Welcome to Artifical Intelligence

Preparation notes for final term exams.

I'm not responsible for anything!

What is AI?

The study and design of computing systems that perceives its environment and takes actions like human beings.
The term was introduced by John McCarthy in `1956` in the well-known Dartmouth Conference.

Difference b/w Intelligence & AI?

Intelligence: The ability to learn and solve problems.
Artifical Intelligence: is the intelligence exhibited by machines or software.

Four Schools of Thoughts

Thinking humanly	Thinking rationally	Acting humanly	Acting rationally
The exciting new effort to make computers think.	The study of mental faculties through the use of computational models.	The study of how to make computers do things which, at the moment, people are better.	Computational Intelligence is the study of the design of intelligent agents.

how to make a program to think like a human?
We need to get inside the actual workings of human minds. There are three ways:

1. **Introspection:** trying to catch our own thoughts as they go by;
2. **Psychological Experiments:** observing a person in action.
3. **Brain Imaging:** observing the brain in action.

Turning Test AI?

A Turing Test is a method of inquiry in artificial intelligence (AI) for determining whether or not a computer is capable of thinking like a human being.

The original Turing Test requires `3` terminals, each of which is physically separated from the other two. One terminal is operated by a computer, while the other two are operated by humans.

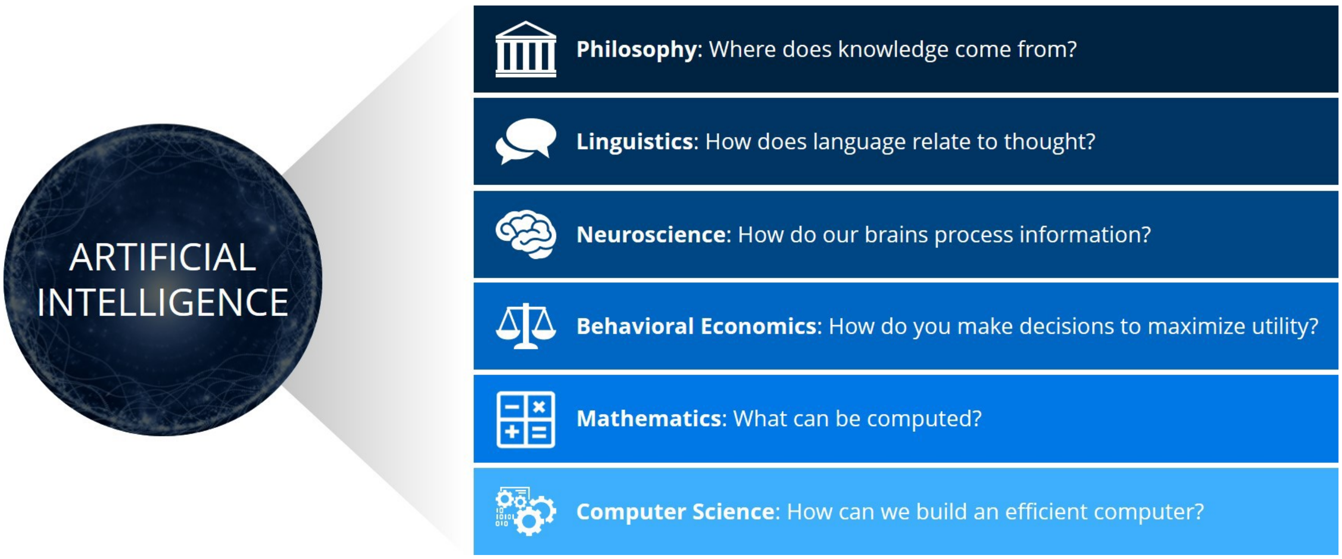
When to Stop Turning Test:

The test is repeated many times. If the questioner makes the correct determination in half of the test runs or less.

Requirements to Pass Turing Test:

Natural Language Processing: to enable it to communicate successfully in English.
Knowledge Representation: to store what it knows or hears.
Automated Reasoning: to use the stored information to answer questions and to draw new conclusions.
Machine Learning: to adapt to new circumstances and to detect and extrapolate patterns.
Computer Vision: to perceive objects.
Robotics: to manipulate objects and move about.

Foundation of Artificial Intelligence



Pros & Cons of AI

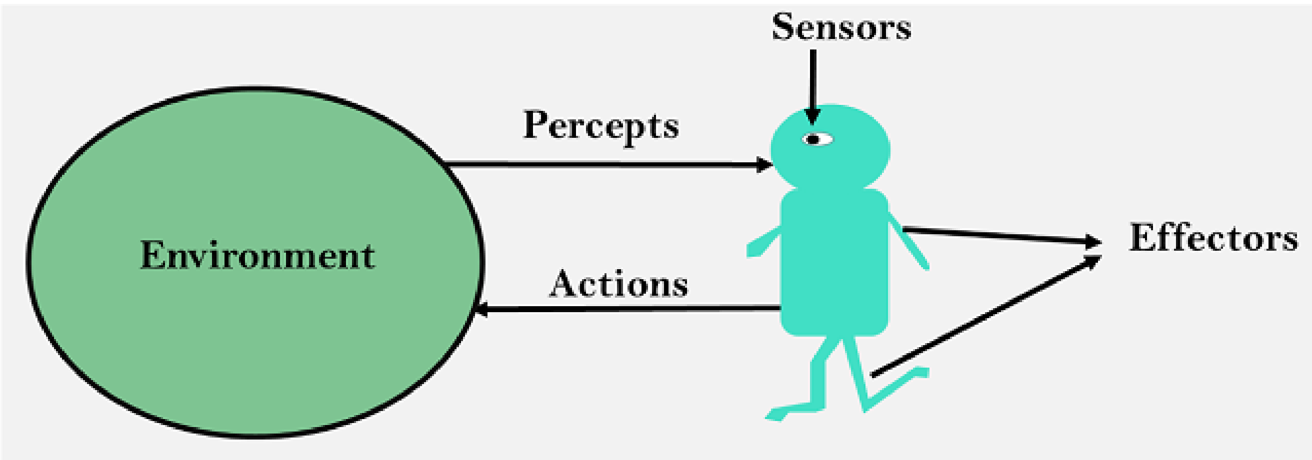
Pros	Cons
Reduction in Human Error	High Costs of Creation
Takes risks instead of Humans	Making Humans Lazy
Available 24x7	Unemployment
Helping in Repetitive Jobs	No Emotions
Digital Assistance	Lacking Out of Box Thinking
Faster Decisions	--

