ARTIFICIAL INTELLIGENCE & EXPERT SYSTEMS

CSE-321 || Lecture-02

CONTENT

- Knowledge
- Intelligent Agent
- Specifying the Task Environment
  Types of Agent
- supervised and unsupervised learning
- Classification

KNOWLEDGE

- Can be defined as the body of facts & principles accumulated by humankind or the act, fact, or state of knowing.
- It is having a familiarity with **language, concepts, procedures, rules, ideas, abstractions, places, customs, facts, & associations**, coupled with an ability to use these notions effectively in modeling different aspects of the world.
- The meaning of knowledge is closely related to the meaning of intelligence.
- A characteristic of intelligent people is that they posses much knowledge.
- Knowledge is likely stored as **complex structures of interconnected neurons**.

KNOWLEDGE

Declarative vs. procedural

Procedural: **Compiled** knowledge related to the performance of **some tasks**

- The steps used to solve an algebraic equation

Declarative: passive knowledge expressed as statements of facts about the world.

- Personal data in a database

Heuristic Knowledge

Special type of knowledge used by humans to solve complex problems.

- The knowledge used to make good judgments, or strategies, tricks, or ‘rules of thumb’ used to simply the solution of problems.
- Heuristics are usually acquired with much experience
- Fault in a TV set
- an experienced technician will not start by making numerous voltage checks when it is clear that the sound is present but the picture is not
- The high voltage fly back transformer or related component is the culprit
- **May not always be correct**
- But frequently/quickly can find a solution

KNOWLEDGE AND DATA

- Knowledge should not be confused with data
- **Physician treating a patient use both Knowledge & Data**
- **Data:** record: history, measurement of vital sign, drugs given, response to drugs,……
- **Knowledge:** what Physician learned from medical school, internship, residency, specialization, practice.
- Knowledge includes & requires the use of data & information
- **It combines relationship, correlations, dependencies, & notion of gestalt with data & information**

Belief, Hypothesis, & Knowledge

- **Belief:** define as essentially **any meaningful & coherent** expression that can be represented

  It may be true or false

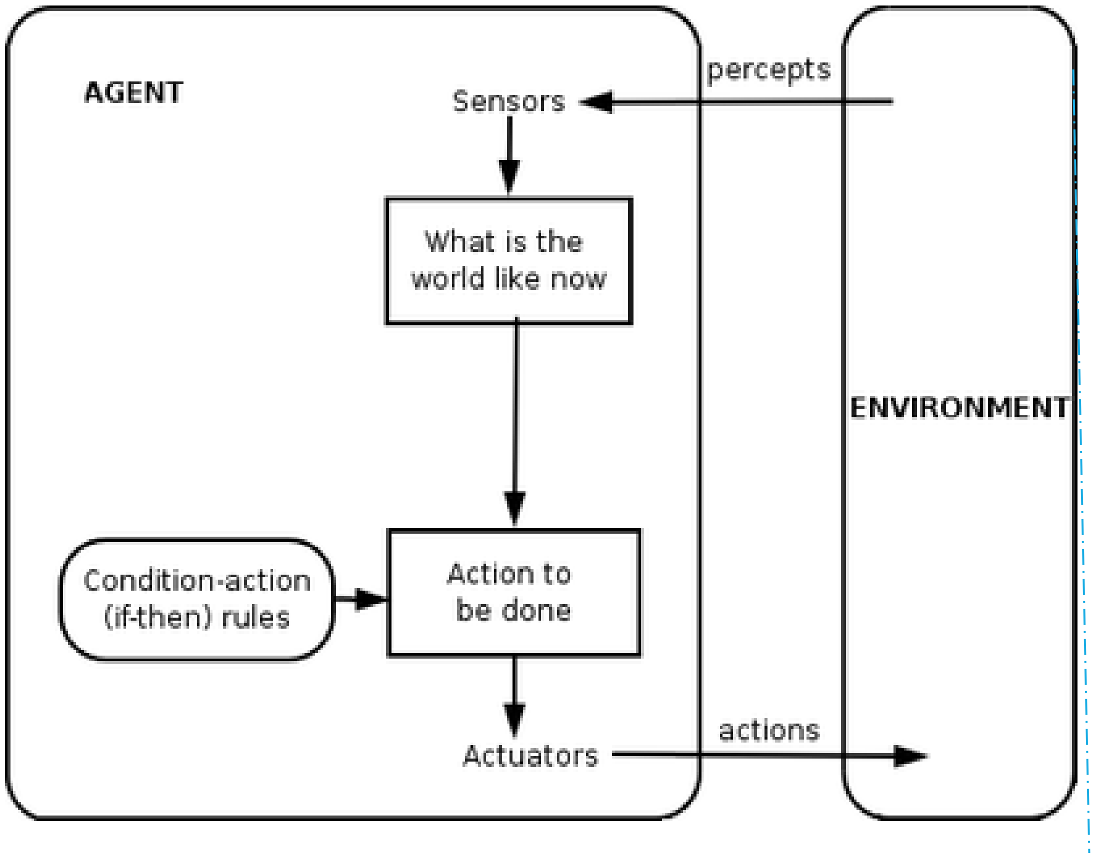
- **Hypothesis:** define as **a justified belief that is not known to be true**

