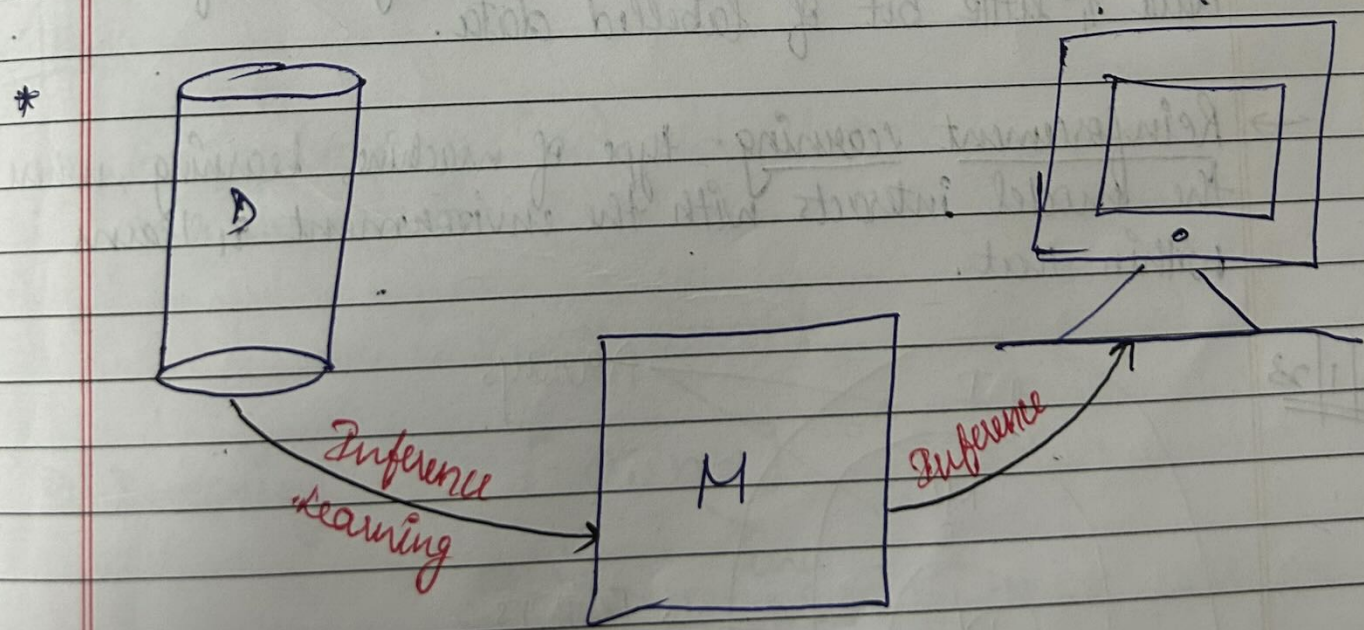
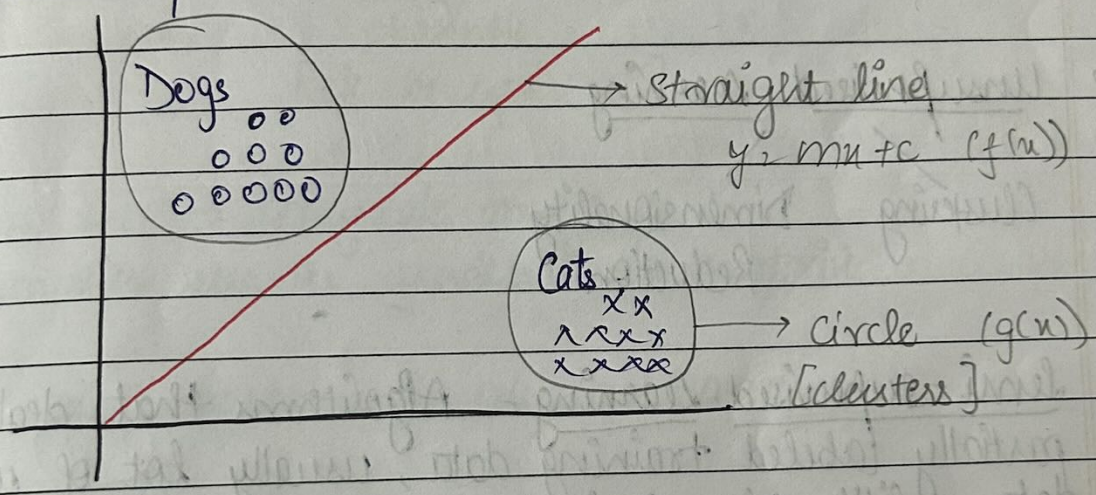


