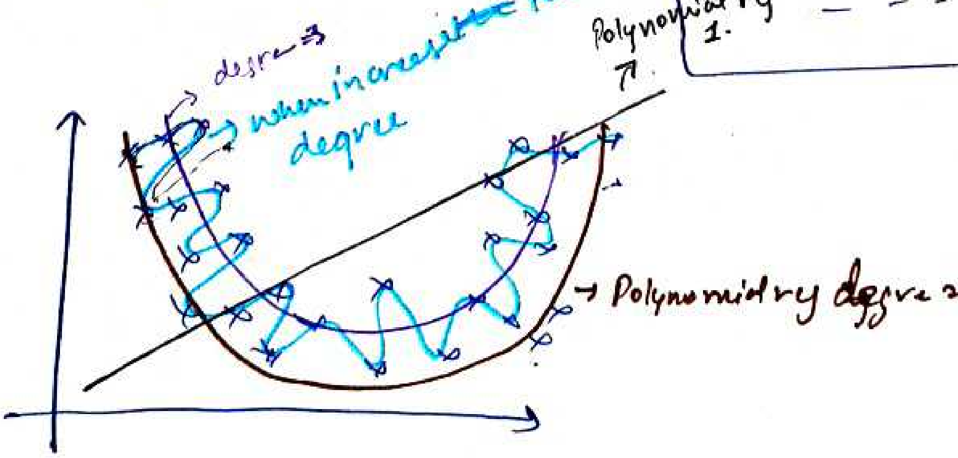
⇒ Constant

Polynomial degree = 1 $\quad h_\theta(x) = \underbrace{\theta_0 x_1^0 + \theta_1 x_1^1}_{\text{Simple Linear Regression}}$

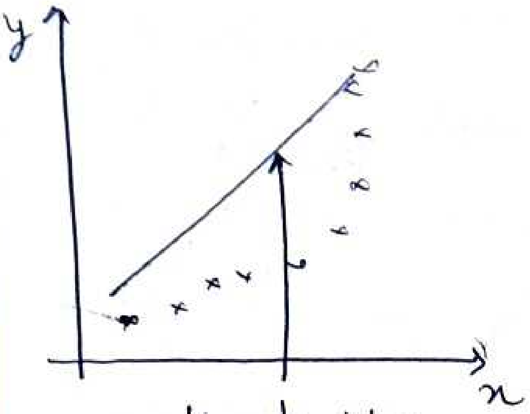
Polynomial degree = 2 $\quad h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2$

⋮

Polynomial degree = n $\quad \boxed{h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2 + \ldots \theta_n x_1^n}$

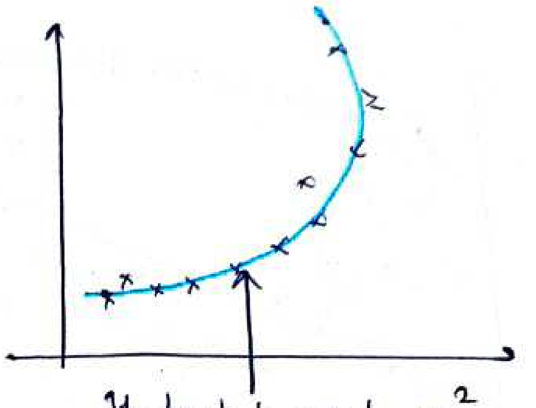


Simple linear reg



$y = b_0 + b_1 x$

Polynomial model



$y = b_0 + b_1 x + b_2 x_1^2$.

