Machine Learning Pipeline

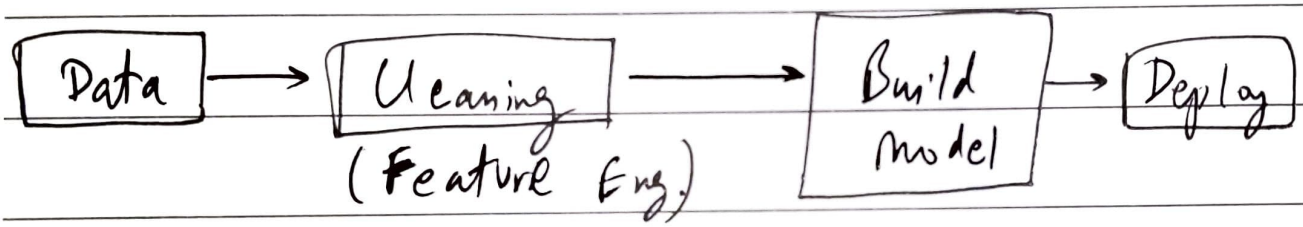
Data ?

↙ Generate

↘ Collect / Scrape

↙ Yourself

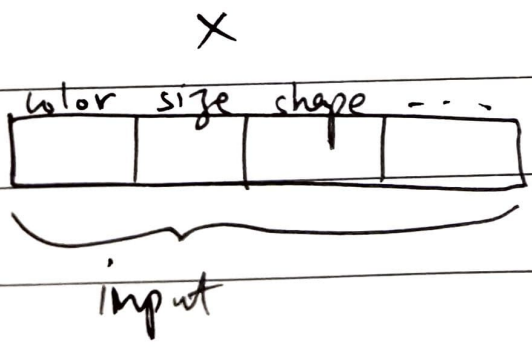
↘ sklearn

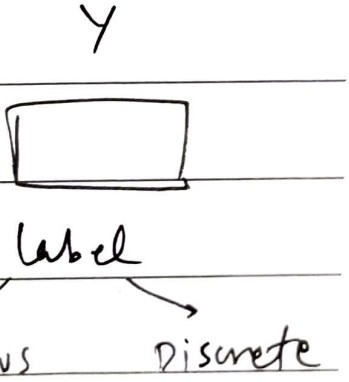


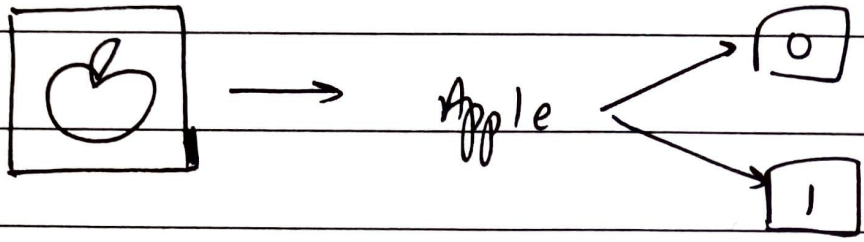
* Supervised learning :

— Learn from data in past (labelled data)

— Generalise predictions on future data and avoid overfitting (This is training)







Representation:

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ & & x_j^{(i)} & & \end{bmatrix}_{m \times n}$$

$x_j^{(i)}$: j^{th} feature of i^{th} example

$$x^{(i)} = < x_1^{(i)} \ldots x_n^{(i)} >$$

m examples & n features

$$y = \begin{bmatrix} \\ y^{(i)} \\ \\ \end{bmatrix}_{m \times 1}$$

Output (Supervised)

Regression

Output is continuous

Ex: Housing, Stock price, CGPA

Role of model is to map a function from $x \rightarrow y$ i.e, $f(x) \rightarrow y$

$x \in R^n$

$y \in R$

Goal: Minimize Loss (Error)

Avoid overfitting (High accuracy in training, low in testing)