What is Agents?

Artificial intelligence is defined as a study of rational agents. A rational agent could be anything which makes decisions, as a person, firm, machine, or software. It carries out an action with the best outcome after considering past and current percepts(agent's perceptual inputs at a given instance).

An AI system is composed of an agent and its environment. The "agents" act in their environment.
The "environment" may contain other agents. An agent is anything that can be viewed as :

- perceiving its environment through sensors
- acting upon that environment through actuators



Human & Robotic Sensor & Acutator

Agent	Sensor	Acutator
Human Agent	Eyes, Ear, & other organs.	Hands, legs, mouth, & other body parts.
Robotic Agent	Camera, Recorder, & infrared range finder.	Various motors.
Software Agent	Keystrokes, File contents, Received Network Packages.	Screen, Files, Sent Network Packets.

Structure of Intelligent Agent

`Agent = Architecture + Agent Program`

Architecture: is the machinery that the agent executes on. It is a device with sensors and actuators, for example : a robotic car, a camera, a PC.
Agent Program: is an implementation of an agent function.

Agent Terminology

- **Performance Measure of Agent:** It is the criteria, which determines how successful an agent is.
- **Behavior of Agent:** It is the action that agent performs after any given sequence of percepts.
- **Percept:** It is agent's perceptual inputs at a given instance.
- **Percept Sequence:** It is the history of all that an agent has perceived till date.
- **Agent Function:** It is a map from the precept sequence to an action.

What is Rationality?

Rationality is nothing but status of being **reasonable**, **sensible**, and having good sense of **judgment**.

Rationality is concerned with expected actions and results depending upon what the agent has perceived. Performing actions with the aim of obtaining useful information is an important part of rationality.

What is Ideal Rational Agent?

Ideal Rational Agent is capable of doing expected actions to maximize its performance measure, on the basis of −

- Its percept sequence
- Its built-in knowledge base

Rationality of an agent depends on the following −

- The performance measure, which determine the degree of success.
- Agent’s Percept Sequence till now.
- The agent’s prior knowledge about the environment.
- The actions that the agent can carry out.

Rationality Vs Omniscience

Rationality	Omniscience
all knowing with infinite knowledge	only knows the actual outcome of its actions and can act accordingly

Rationality Vs Perfection

Rationality	Perfection
maximizes expected performance	maximizes actual performance

Proposed definition required

- Information Gathering
  - Agents can perform actions in order to modify future percepts so as to obtain useful information.
- Learning
  - An agent can also learn from what it perceives.
- Autonomy
  - An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt).

PEAS

Specifying the task environment is always the first step in designing agent.
`Performance, Environment, Actuators, Sensors`

Taxi Driver Example

Performance	Environment	Actuators	Sensors
safe, fast, legal, comfortable trip	roads, other traffic, pedestrians, customers	steering, accelerator, brake, signal, horn, display	camera, sonar, speedometer, GPS, odometer, engine sensors, keyboard, accelerator

Medical Diagnosis System Example

Performance	Environment	Actuators	Sensors
healthy patient, minimize costs, lawsuits	patient, hospital, staff	display questions, tests, diagnosis, treatments, referrals	keyboard entry of symptoms, findings, patient’s answers

Part Picking Robot Example

Performance	Environment	Actuators	Sensors
Percentage of parts in correct bins	Conveyer belt with parts, bins	Jointed arm and hand	camera, joint angle sensors

Properties of Task Environment

Single-Agent Vs Multi-Agent

Single-Agent	Multi-Agent
environments are those on which a finite set of possibilities can drive the final outcome of the task.	Environments rely on unknown and rapidly changing data sources.

Fully Observable Vs Partial Observable

Fully Observable	Partial Observable
if the sensors can always see the entire state of the environment.	if the sensors can see only a fraction of the state. Partial Observable agent needs memory.

Deterministic Vs Stochastic

Deterministic	Strategic	Stochastic
Where your agent's actions uniquely determined the outcome.	Environment is deterministic except for the actions of other agents.	We can't predict the outcome.

Episodic Vs Sequential

Episodic	Sequential
The agent's experience is divided into atomic "episodes".	The current decision could affect all future decisions.
Each episode consists of the agent perceiving and then performing a single action.	--
The choice of action in each episode depends only on the episode itself.	--

Static Vs Dynamic Vs Semi-Dynamic

Static	Dynamic	Semi-Dynamic
Environment cannot change while an agent is deliberating	Environment can change while an agent is deliberating	If the environment itself doesn't change with time but the agent's performance score does.

