

CSC 402/502:

**INTRODUCTION TO
ARTIFICIAL INTELLIGENCE &
EXPERT SYSTEMS**

CSC 402/502: INTRODUCTION TO ARTIFICIAL INTELLIGENCE & EXPERT SYSTEMS

CONCEPT OF ARTIFICIAL INTELLIGENCE (AI)

Since the invention of computers or machines, their capability to perform various tasks went on growing exponentially. Humans have developed the power of computer systems in terms of their diverse working domains, their increasing speed, and reducing size with respect to time.

A branch of Computer Science named *Artificial Intelligence* pursues creating the computers or machines as intelligent as human beings.

What is Artificial Intelligence?

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

AI is accomplished by studying how human brain thinks and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Goals of AI

- To Create Expert Systems – The systems which exhibit intelligent behavior, learn, demonstrate, explain, and advice its users.
- To Implement Human Intelligence in Machines – Creating systems that understand, think, learn, and behave like humans.

AN OVERVIEW OF ARTIFICIAL INTELLIGENCE (AI)

According to the father of Artificial Intelligence, John McCarthy, it is “*The science and engineering of making intelligent machines, especially intelligent computer programs*”.

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think. AI is accomplished by studying how human brain thinks and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Artificial Intelligence (AI) is a multidisciplinary field of computer science that focuses on creating systems and machines capable of performing tasks that typically require human intelligence. These tasks include problem-solving, learning, reasoning, perception, language understanding,

and decision-making. AI has gained significant prominence in recent years due to advances in machine learning, deep learning, and the availability of vast amounts of data.

What is Artificial Intelligence AI?

Artificial Intelligence (AI) is the capability of machines or computers to exhibit human-like intelligence. This involves a range of technologies that empower machines to perceive, reason, act, and learn similarly to humans.

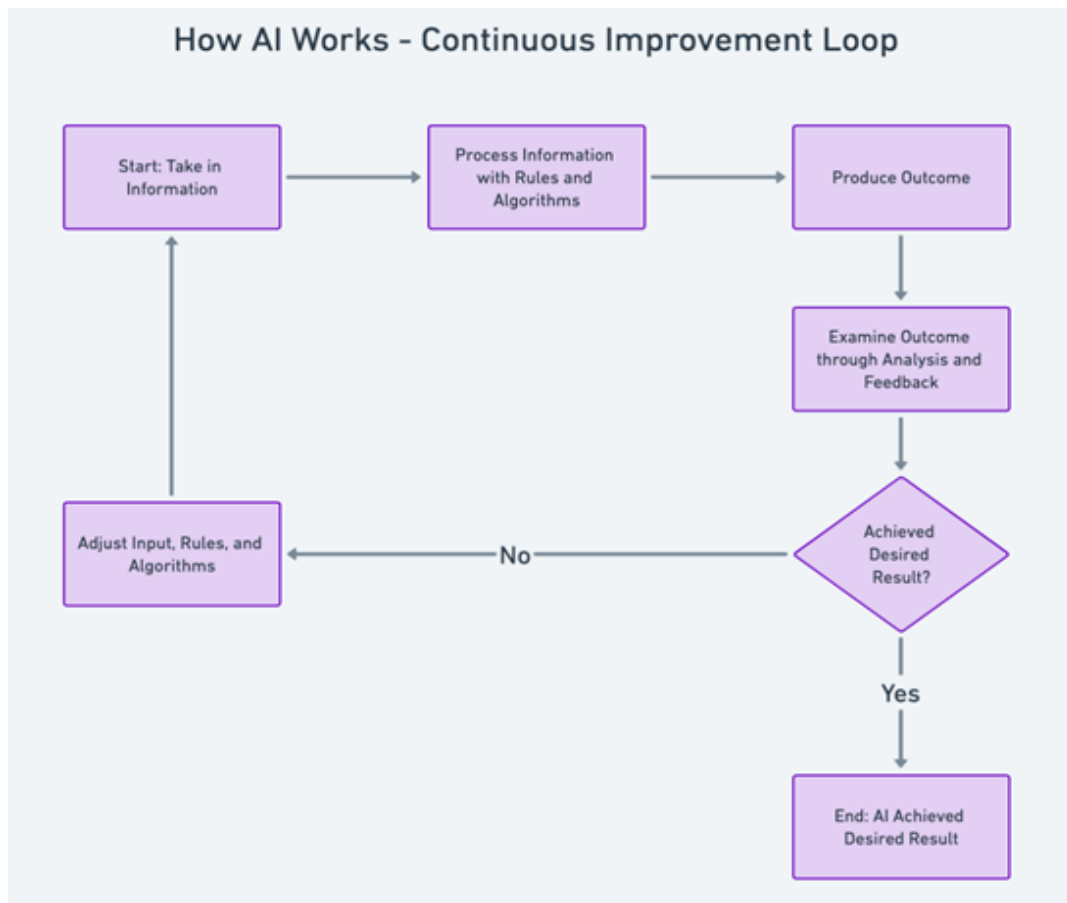
AI systems are designed to recognize and interpret their environment, make decisions, tackle complex challenges, learn from past data, and mimic behavioral patterns. These abilities enable them to perform tasks such as autonomously driving cars or using facial recognition for device security.

The AI world involves a mix of technologies including *machine learning*, *natural language processing*, and *computer vision*. These sophisticated technologies enable computers to process and understand human language, learn from examples, and make informed predictions.

While each technology within AI is advancing independently, its collective application alongside other technologies, data, analytics, and automation, has the potential to transform business operations. From enhancing supply chain efficiency to elevating customer service experiences, AI offers transformative solutions for businesses to achieve their objectives more effectively.

How does AI work?

AI systems start by taking in information, such as speech, text, or images. Once the data is received, the system uses rules and algorithms to process it. This involves interpreting the information, making predictions, and taking actions based on the input.



After processing, the AI system produces an outcome, which could be a success or failure in handling the input data. This result is then carefully examined through analysis, discovery, and feedback to understand how well the AI performed.

Using these assessments, the AI system makes adjustments to its input data, rules, and algorithms. This iterative process continues until the AI achieves the desired result.

In essence, AI works through a continuous loop of taking in data, processing it, evaluating the outcome, and refining its approach to improve over time.

Why is Artificial Intelligence Important?

Artificial intelligence (AI) holds significant importance due to its transformative impact on various aspects of our lives. It has revolutionized how we work, live, and engage in activities.

Businesses are increasingly relying on AI based solutions, such as AI based automation testing tools, to streamline operations and enhance efficiency. This technology has proven effective in automating tasks traditionally performed by humans, such as customer service, fraud detection, and quality control.

In many cases, AI surpasses human capabilities, especially in tasks that require precision and attention to detail. For instance, analyzing vast amounts of legal documents for accurate information is accomplished swiftly and with minimal errors by AI tools.

The ability of AI to process massive datasets provides valuable insights into business operations, uncovering aspects that might have gone unnoticed.

The growth of generative AI tools is particularly noteworthy, impacting diverse fields like education, marketing, and product design. These tools contribute to advancements by offering innovative solutions and expanding possibilities in various industries.

Prominent companies, including Alphabet, Apple, Microsoft, and Meta, recognize the crucial role of AI in maintaining competitiveness and improving operations. Google, an Alphabet subsidiary, integrates AI into its core functions, such as the search engine, self-driving cars through Waymo, and groundbreaking advancements in natural language processing, like the transformer neural network architecture developed by Google Brain.