Classification

Discrete Value

Ex: Disease or not, Apple or mango or Guava, Object Classification

$f(x) \rightarrow y$

$x \in R^n$

$y \in [1 \ldots k]$

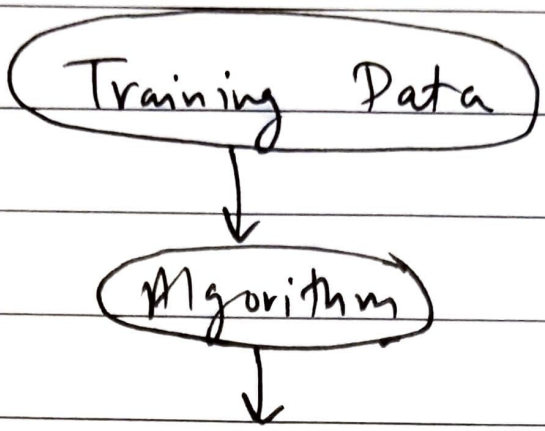
↓

Output is one of the classes

$x^{(i)} \rightarrow$ n features

Goal: Minimise no. of misclassifications

Avoid overfitting



x_{test} (unseen data) → Hypothesis → y_{test} (prediction)

Hypothesis: Function having some parameters which will map input to desired output.

Multi label Classification:

Say you are given a text about mars Rover, it can have labels such as Science, Tech, etc. i.e. more than one labels. You get multiple outputs for same class. i.e. multiple outputs.

Examples of Supervised learning:

→ Spam Detection

→ Document Classification (multilabel)

→ NLP

→ Image Classification

→ Sequence (music / Text / Speech)

→ Predictive.

---

* Unsupervised Learning:

- No supervision
- Algorithm finds pattern, ex: Clustering (grouping similar data)
- No labelled data (only X, no y)

Semi-supervised: we manually label few data points.

Ex:
Say you 10 mil tweets, labeling is difficult. But if you label some data yourself, it can label others now.

Applications of Unsupervised Learning:

→ object segmentation

→ Similarity detection

→ Cluster Automatic labeling

★ Reinforcement Learning :

[Computer Games]

- Learns how humans learn (like riding a bicycle)
- Hit & trial method is used for learning
- Person who rides the bike is called Agent

In programs it is an object which interacts with the environment

- Environment : Agent interacts with it, a simulation, can be a game, etc.
- Goal : You will have agent and environment, environment has a state

ex of state: speed, x, y position, etc.

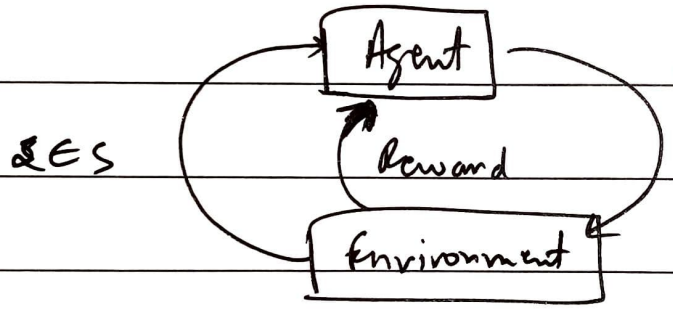
Goal of agent is to interact with environment by taking some action a.

$a \in A$ [A: set of actions]

Like ex: You can move, jump, etc.

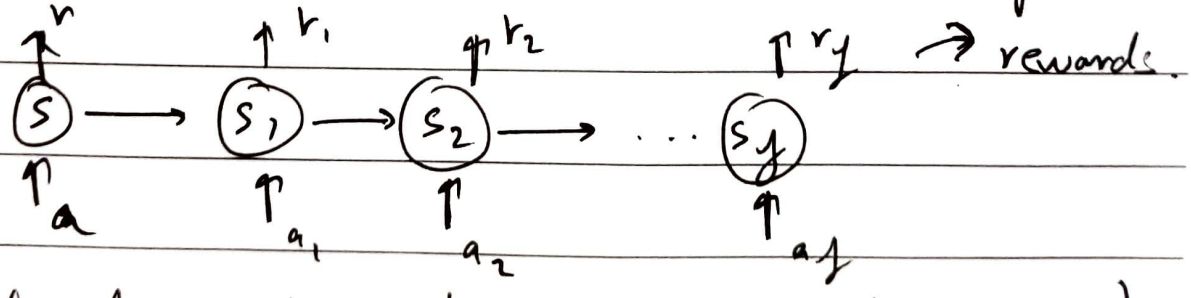
Now, environment reaches state S, $S \in S$

Each state gives you reward.



So, the agent is going to take some action, on the environment and environment will reach state S_1, repeat till final state (Game Over).

Game over : Killed or Game Complete



(Goal of agent is to maximise total reward)

∴ we need to design a policy / algorithm that helps in getting maximum reward.

