8/10/2022

MACHINE LEARNING

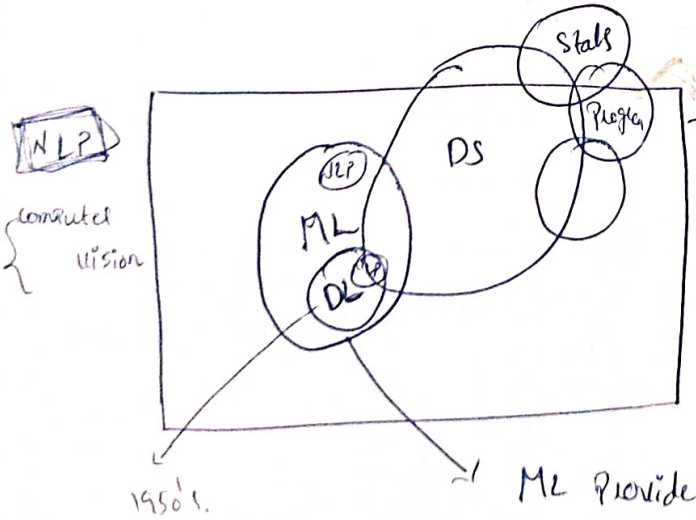
Agenda :-

1. Machine learning Introduction
2. AI Vs ML Vs DL Vs DS
3. Simple linear Regression → Mathematical Intuition

AI Vs ML Vs DL Vs DS :- aim is to create AI Application.

Ex: Netflix → recommendation system
→ Software Product.
Streaming platform
↳ AI module

at the end of the day, we can create AI application.



→ Artificial Intelligence.

It is creating an application where it performs all its task without any human intervention.

ML Provides us statistical-tools to explore, visualize, analyze and perform prediction and other tasks with the help of data.

Statistical tools :- algorithms.

Ex: Amazon.in → recommendation

- Chatbots → AI
- Google Assistant
- Self driving cars
- Alexa.
- Video streaming recommendation.