Components of Artificial Intelligence (AI)

1. Learning

Learning in the context of AI is similar to how humans acquire knowledge but implemented computationally. One fundamental aspect of AI learning is the trial-and-error method. The AI system attempts various solutions to a problem and retains successful strategies in its database for future use.

Another form of learning is rote learning, where the AI memorizes specific items, such as problem-solving approaches, vocabulary, or foreign languages. This information is later generalized and applied in diverse contexts.

Example: In image recognition, AI learns to identify objects by analyzing and memorizing patterns in a vast dataset.

2. Reasoning and Decision Making

AI analyzes information and makes decisions through reasoning. This involves drawing inferences from given situations, categorized as inductive or deductive. Deductive inferences involve providing guaranteed conclusions, while inductive inferences deal with situations where outcomes are not certain.

Example: Chess-playing programs use reasoning to evaluate possible moves and make decisions based on the likely outcomes.

3. Problem Solving

AI's problem-solving ability involves techniques like planning, search, and optimization. Special-purpose methods tailor solutions to specific features of a given problem, while general-purpose methods address a wide range of diverse issues. Problem-solving in AI includes step-by-step reduction of differences between the current state and the goal state.

Example: Route optimization algorithms in navigation systems solve the problem of finding the most efficient path between two points.

4. Perception

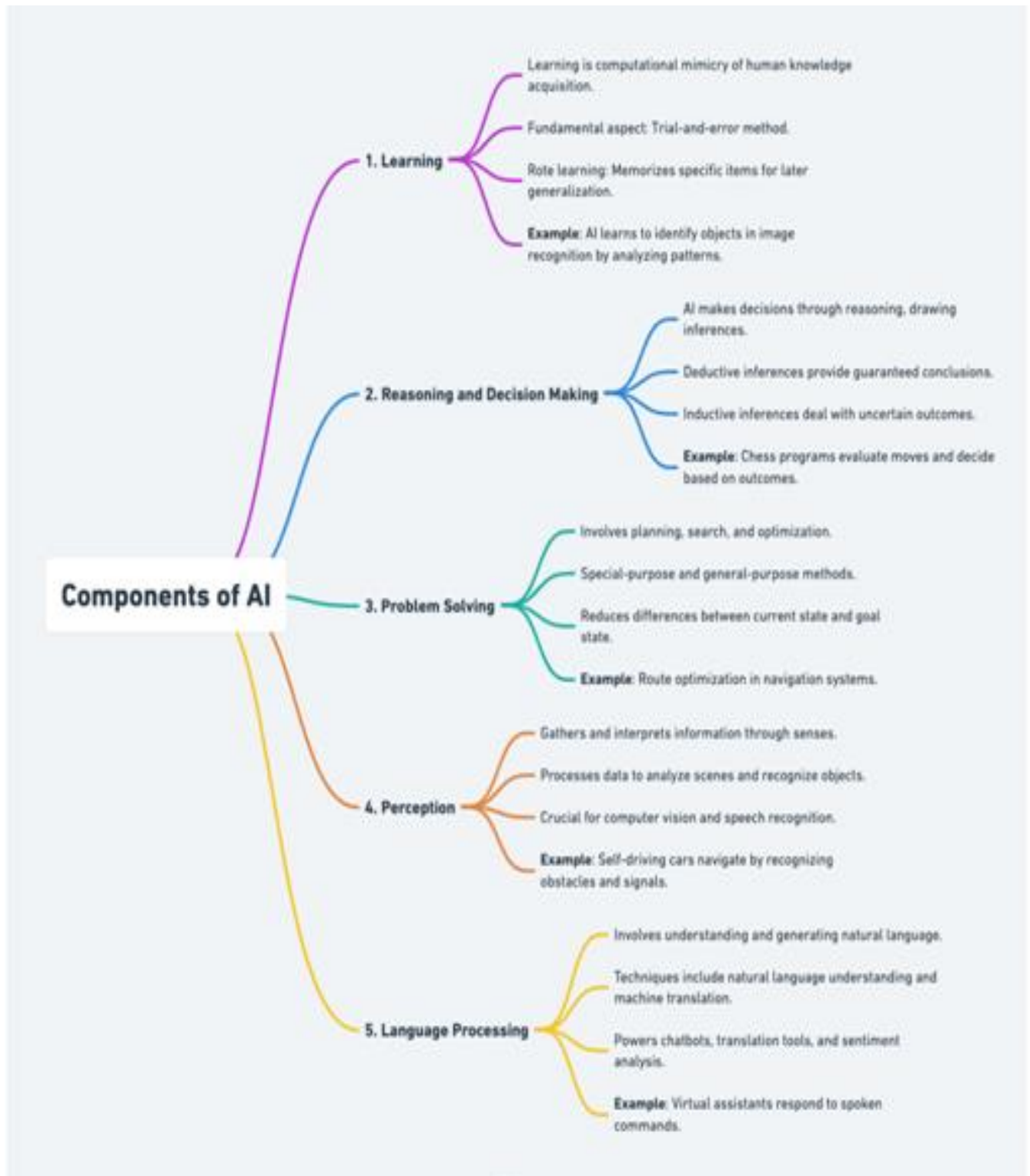
AI perceives its environment by gathering and interpreting information through sense organs, whether artificial or real. The system internally processes this data to analyze scenes, recognize objects, and understand their relationships and features. Perception is crucial for tasks like computer vision and speech recognition.

Example: Self-driving cars use perception to navigate roads by recognizing obstacles, pedestrians, and traffic signals.

5. Language Processing

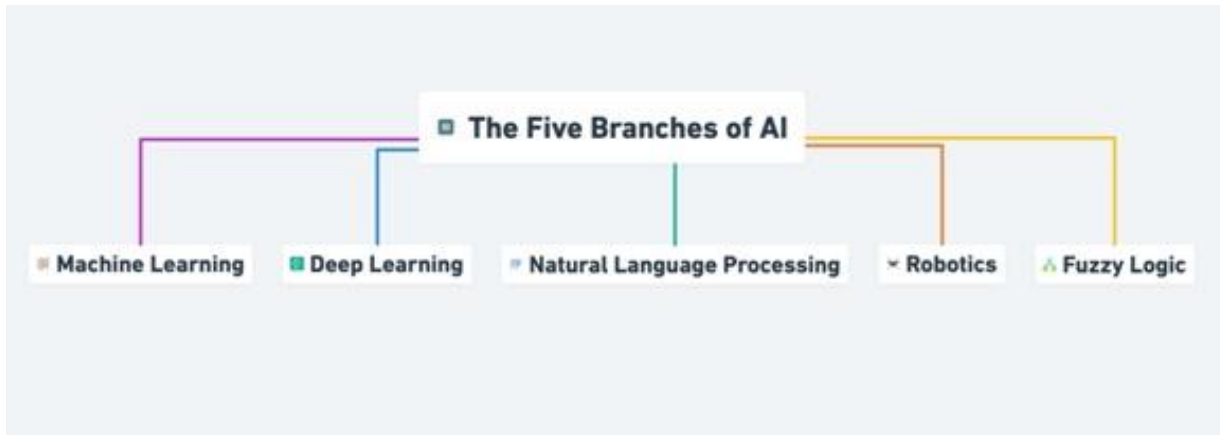
Language processing in AI involves understanding and generating natural language. Techniques like natural language understanding, machine translation, and text generation enable AI to interact with language effectively. This allows applications like chatbots, language translation tools, and sentiment analysis to function seamlessly.