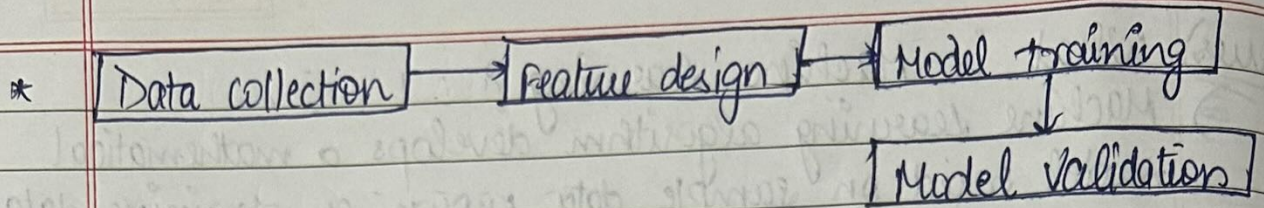
Ques) What is Machine Learning?  
 Machine learning algorithm develops a mathematical model based on sample data known as training data, in order to make predictions or decisions without being explicitly programmed to do so.

\* Large dataset  $\rightarrow$  Deep learning  
 Small dataset  $\rightarrow$  Machine Learning

\* Data Separation  $\rightarrow$  either clusters or.

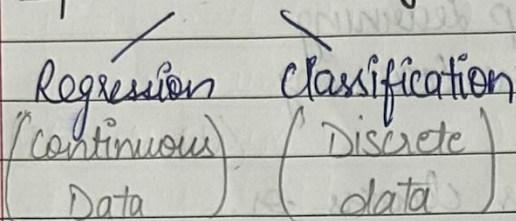




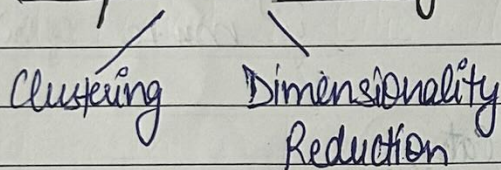


## → Types of Machine Learning Algorithm

### → Supervised Learning



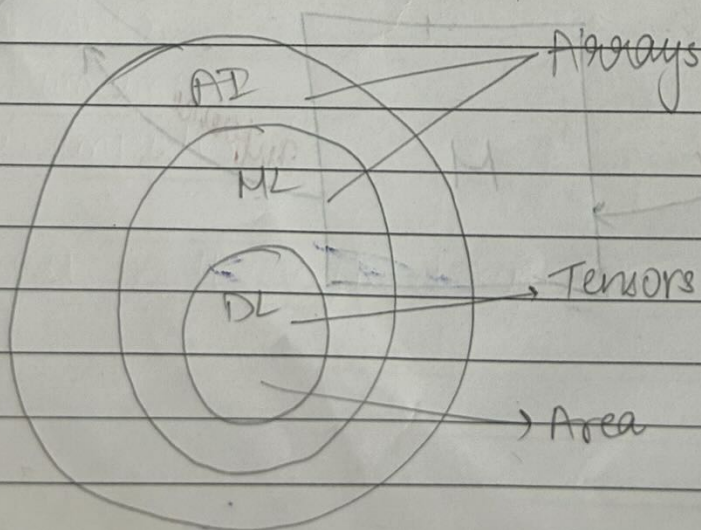
### → Unsupervised Learning



→ Semi-supervised Learning - Algorithms that deal with partially labeled training data, usually lot of unlabelled data & little bit of labelled data.

→ Reinforcement learning - type of machine learning, where the model interacts with the environment & learns within that.

4/1/23





⇒ Element wise ,  $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix}$   
 $\begin{bmatrix} 2 & 4 & 6 \end{bmatrix}$

⇒ np.dot tells 2 vector kithe similar hai

⇒ Images

- Grayscale  $\begin{bmatrix} 0 & 255 \end{bmatrix} [1]$   
2 channels
- Coloured  $\begin{bmatrix} \square & \square & \square \end{bmatrix}$   
3 channels  
[ek ke upar ek]

⇒ we see → grayscale image  
computer sees & understands matrix

⇒ Basically, matrix is also an array.