Game Policy:

Ex: for racing game, car can go left(1) or right(0)

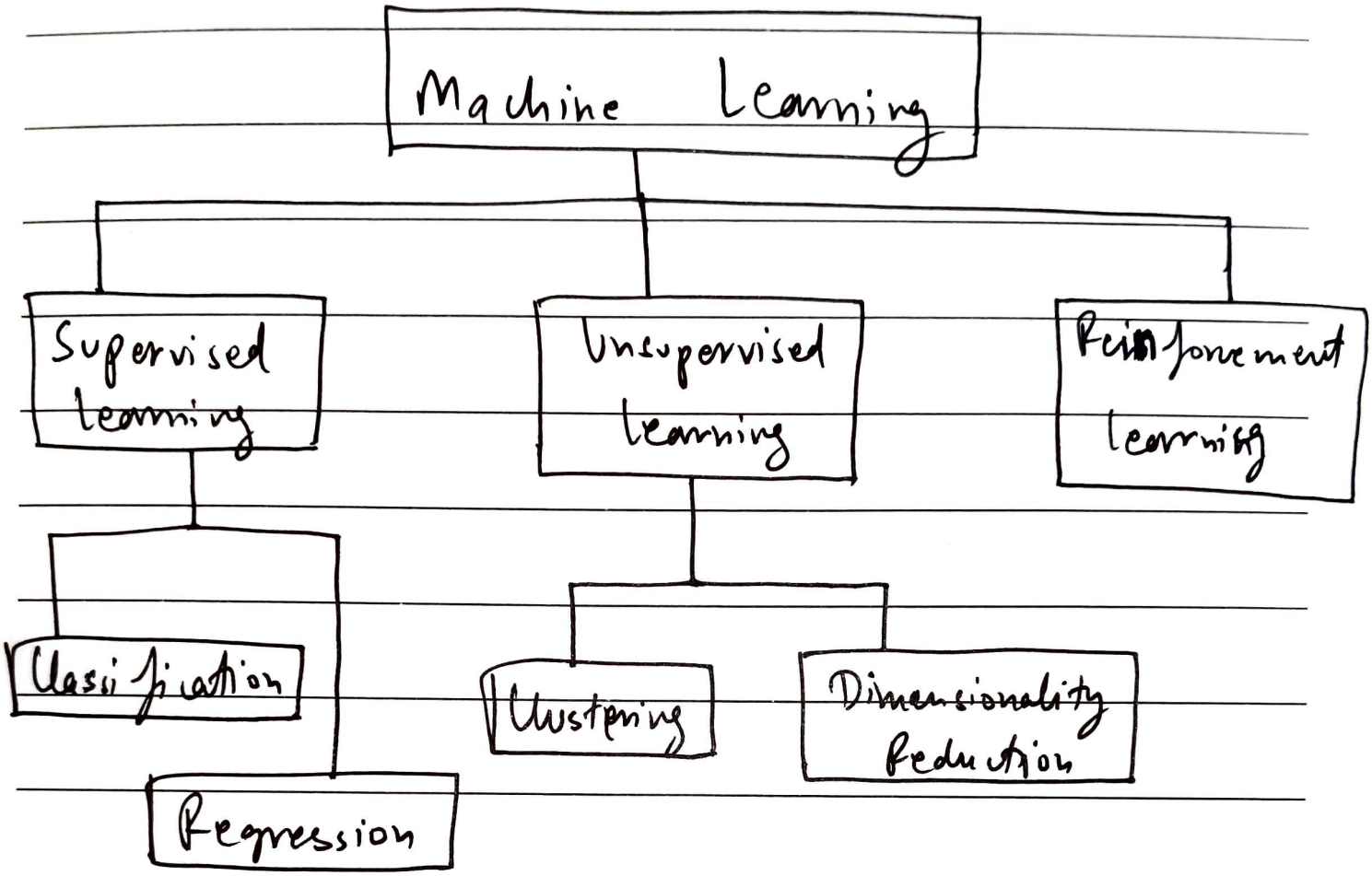
Find Probability of ~~right~~ Correct action A:

$P(A/s)$ = Posterior Probability

↓

** Best action to take in given state S.

* Method / Algorithm to get best possible action, one example for this is, Deep Q-learning.


Machine Learning
Supervised learning
Unsupervised learning
Reinforcement learning
Classification
Regression
Clustering
Dimensionality Reduction