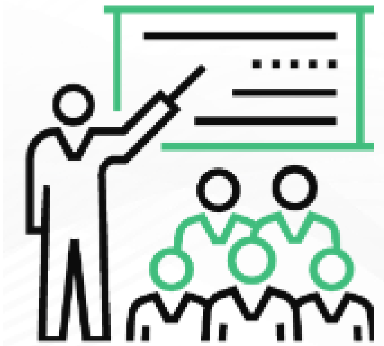
MODULE 1

Foundations of AiML

You will learn about an 'Overview of AiML' in this module.

Module Learning Objectives



- At the end of this module, you will be able to:
- Understand the basics of Artificial Intelligence (AI) and its evolution
- Understand the basics of Machine Learning (ML), its origin and how it has evolved over time
- Enumerate the factors that connect and the commonalities that exist among Machine Learning (ML) and Artificial Intelligence (AI)
- Explain the applications of AI and ML in the modern context

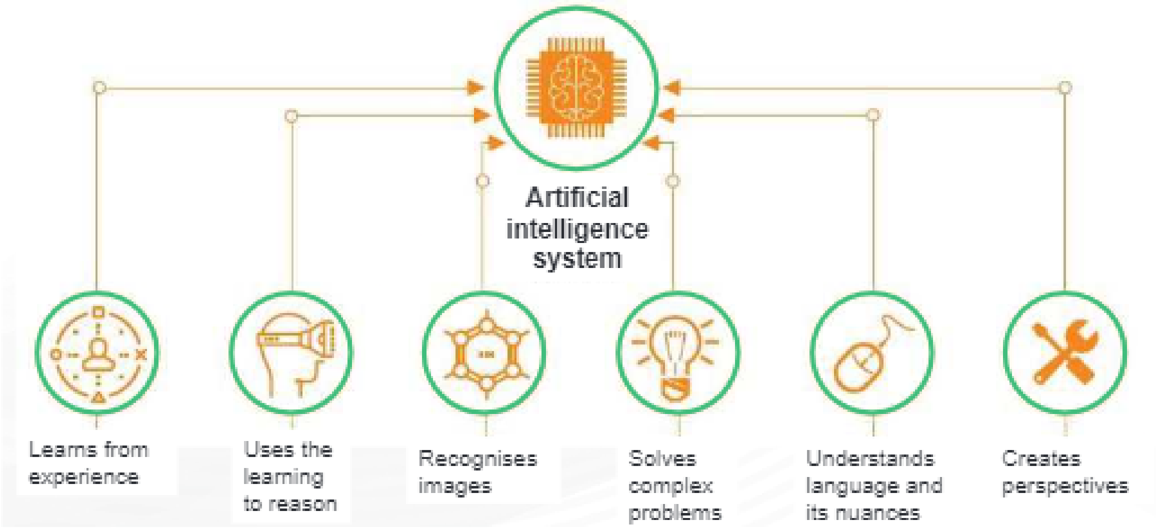
Module Topics

Let us take a quick look at the topics we will cover in this module:

- Introduction and history of AI
- Introduction and history ML
- Overlap between ML and AI
- Applications of AI and ML in the modern context

1. An Introduction to AI

Artificial intelligence is the ability of a computer program or a computer-assisted system to process the information and extract outcomes in a manner similar to the thought process of a human being.

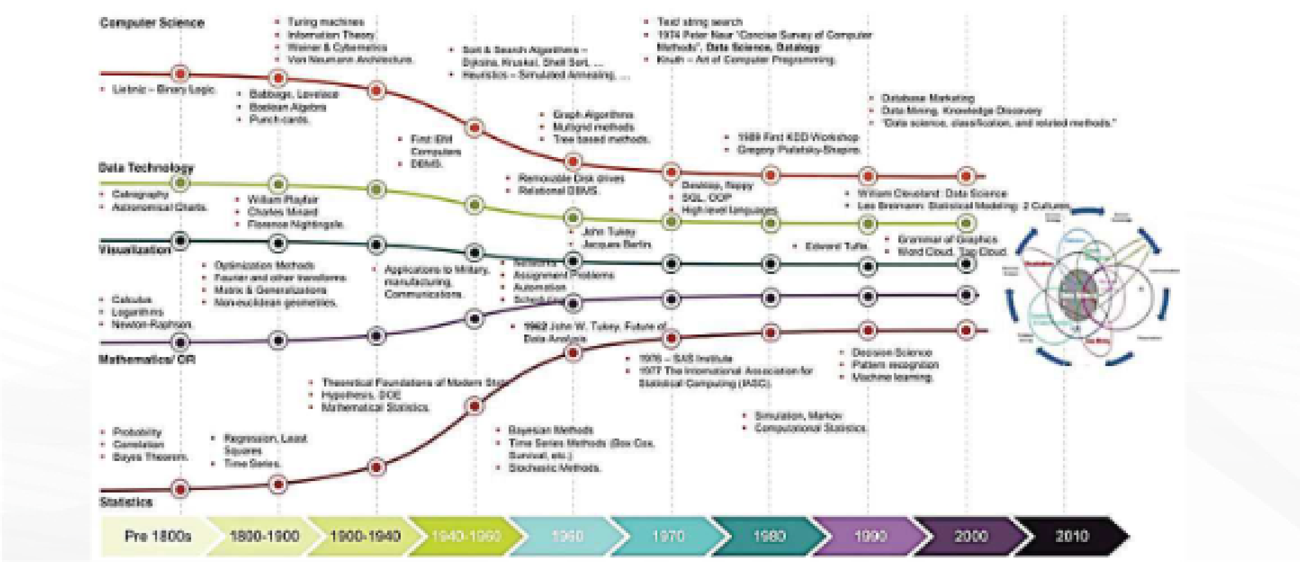


The major goal of AI is to develop the systems capable of handling complex problems in a way similar to human logic and reasoning. Thus, AI refers to the capability of computers or computer-enabled systems like robots to process the information and derive outcomes in a way that resembles human thought process in learning, decision making and solving problems.

John McCarthy (founder of MIT AI lab), who is regarded as the father of AI, defines AI as "the science and engineering of making intelligent machines, especially intelligent computer programs."

A report by Frost & Sullivan estimates that AI has the capacity to automate around 50% of jobs in the Western world within the next two decades.

1.1 History of AI



Intelligence is said to be a rare gift that humans have among all other creatures on the planet. The ability to reason, solve problems, understand, learn and remember language, to dream and remember skills are all a gift of evolution.

History shows that it is possible to create super intelligence through technological advancements. What would super intelligence be like?

Nick Bostrom defined it as "an intellect that is much smarter than the best human brains in practically every field, including scientific creativity, general wisdom and social skills."

Before understanding the ability to create intelligence, scientists discovered the ways in which machines learned. This process of learning is what became the stepping stone to artificial intelligence development. Instead of writing rules and algorithms are created to process information and learn from data. Instead of command and output, a logic is written for the machine to use as a method of processing a given data. This way the machine takes an input and produces an output using the logic or algorithm. A set of input and output associations are given to a machine, the machine learns these associations and produces a code. Once the machine assimilates this code, it can produce outputs based on this logic to other external inputs. This is very similar to how data mining works. Except that data mining seeks to derive knowledge out of existing information whereas machine learning works towards predicting an output or a future data from existing information and associations.

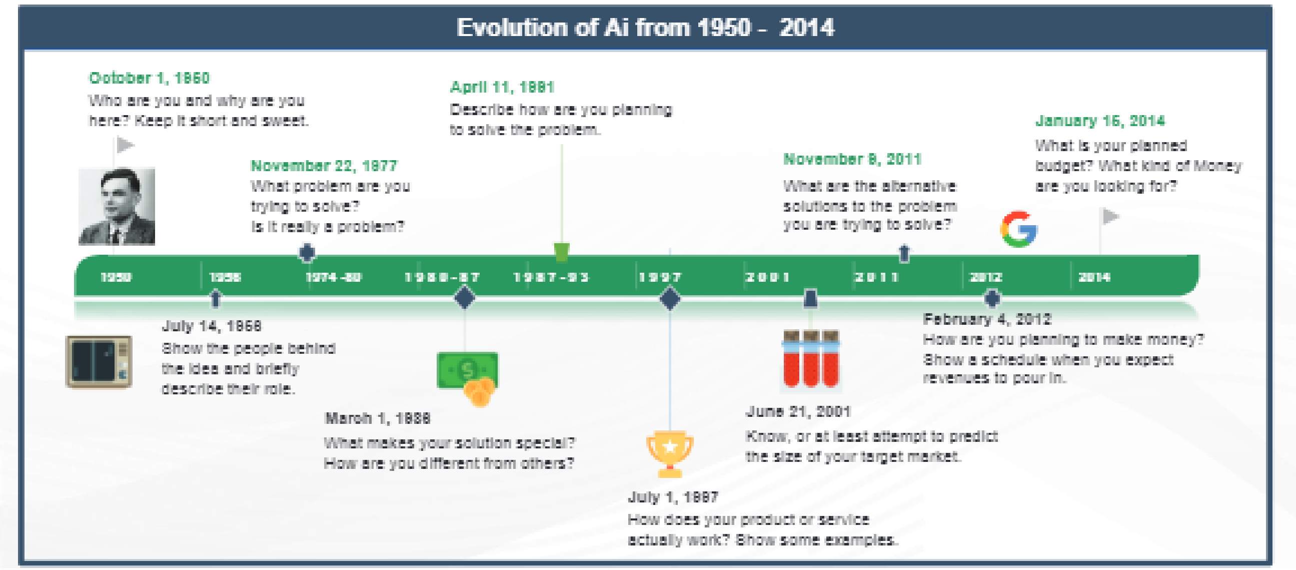
Over the years, this super intelligence has been defined as different aspirational goals by scientists and achieved segmentally.

Artificial general Intelligence – This would be a machine that can learn, recall, makes decisions and communicate in the way that humans do. The singularity summit conducted in 2012 predicted that this would be achieved by 2040.

Weak/narrow artificial intelligence- This is designed to solve specific problems. For example, playing chess, or learning a language, translating, etc.