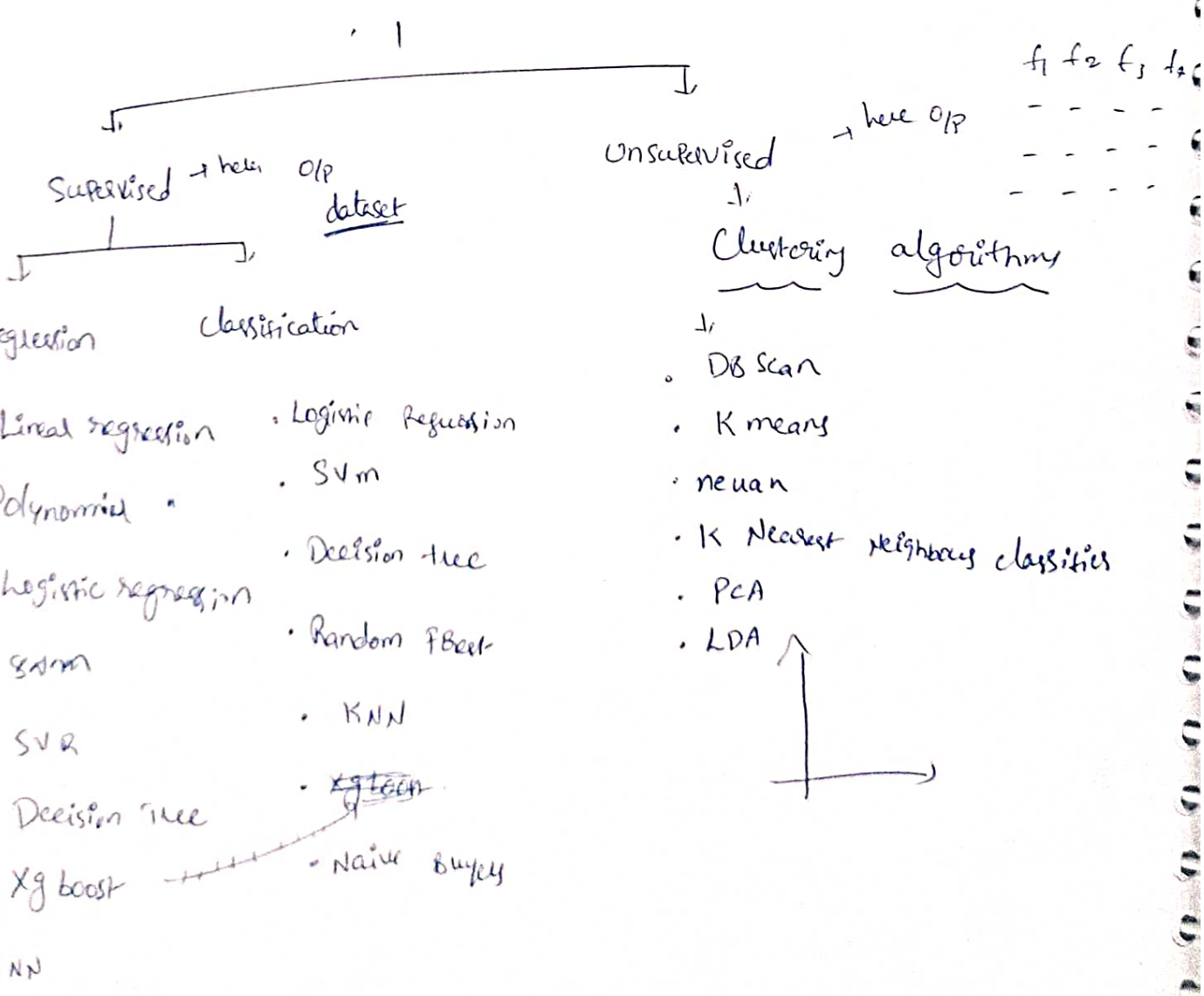
DL :-

↓ → to mimic the human brain.
Multinural n.w.

ANN, CNN, object detect problems.

NLP → where we work with text data

ML and DL



Supervised → here we found continuous variable in o/p. Unsupervised

↙ predict

Degree	EXP	Salary
BE	7	50K
PHD	2	70K
-	-	-
-	-	-
-	-	-

Regression

here datapoint is continuous

→ Independent features

Eg

No. of Play hrs	No. of Study hrs	Pass/Fail
9	1	0
7	2	0
3	5	1

dependent feature

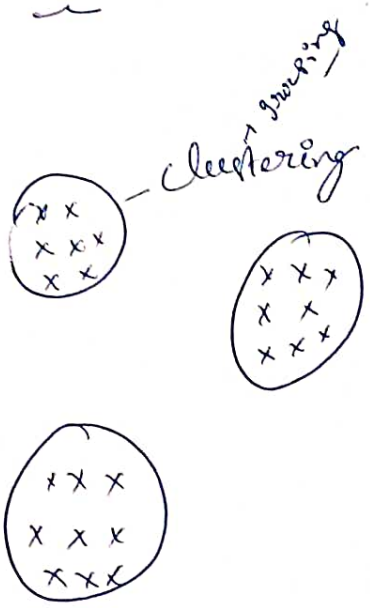
categorical → here we found fixed no. of outputs.

classification problem

Eg
- Flight Price Prediction → Regression
- Algerian Fire Forest → classification
- Air Quality Index → Regression
- Tom Rain / Not → classification
- Buy day of the Person → classification

~~Simple Linear Regression~~ :- → Unsupervised ML

Age	Salary	Spending - Score (1-10)
24	70K	1
26	100K	9
-	-	-
21	20K	9
25	120K	2



Simple Linear Regression :- → 1. Independent feature 2. Dependent feature.

Aim :- To create a model

I/p → Weight Predict → weight

Data set-

Height	Weight

Eg No. of rooms (x) Price (y)

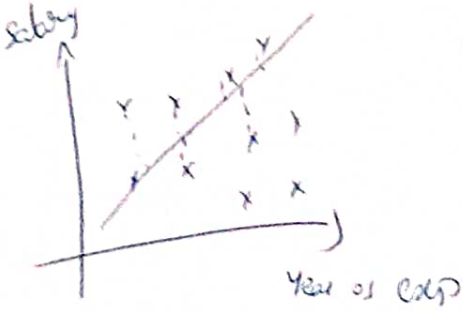
Eg Year of exp Salary (continuous)

Model → Year of exp → Salary

Predict → Salary based on I/p Year

Training dataset.

