

Q1- Machine Learning -

- sub field of AI concerns with developing computational theories of learning & building learning machines.
- imitate the way that human learn, gradually improve accuracy (use of data & algo).

Q2

Imp of ML:

- 1) Data Driven Decision Making.
- 2) Automation
- 3) Personalization (User experiences)
- 4) Improved Accuracy -
- 5) Cost Reduction.
- 6) Healthcare Advances.
- 7) Accessibility
- 8) Scientific Discovery.
- 9) Economic Growth
- 10) Education & Skill Development.



Q3 ML process (steps)

- ① Data acquisition
- ② Data processing
- ③ Model engineering
- ④ Execution
- ⑤ Deployment

Q4 Appl<sup>n</sup> of ML.

- ① optical character recognition
- ② Face detection.

- ③ spam filtering
- ④ Spoken Language Understanding
- ⑤ UBER
- ⑥ Healthcare Industry

Q6 Supervised learn

Q7 Q8 Q9 Q10 Q11 Q12

Q17 Q18

supervised & Unsupervised.  
(Next)

Models of ML

→ Geometric

2) Probabilistic

3) Logical Models.

→ Naive Bayes Classifier



## Supervised

- Input Data - labelled
- predict a specify quantity / outcome / <sup>predict</sup>
- Fraud detection.
- subtype - classification & regression
- + higher accuracy
- more time consuming
- has feedback mechanism
- we can test our model
- Desired output is given
- Algorithms in  $\rightarrow$  .  
decision trees,  
logistic regressions,  
support vector machine.

## Unsupervised.

- unlabelled.
- Understand data <sub>insight</sub>
- Identify Managem
- clustering & association.
- Lower accuracy
- less time consuming.
- NO feedback mechanism -
- we can't test model
- Desired output is not given.
- Algorithms -  
• k-means clustering  
• hierarchical clustering  
• apriori algorithms



# Q13 - Reinforcement Learning

- science of decision making
- learning optimal behavior in env<sup>i</sup> → obtain max reward
- Data accumulated → use trial & error method.
- Points in RL

\* Input → initial state from which model will start.

\* Output → many possible outputs are a variety of outputs / sol<sup>n</sup> for particular problem.

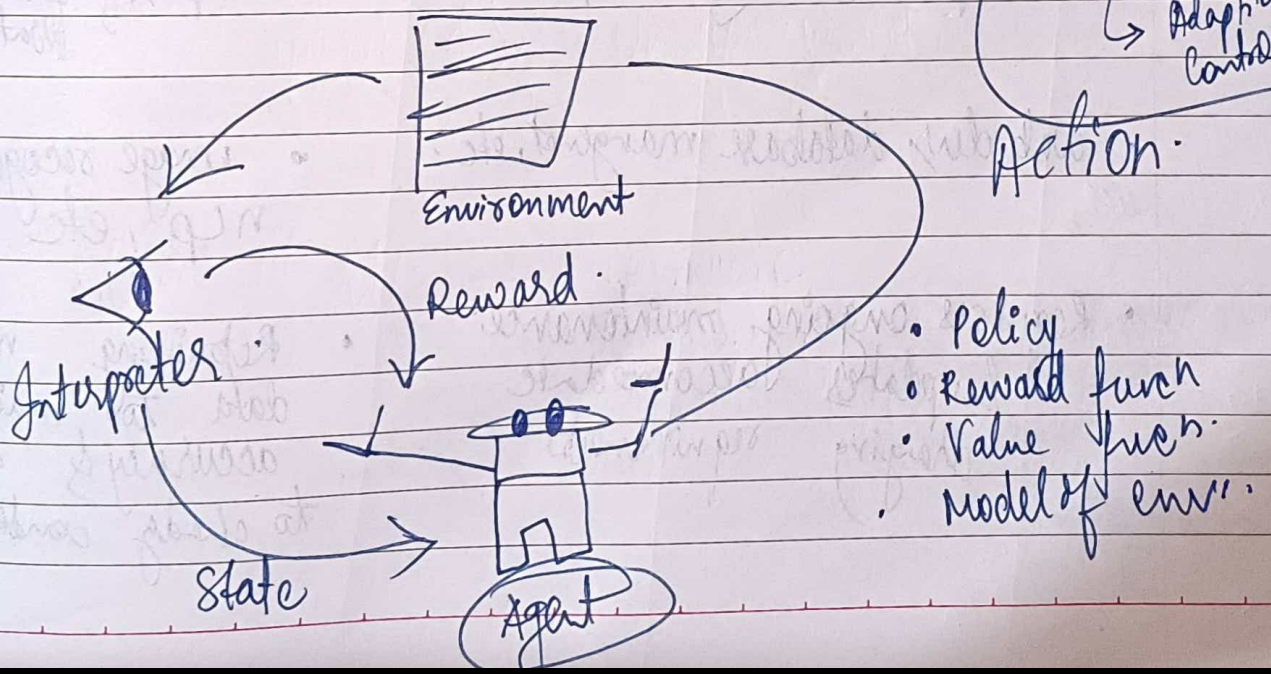
\* Training - Based on input, Model returns a state, user decides to reward or punish model based on its output.

• Model keeps learning

• Best sol<sup>n</sup> → Maximum Rewards.

App<sup>n</sup> → Robots  
 → Chess  
 → Adaptive Controller.

Action.





Q16

## Tradition programming

- developers → write algo & rules to solve
- Behaviour determined → predefined logic & rules
- Doesn't rely on data.
- manually write rule & instructions
- static & require code modification.
- Human programmes design & implement
- Code is transparent & easily interpretable
- web dev, database management, etc.
- Requires ongoing maintenance & updates to accommodate changing requirements

## Machine learning

- Learns from data
- discovers patterns, correlations & rules from data to make decisions
- Rely on data. (Large DS for training)
- automatic generation of rules & logic
- Ideal for problem with complex patterns, uncertainty, etc.
- Human required in data preparation, feature engineering, etc.
- learning is automated
- Complex & difficult to interpret
- image recognition, NLP, etc.
- Requiring new data to maintain accuracy & adapt to changing conditions



Q18

## Parametric

- learner  $\rightarrow$  summarized data through collection of parameters.
- Test group means
- need larger samples
- info  $\rightarrow$  popul<sup>n</sup>  $\rightarrow$  known
- Interval & ratio scale data
- Samples are independent.
- eg  $\rightarrow$  logistic regression  
SVM

## Non-Parametric

- no assumptions or mapping func<sup>n</sup>.
- Test group medians
- small samples.
- info  $\rightarrow$  popul<sup>n</sup>  $\rightarrow$  unknown
- Ordinal & nominal data.
- not necessary  $\rightarrow$  sample  $\rightarrow$  indep<sup>dat</sup>
- eg  $\rightarrow$  K-nearest neighbors.

Q19

## Imp elements of ML:

- 1) Training
  - 2) Validation
  - 3) Appl<sup>n</sup>.
- } Test

GPT

- 1) Data
- 2) Features
- 3) Algorithms
- 4) Model
- 5) Training
- 6) Validation
- 7) Hyperparameters
- 8) Loss func<sup>n</sup>
- 9) Testing & evaluation
- 10) Deployment



## ML