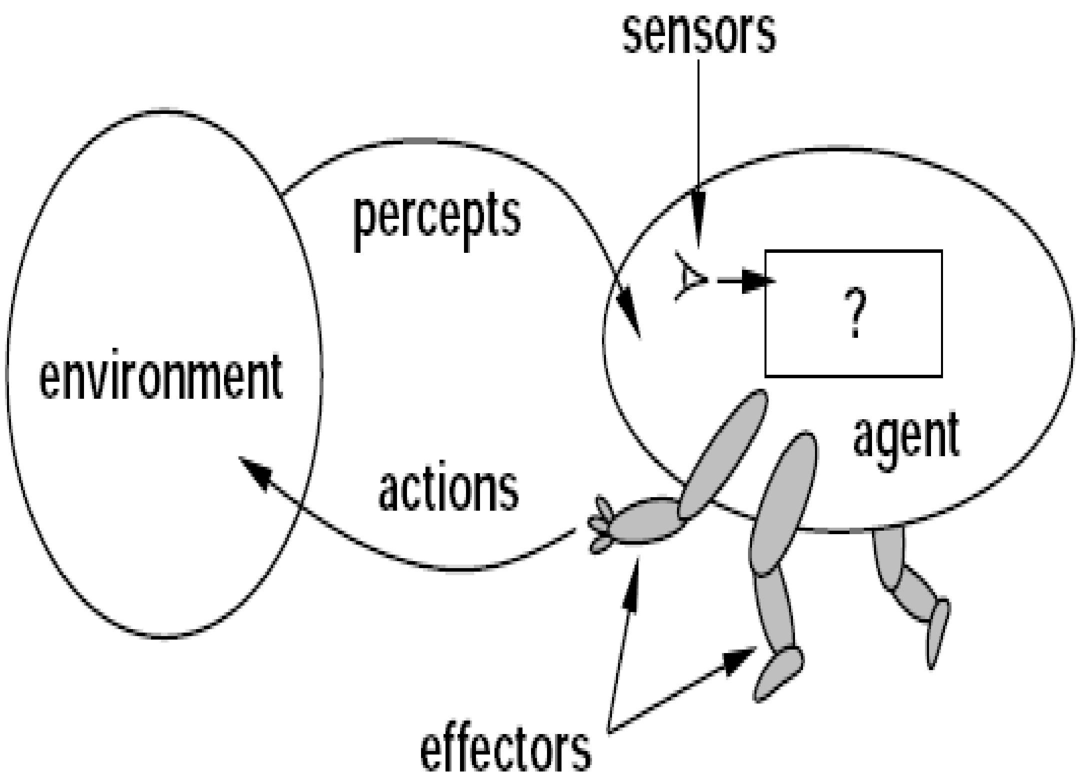
  Thus a hypothesis is a belief which is backed up with some supporting evidence, but it may still be false

- **Knowledge:** define as **true justified belief.** It is a belief that is **supported by strong evidence and proven to be true**.

INTELLIGENT AGENT

"An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators.**"





INTELLIGENT AGENT

- A *human agent* has **eyes, ears, and other organs for sensors** and **hands, legs, vocal tract,** and so on **for actuators**.

- A *robotic agent* might have **cameras and infrared range finders** for **sensors** and **various motors for actuators.**

- A *software agent* receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

SOME DEFINITIONS

Perception

*"The term **percept** to refer to the agent's perceptual inputs at any given instant."*

Agent Function

*"Mathematically speaking, we say that an agent's behavior is described by the **agent function** that maps any given percept sequence to an action".*

agent = architecture + program

SPECIFYING THE TASK ENVIRONMENT

- For the acronymically minded, we call PEAS this the PEAS(Performance, Environment, Actuators, Sensors) as task environment in designing an agent.
- For example, we focus an automated taxi driver. The full driving task is extremely open-ended.