Year of exp	Salary
-	-
-	-
-	-
-	-



Error
residuals

find the best fit line, where the diff. btn Real Point & Predicted Point training model is

it should be very minimal :

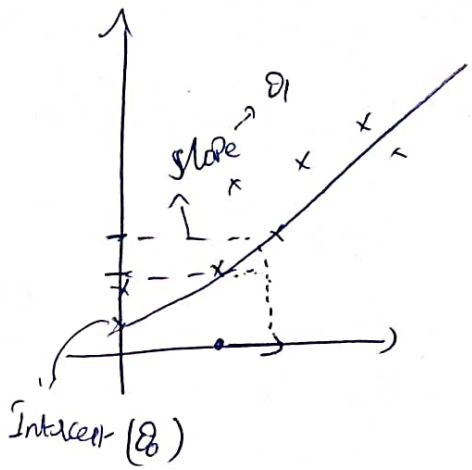
Equation of a Straight line:

$y = mx + c$

$y = \beta_0 + \beta_1 x_1$

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

θ_0 → intercept, θ_1 → slope



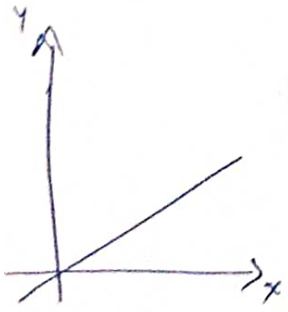
exp	Salary
0	3.25 lakh

→ With the Unit movement in x-axis, what's the movement in Y-axis → Slope

Error / Residual

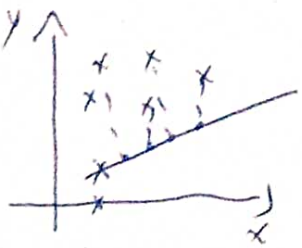
θ_0, θ_1 → Value

Training of the model



$\theta_0 = 0$

it best-fit line passes through origin, hence $\theta_0 = 0$, $\theta_1 = 0$



Minimize the error

Cost function — Predicted / actual

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

↓↓↓

m = all the data points

Mean Squared Error; ½

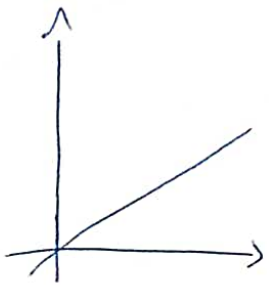
Final Aim :-

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

(increasing; J: slope)

MSE = Mean Squared Error.

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



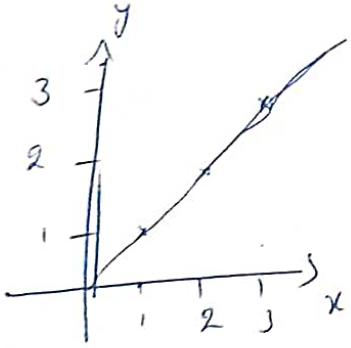
Let us consider

here $\theta_0 = 0$

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

Train

x	y
1	1
2	2
3	3



$\theta_1 = 1$

when

what is $h_\theta(x) = \theta_1$ $x = 1$

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

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

$\theta_0 = 0$. $\theta_1 = 1$.

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

here $m = 3$

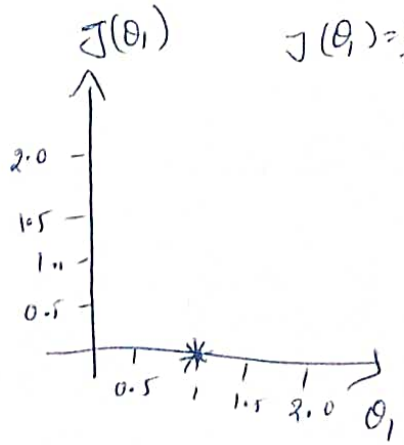
$= \frac{1}{3} \left[(1-1)^2 + (2-2)^2 + (3-3)^2 \right]$

$= \frac{1}{3} \left[0 + 0 + 0 \right]^2$

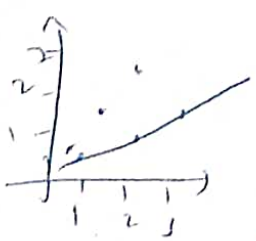
$= 0$.

$\theta_1 = 1$

$J(\theta_1) = 0$.



if $\theta_1 = 0.5$.