Multiple polynomial Regression

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ {multiple linear regression}

Polynomial degree = 2

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_3^2}$$

Performance Metrics Used in Regression

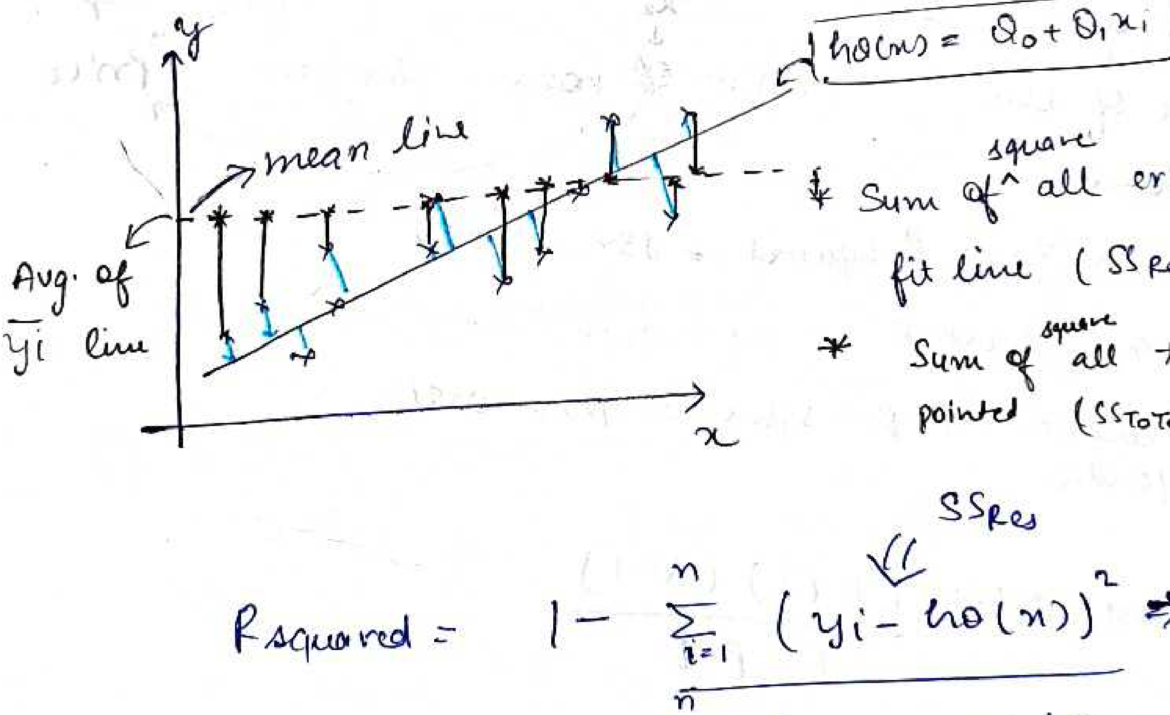
① R squared   ② Adjusted R squared

R squared

$$R\text{squared} = 1 - \frac{SS_{Res}}{SS_{Total}} \quad \{\text{Average of } y_i \text{ line}\}$$

SS_{Res} = Sum of square Residual (Error)

SS_{Total} = Sum of square Total



* Sum of square all error after fit line (SS_{Res})

* Sum of square all the predicted pointed (SS_{Total})

$$R\text{squared} = 1 - \frac{\overbrace{\sum_{i=1}^{n} (y_i - h_\theta(x))^2}^{SS_{Res}}}{\sum_{i=1}^{n} (y_i - \bar{y_i})^2} \quad \begin{matrix} \Rightarrow \text{small value} \\ \Rightarrow \text{large value} \end{matrix}$$

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

R squared ranges between 0 to 1

f0 f1 f2 f3 f4 → O/P

Correlation to output

R squared = 75% = 0.5

R squared = 80%

R squared = 90% ↑↑

No. of feature ↑ → Output Correlation

* add any feature Accuracy also ↑↑
r squared value increase

(2) Adjusted R squared

x1 x2 y

not important
gender Size of the house No. of rooms location → Price

$x_1, x_2 \rightarrow$ R. squared = 85% = 0.85

x_1, x_2 location R. squared = 90% = 0.90

x_1, x_2 location gender → R. squared = 91% = 0.91

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

N = No. of data points

R^2 = R squared

P = No. of Independent feature

Adjusted R squarr << R squared

$R^2 = 80\%$ Adjusted $= 75\%$

$P=3$ $R^2 = 85\%$ Adjusted $= 78\%$

$P=4$ $R^2 = 87\%$ Adjusted $= 76\%$

$\Downarrow$ $\Downarrow$

* Independent feature is not that important

* Use both R^2 and Adjusted R^2

MAE, MSE, RMSE [Cost function]