Discrete Vs Continuous

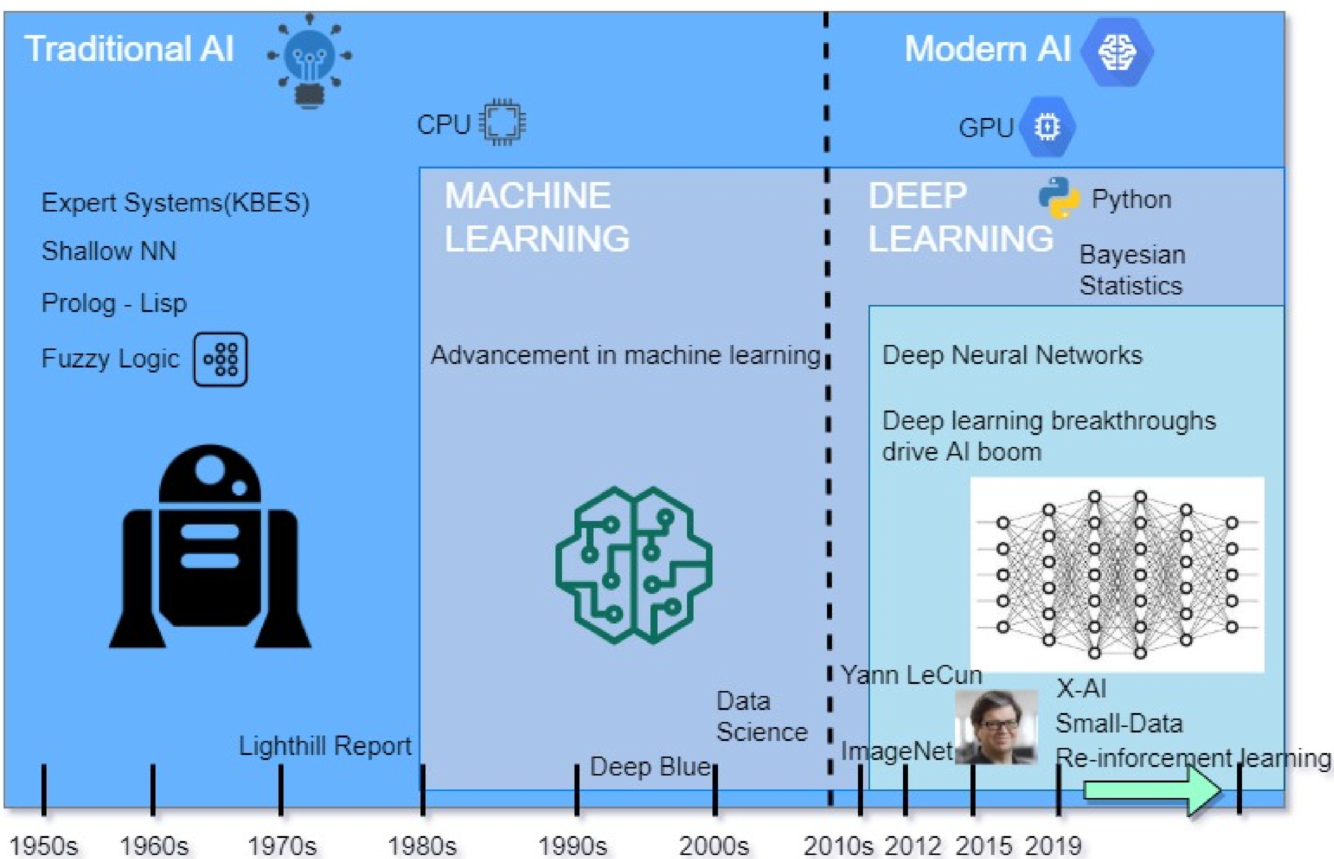
Discrete	Continuous
Environments are those on which a finite set of possibilities can drive the final outcome of the task.	Environments rely on unknown and rapidly changing data sources.

Example Task Environment Table

Task Environment	Observable	Determines	Episodic	Static	Discrete	Agent
Crossword Puzzle	Fully	Determines	Sequential	Static	Discrete	Single
Chess with clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Con	Multi
Medical Diagnosis	Partially	Stochastic	Sequential	Dynamic	Con	Single
Image Analysis	Fully	Deterministic	Episodic	Semi	Con	Single
Part pick mg robot	Partially	Stochastic	Episodic	Dynamic	Con	Single
Refing controller	Partially	Stochastic	Sequential	Dynamic	Con	Single
Interstice English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Classical & Modern AI

- **Classical AI**
  - Classic AI consists mainly in inference engines used for rule based systems. The main goal was to map "human reason" and "process data" with semantic content. The main early results were fraud detection, medical diagnose and automating expert knowledge about specific domains.
- **Modern AI**
  - Modern AI "Deep Learning" encloses also computational intelligence. Neural Networks, Genetic Algorithms, Sworm Algorithms and Fuzzy Logic are the pillars of computational intelligence.



First-Order Logic (FOL)

- FOL is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- FOL is also known as Predicate logic or First-order predicate logic. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.
- FOL (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world
    - Objects: A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, etc.
    - Relations:
        - It can be unary relation such as: red, round, is adjacent,
        - n-any relation such as: the sister of, brother of, has color, comes between
    - Function: Father of, best friend, third inning of, end of, etc.

As a natural language, first-order logic also has two main parts:

1. Syntax
2. Semantics

Syntax of First-Order logic

The syntax of FOL determines which collection of symbols is a logical expression in FOL. The basic syntactic elements of FOL are symbols. We write statements in short-hand notation in FOL.

Basic Elements of First-order logic

Constant	1, 2, A, John, Mumbai, cat,
Variables	x, y, z, a, b,
Predicates	Brother, Father, Uncle,
Function	sqrt, LeftLegOf, pow,
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀, ∃

Atomic Sentence

- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as `Predicate (term1, term2, ......, term n)`.

