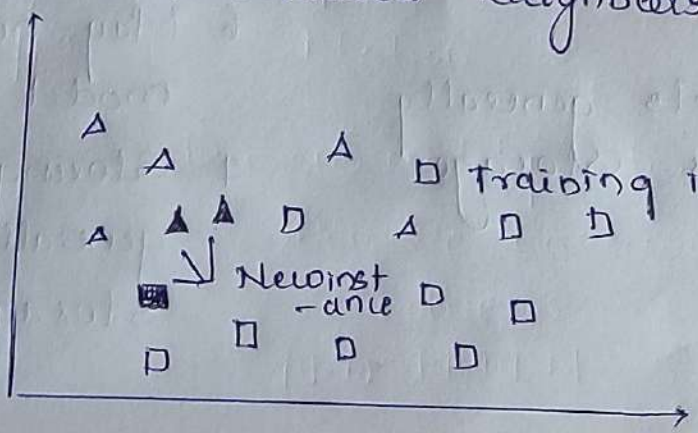
## * Instance Vs Model Based learning

1. Instance based learning :→
(Memorizing)

- It stores training examples & uses them to compare new data points

It follows K-Nearest Neighbours (K-NN)



2. Model Based learning :→

Builds a generalized model from the data to make predictions.

* Difference :-

| Model | Instance |
|---|---|
| 1. prepare the data for model training | 1. prepare the data for model training No difference here. |
| 2. Train model from training data to estimate model parameters i.e discover patterns | 2. Do not train model pattern discovery postponed until scoring query received. |
| 3. Store the model in suitable form | 3. There is no model to store. |
| 4. Generalize the rules in form of model even before scoring instance using model | 4. No generalization before scoring. Only generalize for each scoring instance individually as & when seen |
| 5. predict for unseen scoring instance using model | 5. predict for unseen scoring instance using training data directly |
| 6. Requires a known model form | 6. May not have explicit model form |
| 7. Storing models generally requires less storage | 7. Storing training data generally requires more storage. |

* Challenges in Machine learning :→

1. Data Collection
2. Insufficient data / labelled data
3. Non - representative data
   - Sampling Noise
   - Sampling Bias
4. poor Quality data
5. Irrelevant features
6. Overfitting
7. Underfitting
8. Software Integration
9. offline learning
10. Cost involved

* Machine learning Development lifecycle :→

1. Framing the problem
   Data collection, Algorithm
2. Gathering data
   Kaggle, CSV, API
3. Data preprocessing
   Clean the data
4. Exploratory data Analysis (EDA)
   univariate / Bivariate
5. Feature Engineering & Selection
6. Model training Evalation & Selection
7. Model deployment
8. Testing
9. Optimize

* Tensor

- It is a container that stores number in multiple dimensions

# 0D Tensor / Scalar :→

Holds one number, no dimensions

. Scalar Size is always = 1

2. 1D Tensor / Vector :→

[1, 2, 3, 4] → Vector / 1D array / array / 1D tensor

3 2D Tensor / Matrices :→

[1, 2, 3] [4, 5, 6] [7, 8, 9]

$$\begin{bmatrix} [1,2,3] \\ [4,5,6] \\ [7,8,9] \end{bmatrix} \rightarrow 2D$$

Rank = 2 = ndim

```
mat = np.array([[1,2,3], [4,5,6], [7,8,9]])
mat
o/p = array([[1,2,3],
             [4,5,6],
             [7,8,9]])
mat.ndim
```

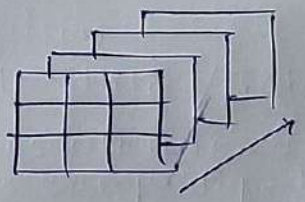
0D :→

```
import numpy as np
arr = np.array([1])
arr
o/p array([1])
arr.ndim
1
```

1D :→

```
arr = np.array([[1,2,3],
                [4,5,6],[7,8,9]])
arr
o/p = array([1,2,3],
            [4,5,6]
            [7,8,9]])
arr.ndim
2
```

## ND tensors :→



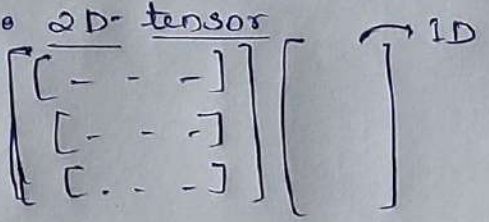
4x3x3

- 1D-tensor

[8.1, 91, 0] → 1D

↳ vector

- 2D- tensor



- 3D - tensor

timeseries data

| | highest | lowest |
|---|---|---|
| Day1 | - | - |
| : | : | : |
| 365 | - | - |

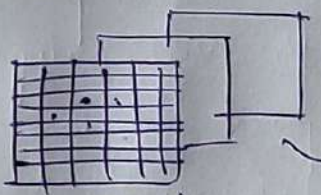
trading

[365, 2]

↳ 10yrs = [10, 365, 2]

↳ 3D

- 4D-tensor



(1200, 800)

(3, 1200, 800) ← 3D

→ If I require 50 color images

(50, 3, 1200, 800)

- 5D - tensor

Videos

60 sec → 4 videos

↓

30 fps

↳ 480p → 480 x 720 (3 channels)

(4, 1800, 480, 720, 3)

↳ 5D