SPECIFYING THE TASK ENVIRONMENT

There is no limit to the novel combinations of circumstances that can arise another reason we chose it as a focus for discussion, in following figure summarizes the PEAS description for the taxi's task environment.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

SPECIFYING THE TASK ENVIRONMENT

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts: bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, beaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises. suggestions, corrections	Keyboard entry

INTERNET SHOPPING AGENT

Performance measure?? price, quality, appropriateness, efficiency

Environment?? current and future WWW sites, vendors, shippers

Actuators?? display to user, follow URL, fill in form

Sensors?? HTML pages (text, graphics, scripts)

ENVIRONMENT TYPES

Fully observable vs. partially observable:

- An agent's sensors give it access to the complete state of the environment at each point in time.
- An environment might be partially observable because of noisy & inaccurate sensors or because parts of the state are simply missing from the sensor data.

ENVIRONMENT TYPES

Deterministic vs. stochastic vs strategic :

- The next state of the environment is completely determined by the current state and the action executed by the agent.
- A stochastic environment is one in which the outcome of an action is uncertain and involves probability.
- Organizations should establish clear goals and objectives for AI initiatives that align with their business strategy.
- Additionally, conducting thorough market research and competitive analysis is essential **to identify opportunities and mitigate risks.**

ENVIRONMENT TYPES

Episodic vs. Sequential:

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

ENVIRONMENT TYPES

Static vs. dynamic: The environment is unchanged while an agent is deliberating.

(The environment is semi-dynamic if the environment itself does not change with the passage of time but the agent's performance score does)

Discrete vs. continuous: A limited number of distinct, clearly defined percepts and actions.

Single agent vs. multiagent: An agent operating by itself in an environment.

ENVIRONMENT TYPES

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??	Yes	No	Yes(except auctions)	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

TYPES OF AGENT

There have four basic kinds of agent programs that embody the principles underlying almost all intelligent systems:

1. Simple reflex agents
2. Model-based reflex agents
3. Goal-based agents
4. Utility-based agents

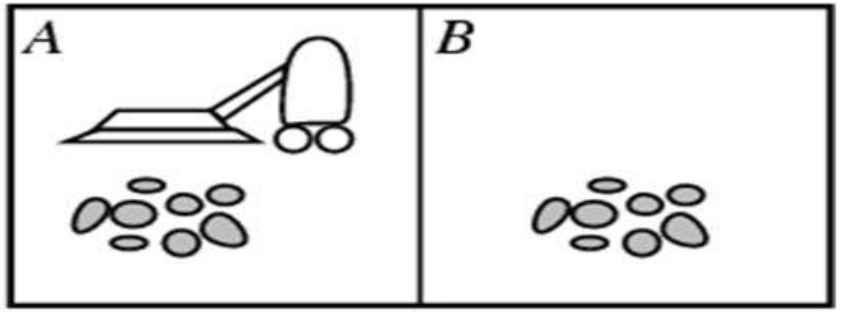
SIMPLE REFLEX AGENT

- The simplest kind of agent is the simple reflex agent.
- These agents select actions on the basis of the current percept, ignoring the rest of the percept history.
- For example, the vacuum agent is a simple reflex agent, because its decision is based only on the current location and on whether that location contains dirt.
- The agent will work *only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable*.

SIMPLE REFLEX AGENT EXAMPLE

Vacuum-cleaner world

- **Percepts:**
  Location and status,
  e.g., [A,Dirty]
- **Actions:**
  Left, Right, Suck, NoOp



Example vacuum agent program:

function Vacuum-Agent([location,status]) returns an action

- *if* status = Dirty *then* return Suck
- *else if* location = A *then* return Right
- *else if* location = B *then* return Left

MODEL-BASED REFLEX AGENTS

- The most effective way to handle partial observability is for the agent to keep track of the part of the world it can't see now.
- That is, the agent should maintain some sort of internal state that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