Example:
```
Fahan and Nauman are brothers: => Brothers(Farhan, Nauman).
Chinky is a cat: => cat (Chinky).
```

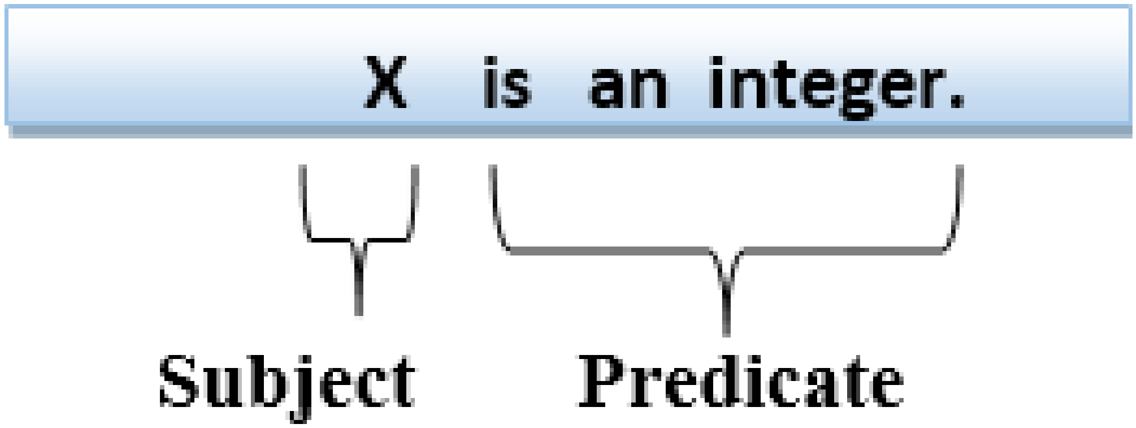
Complex Sentence

- Complex sentences are made by combining atomic sentences using connectives.

First-order logic statements can be divided into two parts:

1. Subject: Subject is the main part of the statement.
2. Predicate: A predicate can be defined as a relation, which binds two atoms together in a statement.

Consider the statement: "x is an integer." it consists of two parts, the first part `x` is the subject of the statement and second part `is an integer`, is known as a predicate.



Quantifiers in First-order logic:

- A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.

There are two types of quantifier:

1. Universal Quantifier: for all, everyone, everything.
2. Existential Quantifier: for some, at least one.

Universal Quantifier:

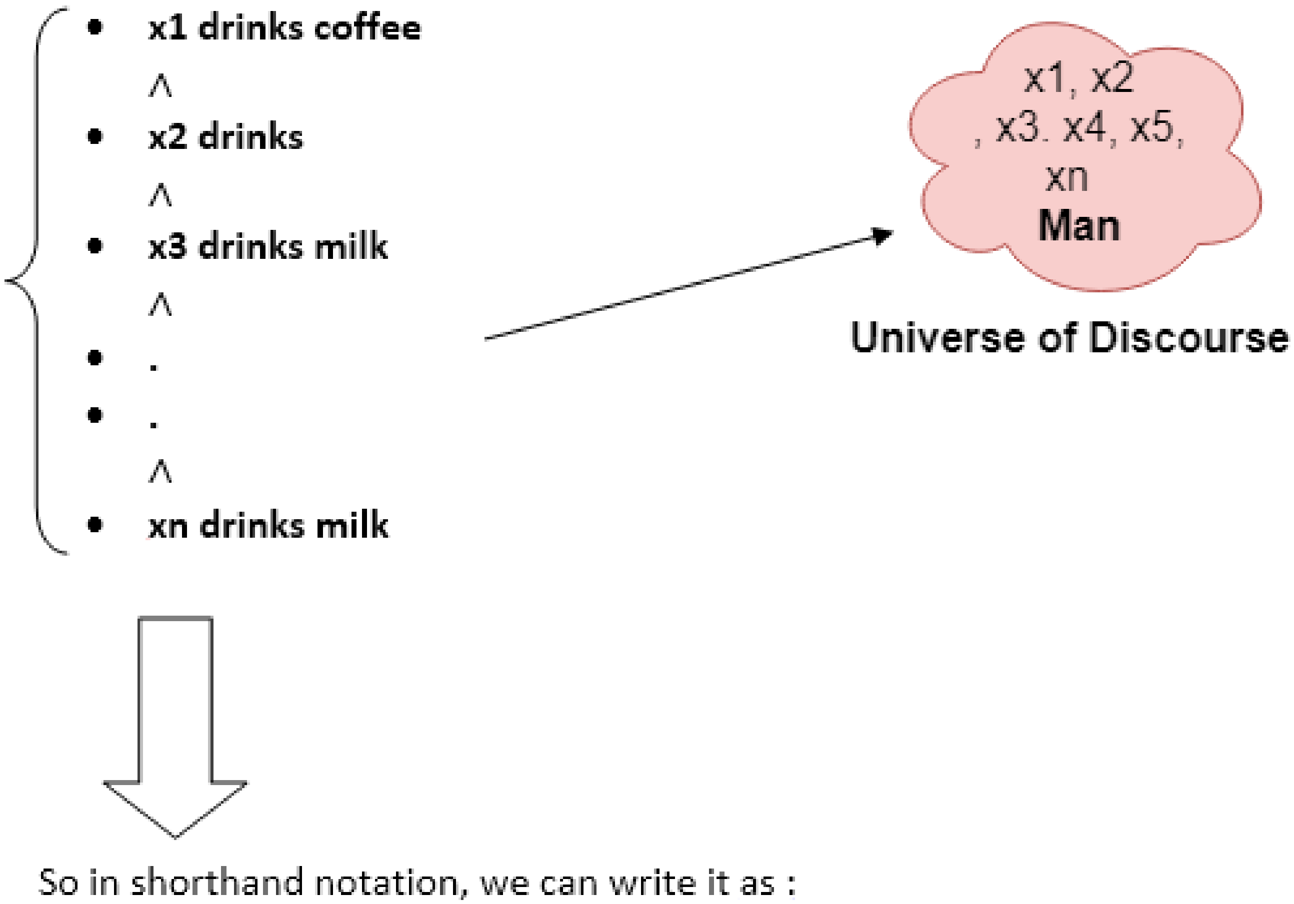
Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