Adapted from Reference- https://hackernoon.com/understanding-understanding-an-intro-to-artificial-intelligence-be76c5ec4d2e

1.2 Timeline of Artificial Intelligence (AI)



1950 - Alan Turing came up with the first Academy paper on artificial intelligence. His major question, " being can machines think?" According to his paper computational logic became the core functioning principle of artificial intelligence. His focus in this paper was majorly on the wariest aspects of machine learning. All approaches were based on the ways in which humans learn. Through rewards and punishment, modeling, deductions based on learned principles also known as computational logic.

The breakthrough in Turing's work was his the first AI test called the imitation game.

Adapted from http://www.lirmm.fr/ecai2012/images/stories/ecai_doc/pdf/Turing/Muggleton_ECAI.pdf

1956- Turing's test stirred up an official meet in Dartmouth College, with eminent experts discussing artificial simulation.

This discussion led to the expansion of the idea and John McCarthy coined the term artificial intelligence for the first time in the paper that was submitted as a result of this rigorous brainstorming on the working of the machine based learning and execution of tasks in a similar manner as humans. The focus now shifted towards the ability of machines or computers to truly understand language and not just use wrote facts to process information. Was there a possibility for computers to go beyond learning and computing? Could computers develop intuitive capabilities after learning languages?

These fundamental questions and discussions led to funded research projects in the field aided by corporates and the government.

1974-1980- The research on AI was not necessarily giving quick results. It led to the AI effect.

The decade saw eager premature promises of a fully functional AI system which did not really materialize. The changes and problems solved by developed machines almost constantly brought in changes to our own understanding of intelligence. As funding sources went down due to the discounting of actual advancements that happened due to these redefined versions of intelligence. The AI ecosystem went on a lowdown and the PC revolution started taking place.

1980-87 – A wave of specific expert systems arose in the UK and Japan. The Japanese government was attempting to make fifth generation computers that specialized in specific stimulation of skills among human experts in a variety of domains. This led to better functioning personalized computers.

1987- 93 – AI came back to the top with an MIT COG project that was a humanoid robot. Using Dynamic analysis and Preplanning Tool the robot could now play chess. This restored faith in the ability of AI and the scope of development. Computer scientist Vernor Vinge talked about the " coming technological Singularity" in his paper.

1997- This was the year when evidences of these specific narrow intelligences were employed in different machines and displayed as samples of working intelligent machines. IBM developed a super computer called DeepBlue that defeated world champion Garry Kasparov in a chess battle. This marked a turning point in the history of AI where the machine proved that it could think more strategically than a human brain.

After this turning point the AI started spreading across through different fields and machines.

Over the years AI has made its way into the process like predicting market trends through big data, targeting the specific sample size, making money using specific targeting and more. The data became the key to trigger and equip AI to produce results based on commands and requirements.

Adapted from - https://www.kdnuggets.com/2017/04/brief-history-artificial-intelligence.html

https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm

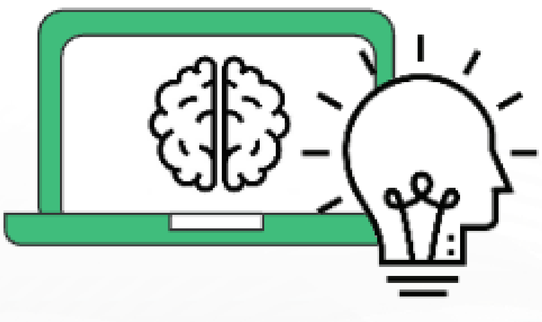
2. Introduction to Machine Learning

Machine Learning

The science of making machines act without being explicitly programmed.

An application of AI that provides machines, the ability to learn automatically and improve from experiences.

Give machines, the access to data and let them learn for themselves.



If you want to make machines act, you need to teach them. You need to make them learn, by providing them with repeating patterns. For example, if you want the system to predict rainfall, you need to supply it at least with a decade's rainfall data, if not more. Using this data, machine will understand the pattern and amount of rainfall over the past years and will predict the rainfall for the current or upcoming years. So, you have to train the ML model with large volumes of similar data.

Arthur Samuel, who is credited with the coining of the term ML, defines ML as "the field of study that gives computers, the ability to learn without being explicitly programmed."

Tom Mitchell in 1997 provided a more engineering-oriented definition for ML - "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

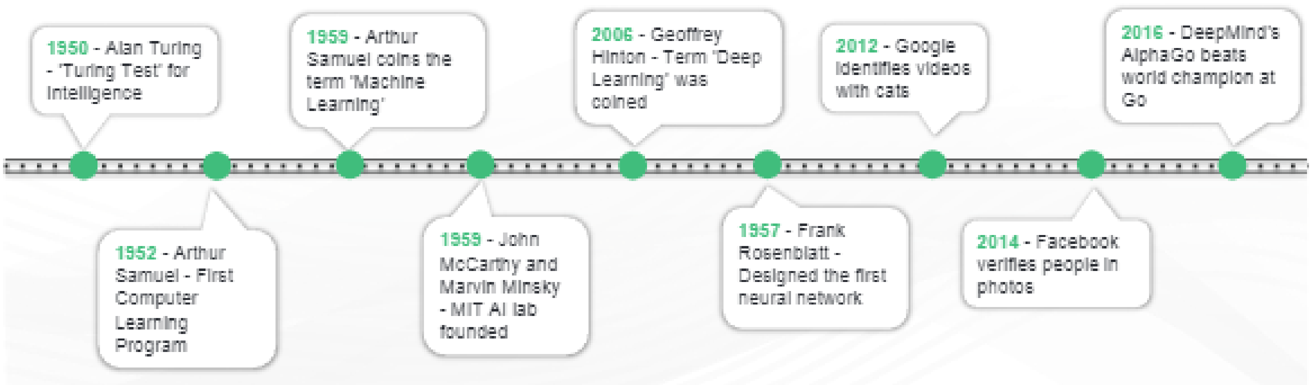
For example, you might have spam filters in your email. A spam filter is the ML program that can learn to flag spam given examples of spam emails (e.g., flagged by users) and examples of regular (non spam, also called "ham") emails. The examples that the system uses to learn are called the ***training set***. Each training example is called a ***training instance*** (or *sample*). In this case, the task T is to flag spam for new emails, the experience E is the ***training data***, and the performance measure P needs to be defined; for example, you can use the ratio of correctly classified emails. This particular performance measure is called ***accuracy*** and it is often used in classification tasks.

If you take AI as a tool, ML is the way to build the tool. ML provides machines, the ability to learn and improve from experiences without the need of being explicitly programmed.

Some of the examples for ML include web search engines, self-driving cars, speech and handwriting recognition, image recognition, recommendation engines, natural language processing, fraud detection, human genome mapping, etc. Learning algorithms are now surpassing human performance in lots of areas these days.

2.1 History of ML

The sequence of events that happened during the evolution of ML is given below:



Way back in 1950, an English computer scientist and mathematician named Alan Turing developed the Turing Test, that evaluates a machine's ability to showcase an intelligent behavior that is equivalent to or comparable with that of a human being. This is a notable incident in the history of Machine learning, as it proved machines' ability to mimic human thinking and intelligence.

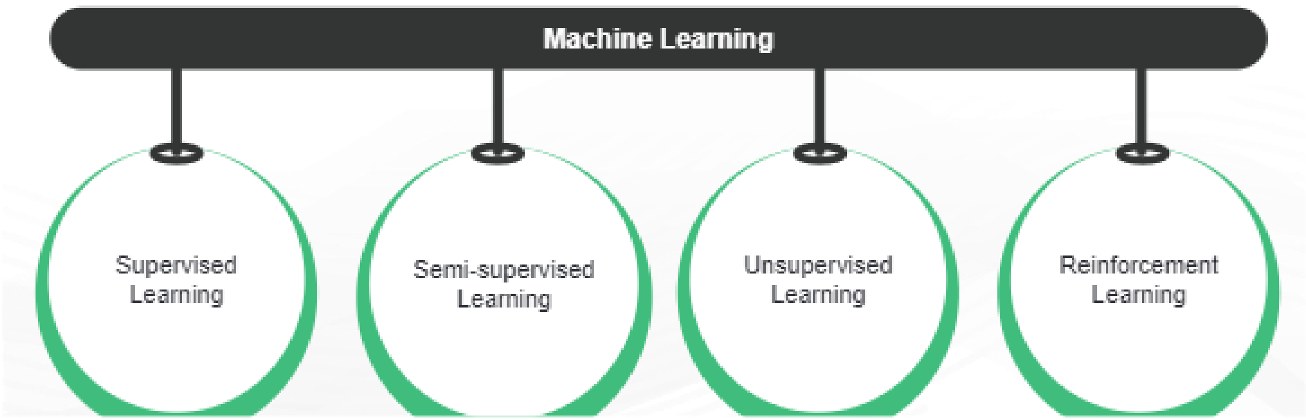
Arthur Lee Samuel was an American pioneer in the field of computer gaming and AI, who coined the term "Machine Learning" in 1959. The Samuel Checkers-playing Program was among the world's first successful self-learning programs, and as such a very early demonstration of the fundamental concept of artificial intelligence (AI).

ML started flourishing as an individual discipline in the 1990s. ML and data mining often employ the same methods and overlap significantly, but while ML focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in data.

Machine learning has been around for decades in some specialized applications, such as Optical Character Recognition (OCR). But the first ML application that really became mainstream, improving the lives of hundreds of millions of people, took over the world back in the 1990s: it was the spam filter.

2.2 Categories of ML Systems

There are the following four major categories of ML systems:



The primary goal of ML is to make computers learn and understand automatically with little or no human intervention and adjust the actions according to changing situations. There are several ways to teach machines. Most commonly used machine learning algorithms are:

- **Supervised Learning** - apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly. A typical supervised learning task is classification. The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

- **Unsupervised learning** - used when the information used to train is neither classified nor labelled. The system tries to learn without a teacher. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

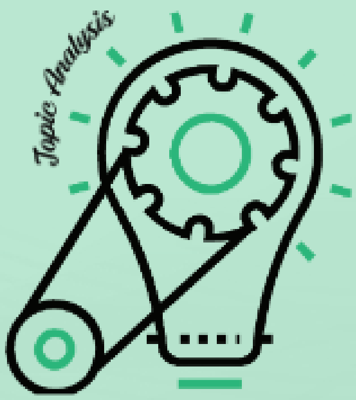
- For example, say you have a lot of data about your blog's visitors. You may want to run a clustering algorithm to try to detect groups of similar visitors. At no point, do you tell the algorithm which group a visitor belongs to, it finds those connections without your help. For example, it might notice that 40% of your visitors are males who love comic books and generally read your blog in the evening, while 20% are young sci-fi lovers who visit during the weekends, and so on.