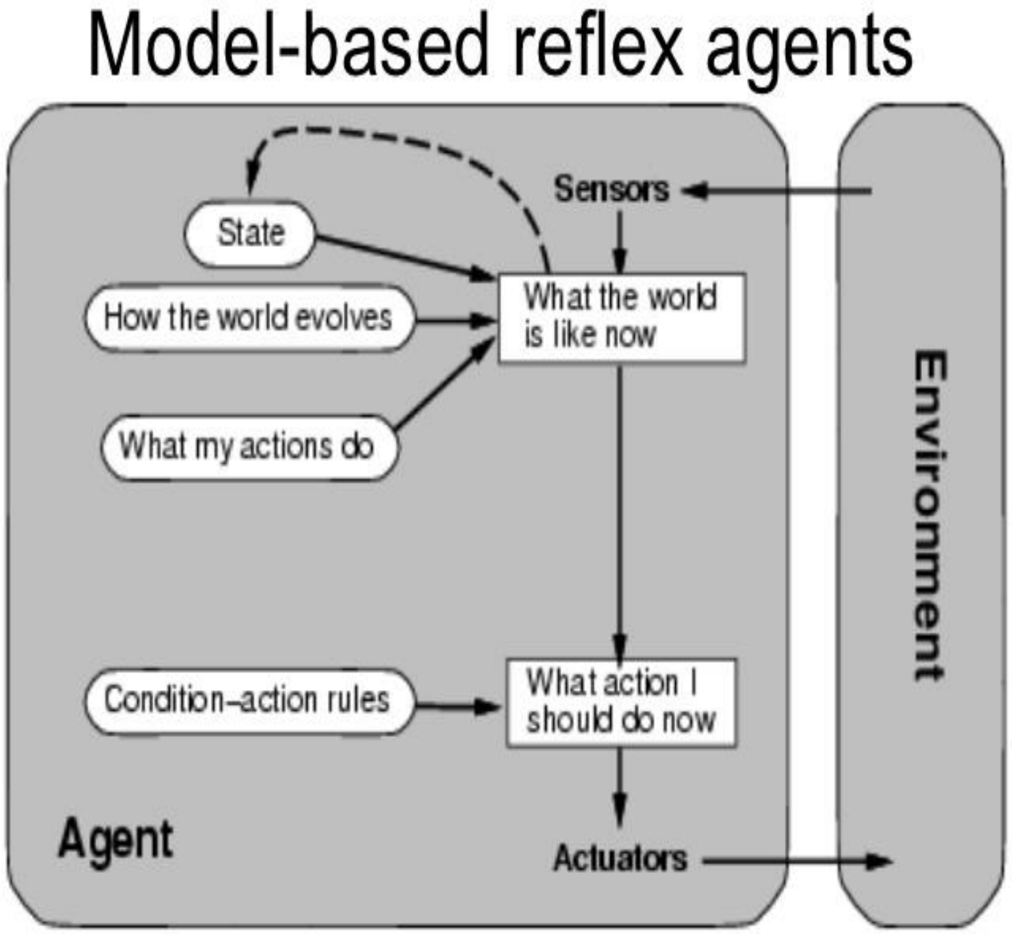
MODEL-BASED REFLEX AGENTS

- Updating this **internal state information as time goes** by requires two kinds of knowledge to be encoded in the agent program.

1. we need some information about **how the world evolves independently** of the agent—for example, that **an overtaking car generally will be closer behind than it was a moment ago**.
2. we need some information about **how the agent's own actions affect the world**—for example, that when the agent turns the steering wheel clockwise, the car turns to the right, or that after driving for five minutes northbound on the freeway, one is usually about five miles north of where one was five minutes ago.

MODEL-BASED REFLEX AGENTS

- This knowledge about "how the world works"—whether implemented in simple Boolean circuits or in complete scientific theories—is called a model of the world.
- An agent that uses such a model is called a model-based agent".



GOAL-BASED AGENTS

- Knowing something about the current state of the environment is not always enough to decide what to do.
- For example- **at a road junction**, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to.
- In other words, as well as a current state description, the agent needs some sort of goal information that describes situations that are desirable.
- For example, being at the **passenger's destination**.
- The agent program **can combine** this with the model (the same information as was used in the model-based reflex agent) **to choose actions that achieve the goal.**

GOAL-BASED AGENTS

- Sometimes goal-based action selection is straight forward, when goal satisfaction results immediately from a single action.
- Sometimes it will be more tricky, when the agent has to consider long sequences of twists & turns in order to find a way to achieve the goal.
- **Search & planning** are the subfields of AI devoted to finding action sequences that achieve the agent's goals.

UTILITY-BASED AGENTS

- Goals alone are not enough to generate high-quality behavior in most environments.
- For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.
- Goals just provide a crude binary distinction between "happy" and "unhappy" states.

UTILITY-BASED AGENTS

- An agent's utility function is essentially an internalization of the performance measure.
- If the **internal utility function and the external performance measure** are in agreement, then an agent that chooses actions to maximize its utility will be rational.