If x is a variable, then ∀x is read as:

- For all x
- For each x
- For every x

Example:
All man drink coffee:

Let a variable x which refers to a cat so all x can be represented in UOD as below:



`∀x man(x) → drink (x, coffee).`
It will be read as: There are all x where x is a man who drink coffee.

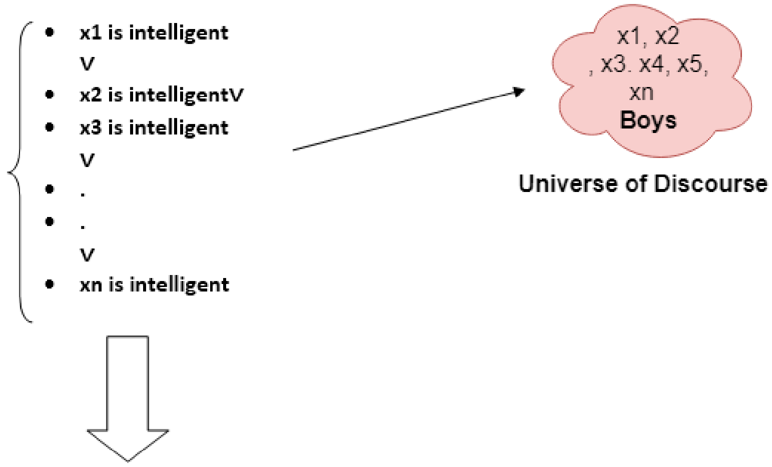
Existential Quantifier:

Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.
It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

- There exists a 'x.'
- For some 'x.'
- For at least one 'x.'

Example:
Some boys are intelligent.



So in short-hand notation, we can write it as:

`∃x: boys(x) ∧ intelligent(x)`
It will be read as: There are some x where x is a boy who is intelligent.

Points to remember:

- The main connective for universal quantifier ∀ is implication →.
- The main connective for existential quantifier ∃ is and ∧.

Properties of Quantifiers:

- In universal quantifier, ∀x∀y is similar to ∀y∀x.
- In Existential quantifier, ∃x∃y is similar to ∃y∃x.
- ∃x∀y is not similar to ∀y∃x.

Some examples FOL using quantifier:

1. All birds fly.
   In this question the predicate is "fly(bird)."
   And since there are all birds who fly so it will be represented as follows.
   `∀x bird(x) → fly(x).`

2. Every man respects his parent.
   In this question, the predicate is "respect(x, y)," where x=man, and y= parent.
   Since there is every man so will use ∀, and it will be represented as follows:
   `∀x man(x) → respects (x, parent).`

3. Some boys play cricket.
   In this question, the predicate is "play(x, y)," where x= boys, and y= game. Since there are some boys so we will use ∃, and it will be represented as: `∃x boys(x) → play(x, cricket).`

Data Types?

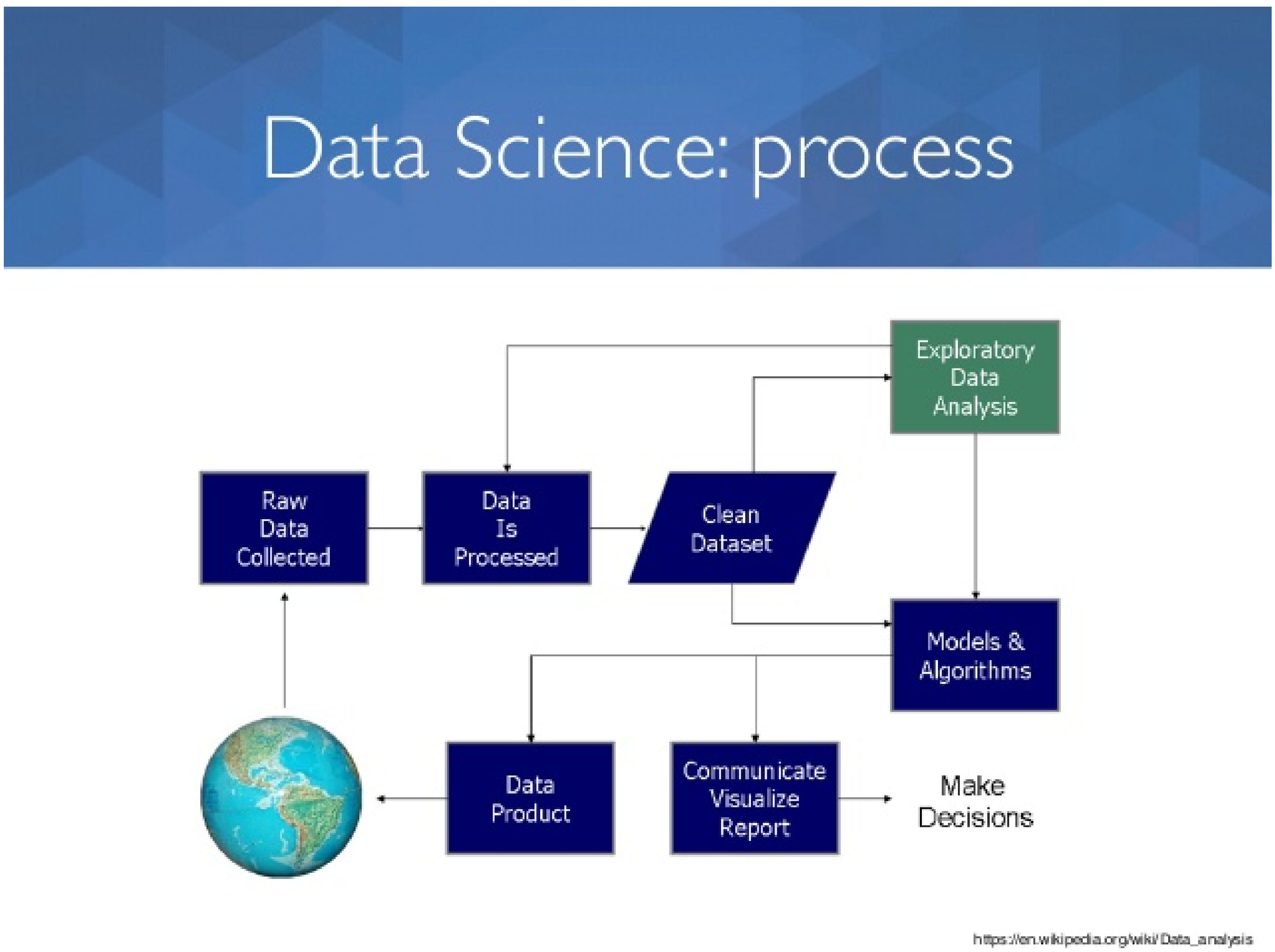
- Text
- Numbers
- Images
- Audios
- Videos
- Tables
- Graphs
- Transactions