- **Semi-supervised Learning** - Some algorithms can deal with partially labelled training data, usually a lot of unlabelled data and a little bit of labelled data. This is called semi-supervised learning. Some photo-hosting services, such as Google Photos, are good examples of this. Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in photos 2, 5, and 7. This is the unsupervised part of the algorithm (clustering). Now all the system needs are for you to tell it who these people are. Just one label per person, and it is able to name everyone in every photo, which is useful for searching photos.

- **Reinforcement learning** - a learning method that interacts with its environment by producing actions and discovers errors or rewards. The learning system, called an *agent* in this context, can observe the environment, select and perform actions, and get *rewards* in return (or penalties in the form of negative rewards). It must then learn by itself what is the best strategy, called a *policy*, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal. For example, many robots implement Reinforcement Learning algorithms to learn how to walk. DeepMind's AlphaGo program is also a good example of Reinforcement Learning: it made the headlines in March 2016 when it beat the world champion Lee Sedol at the game of Go. It learned its winning policy by analyzing millions of game and then playing many games against itself. Note that learning was turned off during the games against the champion; AlphaGo was just applying the policy it had learned.

What did You Grasp?



1. Which of the following options is an example for supervised learning?

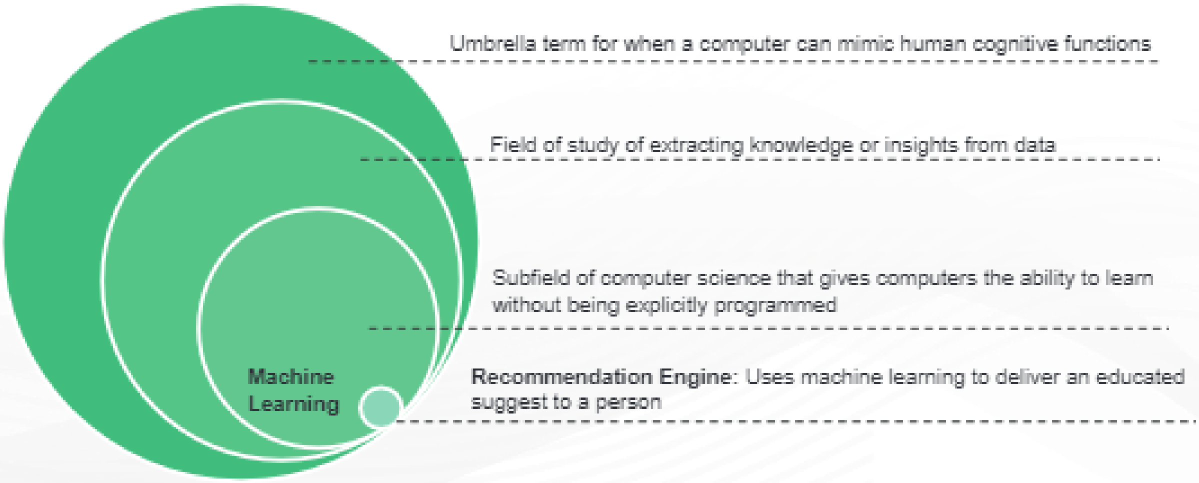
   A) Grouping of blog visitors
   B) Robotics
   C) Spam filters
   D) Photo sorting

2. Who among the following, coined the term Deep Learning?

   A) John McCarthy
   B) Geoffrey Hinton
   C) Frank Rosenblatt
   D) Alan Turing

3. How do DS, ML and AI Overlap with Each Other?

The following diagram illustrates how these terms overlap each other.



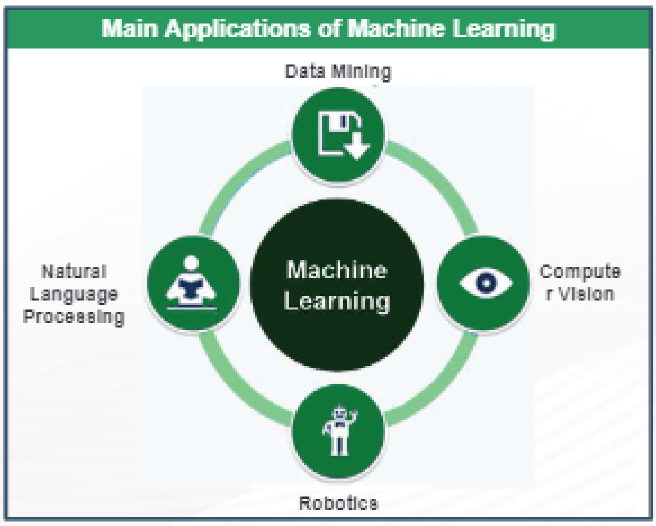
There is so much of hype these days around DS, ML and AI. Let's see, how these terms overlap with each other.

We learnt in an earlier section that DS encompasses a number of disciplines like mathematics, statistics, computer science, information science. Using ML is the practical application of AI. ML is the technology or enabler for AI. DS is likely, how you apply ML, i.e. the real-world application of ML, the goal being the creation of smart products that enrich people's lives.

AI is the broader concept of machines being able to carry out tasks in a way that we would consider "smart". ML is a current application of AI based on the idea that we should just be able to give machines access to data and let them learn for themselves.

4. Applications of ML

The following diagram illustrates the applications of ML.



Source: Frost & Sullivan

As you can see in the picture, there are four major applications of ML:

- Data Mining
- Computer Vision
- Robotics
- Natural Language Processing

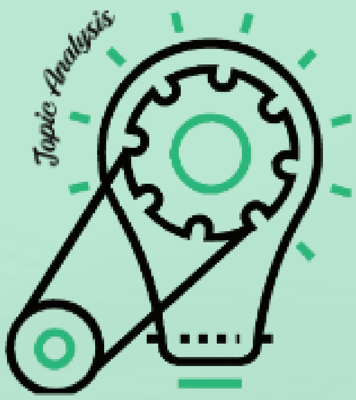
Various sectors in which ML finds its applications:

- Agriculture - Farm mechanization, Precision agriculture
- Healthcare - Diagnostic APIs, Personalized treatment
- Manufacturing - Edge computing
- Automotive - Self-driving cars, Infotainment systems
- Social Media - Intelligent filters, Facial identification
- Financial Services - Innovative payment modes, Fraud detection

ML is the best choice for:

- The problems for which existing solutions require a lot of hand-tuning or long lists of rules: One ML algorithm can often simplify code and perform better.
- Complex problems for which there is no good solution at all using a traditional approach: The best ML techniques can find a solution.
- Fluctuating environments: a Machine Learning system can adapt to new data.
- Getting insights into complex problems and large amounts of data.

What did You Grasp?



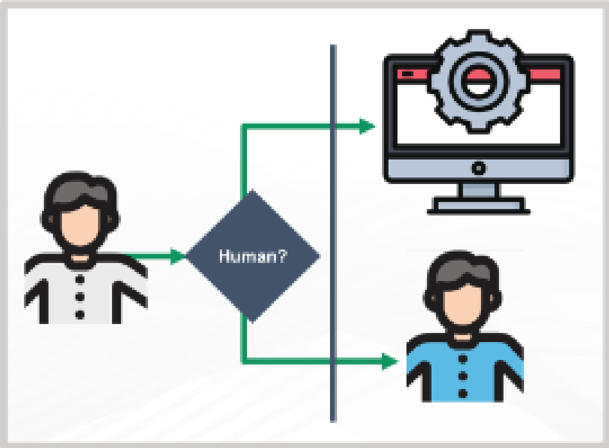
1. State True or False.

   Artificial Intelligence is the application of machine learning for creating smart systems.

   A) True
   B) False

5. Artificial Intelligence with Turing Test Problem

- First test ever to check intelligence of a Machine. Determine whether a computer has achieved the intelligence of humans?
- Alan Turing (Father of Modern Computer)
- An English Mathematician wrote a landmark paper in 1950 that asked about the questions. Can machines think?
- He proposed a test to answer the question: " How will we know when we've succeeded?"
- Its Human vs the Machines!
- Can you guess who is who?



Introduction to Turing Test (1950)

So basically this test was about checking the level of intelligence machine has achieved till now. Is it equivalent to what humans have today? Can we distinguish between machine and humans without seeing who is on the other side? The test was based on a natural language conversation with a human and a machine. The evaluator is aware that one of them is surely a machine. All three people in this scenario are totally separated from each other. However, the so-called **total Turing Test** includes a video signal so that the interrogator can test the subject's perceptual abilities, as well as the opportunity for the interrogator to pass physical objects "through the hatch." To pass the total Turing Test, the computer will need

- **computer vision** to perceive objects, and
- **robotics** to manipulate objects and move about.

The machine will pass if the evaluator is unable to distinguish between both the participants.

6. Difference between Thinking Humanly and Thinking Rationally

THINKING HUMANLY	THINKING RATIONALLY
"The exciting new effort to make computers think...machines with minds, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning..." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
ACTING HUMANLY	**ACTING RATIONALLY**
"The art of creating machines that per-form functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole et al., 1998) "AI...is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

The definitions on top are concerned with *thought processes* and *reasoning*, whereas the ones on the bottom address *behaviour*. The definitions on the left measure success in terms of fidelity to *human* performance, whereas the ones on the right measure against an *ideal* performance measure, called **rationality**. A system is rational if it does the "right thing," given what it knows.

Deep dive into the above table.

1) Acting Humanly - The computer would need to possess the following capabilities:

- **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.

2) Thinking Humanly – If we say a program can think like humans, then firstly we must find out how humans think. Three ways to find that out :

- Through introspection—trying to catch our own thoughts as they go by.
- Through psychological experiments—observing a person in action.
- Through brain imaging—observing the brain in action.