Example: Virtual assistants understand and respond to spoken commands, showcasing language processing capabilities in AI.



The Five Branches of AI

Below are the five primary branches or subfields of Artificial Intelligence (AI), each contributing uniquely to the development and capabilities of intelligent systems.



1. Machine Learning

Machine Learning (ML) stands as a vital subset within AI, focusing on machines' capacity to learn autonomously from data and algorithms. ML leverages the foundational elements of AI to make decisions without explicit programming by humans, enhancing its adaptability and problem-solving capabilities.

2. Deep Learning

Deep Learning (DL) operates as a subset of machine learning, utilizing artificial neural networks (ANNs) inspired by the human brain. DL excels at extracting intricate features from data, leading to superior performance compared to traditional machine learning.

It minimizes human intervention further, although it requires substantial amounts of data. Common applications include natural language processing improvements in technologies like Amazon Alexa or Google Home.

3. Natural Language Processing

Natural Language Processing (NLP) is a critical aspect of AI enabling computers to comprehend spoken words and written text. NLP is pervasive in digital assistants, chatbots, virtual assistants, and spam detection. It plays a vital role in sentiment analysis, extracting emotions and attitudes related to a product or service from textual data.

4. Robotics

Robotics integrates AI to create and design autonomous or semi-autonomous robots and machines. This field often incorporates other AI technologies such as NLP and ML to enhance the capabilities of robots. AI-based robots are already making significant contributions to various industries, including healthcare, retail, and manufacturing, performing tasks with precision and efficiency.

5. Fuzzy Logic

Recognizing that the world is not always binary, Fuzzy Logic comes into play in AI to handle conditions that are not strictly true or false. Fuzzy Logic resolves uncertainties by employing if-then statements or rules, utilizing linguistic variables and fuzzy rules. An example is an automatic braking system, where fuzzy logic helps determine the optimal braking force based on imprecise information, contributing to the effectiveness of AI systems in real-world, less predictable scenarios.

Common AI Applications

1. *Smart Assistants*

Smart assistants, powered by AI, have become an integral part of daily life. These virtual companions, like Siri or Google Assistant, utilize natural language processing to answer queries, set reminders, and perform tasks, simplifying user interactions with technology.

Currently, chatbots are providing dynamic and intelligent analytics by interacting with visitors through engaging conversations. Notably, more than 67% of online visitors express a preference for chatbots.

2. *Financial Fraud Detection*

AI plays a crucial role in safeguarding financial transactions. By analyzing patterns, anomalies, and user behaviors, AI algorithms can detect and prevent fraudulent activities, ensuring the security of online banking, credit card transactions, and digital payments.

3. *Autonomous Vehicles*

AI is steering the future of transportation with the development of autonomous vehicles. Through advanced sensors and machine learning algorithms, self-driving cars can navigate roads, make real-time decisions, and enhance overall road safety, paving the way for a revolution in the automotive industry.

4. *Predictive Maintenance in Manufacturing*

In the manufacturing sector, AI is employed for predictive maintenance. By analyzing data from sensors and equipment, AI algorithms can predict when machinery is likely to fail, allowing businesses to schedule maintenance proactively, minimize downtime, and optimize production processes.

5. *Educational Technology (EdTech)*

AI is transforming the education landscape through personalized learning experiences. Educational technology platforms use AI to analyze students' progress, adapt learning materials to individual needs, and provide targeted feedback, enhancing the effectiveness of education in a digitally driven era.

The Key Challenges of AI

1. *AI Algorithm Bias*

AI systems rely on trained data, and as we delve into AI, biases within the data become apparent. These biases can include factors like race, gender, or ethnicity, affecting critical decisions like job interviews or loan approvals. To address this, companies like Microsoft and Facebook are developing anti-bias tools to identify and rectify biased perspectives in AI algorithms.

2. *Black Box Problem*

AI algorithms often function as black boxes, making their inner workings unclear. While we can see the predictions, understanding how the algorithm reaches those predictions is challenging. Techniques like 'local interpretable model-agnostic explanations' (LIME) are being employed to provide additional information for each prediction, enhancing algorithm transparency and reliability.

3. *Requirement of High Computing Power*

AI demands substantial computing power, especially for training models. The rise of deep learning algorithms necessitates additional cores and GPUs, often requiring supercomputers. Limited global availability and high costs of supercomputers hinder AI implementation in areas such as astronomy, where it could be valuable for tasks like asteroid tracking.

4. *Complicated AI Integration*

Integrating AI into existing corporate infrastructure is more complex than typical software updates. Compatibility with current programs and ensuring AI integration doesn't adversely affect existing processes are crucial. Establishing an efficient AI interface is necessary for the smooth management of AI infrastructure, but the transition can be challenging.

5. *Lack of Understanding of Implementation Strategies*

Despite AI's transformative potential, there is a lack of clear understanding regarding its implementation strategies. Businesses must identify areas benefiting from AI, set realistic goals, and establish feedback loops for continuous improvement. Corporate managers need a comprehensive understanding of AI technologies, trends, and limitations to implement AI solutions effectively.

6. *Legal Concerns*

Organizations must navigate legal concerns related to AI, especially concerning sensitive data collection. Even if data collection is legal, its aggregation can have negative implications.

Latest trends in AI in 2024

The AI landscape in 2024 is witnessing notable trends that impact various sectors:

1. *Advancements in IoT Integration:*

AI's influence on the Internet of Things (IoT) continues to expand. IoT devices, which connect to the internet, are becoming more adept at collecting and sharing data, fostering a seamless integration of smart products.

2. *Innovations in Medical Sensory Perception:*

A groundbreaking development in medicine involves the integration of artificial sensors with human senses. Technologies like corneal implants and laser tactile canes are emerging, showcasing AI's role in enhancing sensory perception for medical applications.

In the healthcare sector, the artificial intelligence (AI) market was valued at approximately 11 billion U.S. dollars globally. Projections indicate substantial growth, with expectations that the worldwide healthcare AI market will reach nearly 188 billion U.S. dollars by 2030.

Anticipated to expand at a compound annual growth rate of 37 percent from 2022 onwards, the sector is poised for significant advancements in the next decade according to Statista.

3. *Evolving Educational Technologies:*

AI's prominent role in education is growing, leading to an expansion of online and virtual learning options for individuals of all ages. This trend also involves the integration of AI-powered robots in educational settings, potentially transforming teaching methodologies.

4. *Enhancements in Business Analytics:*

Machine learning algorithms are increasingly refining Customer Relationship Management (CRM) platforms. The goal is to optimize customer service and satisfaction by leveraging AI's

capabilities for data analysis and insights. Anticipated to attain a market value of \$49 billion, the most promising developments are still ahead!

5. *Optimizing Manufacturing Processes:*

AI's impact on manufacturing is significant, with projections suggesting substantial market growth. Current applications include improving daily operations, introducing innovative products, and providing accurate forecasts for future financial needs within the manufacturing sector.

6. *Productivity Boost in Legal Practices:*

Attorneys are leveraging AI to enhance productivity and efficiency in legal processes. From data collection for contracts to identifying optimal strategies for specific legal situations, AI platforms are becoming invaluable tools for legal professionals.