DataFied?

Wherever we go, we are “datafied"

- Smartphones are tracking our locations.
- We leave a data trail in our web browsing.
- Interaction in social networks.

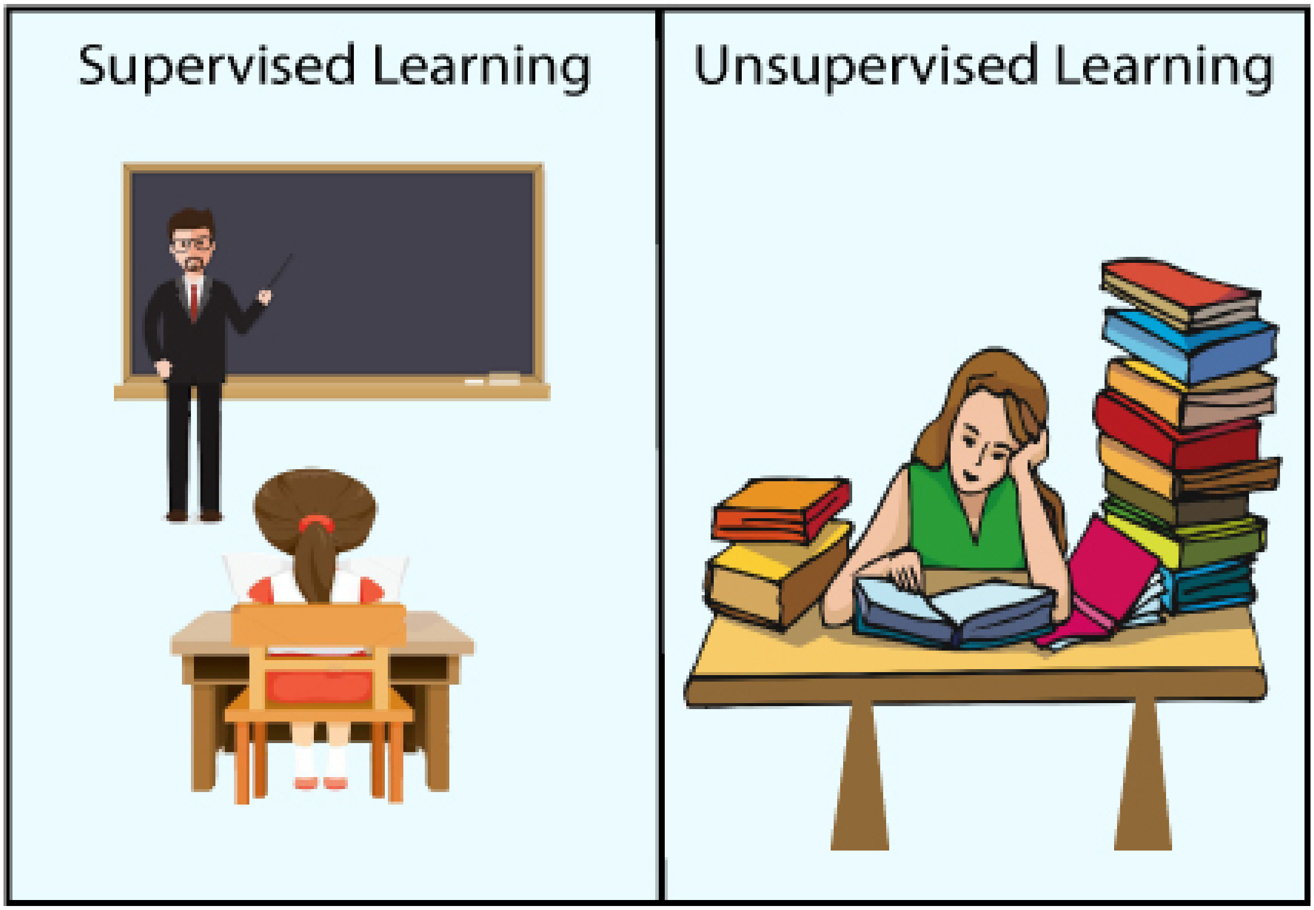
Data Science Process



Machine Learning Vs Statistics

Machine Learning	**Statistical**
Decision trees	Hypothesis testing
Rule induction	Experimental design
Neural Networks	Linear regression
Clustering method	Logistic regression
Association rules	--
Visualization	--
Genetic algorithm	--
Graphical models	--

Supervised & Unsupervised



What is Supervised Machine Learning?