Supervised Vs Unsupervised Learning

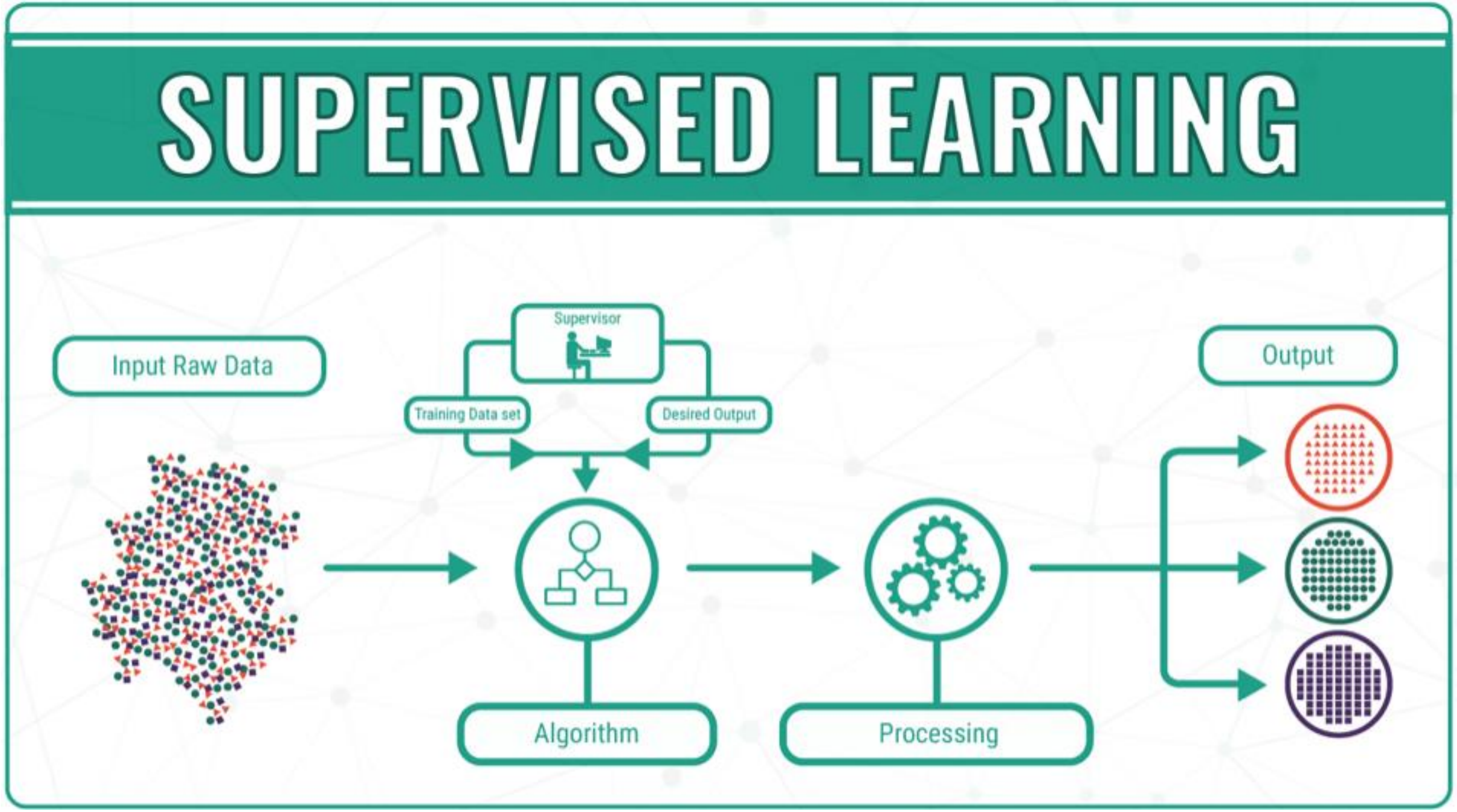
- One of the most common ways to impart artificial intelligence into a machine is through machine learning.
- The world of machine learning is broadly divided into supervised and unsupervised learning.

SUPERVISED LEARNING

Supervised learning refers to the process of building a machine learning model that is **based on labeled training data.**

- For example, let's say that we want to build a system to automatically predict the income of a person, based on various parameters such as age, education, location, and so on.
- To do this, we need to create a database of people with all the necessary details and label it. By doing this, we are telling our algorithm what parameters correspond to what income.
- Based on this mapping, the algorithm will learn how to calculate the income of a person using the parameters provided to it.