Looking into the various facets of intelligence, unraveling the workings of AI, and exploring its diverse learning methods, we can affirm with certainty that AI stands poised as the harbinger of the future. It emerges as a steadfast ally in intelligent automation, making it a worthwhile pursuit to grasp the fundamentals of AI.

The impact of Artificial Intelligence extends beyond the realm of business, offering a gateway for enhancing services and building stronger connections with audiences. Beyond commercial applications, AI serves as the linchpin for automation, ranging from self-driving cars to voice-command-operated home systems.

Embracing a foundational understanding of AI is not just an investment in knowledge but a pathway to navigating the evolving landscape of innovation and efficiency.

What Contributes to AI?

When we think of artificial intelligence (AI), we think of machines, computer systems, algorithms, and models trying to get closer to human intelligence. Today, this space straddles many areas, like natural speech recognition, expert systems, machine learning, general AI applications, and more. The future, however, is mapped towards singularity:

There would come a time when AI would exceed our expectations and assumptions. We would meet the realm of Strong AI. Humans could leverage this form of intelligence at an unprecedented level and in exponential ways. But to get there, a lot of pieces of the AI puzzle would have to come together. Here are a few important ones.

Artificial Intelligence (AI) and Big Data

AI is nothing but intelligence that learns, re-learns, and reinforces its learning through the data that it gets. This data can either be specifically fed into the modes or designed to collect this data independently. Hence, the volume and quality of data are extremely important in determining the efficacy and relevance of AI for any purpose.

Big data is a new form of information asset, and it requires massive processing models and the right computing muscle to take care of its flow, velocity, scale, and density. For Big Data to expand and extend human intelligence through AI, we would need to master many dimensions of Big Data. They would include, but would not be limited to, speech recognition, IoT and sensor data, image recognition, unstructured data, and real-time data of many formats.

The space of Big Data is growing rapidly and in multiple directions. As per reports from MarketsandMarkets, it can grow from \$138.9 billion in 2020 to \$229.4 billion by 2025 and enable companies to increase operational efficiencies and reduce costs. Of course, that should help the growth of AI too.

Artificial Intelligence (AI) and innovation

AI is much more than the fundamentals of the technology that power it. Its real impact and possibility lie in the way it is augmented consistently. It has to get the right research and creative impetus to explore new models, processing modes, algorithm designs, and enablers. There are many constraints for AI, as we see them today. They deter many enterprises from leveraging AI with its full potential.

Artificial Intelligence (AI) and talent

The first reflex thought that most professionals feel when they hear of AI is displacement because of AI. This is not just because AI can take over repetitive jobs. But because AI applications would need a new set of skills and mindset, professionals would have to get ready for it. According to a Gartner Research Circle survey, about 56 percent of respondents felt that acquiring new skills will be required to do both existing and newly created jobs. Interestingly, the O'Reilly annual AI Adoption in the Enterprise 2021 survey says that demand for AI expertise has exceeded supply. Companies are feeling the skills shortage most acutely in the areas of ML modeling and data science (52 percent), understanding business use cases (49 percent), and data engineering (42 percent). Another reflection of the AI skills gap is the finding that the percentage of companies with AI products in production over the last year (25 percent) is flat when compared with 2020 (26 percent) and 2019 (27 percent).

Artificial Intelligence (AI) and investments

A lot hinges on how much the top leaders of an organization feel excited about AI. Investments will drive more innovation and more capabilities ahead. For instance, the McKinsey report shows that AI top performers invest more of their digital budgets in AI than their counterparts and are more likely to increase their AI investments in the next three years. As to the distinct aspects of these high performers, there is another point worth noting. The ability to develop AI solutions in-house, instead of buying solutions – and they typically employ more AI-related talent, such as data engineers, data architects, and translators, than their counterparts. These companies have also built a standardized end-to-end platform for AI-related data science, data engineering, and application development. So enterprises would have to find that kind of enthusiasm as budgets to leverage AI strongly.

Artificial Intelligence (AI) and processing power

AI depends a lot on how models use data, learn, and apply their intelligence. That is where AI would need higher and wider processing alternatives. Especially in the area of real-time data. The McKinsey report also talks about the adoption of deep learning—a type of machine learning that uses neural networks and can sometimes deliver superior results. As of now, just 16 percent of companies seem to have taken deep learning beyond the piloting stage. Only high-tech and

telecom companies are leading the charge here, with 30 percent embedded deep-learning capabilities. Opening up to new technology paradigms would be a precursor to real growth in AI.

Artificial Intelligence (AI) And Improved Algorithms

Companies are still not sure about handling a lot of AI ramifications. That's where the algorithms need to evolve and iron out these kinks. In the McKinsey report, we see how companies are still worried about abort risks like explainability (41 percent), privacy (39 percent), equity and fairness (24 percent), and cybersecurity (62 percent). In fact, as per the O'Reilly survey, mature organizations checked for unexpected outcomes or predictions, interpretability and transparency, and model degradation. Privacy and fairness, bias, and ethics were also ranked above 50 percent in terms of concerns. We need algorithms that can address these issues and bolster the excitement that enterprises have towards AI. If all these ducks fall in a row, AI would swim to new shores of possibility and impact soon.

Applications of AI

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc. AI is making our daily life more comfortable and faster.

Following are some sectors which have the application of Artificial Intelligence:



1. AI in Astronomy

- ***Automated Celestial Object Identification:*** AI systems can automatically identify and classify celestial objects in astronomical images, aiding in discovering new stars, galaxies, and other cosmic phenomena. In simple words, AI can spot and sort out things in space by

looking at pictures. It's like having a cosmic detective that finds new stars, galaxies, and other mysterious objects without human help.

- **Exoplanet Hunting:** AI helps astronomers find planets outside our solar system by looking at lots of data. It can notice tiny changes in the light from stars, which tell us there might be planets around them, such as those caused by exoplanet transits.
- **Analyzing Space Information:** AI plays a crucial role in the study of space. It assists scientists by carefully examining vast amounts of complex data gathered from space observations. This helps astronomers uncover sophisticated patterns, unusual phenomena, and connections that might be otherwise very difficult to notice. Essentially, AI acts as a dedicated assistant, sifting through the cosmic data haystack to find the valuable needles of knowledge.
- **Watching Space Events in Real-time:** AI-powered tools can keep a constant eye on the night sky, looking out for sudden happenings like exploding stars (supernovae) or bursts of powerful gamma rays. This allows scientists to quickly study these events in more detail when they occur.
- **Making Telescopes Smarter:** AI is like a brain for telescopes. It helps them work better by changing their settings on the fly. For example, if the weather gets cloudy or if scientists want to study something specific in space, AI can adjust the telescope to get the best results. It's like having a telescope that can think and adapt to the situation.