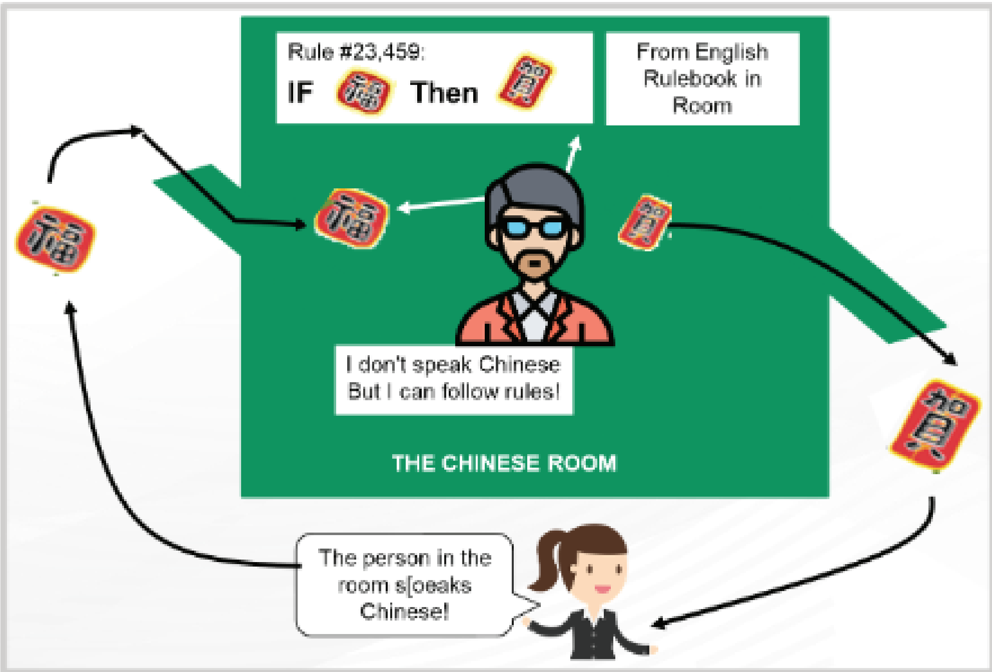
If the program's input–output behaviour matches the corresponding human behaviour, that is evidence that some of the program's mechanisms could also be operating in humans.

3) Thinking Rationally: (Right Thinking) – A Greek philosopher Aristotle can up with the first code of right thinking. His **syllogisms** provided patterns for argument structures that always yielded correct conclusions when given correct premises—for example, "John is a man; all men are mortal; therefore, John is mortal." These laws of thought were supposed to govern the operation of the mind; their study initiated the field called **logic**. In 19th Century people developed a precise notation for all kinds of location statements like and, or and negation. But was the computer program able to understand and decode it? There are two main obstacles to this approach. First, it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. Second, there is a big difference between solving a problem "in principle" and solving it in practice. Even problems with just a few hundred facts can exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first. Although both of these obstacles apply to *any* attempt to build computational reasoning systems, they appeared first in the logicist tradition.

4) Act rationally – An agent (Machine) is asked to do something more than just a simple computer program would do like operate autonomously, perceive the environment, adapt to changes or try to achieve certain goals. A **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome. It works on laws of thoughts which means that the agent should be able to make correct inferences because along with acting rationally the agent should also be able to reason logically to the conclusion that a given action will achieve one's goal

and then act on it with its conclusion. On the other hand, correct inference is not *all* of rationality, in some situations, there is no provably correct thing to do, but something must still be done. There are also ways of acting rationally that cannot be said to involve inference. For example, recoiling from a hot stove is a reflex action that is usually more successful than aslower action taken after careful deliberation.

7. Chinese Room Problem



Let's discuss some of the key details of the Chinese room problem:

- We just learnt about the Turing test. Now John Searle in 1980 said that the Turing Test is not the right way to judge Intelligence of the machine.
- The **Chinese room argument** holds that a digital computer executing a program cannot have a "mind", "understanding" or "consciousness regardless of how intelligently or human-like the program may make the computer behave.
- According to him machines which don't have any understanding about the input and they are just processing the output to the program cannot be considered as intelligent.
- The Chinese Room name comes from the scenario in which he performed the test in order to put his point forward and contradict the Turing test.
- Simple Idea behind the test was to act as an translator from Chinese to English without understanding the context behind the communication.

- A machine programmed with the right inputs and right outputs having a mind of exactly the same sense human minds have is called a Strong AI.
- A machine that is only processing input and giving a corresponding output without getting the exact sense or understanding is called a weak AI.

John imagines himself (instead of machine) as non-Chinese person sitting inside the room isolated from another Chinese person who is outside the room tries to communicate. He has provided a list of Chinese characters and an instruction book explaining in detail the rules according to which strings (sequences) of characters may be formed, but without giving the meaning of the characters. That means he has a book with an English version of the computer program, along with sufficient paper, pencils, erasers, and filing cabinets.

In this thought experiment, a person in the "Chinese room" is passed questions from outside the room, and consults a library of books to formulate an answer.

Now, he receives all the messages posted through a slot in the door written in Chinese language. He will process all the symbols according to program instructions and produces the Chinese characters as output like:

- If he finds a Chinese symbol like , he returns symbol
- If he finds a Chinese symbol like ДКЯЕ, he returns symbol Ꮑᑌᑯ

8. Agents

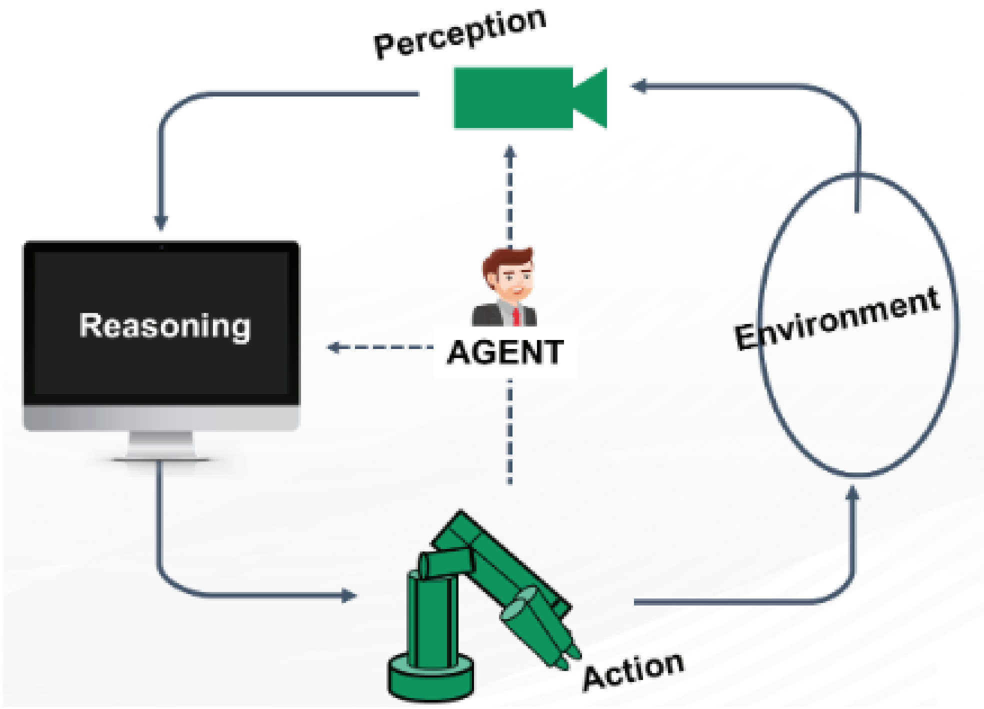
An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.

- 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.
- **Percept** to refer to the agent's perceptual inputs at any given instant.
- An agent's behaviour is described by the **agent function** that maps any given percept sequence to an action.
- The agent function for an artificial agent will be implemented by an **agent program.**
- The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

Adapted from: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments

Go through the points mentioned on the slide.

8.1 Rational Agent



A **rational agent** is one that does the right thing—conceptually speaking.

Obviously, doing the right thing is better than doing the wrong thing, but what does it mean to do the right thing?

Considering the *consequences* of the agent's behavior. When an agent is plunked down in an environment, it generates a sequence of actions according to the percepts it receives. This sequence of actions, causes the environment to go through a sequence of states. If the sequence is desirable, then the agent has performed well. This notion of desirability is captured by a **performance measure** that evaluates any given sequence of environment states.

As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.

Notice that we said *environment* states, not *agent* states. If we define success in terms of agent's opinion of its own performance, an agent could achieve perfect rationality simply by deluding itself that its performance was perfect.

8.2 Task Environment

PEAS (Performance, Environment, Actuators, Sensors)	
Performance measure	How will you judge the performance of the agent
Environment	Place around where the agent is put in to perform
Actuators	Hardware / Device through which agent will interact with the environment
Sensors	Hardware / Devices through which agent will observe or take input from the environment

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

The facilitator will explain you about the task environment.