SUPERVISED LEARNING

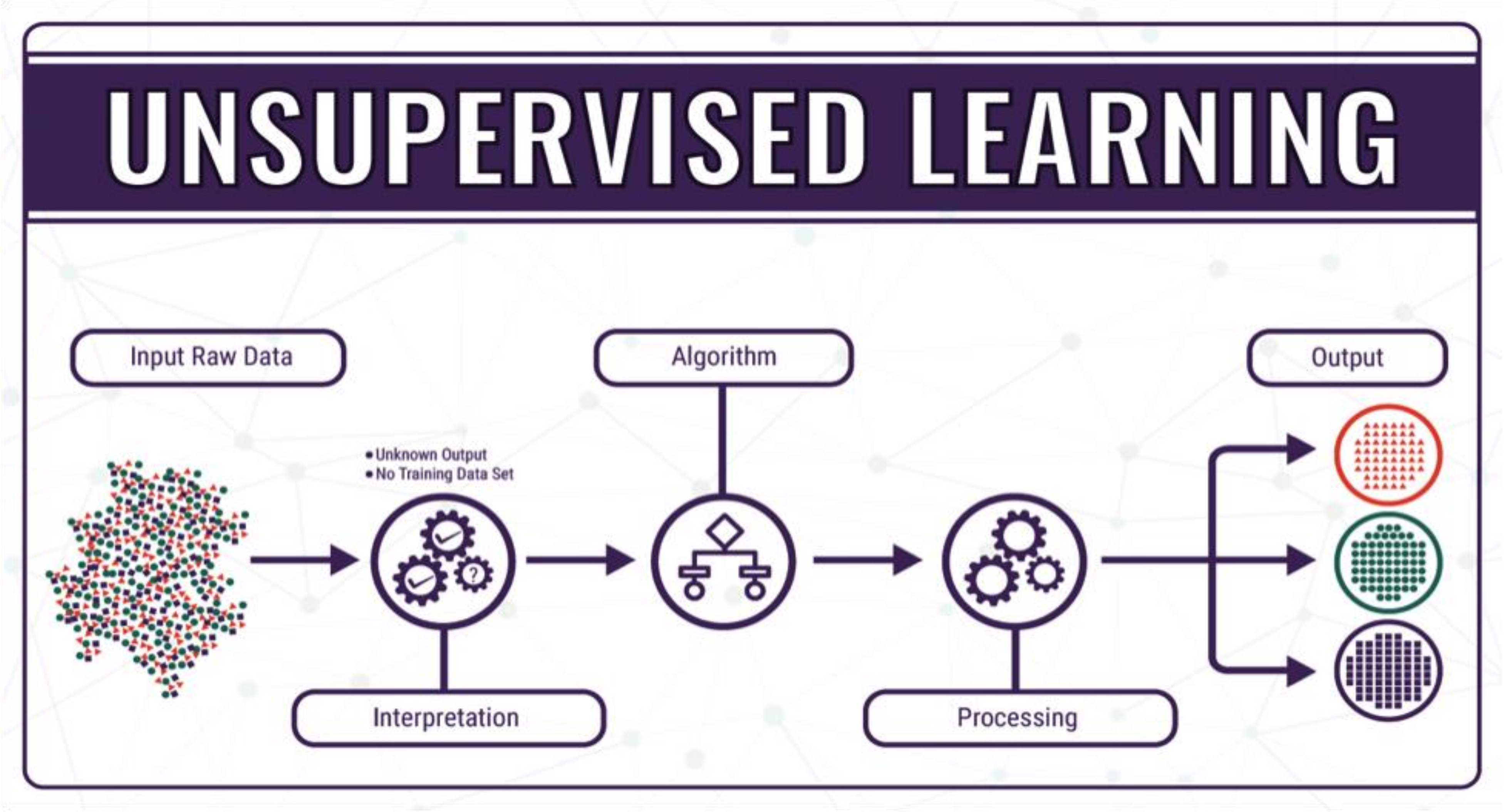


UNSUPERVISED LEARNING

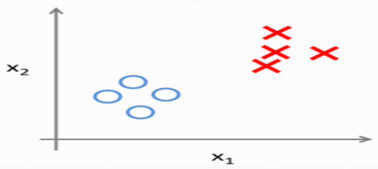
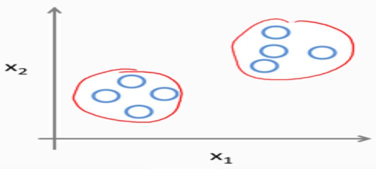
Unsupervised learning refers to the process of building a machine learning model **without relying on labeled training data**.