2. AI in Healthcare

- **Helping Doctors See inside the Body Better:** AI is like a super helper for doctors when they look at pictures of the inside of a patient's body, like X-rays or MRIs. It uses smart algorithms to find things like problems, tumors, or broken bones very accurately. This means doctors can figure out what's going on faster and more accurately, which is great for patients and for better diagnosis.
- **Detecting Health Problems Early:** AI acts as a health detective. It looks at your health information to find out if you might get certain diseases in the future. When it sees a high risk, doctors can step in early to help you stay healthy. This is really important for conditions like diabetes and heart problems because catching them at this time means better treatment and less trouble for the patient.
- **Developing Medications Quickly and Cost-Effectively:** AI acts like a super scientist in the lab. It uses certain algorithms to predict how different chemicals can fight diseases. This helps us make new medicines much quicker and at a low cost. So people can get the treatments they need sooner, and it doesn't cost as much money to manufacture them.
- **Personalized Treatment Plans:** AI looks at your health information, like your genes, what happened to you before, and how you've responded to treatments. Then, it makes a special plan just for you. This means your treatment works better and doesn't give you as many problems. It's like having a personalized health coach, which helps in avoiding complications caused by improperly prescribed medicine.
- **Managing Hospital Functions and Resources:** AI acts like a manager for hospitals. It helps with things like when patients come in, where to put resources like doctors and

supplies, and how to make sure everything runs well. It can even guess how many patients might come in ahead of time, so hospitals use their staff and resources in the best way possible.

3. AI in Gaming

- **Smart Game Characters:** AI is like the brains behind game characters that aren't controlled by players. They make these characters, called NPCs, act more like real people or clever enemies. They can learn from what players do and change their behavior, which makes games more exciting and lifelike. Imagine playing a game where the bad guys learn and adapt to your moves - that's what AI does.
 - **Creating Game Worlds with AI:** AI can make parts of video games all on its own. It can create levels, maps, and places to explore without people having to make them by hand. This means games can have bigger and more interesting worlds because AI does a lot of the work, kind of like a game world builder. It helps game developers, too.
 - **Making Games Look and Feel Real:** AI helps to make games look and act more like the real world. They create graphics that look just like the things we see, and they make how things move in games feel realistic, like in real life. They even guess what players might do next so the game looks smooth and natural.
-

4. AI in Finance

- **Identifying and Prevention of Fraud:** AI keeps an eye on bank transactions all the time. They act like super detectives who can spot strange things happening with money, like someone using a credit card in a weird way. When they see something fishy, they raise the alarm and help the bank stop bad people from stealing money. This happens really fast, without needing people to check every transaction.
 - **Automated Trading:** AI helps a skilled trader who works automatically. It uses various algorithms to swiftly buy and sell stocks while analyzing all the market information. This boosts trading strategies, making investments more efficient and profitable.
 - **Risk Control:** AI helps in examining lots of data to check how risky something is, like giving out loans or making investments. It looks at things like whether someone can pay back a loan or how safe an investment is. This helps banks and investment firms make smarter choices so they don't lose money and can help others save and grow their money.
-

5. AI in Data Security

- **Anomaly Detection:** AI works as a digital detective. It looks at big piles of data and watches for anything strange or out of the ordinary, like someone sneaking into a digital vault or trying to steal secrets. When it sees something fishy, it raises the alarm, helping to keep important data safe from cyber-attacks.
- **Predicting Threats:** AI looks at past troubles and keeps an eye on new dangers that are popping up. By doing this, it can predict what bad things might happen in the future, like a security breach or a cyberattack. This way, companies can get ready in advance to protect their important data, sort of like putting up a strong fortress before any attack happens.

- **Automated Safety Response:** AI acts like a digital guardian that can respond when there's trouble. If it sees something bad happening, like a cyberattack, it can automatically take action. It might isolate the part that's under attack. This way, it keeps your important stuff safe in the digital world.
-

6. AI in Social Media

- **Smart Suggestions:** AI helps as a guide on social media. It watches what you like and what you do, and then it suggests things you might enjoy, like posts, videos, or ads. It acts as someone who knows your tastes and shows you stuff you're really into, making your social media experience more enjoyable and personalized.
 - **Virtual Assistants and Chatbots:** AI chatbots and virtual assistants act as digital helpers on social media. They're quick to respond and can talk to you just like a real person. They answer your questions, share information, and even help with problems. It's like having an assistant available 24/7, making your social media experience smoother and more helpful.
 - **Sentiment Analysis:** AI can figure out how people feel on social media. It looks at what they say in comments and posts and decides if it's a happy, sad, or neutral kind of message. This helps companies understand what people think so they can react in the right way. It's like having a mood gauge for the internet so businesses can make their customers happier.
 - **Trend Analysis:** AI keeps track of all the chats and what's popular right now. This helps companies and regular folks understand what everyone's thinking and talking about. It acts as a social media news reporter that keeps customers in the loop about what's hot and what people are buzzing about.
-

7. AI in Travel & Transport

- **Optimization of Route:** AI plays a crucial role in optimizing travel routes, be it for parcel deliveries, public transportation, or personal trips. It efficiently calculates the swiftest and most economical paths from one point to another point, resulting in reduced travel time, minimized fuel consumption, and cost savings. Essentially, it serves as a pocket-sized travel advisor, enhancing the speed and budget-friendliness of your journeys.
- **Smart Security Screening:** AI helps in keeping traveling safely. It uses special skills to scan bags and people quickly. It can spot things that might be dangerous and make security checks faster and smoother. This means you can fly knowing that the airport is working hard to keep you safe without making your travel a hassle.
- **Chatbots for Travel Support:** AI chatbots are like digital travel helpers. These chatbots are capable of aiding you in various tasks such as reserving tickets, suggesting interesting destinations to explore, and providing responses to your inquiries, much like an affable travel consultant. This elevates the convenience and pleasure of your travel adventures, as you can access assistance whenever it's required, even during late-night hours.
- **AI Prevents Breakdowns:** AI works like a fortune teller for machines like cars, planes, and roads. It predicts when they might get sick and need fixing. This way, we can fix them

before they break down and cause problems. It keeps everything running smoothly, making travel safer and saving a lot of time and money.

8. AI in Automotive Industry

- **Self-Driving Cars:** AI is like the brain of self-driving cars. It looks at what's happening around the car using various sensors and decides what the car should do, like turning or stopping. It's like having a super-smart driver that doesn't need a person. This makes cars drive on their own, making travel more convenient and safer because there's no need for a human to steer.
 - **Advanced Driver Assistance Systems (ADAS):** AI adds extra smarts to your car to keep you safe. It possesses the capability to autonomously adjust your vehicle's speed while on the highway, assist in maintaining your lane, and swiftly engage the brakes when detecting potential hazards. These intelligent functionalities function akin to a co-pilot, ensuring your safety by preventing accidents and ensuring your safe arrival at your intended destination.
 - **Streamlining Production Processes:** AI watches over machines, checks if they're healthy, and makes sure they don't break. It also helps with ordering materials and makes sure everything is made just right. This makes things faster, cheaper, and better quality, like having a super factory manager.
 - **Voice Recognition:** AI-driven voice recognition systems allow drivers to control various functions in their vehicles, such as navigation, music, and communication, using natural language.
-

9. AI in Robotics:

- **Self-Moving Robots:** AI makes robots really smart at moving around on their own. It's like giving them a built-in GPS and a clever brain. They can figure out where to go and how to get there without bumping into things or needing a person to show them the way. This helps them do tasks like delivering packages or exploring places on their own, making them super independent.
 - **Object Recognition and Manipulation:** AI gives robots sharp eyes and clever hands. It helps them see objects clearly and then pick them up and move them just right. This is super useful, especially in places like warehouses, where they can do things like sorting and packing items accurately.
 - **Collaboration of Humans and Robots:** AI makes it possible for robots to be great team players with people. They can work alongside humans, helping out and learning from them. If a person does something, the robot can understand and follow their lead. This makes workplaces safer and more efficient, like having a trusty robot colleague who understands and supports you.
-

10. AI in Entertainment

- **Recommendation of Content:** AI looks at what customers have liked before, such as movies or music, and suggests new things that they might enjoy. It's like having a personal

entertainment guide, making their experience more enjoyable by offering just what they like.