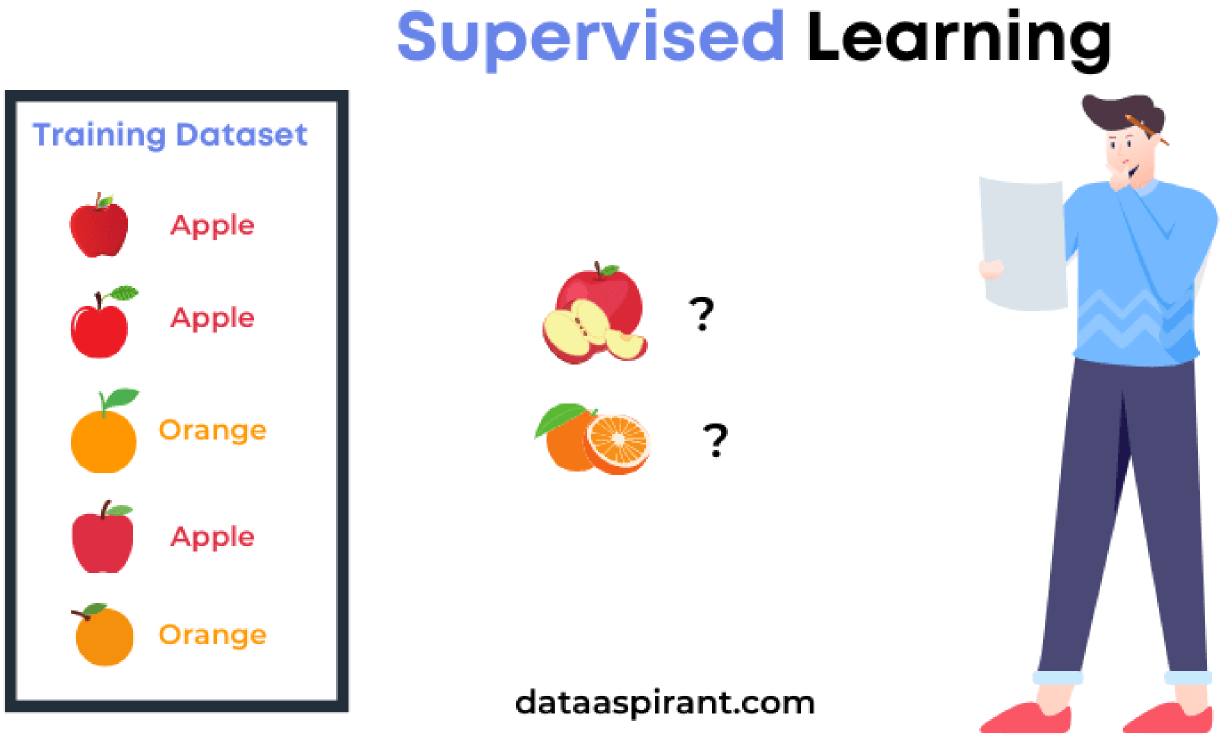
In Supervised learning, you train the machine using data which is well "labeled." It means some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher.

What is Unsupervised Machine Learning?

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.