Activity

Internet Shopping Agent

- Performance measure- price, appropriateness, efficiency
- Environment- current and future websites, vendors, shippers

- Actuators- display to user, follow URL, fill in the form
- Sensors- software for processing webpages

Interactive English Tutor

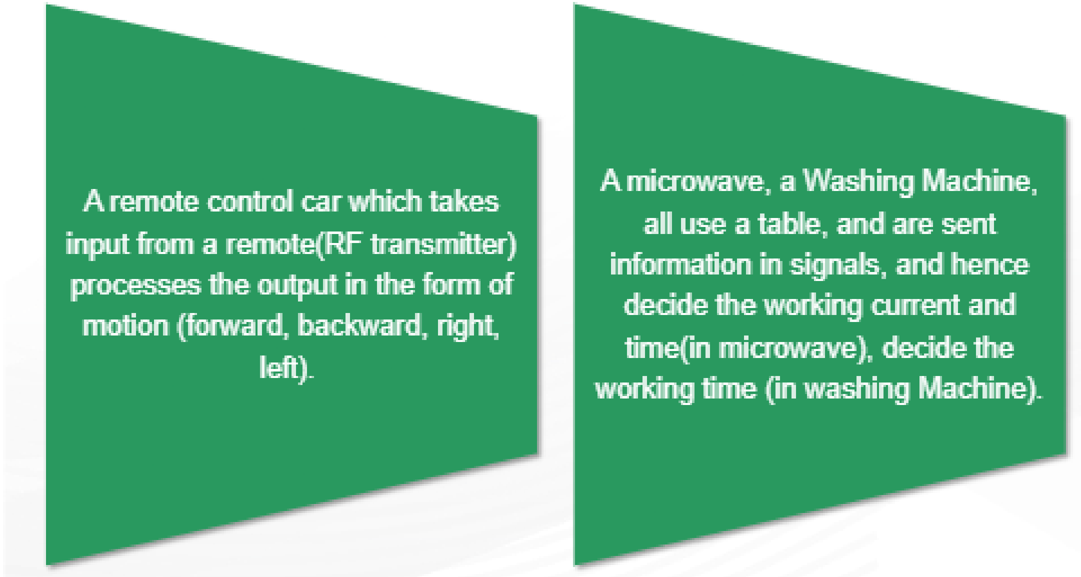
- Performance measure: Maximize student's score on the test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

8.3 Types of Agents

Following are the different types of agents:

1. Table Driven Agents
2. Simple Reflex Agents
3. Model Based Agents
4. Goal Based Agents
5. Utility Based Agents
6. Learning Agents

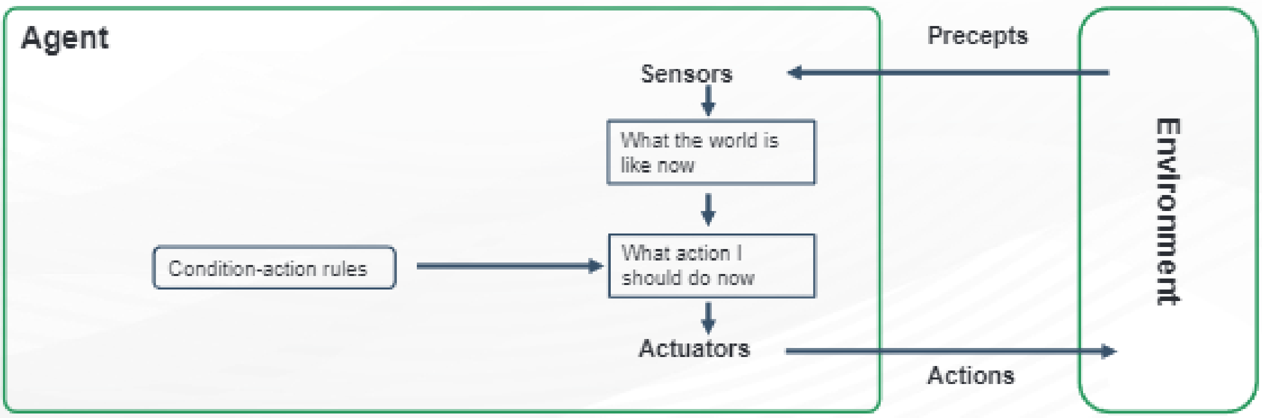
8.3.1 Table Based Agents



In a simple **Table**-driven **Agent**, a lookup **table** is used to match every possible percept sequence to the corresponding action.

8.3.2 Simple Reflex Agents

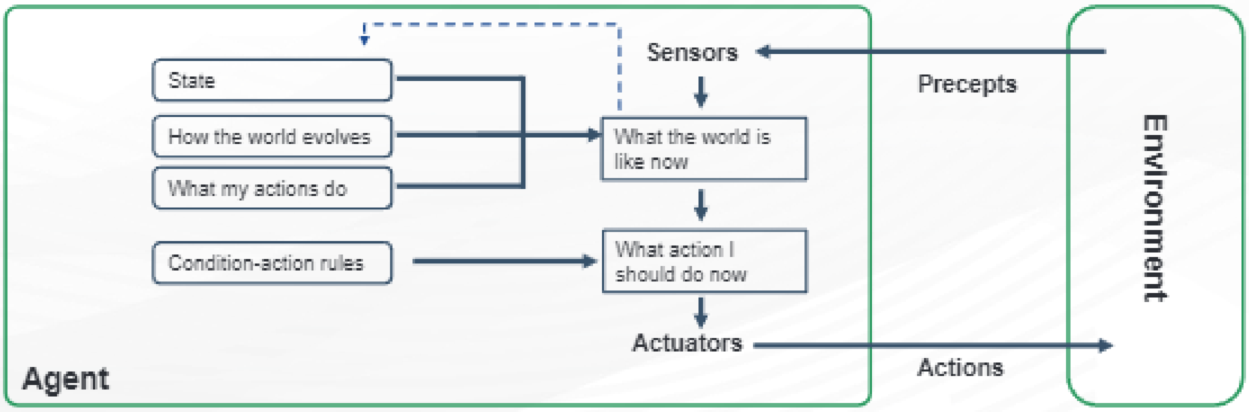
- A proportional controller which produces proportional output
- Input = x, Output = p * x, here p is the proportionality constant
- An A/C, keep the room temp in a close range of the set temperature (in A/C)
- If the temp goes above the set point the Compressor is turned off, if the temp goes below it, Compressor is turned on.



Imagine yourself as the driver of the automated taxi. If the car in front brakes and its brake lights come on, then you should notice this and initiate braking. In other words, some processing is done on the visual input to establish the condition we call "The car in front is braking." Then, this triggers some established connection in the agent program to the action "initiate braking." We call such a connection a **condition–action rule**, written as "if car-in-front-is-braking then initiate-braking."

8.3.3 Model Based Agent

- A Prediction system that uses a part of the given data. when predicting Share value of a certain company an intelligent software doesn't observe the entire data.. it takes a look at the partial information for the last week or last 2 months, and then decides if the market is bullish or bearish.
- A data entry system, or an attendance system this system doesn't look if the person returning the book or making the attendance is actually the person. So, theoretically we can cheat the machine by producing a fake fingerprint, and get attendance. It looks at the partial data.

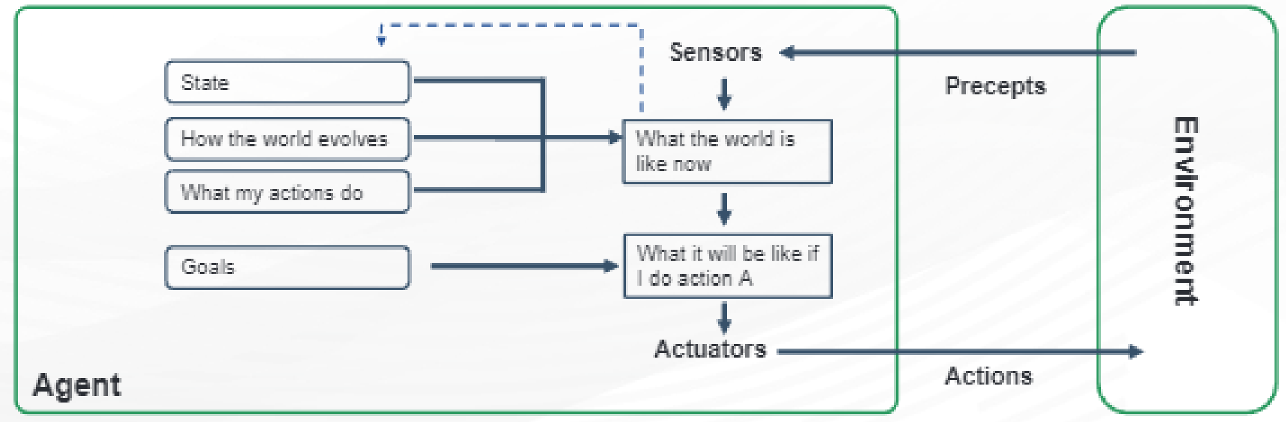


The most effective way to handle partial observability is for the agent to *keep track of the part of the world it can't see now*. That is, the agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. For other driving tasks such as changing lanes, the agent needs to keep track of where the other cars are if it can't see them all at once. And for any driving to be possible at all, the agent needs to keep track of where its keys are. Updating this internal state information as time goes by requiring two kinds of knowledge to be encoded in the agent program.

First, 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. Second, 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. 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.**

8.3.4 Goal Based Agents

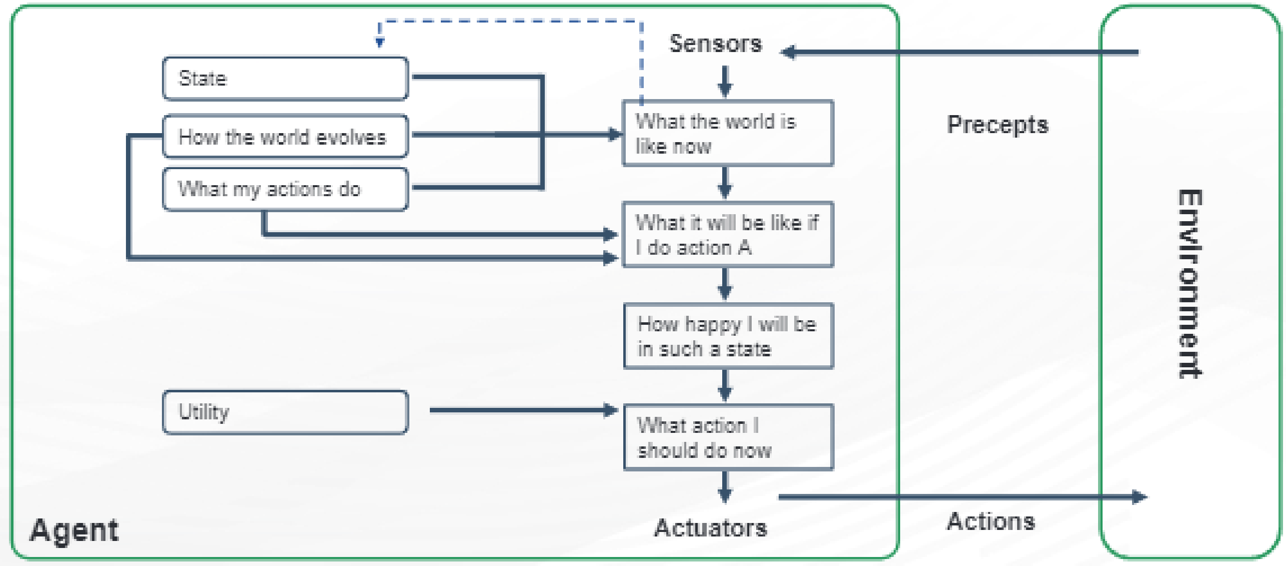
- The goal is fixed, but it can take any path to reach the goal, the goal here helps the system understand which paths are required and which can be ignored, Agent knows which actions it takes are bringing it closer to the goal and which are not needed.
- This is used in games where a fixed goal is to be achieved.
- A Simple Road tracking Agent, uses a fixed goal and suggests optimal ways of reaching said goal.