- **AI as a Creative Assistant:** AI acts as a creative sidekick for artists and creators. It can make music, art, and videos or help improve what they create. It's like having a helper that speeds up the creative process, making it easier to bring new ideas to life. This way, artists can focus more on their vision, and AI handles the technical bits.
 - **Live Event and Performance Enhancements:** AI makes live events and performances even cooler. It can translate what people are saying in real time, add cool effects that blend with what's happening, and even predict what the audience will like. This makes shows and events more exciting and enjoyable for everyone there. It's like having a magic touch that brings performances to life in new and amazing ways.
-

11. AI in Agriculture

- **Crop Observation and Control:** AI, with the help of various sensors, acts as a guardian for crops on the farm. It keeps an eye on them, making sure they're healthy and growing well. It tells farmers when it's the best time to plant, water, and harvest the most crops. It's like having a farm expert who ensures the fields are super productive so farmers can get the most out of their hard work.
 - **Smart Farming for Efficiency:** AI makes farming super efficient. It helps farmers use just the right amount of things like fertilizer and pesticides, not too much and not too little. This means there's less waste, and the crops grow better. It's like having a precise chef in the field, making sure everything is just perfect for the plants to thrive and produce lots of food.
 - **Automated Farming:** AI controls a number of machines like tractors and drones. These machines can plant seeds, remove weeds, and spray stuff on crops all by themselves. They do it super well and exactly as needed, like having expert farmers who never get tired and work perfectly, making farming easier and more efficient.
 - **Monitoring Livestock:** AI uses special sensors and smart data analysis to make sure they're healthy and happy. If anything is wrong, it alerts the farmer. This way, the animals are well taken care of, and the farm can run smoothly. It's like having a watchful friend for the animals, making sure they're okay and the farm works better.
-

12. AI in E-commerce

- **Personalized Product Suggestions:** AI looks at what you've looked at and bought before and suggests things you might really like. It's like having a personal shopper who knows your style, making your online shopping more fun and helping you discover new things you might want to buy. Plus, it's great for the store because it helps them sell more, and as a customer, it saves your time.
- **Managing Inventory:** AI takes care of a store's shelves. It predicts how much of each product people will buy and automatically orders more when needed. In this manner, there exists an optimal balance of products, preventing excessive stock that ties up funds while

also ensuring an adequate supply to prevent customers from leaving without making a purchase.

- **Dynamic Pricing:** Artificial intelligence dynamically adjusts pricing according to demand, market competition, and inventory levels, ensuring customers receive optimal value while enhancing the store's profitability.
-

13. AI in education:

- **Education Content Creation:** AI acts as a teaching assistant for educators. It helps them make things like quizzes, lesson plans, and study materials. This makes teaching easier and better because educators have more time for students, and the materials are top-notch. It's like having a super-efficient helper who does the paperwork, leaving teachers more time to inspire students.
 - **Virtual Learning Assistants:** AI is there to answer questions, explain things, and offer help whenever students need it, day or night. This makes learning easier and more fun because students have someone to turn to whenever they're stuck. It also takes some pressure off teachers because AI can handle common questions, leaving more time for personalized teaching.
 - **Automated Assessment and Instant Feedback:** AI acts like a super-speedy homework checker. It looks at your assignments and tests and gives you grades and feedback right away. This aids in gauging your progress and pinpointing areas for potential enhancement. Furthermore, it alleviates some of your teacher's grading responsibilities, allowing them to dedicate more time to teaching rather than paper evaluation.
 - **Customized Learning Routes:** AI figures out what you're good at and where you might need extra help. Then, it gives you the right stuff to learn and the best way to learn it. This makes learning easier and more fun.
-

The applications of AI are vast and diverse, touching nearly every aspect of our lives. From healthcare to finance, astronomy to gaming, and transportation to entertainment, AI is reshaping industries and propelling us into a future where the possibilities seem limitless. As AI continues to advance, its impact on society is poised to grow, promising increased efficiency, better decision-making, and innovative solutions to some of our most pressing challenges. Embracing and responsibly harnessing the power of AI will be key to unlocking its full potential and ensuring a brighter future for all.

INTELLIGENT SYSTEMS IN AI

Intelligent systems in artificial intelligence (AI) represent a broad class of systems equipped with algorithms that can perform tasks typically requiring human intelligence. These systems span various domains from robotics to data analysis, playing a pivotal role in driving innovation across industries. Here, we delve into the essence of intelligent systems, their core components, applications, and the future trajectory of this transformative technology.

An intelligent system in AI is a technology equipped with the capability to gather data, process it, and make decisions or perform actions based on that data. At its core, an intelligent system mimics the cognitive functions of human beings, such as learning from experience, understanding complex concepts, solving problems, and making decisions.

Understanding Intelligence

The notion of intelligence used in reference to both men and machines entails the capacity to acquire knowledge, perceive and comprehend information, deduce, rectify problems, educate oneself, and take charge of a new situation. In AI, "intelligence" is not merely the capacity to process data but more of making good or profound insights and decisions to their information.

Applications of Intelligent Systems

Intelligent systems are poised to fill a growing number of roles in today's society, including:

- Factory automation
- Field and service robotics
- Assistive robotics
- Military applications
- Medical care
- Education
- Entertainment
- Visual inspection
- Character recognition
- Human identification using various biometric modalities (e.g. face, fingerprint, iris, hand)
- Visual surveillance
- Intelligent transportation

Challenges in Intelligent Systems

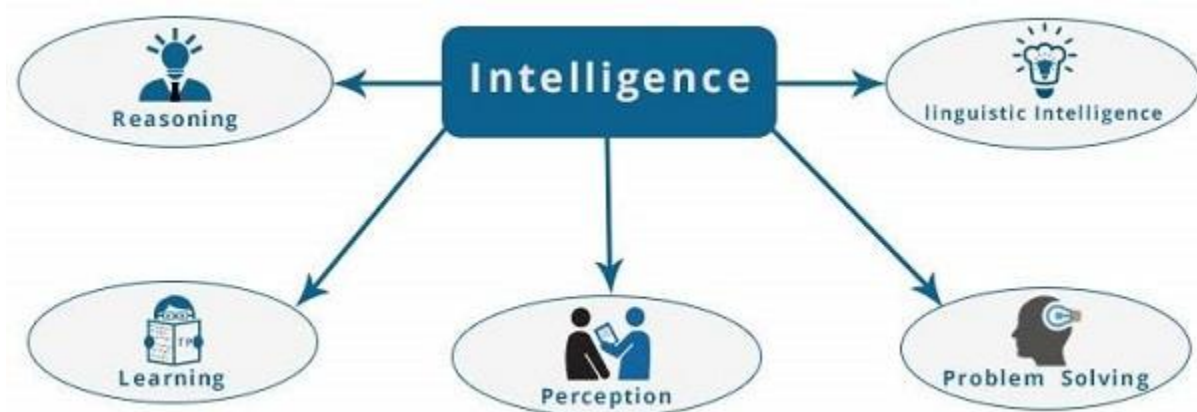
Research in intelligent systems faces numerous challenges, many of which relate to representing a dynamic physical world computationally.