6/1/23

⇒  $a = \text{np.arange}(15)$  {15 elements}  
 $a.\text{reshape}(3, 5) \mid (5, 3)$   
 product should be no. of elements

⇒ Indexing  $[0 : n-1]$   
 starting value → end value



## → Functions in Numpy:

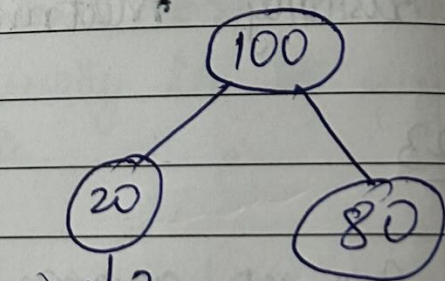
- ⇒ `a.sort()`
- ⇒ `np.square(a)`
- ⇒ `np.abs`
- ⇒ `np.sqrt`
- ⇒ `np.add(a1, a2)`
- ⇒ `np.subtract(a1, a2)`
- ⇒ `np.maximum(a1, a2)`
- ⇒ `np.minimum(a1, a2)`
- ⇒ `np.power(a1, a2)`
- ⇒ `np.greater(a1, a2)` {returns True/False}

## → Mathematical & Statistical Functions in Numpy:

- `np.mean(a)` / `a.mean()` {a can be array as well}

\* Mean, Median, mode,  
ek hi value deta hai

- `a.var()`
- `a.std()`
- `a.sum()`
- `a.max()`
- `a.min()`
- `a.cumsum()`
- `a.cumprod()`



nan?

⇒ 20 values → 0

⇒ 20 values → drop {data is less}

⇒ 20 values → mean

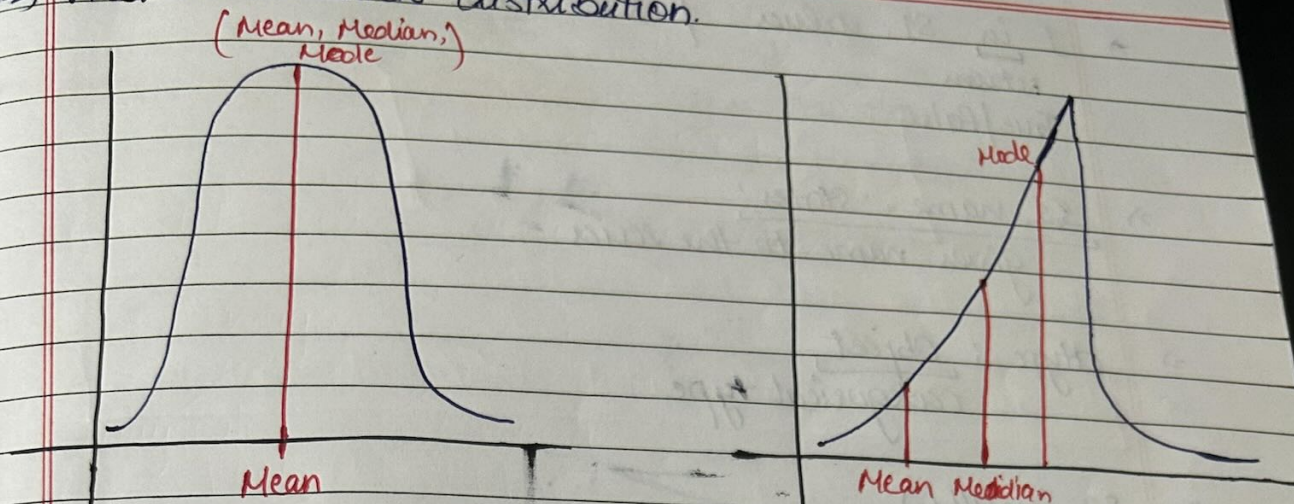
but can't use

prime/median  
why?

→ ~~first mode check distribution~~



⇒ First we'll check distribution.



→ Outlier - koi bhi data ho, ek koi aisi value jo out of box ho, mtlb range se bahar ho, us value ko outlier bolte hai

Eg)  $\frac{1-5}{\text{range}}$  (19) → outlier

Outlier → Genuine outlier  
↳ Fake outlier

⇒ Pandas -

→ dataframe - collection of series

⇒ SL. values

⇒ SL. index / SL. index : ['a', 'b', 'c', 'd']

⇒ Row → iloc, loc  
Column

df['Age'] | df.Age

iloc  
index  
location  
(index)

loc  
location  
(label)



→ 1 in S1, values | S1.index  
returns  
True/False

→ S2.name = 'States'  
gives name to the series

→ dtype : Object  
categorical type

→ S3.isnull()  
checks null value

2) Dataframe

→ pd.DataFrame

→ .head() { default gives first 5 values }  
5/10/15

→ .tail() { default give last 5 values }  
5/10/15