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. Here in this case that goal can be a destination where the passenger is heading.

8.3.5 Utility Based Agents

- The utility function is a function that maps a state to a measure of the utility of the state.
- An autonomous car.



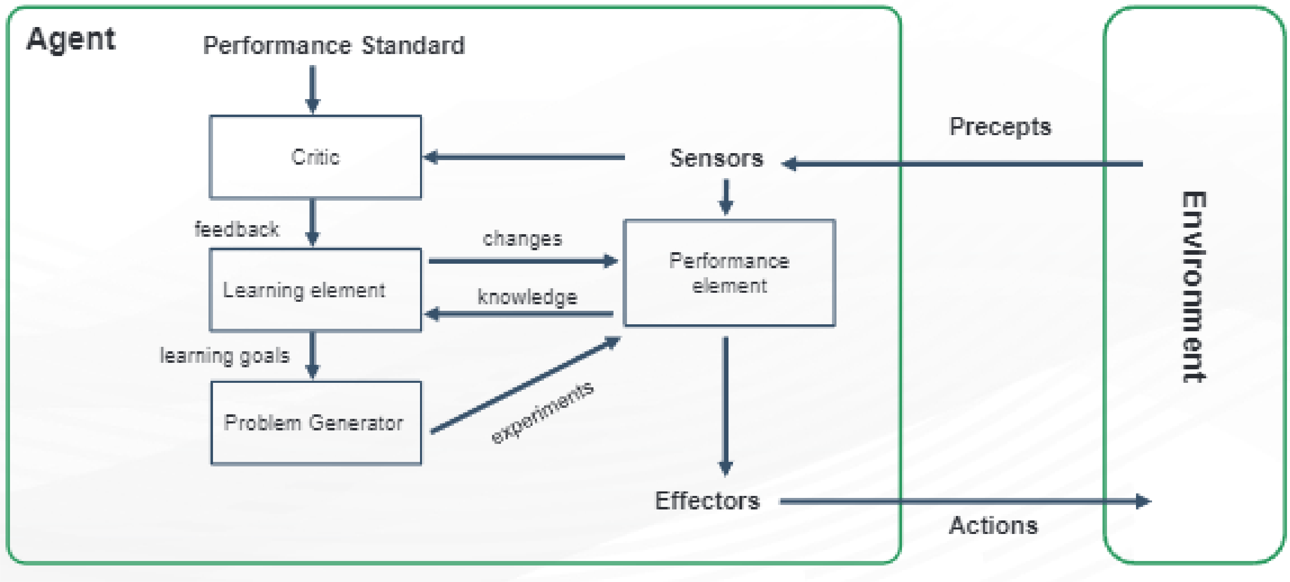
Goals alone are not enough to generate high-quality behavior in most environments. For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others. Goals just provide a crude binary distinction between “happy” and “unhappy” states. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. Because “happy” does not sound very scientific, economists and computer scientists use the term **utility** instead.

We have already seen that a performance measure assigns a score to any given sequence of environment states, so it can easily distinguish between more and less desirable ways of getting to the taxi’s destination. An agent’s **utility function** is essentially UTILITY FUNCTION 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 according to the external performance measure.

Adapted from - https://www.passeidireto.com/arquivo/18996209/livro-artificial-intelligence-a-modern-approach-3rd-edition-/26

8.3.6 Learning Based Agents

⇥ **Supervised Learning. Teaching using Examples and Data and Relative Output.**
⇥ **Autonomous house system like JARVIS in Ironman Movie.**



In many areas of AI, this is now the preferred method for creating state-of-the-art systems. Learning has another advantage, as we noted earlier: it allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone might allow.

A learning agent can be divided into four conceptual components:

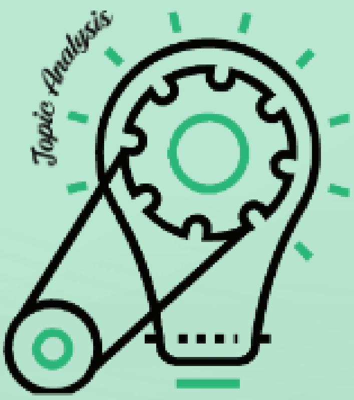
The most important distinction is between the **learning element**, which is responsible for making improvements, and the **performance element**, which is responsible for selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions. The learning element uses feedback from the **critic** on how the agent is doing and determines how the performance element should be modified to do better in the future.

The critic tells the learning element how well the agent is doing with respect to a fixed performance standard. The critic is necessary because the percepts themselves provide no indication of the agent's success.

The last component of the learning agent is the **problem generator**. It is responsible for suggesting actions that will lead to new and informative experiences. The point is that if the performance element had its way, it would keep doing the actions that are best, given what it knows. But if the agent is willing to explore a little and do some perhaps suboptimal actions in the short run, it might discover much better actions for the long run. The problem generator's job is to suggest these exploratory actions.

Adapted from - https://www.ques10.com/p/13489/what-are-the-basic-building-blocks-of-learning-a-1/

What did You Grasp?



1. What type of Agent is a:
   A) Loan Default Prediction System
   B) Google Maps route detection Algorithm
   C) Determining the shortest path in Google Maps
   D) An IR sensor that sends an alert every time someone crosses it
2. The performance of an agent can be improved by
   A) Learning
   B) Observing
   C) Perceiving
   D) None of the mentioned

9. Types of Environments

Fully observable vs. partially observable

If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

Single agent vs. multiagent

An agent solving a crossword puzzle by itself is clearly in a single-agent environment, whereas an agent playing chess is in a two agent environment.

Deterministic vs. stochastic

If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

Adapted from: https://enacademic.com/dic.nsf/enwiki/4030454

A task environment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure. Fully observable environments are convenient because the agent need not maintain any internal state to keep track of the world.

If the environment is partially observable, however, then it could *appear* to be stochastic. Most real situations are so complex that it is impossible to keep track of all the unobserved aspects; for practical purposes, they must be treated as stochastic.

Episodic vs. sequential	In an episodic task environment, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. Crucially, the next episode does not depend on the actions taken in previous episodes. Many classification tasks are episodic. In sequential environments, on the other hand, the current decision could affect all future decisions.
Static vs. dynamic	If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static. If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is semidynamic.
Discrete vs. continuous	The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.
Known vs. unknown	In a known environment, the outcomes for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.

Adapted from- https://enacademic.com/dic.nsf/enwiki/4030484

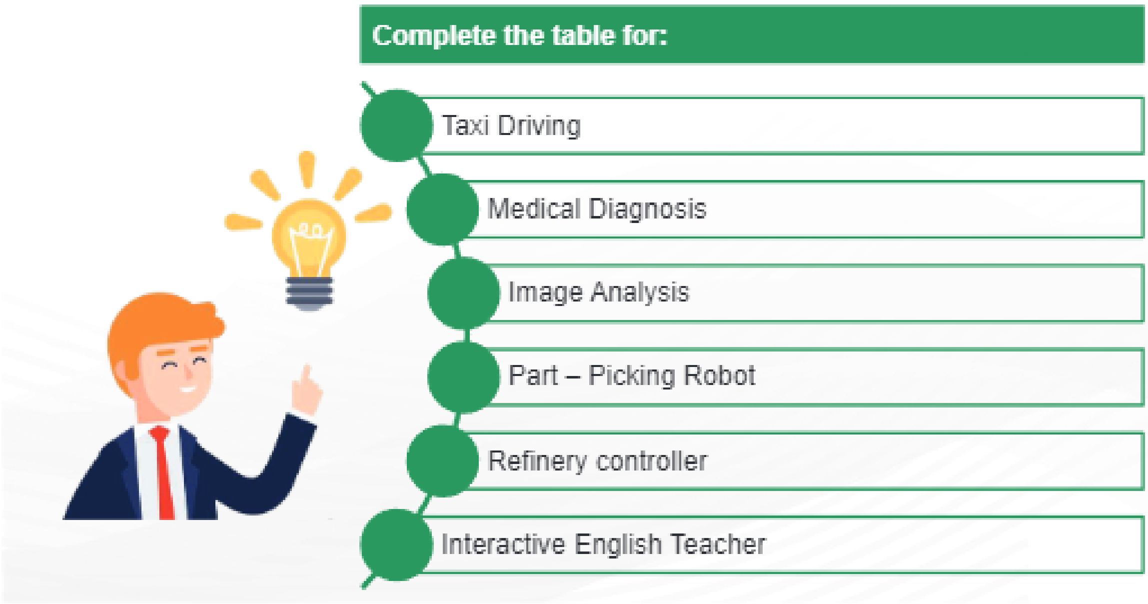
Chess and taxi driving are sequential: in both cases, short-term actions can have long-term consequences. Episodic environments are much simpler than sequential environments because the agent does not need to think ahead.

Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action, nor need it worry about the passage of time. Dynamic environments, on the other hand, are continuously asking the agent what it wants to do; if it hasn't decided yet, that counts as deciding to do nothing.

Taxi-driving actions are also continuous (steering angles, etc.). Input from digital cameras is discrete, strictly speaking, but is typically treated as representing continuously varying intensities and locations.