1. **Uncertainty:** Physical sensors/actuators provide limited, noisy and inaccurate information/action. Therefore, any actions the system takes may be incorrect both due to noise in the sensors and due to the limitations in executing those actions.
2. **Dynamic world:** The physical world changes continuously, requiring that decisions be made at fast time scales to accommodate for the changes in the environment.
3. **Time-consuming computation:** Searching for the optimal path to a goal requires extensive search through a very large state space, which is computationally expensive. The drawback of spending too much time on computation is that the world may change in the meantime, thus rendering the computed plan obsolete.
4. **Mapping:** A lot of information is lost in the transformation from the 3D world to the 2D world. Computer vision must deal with challenges including changes in perspective, lighting and scale; background clutter or motion; and grouping items with intra/inter-class variation.

Components of Intelligence

The components of intelligence, as understood in the context of psychology and cognitive science, are the fundamental elements that collectively define and influence the capabilities and performance of human intelligence. The intelligence is intangible. It is composed of:

- Reasoning
- Learning
- Problem Solving
- Perception
- Linguistic Intelligence



1. **Reasoning** – It is the set of processes that enables us to provide basis for judgment, making decisions, and prediction. Reasoning involves drawing conclusions from evidence or arguments. It includes inductive reasoning, which builds general conclusions from specific examples, and deductive reasoning, which applies general principles to specific cases. There are broadly two types :

<i>Inductive Reasoning</i>	<i>Deductive Reasoning</i>
It conducts specific observations to makes broad general statements.	It starts with a general statement and examines the possibilities to reach a specific, logical conclusion.
Even if all of the premises are true in a statement, inductive reasoning allows for the conclusion to be false.	If something is true of a class of things in general, it is also true for all members of that class.
Example – "Nita is a teacher. Nita is studious. Therefore, All teachers are studious."	Example – "All women of age above 60 years are grandmothers. Shalini is 65 years. Therefore, Shalini is a grandmother."

2. **Learning**

In intelligent systems, learning is pivotal for adapting to new environments and improving decision-making. Learning is the process by which we acquire new

information or modify existing knowledge, skills, and behaviors. It can occur through direct experience, observation, or instruction, and is fundamental to adapting to new situations. The following are brief overview of common learning paradigms:

- i. **Supervised Learning:** Involves training a model on a dataset that includes both inputs and expected outputs, enabling the system to predict outcomes based on past data. Common applications include facial recognition and spam filtering.
- ii. **Unsupervised Learning:** Focuses on identifying patterns and structures in data without predefined labels. It's used for clustering and anomaly detection, such as in market segmentation or fraud detection.
- iii. **Reinforcement Learning:** Employs a system of rewards and penalties to foster environment-specific decision-making. This method is vital in robotics and complex game systems where the AI must adapt strategies based on dynamic conditions.
- iv. **Deep Learning:** Utilizes neural networks with multiple layers to analyze large volumes of data, enhancing capabilities in image and speech recognition technologies.
- v. **Transfer Learning:** Applies knowledge acquired from one task to different but related problems, enhancing efficiency and adaptability across various applications with minimal additional training.

3. Perception

Being able to perceive means that intelligent systems are able to give sense to the data received via their overwhelming number of senses and comprehend their surroundings. It is the process of acquiring, interpreting, selecting, and organizing sensory information. Perception is the cognitive process of interpreting and organizing sensory information to understand the environment. It allows us to take in sensory data through our sense organs and make sense of the world around us.

Perception presumes sensing. In humans, perception is aided by sensory organs. In the domain of AI, perception mechanism puts the data acquired by the sensors together in a meaningful manner. This includes:

- i. **Computer Vision:** The capacity of being able to take in and interpret images and consequently sift and classify different objects, facial details, and scenes.
- ii. **Speech Recognition:** The possibility that a machine can transcribe language into text and give a machine the capability to and how he or she should respond to human speech.
- iii. **Sensor Integration:** Employing a sensor with multiple point of view helps to integrate different outputs, thereby leading more in-depth data. Cameras, microphones, and touch sensors contribute to the creation of a method for determining the surrounding circumstances.

4. Linguistic Intelligence

Linguistic intelligence (AI) includes the capability to grasp, decipher and produce language that a human being can understand. It is one's ability to use, comprehend, speak, and write the verbal and written language. It is important in interpersonal communication. Linguistic intelligence refers to the capability to use language—both written and

spoken—effectively. People with high linguistic intelligence are skilled at reading, writing, telling stories, and memorizing words. This is primarily achieved through Natural Language Processing (NLP), which encompasses:

- i. **Text Analysis:** Use of NLP in doing main text analysis such as sentiment analysis and topic modeling.
- ii. **Machine Translation:** Tend to propose a solution that is done by a machine system when it tries to translate text from one language to another like Google Translate.
- iii. **Dialogue Systems:** Bringing up language conversational agents or chatbots that can have interaction with humans using natural language, such as virtual assistants like Siri and Alexa.

5. Problem Solving

Intelligent systems perform problem-solving for the sake of it. The first cognitive function is the recognition of a problem, devising a plan, and executing it in order to find a solution. Problem solving is the ability to process information and find solutions to complex or challenging situations. It involves identifying the problem, generating potential solutions, and implementing the best solution effectively. Problem Solving – It is the process in which one perceives and tries to arrive at a desired solution from a present situation by taking some path, which is blocked by known or unknown hurdles.

Problem solving also includes decision making, which is the process of selecting the best suitable alternative out of multiple alternatives to reach the desired goal are available.

Techniques used include:

- i. **Search Algorithms:** Explore techniques for example, depth-first search, breadth-first search, and A* algorithms, which are used to criss-cross the possible solutions, in order to find the optimal solution.
- ii. **Heuristics:** Generally, these heuristics give methodology of the process some boundary which is key to such problems.
- iii. **Optimization Techniques:** Methods functioning as genetic algorithms and simulated annealing to optimize the search through most of the available possibilities.

Core Components of Intelligent Systems

Intelligent systems integrate various components of AI technology, including machine learning algorithms, natural language processing, robotics, and expert systems, among others:

i. Machine Learning (ML)

Machine learning algorithms allow systems to learn from data patterns and improve their decision-making over time without being explicitly programmed. ML models are used in applications ranging from predictive analytics in business to diagnostic systems in healthcare.

ii. Natural Language Processing (NLP)

NLP enables a system to understand, interpret, and generate human language in a way that is both meaningful and useful. This component is crucial in applications like chatbots, voice-operated GPS systems, and customer service AI agents.

iii. Robotics

Robotics combines AI techniques with mechanical engineering to create robots capable of performing tasks that require human-like dexterity and decision-making. These are often employed in manufacturing, where they work alongside human operators to enhance productivity and safety.

iv. Expert Systems

Expert systems use rule-based algorithms to emulate the decision-making ability of a human expert. In fields like law and medicine, they provide critical insights based on a set of rules derived from the knowledge of experts.

Applications of Intelligent Systems

Intelligent systems are transforming numerous industries by enhancing efficiency, accuracy, and productivity:

- **Healthcare:** From diagnostics to robotic surgeries, AI systems are providing solutions that help in early diagnosis and personalized treatment plans.
- **Automotive:** Intelligent systems power autonomous vehicles that can perceive their environment and make driving decisions, significantly enhancing road safety.
- **Finance:** In finance, these systems are used for algorithmic trading, risk assessment, and customer service automation.
- **Retail:** AI enhances customer experience through personalized shopping recommendations and inventory management using predictive analytics.