① Mean Squared Error (MSE)

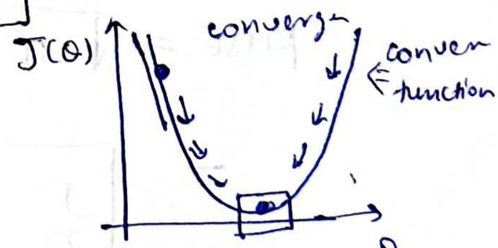
② Mean Absolute Error (MAE)

③ Root Mean Squared Error (RMSE)

① Mean squared Error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2$$

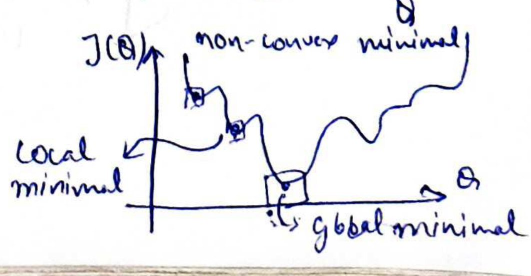
$$\boxed{MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$ → Quadratic equation $\Downarrow$



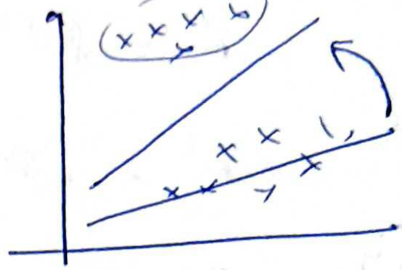
Advantage

(i) This equation is differentiable

(2) It has only one local or global minima



Disadvantage



① Not Robust to outlier.

② It is not in the same unit.

② Mean Absolute Error (MAE)

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

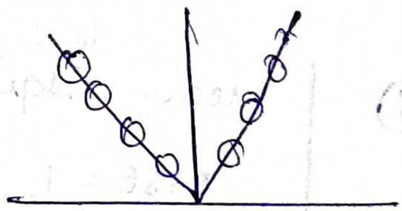
Advantage

① Robust to outlier.

② It is the same unit.

Disadvantage

① Convergence usually takes more time ⇒



③ Root means squared error (RMSE)