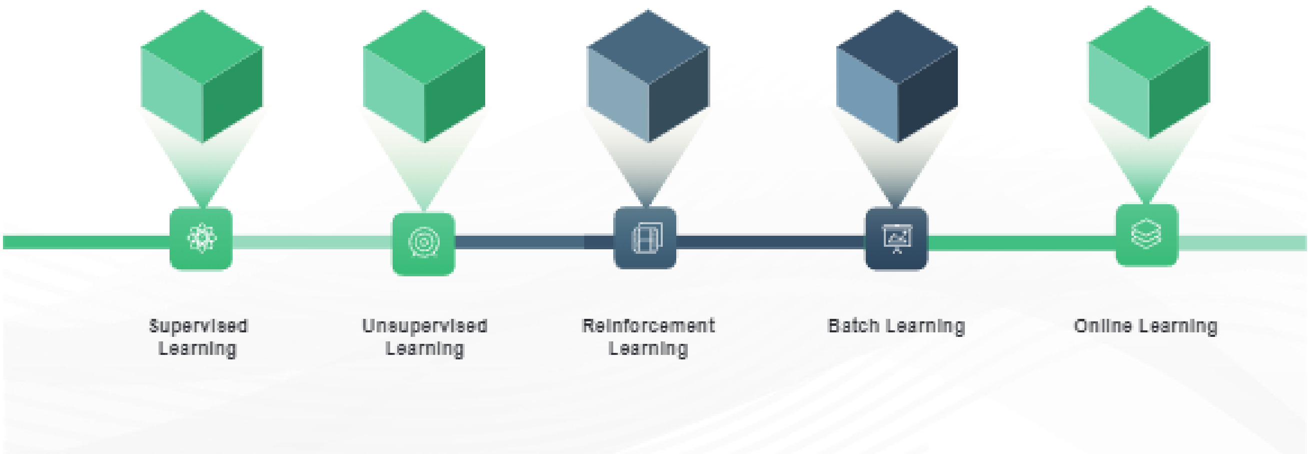
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle Chess with a clock	Fully Fully	Single Multi	Deterministic Deterministic	Sequential Sequential	Static Semi	Discrete Discrete
Poker Backgammon	Partially Fully	Multi Multi	Stochastic Stochastic	Sequential Sequential	Static Static	Discrete Discrete

Quiz



10. Types of Learning

The most important you need to focus on are as follows:



There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)

- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning)

10.1 Supervised Learning

Supervised Learning

Training data is fed with desired solutions, called *labels*. Supervised learning typically includes the task of *classification*.

Example: ***Spam filter***

In *supervised learning*, the training data you feed to the algorithm includes the desired solutions, called *labels*. A typical supervised learning task is *classification*. The spam filter is a good example of this: it is trained with many example emails along with their *class* (spam or ham), and it must learn how to classify new emails.

Another typical task is to predict a *target* numeric value, such as the price of a car, given a set of *features* (mileage, age, brand, etc.) called *predictors*. This sort of task is called *regression*. To train the system, you need to give it many examples of cars, including both their predictors and their labels.

10.2 Unsupervised Learning

Unsupervised Learning

In *unsupervised learning*, as you might guess, the training data is unlabeled. The system tries to learn without a teacher.

In *unsupervised learning*, as you might guess, the training data is unlabeled. The system tries to learn without a teacher.

For example, it might notice that 40% of your visitors are males who love comic books and generally read your blog in the evening, while 20% are young sci-fi lovers who visit during the weekends, and so on. If you use a *clustering* algorithm, it may also subdivide each group into smaller groups. This may help you target your posts for each group.

10.3 Semi Supervised Learning

Semi Supervised Learning

Some algorithms can deal with partially labelled training data, usually a lot of unlabelled data and a little bit of labelled data. This is called *semi supervised learning*

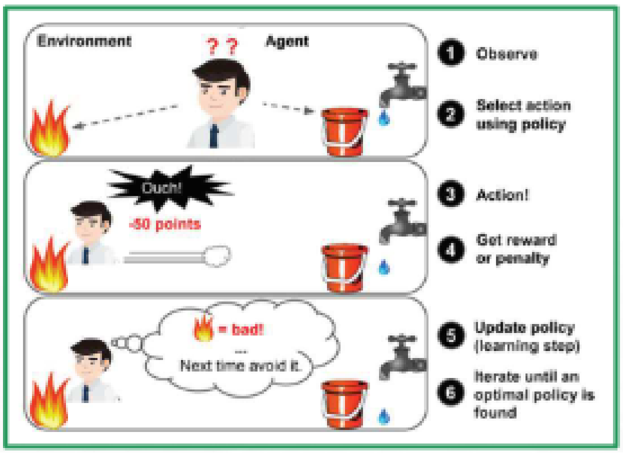
Some photo-hosting services, such as Google Photos, are good examples of semi supervised learning. Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in

photos 2, 5, and 7.

This is the unsupervised part of the algorithm (clustering). Now all the system needs is for you to tell it who these people are. Just one label per person,4 and it is able to name everyone in every photo, which is useful for searching photos.

10.4 Reinforcement Learning

Reinforcement Learning is a very different beast. The learning system, called an *agent* in this context, can observe the environment, select and perform actions, and get *rewards* in return (or *penalties* in the form of negative rewards). It must then learn by itself what is the best strategy, called a *policy*, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.



For example, many robots implement Reinforcement Learning algorithms to learn how to walk. DeepMind's AlphaGo program is also a good example of Reinforcement Learning; it made the headlines in March 2016 when it beat the world champion Lee Sedol at the game of *Go*.

10.5 Batch Learning

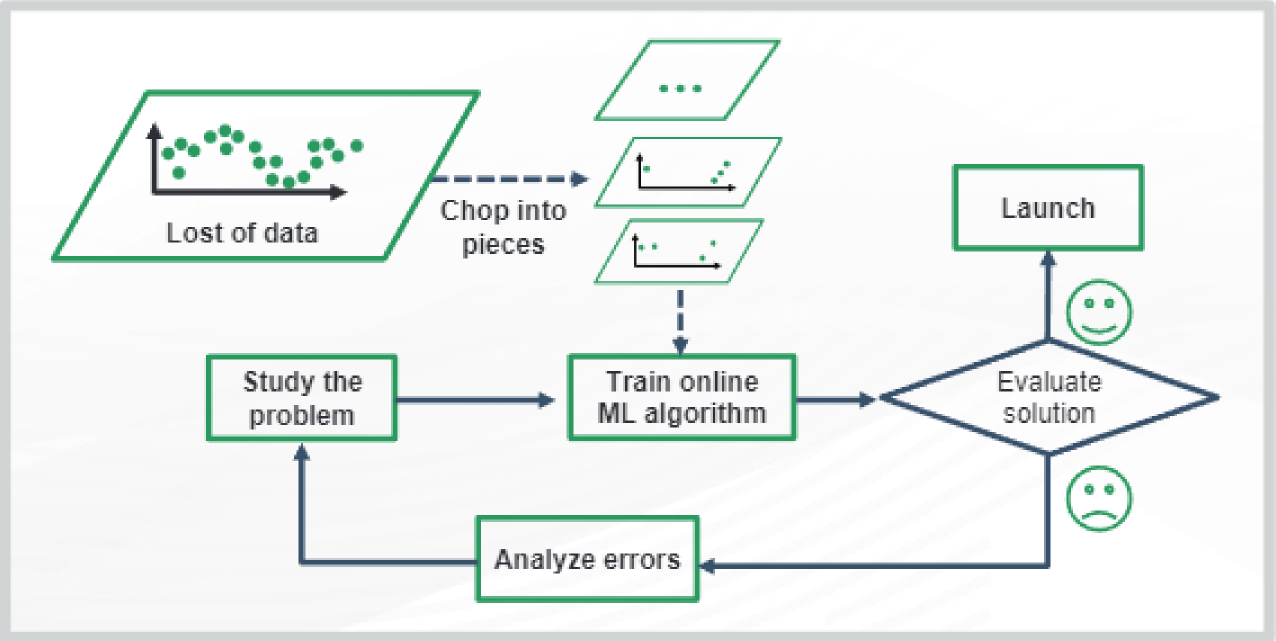
Batch Learning

In *batch learning*, the system is incapable of learning incrementally: it must be trained using all the available data. This will generally take a lot of time and computing resources, so it is typically done offline. First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned. This is called *offline learning*.

If you want a batch learning system to know about new data (such as a new type of spam), you need to train a new version of the system from scratch on the full dataset (not just the new data, but also the old data), then stop the old system and replace it with the new one.

10.6 Online Learning

In *online learning*, you train the system incrementally by feeding it data instances sequentially, either individually or by small groups called *mini-batches*. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives.

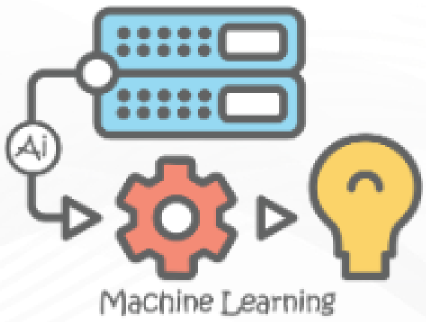


If you have limited computing resources: once an online learning system has learned about new data instances, it does not need them anymore, so you can discard them (unless you want to be able to roll back to a previous state and "replay" the data). This can save a huge amount of space.

Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory (this is called *out-of-core* learning). The algorithm loads part of the data, runs a training step on that data, and repeats the process until it has run on all of the data.