- In some sense, it is the opposite of what we just discussed in the previous paragraph. Since there are no labels available, you need to extract insights based on just the data given to you.
- For example, let's say that we want to build a system where we have to separate a set of data points into multiple groups. **The tricky thing here is that we don't know exactly what the criteria of separation should be.**
- Hence, an unsupervised learning algorithm needs to separate the given dataset into a number of groups in the best way possible.
- Two of the main methods used in unsupervised learning are: Dimensionally Reduction and Clustering.

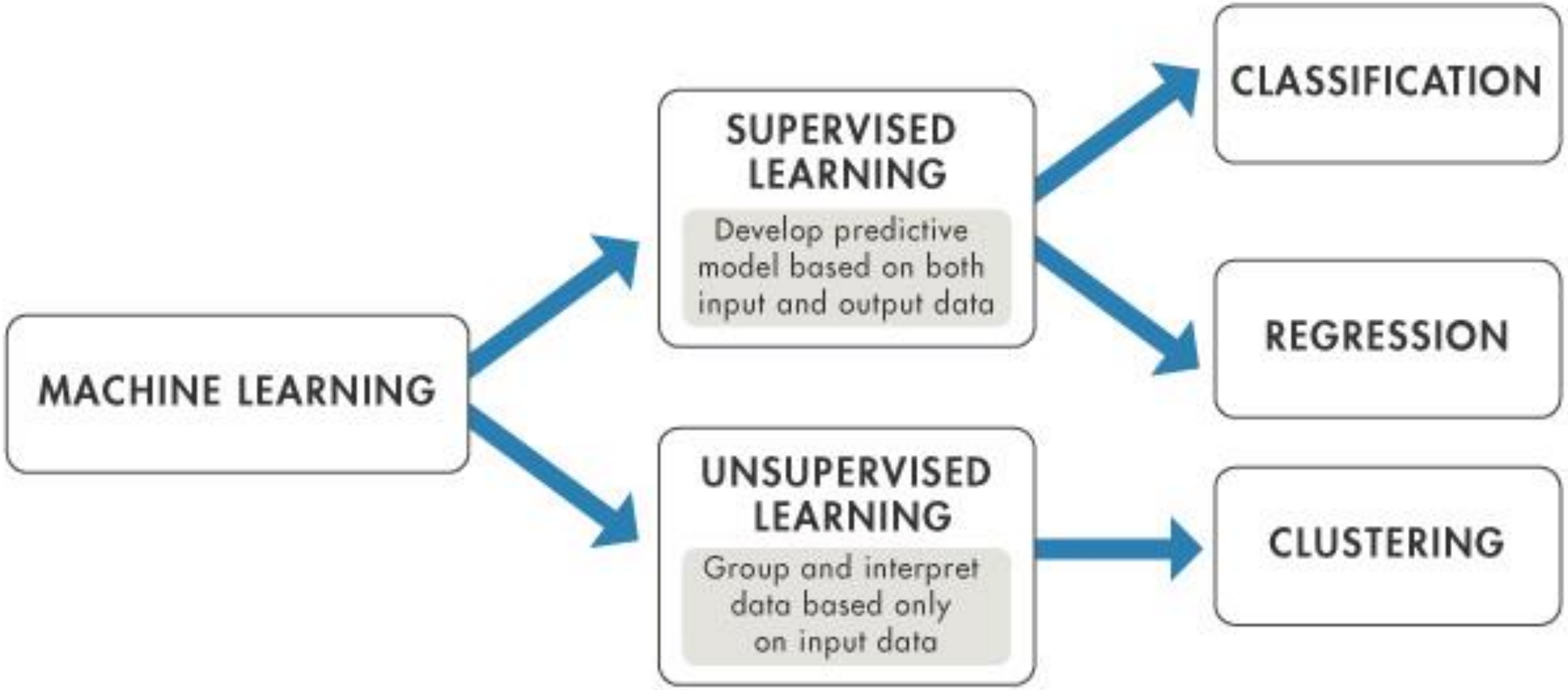
UNSUPERVISED LEARNING



SUPERVISED VS UNSUPERVISED LEARNING

Supervised	Unsupervised
Input Data is labelled	Input Data is Unlabelled
Uses training Dataset	Uses just input dataset
Data is classified based on training dataset	Uses properties of given data to classify it.
Used for prediction	Used for Analysis
Divided into two types Regression & Classification	Divided into two types Clustering & Association
Known number of classes	Unknown number of classes
	
Use off-line analysis of data	Use Real-Time analysis of data

SUPERVISED VS UNSUPERVISED LEARNING



SUPERVISED VS UNSUPERVISED LEARNING

Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised
Continuous	• Clustering & Dimensionality Reduction ○ SVD ○ PCA ○ K-means	• Regression ○ Linear ○ Polynomial • Decision Trees • Random Forests
Categorical	• Association Analysis ○ Apriori ○ FP-Growth • Hidden Markov Model	• Classification ○ KNN ○ Trees ○ Logistic Regression ○ Naive-Bayes ○ SVM

SUPERVISED CLASSIFICATION

We will discuss supervised classification techniques whole of the lecture.