$$RMSE = \sqrt{MSE}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2}$$

Advantage

(1) Same unit

(2) Differentiable

dis advantage

(1) Not Robust to outliers

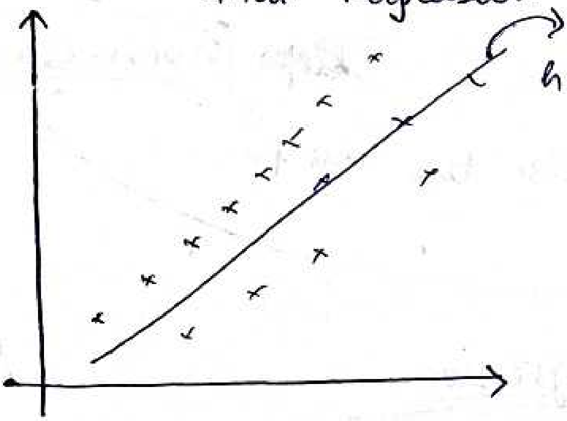
Note :- Linear Regression

Performance matrics :- R^2 and Adjusted R^2 → Acc. of model

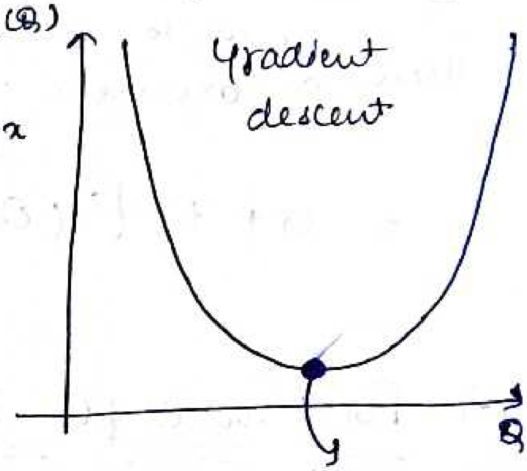
cost function → Error → MSE, MAE, RMSE

Ridge, Lasso, Elastic Regression

Linear Regression

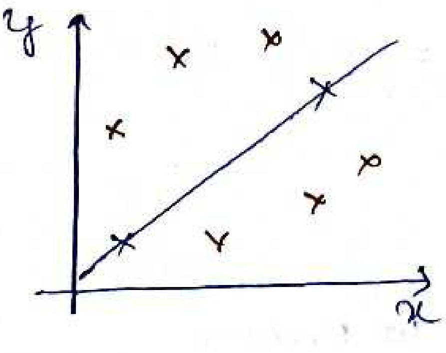


$h_\theta(x) = \theta_0 + \theta_1 x$



$$\text{cost fn} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x)_i\right)^2 \quad [\text{mean squared error}]$$

① Ridge Regression (L_2 Regularization) → Reducing Overfitting



Error = 0 (Train Data)

Error = ↑↑ (Test Data)

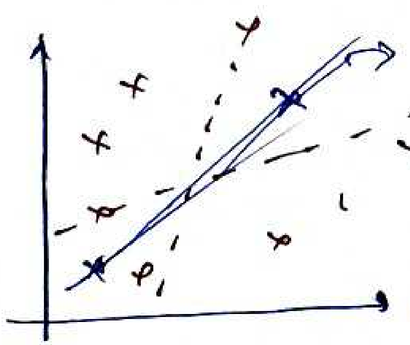
Data → Train, Test

Overfiting

Train → Acc ↑↑ → R^2

Test → Acc ↓↓ → R^2

* In Ridge Regression, Error is not 0.
Error ko zero nhi hone dena. [need little bit error]



Error = 0

Ridge Regression Error ≠ 0

$$\text{Cost fn} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}_{\text{if 0 (zero error) then add}} + \boxed{\lambda \sum_{i=1}^{n}(\text{slope})^2}$$

$\lambda \rightarrow$ Hyper parameter

Let error is 0, and $\lambda = 1$

slope = linear regressio

then accordig to formula

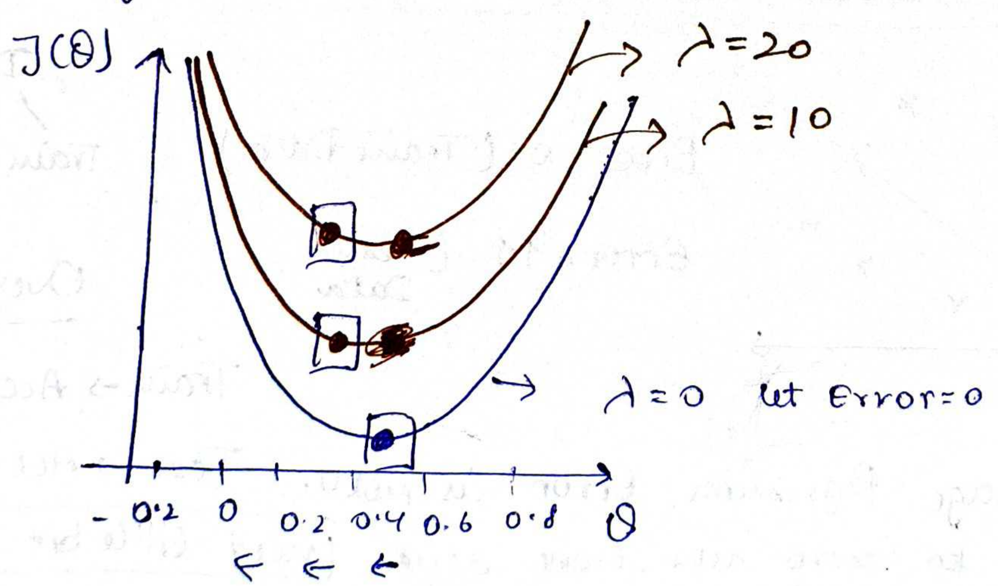
$\boxed{h_\theta(x) = \theta_0 + \theta_1 x}$

$= 0 + 1 \left[(\theta_1)^2 \right] \leftarrow$ Penalize the cost fn.