11. Areas of Artificial Intelligence

- Machine Learning
- Deep Learning
- Natural Language Processing
- Computer Vision
- Robotics

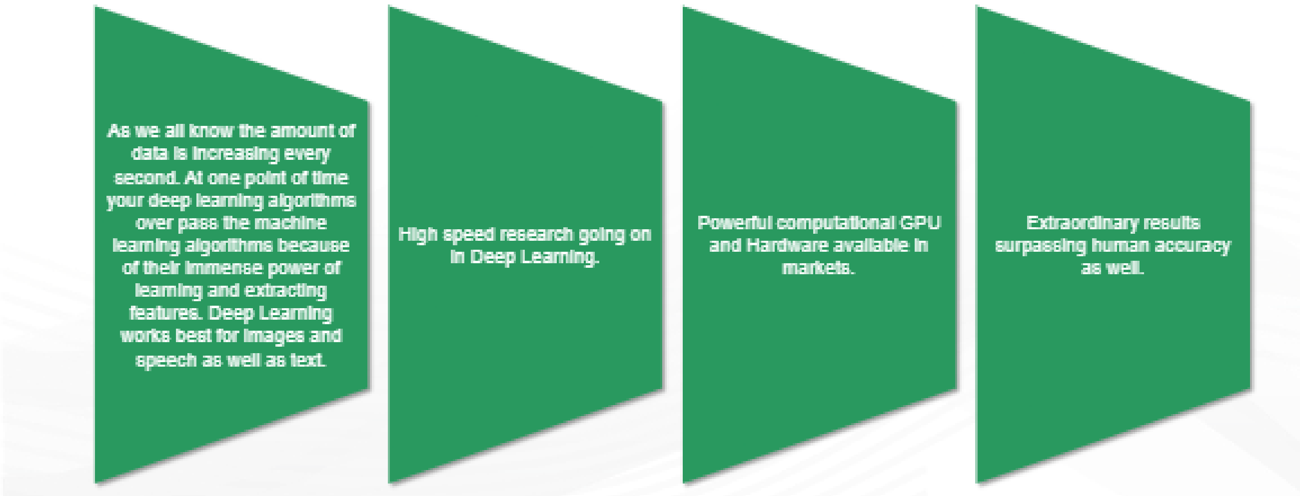


- Machine Learning - Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.
- Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. The basic intuition of deep learning is that we try to replicate working of a human brain. The neuron is the base of our brain. Similarly, we try to make a neural network, which can replicate the working of the human brain. More than one layer of neurons is called as Deep Learning. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labelled data and neural network architectures that contain many layers.

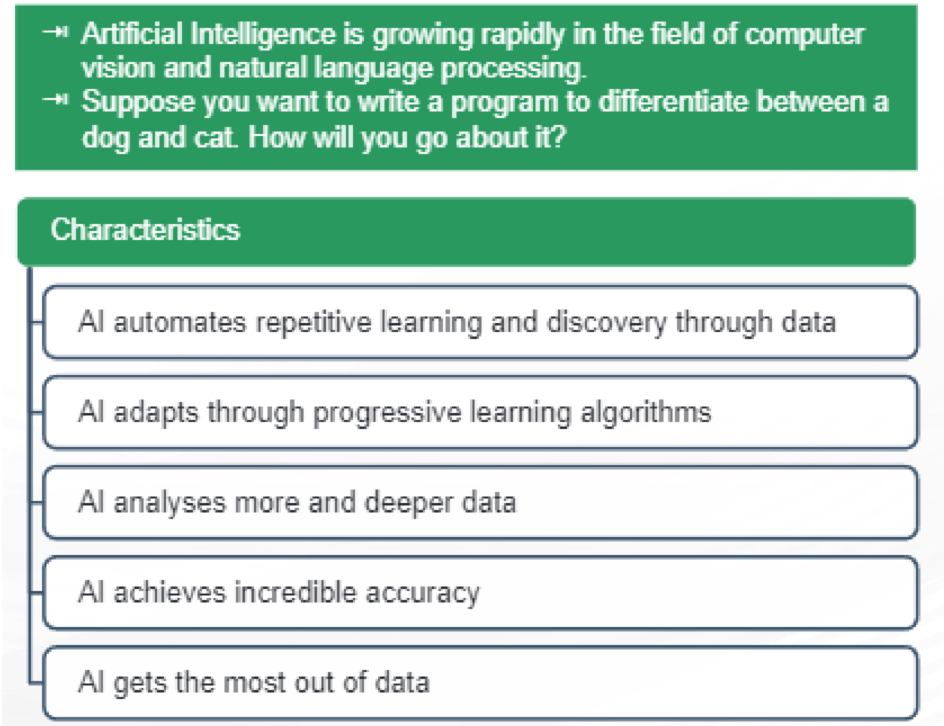
- Natural Language Processing – Performing a task related to test or language in which humans write or read. Like Google Translation, Consumer complaint classification.
- Computer Vision – Giving sight to machines related to images or videos or we can say performing tasks related to images/videos.
- Robotics – Adding Speech for response and vision for sight to any robot makes it even smarter as compared to simple mechanical robots.

11.1 Why is Deep Learning Gaining so much Popularity?



Go through the points mentioned on the slide.

12. Characteristics of Artificial Intelligence



Computer Vision – Performing a task related to images / videos

Natural Language Processing – Performing a task related to test or language in which humans write or read.

Speech Recognition – Performing task related to audio or speech.

Can a machine differentiate between a cat and dog?

You will have to hard code the rules that make a cat and dog different from each other. There could be thousands of rules you could state and still there might be a possibility you would miss a few points. Whereas, using Artificial intelligence techniques you give this power to your program. You have to just provide it with relevant data. It's now the responsibility of your code to learn the differentiating features from your data.

1. Instead of automating manual tasks, AI performs frequent, high-volume, computerized tasks reliably and without fatigue, which brings up better results and improvement in performance.
2. AI finds structure and regularities in data so that the algorithm acquires a skill The algorithm becomes a classifier or a predictor. So, just as the algorithm can teach itself how to play chess, it can teach itself what product to recommend next online.
3. Getting deeper into data helps the machine identify patterns which are not possible even for humans to find out. Also with increase in technologies like Big data and cloud computing, which can handle and process huge amounts of data making an immense rise in Artificial Intelligence.
4. Accuracy is achieved higher and higher with improvement in the algorithms you use. Also, with few self learning algorithms you improve the machine's accuracy itself.
5. When algorithms are self-learning, the data itself can become intellectual property. The answers are in the data; you just have to apply AI to get them out. Since the role of the data is now more important than ever before, it can create a competitive advantage. If you have the best data in a competitive industry, even if everyone is applying similar techniques, the best data will win.

12.1 Applications of Artificial Intelligence

There are immense applications of Artificial Intelligence

- Self driving cars (Tesla and Waymo)
- Chatbots
- Email Spam or Ham detection
- Disease (Cancer, Malaria) detection from Human Cell Images
- Google Assistant, Amazon Alexa, Siri, Google Home
- Sentiment Analysis
- Credit card / Loan default prediction.
- OCR (Optical character recognition)
- Face Recognition
- Recommendation Engines
- Latest and most interesting application bought up by Google

Self driving cars is one of the applications of computer vision where the car has enough intelligence to differentiate between a person, another vehicle or pit hole identifying which it has to change to speed or apply brakes.

Chatbots are one of the most rapidly growing application as it helps a lot of businesses to improve their customer satisfaction and also help exponentially in cost cutting. No actual person is sitting on the other to answer your queries, it's all done by machine. Understanding your problem and providing you with the appropriate solution. Eg., Zomato, Swiggy.

Email Spam Detection – Have you ever noticed how many emails you receive daily in your inbox. Do you open all of them? Thanks to Google (Gmail) that their system is smart enough to segregate your useful mails (ham) and spam. The main idea to look for specific patterns in mails to categorize them. According to Google reports their system is able to identify spam and ham with 99.9% accuracy which is amazing.

Disease Detection – The model is trained on thousands of images and is asked to learn the features which you think might cause cancer. After we are satisfied with the performance of the model we use it to test on the new images and classify them.

Google Assistant, Alexa and Siri are all voices based assistants working on speech to text and text to speech. It is intelligent enough to understand your voice and decode into commands. This will help you do task only with your voice. More the data provided to them the accuracy of the assistant improves. This type of learning is called reinforcement learning where the model is continuously learning with new input data.

Sentiment Analysis – Machine is able to understand the sentiment of the text or audio provided to it. Detecting feelings at a high level makes the machine act in a similar way. Three main sentiments are Positive, Negative or Neutral. Sentiment Analysis is highly used in product/restaurant reviews done by customers on e-commerce websites to find out how people are reacting to their product.

Credit Card default detection – Banks use this model to find out if a person will be able to payback the loan or not using the huge amount of past data and finding out patterns which will be helpful for the bank to save on their assets.

OCR – It is an application of artificial Intelligence where the machine is able to extract text from an image. Like getting your information from your Aadhar card you gave to the bank. Or Extracting text from your personal certificates.

Face Recognition – Smartphones having a face unlock and companies taking attendance using Face recognition.

Recommendations – Movies recommended by Netflix and products recommend by Flipkart and Amazon work on Machine Learning. Based on your past history and your likes they recommend you movies/products.

12.2 Real world Application by Big Companies

1. Banks using Loan and Credit card default prediction
2. Tesla, Uber and Waymo into Self driving cars
3. Haptik.ai – India's leading startup in chatbots
4. Cure.ai – Working in disease detection only with your X ray images.
5. Netflix and Amazon Prime – Movie recommendations
6. Google Duplex, Google Photos tagging

Go through the points mentioned on the slide.

Case Study-I

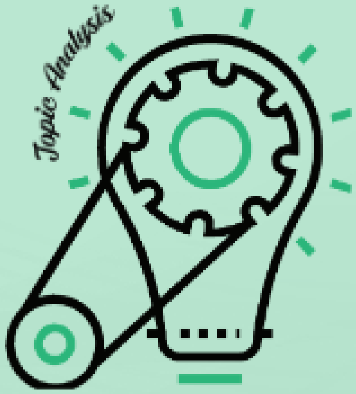
- Artificial Intelligence is highly being used in E-commerce industry and helping them earn millions.
- Here is a small case study how Amazon, Alibaba are using AI to increase their revenue. https://apiumhub.com/tech-blog-barcelona/artificial-intelligence-ecommerce/

Case Study-II

- With the rise of Digital Media, the one thing that also rises is the uses of fake videos and pictures. Adobe, with its tool Photoshop, leads the race of most use image manipulation software in fraud cases.
- Here is a small case study of Adobe coming up with a new feature or tool that uses machine learning to automatically detect when images of faces have been manipulated.

  https://techgrabyte.com/adobes-ai-photoshopped-images/

Test Your Understanding



1. Artificial Intelligence is all about?
   A) Coding
   B) Hardware
   C) Relevant Data
   D) Learning

2. Artificial Intelligence has its expansion in the following application?
   A) Planning and Scheduling
   B) Game Playing
   C) Robotics
   D) All of the Above

In a nutshell, we learnt

1. The basics of AI and its history
2. The basics of ML and its history
3. Different ML algorithms
4. The overlap of ML and AI
5. Practical applications of AI and ML