- The process of classification is one such technique where **we classify data into a given number of classes.**
- During classification, **we arrange data into a fixed number of categories so that it can be used most effectively and efficiently.**
- In machine learning, classification solves the problem of identifying the category to which a new data point belongs.

SUPERVISED CLASSIFICATION

- We build the classification model based on the training dataset containing data points and the corresponding labels.
- For example, let's say that we want to check whether the given image contains a person's face or not.
- We would build a training dataset containing classes corresponding to these two classes: **face and no-face**.
- **We then train the model based on the training samples we have**. This trained model is then used for inference.
- There are two **types of Supervised Learning** techniques: Regression and Classification. Classification separates the data, Regression fits the data.

PREPROCESSING OF DATA

- We deal with a lot of raw data in the real world.
- Machine learning algorithms expect data to be formatted in a certain way before they start the training process.
- In order to prepare the data for ingestion by machine learning algorithms, we have to preprocess it and convert it into the right format
- We will be talking about several different preprocessing techniques.
  - Binarization
  - Mean removal
  - Scaling
  - Normalization

PREPROCESSING OF DATA

Before going more detail, Create a new Python file and import the following packages:

```
import numpy as np
from sklearn import preprocessing
```

Let's define some sample data:

```
input_data = np.array([    [5.1, -2.9, 3.3],
                           [-1.2, 7.8, -6.1],
                           [3.9, 0.4, 2.1],
                           [7.3, -9.9, -4.5]])
```

PREPROCESSING OF DATA : BINARIZATION

This process is used when we want to convert our numerical values into boolean values.

Let's use an inbuilt method to binarize input data using 2.1 as the threshold value.

Add the following lines to the same Python file:

```
# Binarize data
data_binarized = preprocessing.Binarizer(threshold=1.1).transform(input_data)
print("\nBinarized data:\n", data_binarized)
```

PREPROCESSING OF DATA : SCALING

- In our feature vector, the value of each feature can vary between many random values.
- So it becomes important to scale those features so that it is a level playing field for the machine learning algorithm to train on. We don't want any feature to be artificially large or small just because of the nature of the measurements.
- Add the following line to the same Python file:

```
# Min max scaling
data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(1, 2))
data_scaled_minmax = data_scaler_minmax.fit_transform(input_data)
print("\nMin max scaled data:\n", data_scaled_minmax)
```

PREPROCESSING OF DATA : SCALING

If you run the code, you will see the following printed on your Terminal:

```
Min max scaled data:
[[ 0.74117647   0.39548023  1. ]
[ 0.            1.          0. ]
[ 0.6           0.5819209   0.87234043]
[ 1.            0.          0.17021277]]
```

Each row is scaled so that the maximum value is 1 and all the other values are relative to this value.

PREPROCESSING OF DATA : NORMALIZATION

We use the process of normalization to modify the values in the feature vector so that we can measure them on a common scale.

L1 normalization, which refers to **Least Absolute Deviations**, works by making sure that the sum of absolute values is *1* in each row.

L2 normalization, which refers to least squares, works by making sure that the sum of squares is *1*.

PREPROCESSING OF DATA : NORMALIZATION

Add the following lines to the same Python file:

```
# Normalize data
data_normalized_l1 = preprocessing.normalize(input_data, norm='l1')
data_normalized_l2 = preprocessing.normalize(input_data, norm='l2')
print("\nL1 normalized data:\n", data_normalized_l1)
print("\nL2 normalized data:\n", data_normalized_l2)
```

PREPROCESSING OF DATA : NORMALIZATION

If you run the code, you will see the following printed on your Terminal:

L1 normalized data:

```
[[ 0.45132743  -0.25663717   0.2920354 ]
 [-0.0794702    0.51655629  -0.40397351]
 [ 0.609375     0.0625       0.328125 ]
 [ 0.33640553  -0.4562212   -0.20737327]]
```

L2 normalized data:

```
[[ 0.75765788  -0.43082507   0.49024922]
 [-0.12030718   0.78199664  -0.61156148]
 [ 0.87690281   0.08993875   0.47217844]
 [ 0.55734935  -0.75585734  -0.34357152]]
```

Thanks