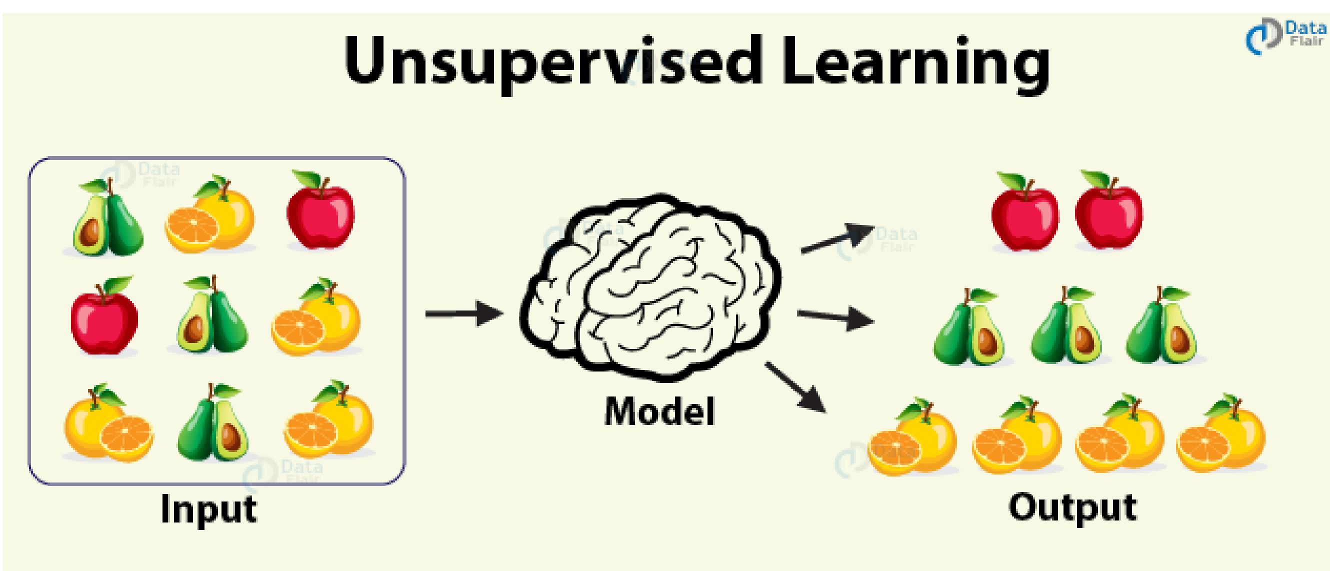
Supervised Machine Learning Techniques

- Regression
    - Regression technique predicts a single output value using training data.
- Classification
    - Classification means to group the output inside a class. If the algorithm tries to label input into two distinct classes, it is called binary classification. Selecting between more than two classes is referred to as multiclass classification.



Unsupervised Machine Learning Techniques

- Clustering
  - Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data.

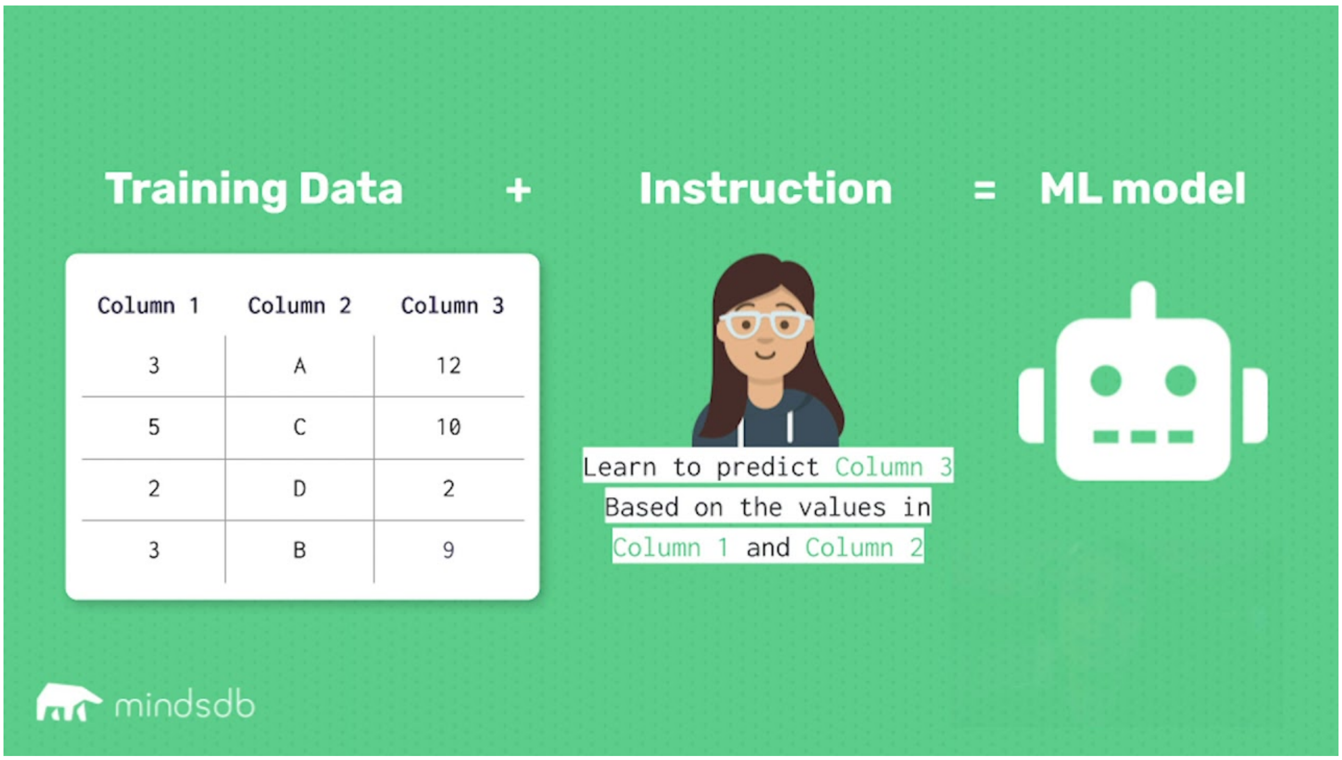


Supervised Vs Unsupervised

Parameters	Supervised	Unsupervised
Process	In a supervised learning model, input and output variables will be given	In unsupervised learning model, only input data will be given
Input Data	Algorithms are trained using labeled data	Algorithms are used against data which is not labeled

Algorithms Used	Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees	Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc
Computational Complexity	Supervised learning is a simpler method	Unsupervised learning is computationally complex
Accuracy of Results	Highly accurate and trustworthy method	Less accurate and trustworthy method.
Real Time Learning	Learning method takes place offline	Learning method takes place in real time
Main Drawback	Classifying big data can be a real challenge in Supervised Learning.	You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known.

Training & Testing in ML



K-Nearest Neighbor(KNN) Algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

How does KNN work?

- Step-1: Select the number K of the neighbors.
- Step-2: Calculate the Euclidean distance of K number of neighbors.
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

Select Value of K in the KNN Algorithm

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Pros Vs Cons of KNN Algorithm

Pros	Cons
It is simple to implement.	Always needs to determine the value of K which may be complex some time.
It is robust to the noisy training data	The computation cost is high because of calculating the distance between the data points for all the training samples.
It can be more effective if the training data is large.	--