$h_\theta(x) = 0.5$ $x = 1$

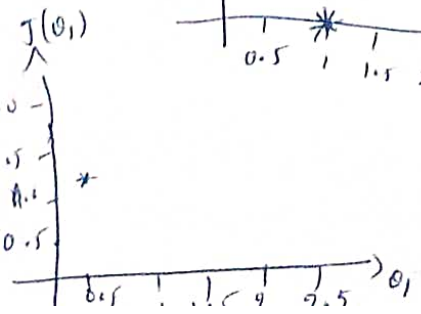
$h_\theta(x) = 1$ $x = 2$

$h_\theta(x) = 1.5$ $x = 3$

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

$= \frac{1}{3} \left[0.25 + 1 + 2.25 \right]$

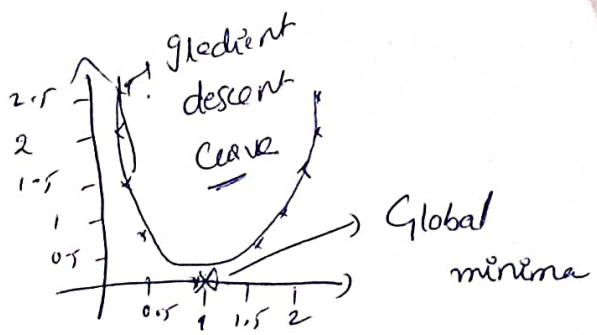
$= 1.16$



When $\theta_1 = 0$

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

$$J\theta_1 = \frac{1}{3}\left[1+4+9\right] = \frac{14}{3}$$



Convergence Algorithm {To Optimize the changes of θ_1 value}

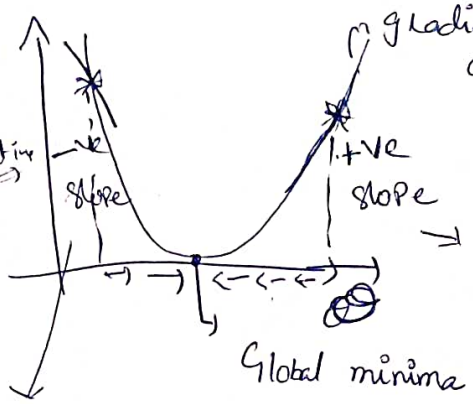
Repeat until Convergence

{

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

learning rate, derivative, slope

}



-ve slope $\theta_j = \theta_j - \alpha(-ve)$

$= \theta_j + \alpha$

if $\theta = 0$ then we can get 3D

+ve slope $\theta_j = \theta_j - \alpha(+ve)$

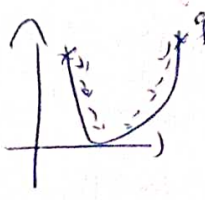
$\theta_j = \theta_j + \alpha$

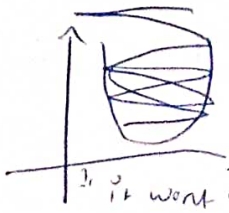
Learning rate (α) :- Decides the speed of convergence

$\alpha = 0.001$

if α is very less it takes lot of time to reach gradient descent

if α is very high, it will jump here & there



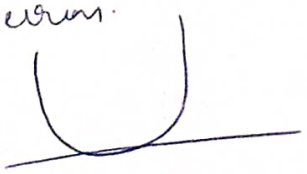


∴ it won't come to global minima

Mean squared error.

MSE

Cost function



MAE

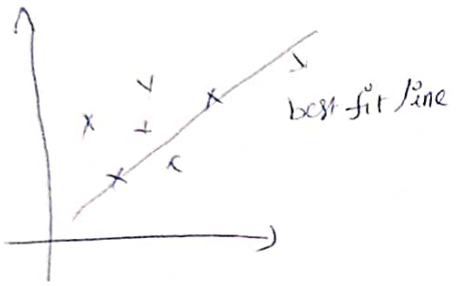
RMSE

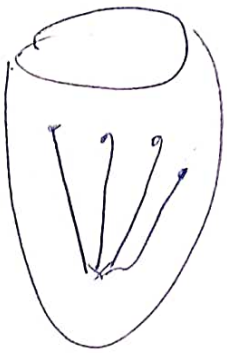
1 Independent 2 dependent

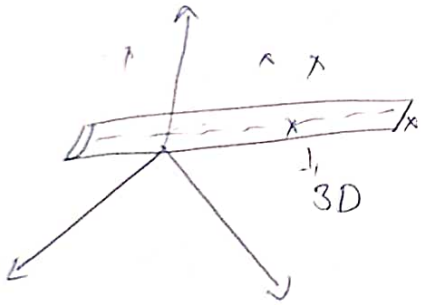
$h_\theta(x) = \theta_0 + \theta_1(x)$

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

x_1	x_2	x_3	
No. of rooms	city	Room size	Price







hyperplane