* For multiple linear regression

$$\text{cost fn} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \left[(\theta_1)^2 + (\theta_2)^2 + (\theta_3)^2 \right]$$

* relation of slop and λ



② Lasso Regression (L_1 Regularization) → Feature selection

Cost fn: $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) + \boxed{\lambda \sum_{i=1}^{n} |\text{Slope}|}$

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$= 0.52 + 0.65 x_1 + 1.5 x_2 + \boxed{0.2} x_3$ → cancel

feature remove

decrease and become zero

Predict with two featu

feature selection

$x_2 \to$ 1 unit

$y \to 1.5 x_2$

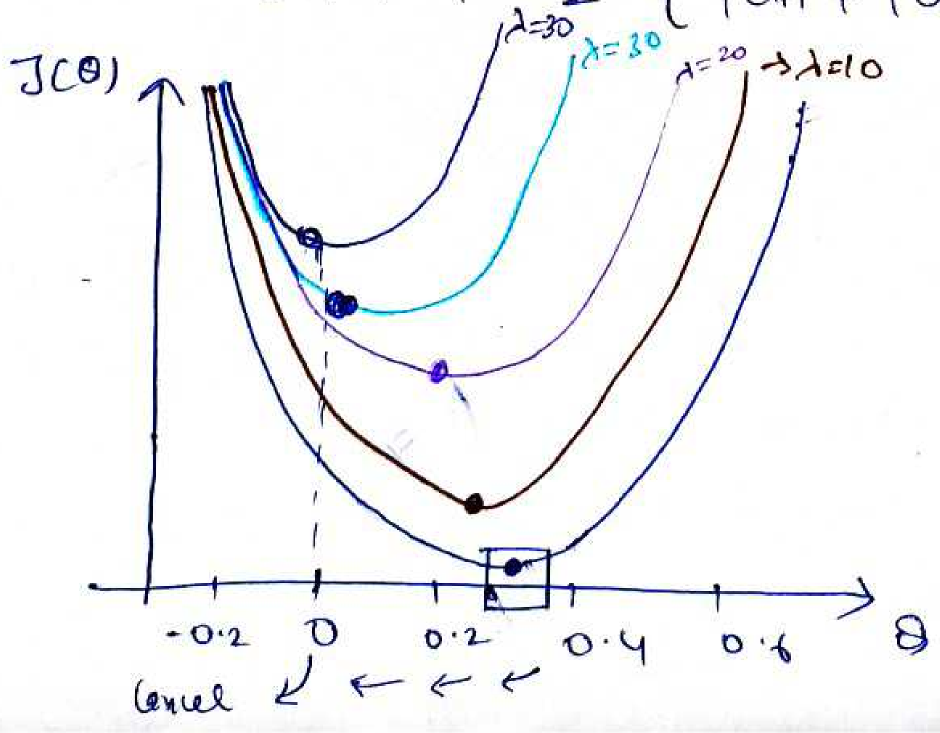
$x_3 \to$ 1 unit

$y = 0.2 x_3$

Cost fn $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \boxed{\lambda \sum_{i=1}^{n} |\text{slop}|}$

$\lambda = 1$

$= \text{Error} + 1\,[\,|\theta_1| + |\theta_2| + |\theta_3|\,]$



(3) Elastic Net Regression

Reducing Overfitting → Ridge

Feature Selection → Lasso

$$\boxed{\text{cost fn} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y_i})^2}_{\text{MSE}} + \underbrace{\lambda_1\sum_{i=1}^{n}(\text{slop})^2}_{\text{Reducing Overfitiy}} + \underbrace{\lambda_2\sum_{i=1}^{n}|\text{slope}|}_{\text{Feature selection}}}$$

$\lambda_1 \lambda_2$ = Hyperparameter