SPEECH AND VOICE RECOGNITION

1. Speech Recognition: - Speech recognition involves converting spoken language into written text. It focuses on understanding *what is being said*.

How It Works:

- **Acoustic Model:** Analyzes the audio signal and identifies phonemes (smallest units of sound).
- **Language Model:** Determines the most probable word combinations based on grammar and context.
- **Speech-to-Text Conversion:** Combines the above models to transcribe spoken words into text.

Applications:

- **Voice Assistants:** Alexa, Siri, Google Assistant use speech recognition to understand user commands.
- **Transcription Services:** Tools like Otter.ai and Rev convert spoken content into text for note-taking or documentation.
- **Customer Service:** Automated IVR (Interactive Voice Response) systems understand user queries in call centers.
- **Accessibility:** Assists individuals with physical disabilities to control devices or compose text using their voice.

Challenges:

- **Accents and Dialects:** Different pronunciations can reduce accuracy.
- **Background Noise:** Interference can make speech harder to interpret.
- **Real-Time Processing:** Fast and accurate recognition in live scenarios is resource-intensive.

2. **Voice Recognition:** - Voice recognition, also called speaker recognition, focuses on identifying *who is speaking*. It analyzes voice patterns unique to an individual to authenticate or differentiate between speakers.

How It Works:

- **Feature Extraction:** Captures characteristics of a person's voice, such as pitch, tone, and cadence.
- **Pattern Matching:** Compares these features against a database of known voice profiles.
- **Authentication or Identification:** Determines if the speaker matches a known voice or identifies a new one.

Applications:

- **Security:** Voice biometrics for secure authentication in banking or sensitive systems.
- **Multi-User Systems:** Devices like smart home assistants identify different users to provide personalized responses.
- **Forensic Analysis:** Used in law enforcement to identify individuals from audio recordings.

Challenges:

- **Voice Imitation:** Advanced mimicking techniques can pose security risks.
- **Health Conditions:** Illness or aging may alter voice characteristics, affecting accuracy.
- **Environmental Factors:** Noise and poor-quality microphones can impact reliability.

Key Differences between Speech and Voice Recognition:

Feature	Speech Recognition	Voice Recognition
Purpose	Converts spoken words to text	Identifies the speaker
Focus	What is being said	Who is speaking
Output	Text transcription	Speaker identity

These both terms are common in robotics, expert systems and natural language processing. Though these terms are used interchangeably, their objectives are different.

Speech Recognition	Voice Recognition
The speech recognition aims at understanding and comprehending WHAT was spoken.	The objective of voice recognition is to recognize WHO is speaking.
It is used in hand-free computing, map, or menu navigation.	It is used to identify a person by analyzing its tone, voice pitch, and accent, etc.
Machine does not need training for Speech Recognition as it is not speaker dependent.	This recognition system needs training as it is person oriented.
Speaker independent Speech Recognition systems are difficult to develop.	Speaker dependent Speech Recognition systems are comparatively easy to develop.

Working of Speech and Voice Recognition Systems

The user input spoken at a microphone goes to sound card of the system. The converter turns the analog signal into equivalent digital signal for the speech processing. The database is used to compare the sound patterns to recognize the words. Finally, reverse feedback is given to the database.

This source-language text becomes input to the Translation Engine, which converts it to the target language text. They are supported with interactive GUI, large database of vocabulary, etc.

Task Classification of AI

AI task classification involves grouping the types of tasks Artificial Intelligence can perform based on functionality, application, or technical approach. Below are some of AI task classification:

1. Based on Functionality

AI tasks can be classified into these broad categories:

1.1 Perception Tasks

AI systems sense and interpret the environment using input data from various sources.

- **Examples:**
 - Image recognition
 - Speech and voice recognition
 - Natural language processing (NLP)
 - Object detection
- **Applications:**
 - Facial recognition in security systems
 - Self-driving cars interpreting road signs

1.2 Reasoning Tasks

AI systems use logic and rules to draw conclusions or make decisions.

- **Examples:**
 - Problem-solving

- Decision-making
- Predictive analytics
- **Applications:**
 - Chess-playing AI like Deep Blue
 - Fraud detection in banking

1.3 Learning Tasks

AI learns from data to improve its performance without being explicitly programmed.

- **Examples:**
 - Supervised learning (classification, regression)
 - Unsupervised learning (clustering, dimensionality reduction)
 - Reinforcement learning (decision-making through rewards)
- **Applications:**
 - Personalized recommendations on platforms like Netflix or Amazon
 - Autonomous robots learning new tasks

1.4 Natural Interaction Tasks

AI systems interact with humans or the environment in natural ways.

- **Examples:**
 - Chatbots
 - Virtual assistants
 - Gesture recognition
- **Applications:**
 - Customer support automation
 - Healthcare assistants

1.5 Generation Tasks

AI creates content such as text, images, audio, or designs.

- **Examples:**
 - Text generation (e.g., GPT models)
 - Image synthesis
 - Music composition
- **Applications:**
 - Content creation for marketing
 - Game design

2. Based on Application Domains

AI tasks can also be classified by their domain-specific use cases:

2.1 Healthcare

- Medical diagnosis (e.g., detecting diseases from X-rays)
- Drug discovery
- Patient monitoring (e.g., wearable devices)

2.2 Finance

- Algorithmic trading

- Risk assessment
- Fraud detection

2.3 Retail and E-Commerce

- Product recommendations
- Inventory management
- Customer behavior analysis

2.4 Transportation

- Route optimization
- Autonomous vehicles
- Traffic prediction

2.5 Manufacturing

- Predictive maintenance
- Robotics in assembly lines
- Quality control

3. Based on Technical Approach

AI tasks can be divided into types based on the methodologies used:

3.1 Rule-Based Systems

- Predefined rules dictate the system's behavior.
- **Examples:** Expert systems for medical diagnosis.

3.2 Machine Learning

- Algorithms that learn patterns from data.
- **Examples:** Image classification, fraud detection.

3.3 Deep Learning

- Neural networks for complex data like images and audio.
- **Examples:** Face recognition, autonomous driving.

3.4 Reinforcement Learning

- Systems learn by trial and error to maximize rewards.
- **Examples:** Robotics, game-playing AI.

3.5 Evolutionary Algorithms

- Inspired by natural selection, AI evolves optimal solutions.
- **Examples:** Genetic algorithms for optimization problems.

4. Based on the Level of AI

AI tasks can be categorized based on the level of sophistication:

4.1 Narrow AI (Weak AI)

- Designed for specific tasks.
- **Examples:** Chatbots, recommendation systems.

4.2 General AI (Strong AI)

- Hypothetical systems capable of understanding and performing any intellectual task.

- **Examples:** Still in research (e.g., AGI projects).

4.3 Super intelligent AI

- AI surpassing human intelligence in all fields.
- **Examples:** A theoretical future state.

5. Based on Real-World Integration

AI tasks are classified into categories reflecting their impact in practice:

5.1 Automation

- Automating repetitive tasks (e.g., data entry, scheduling).

5.2 Optimization

- Improving efficiency and outcomes (e.g., supply chain management).

5.3 Prediction

- Forecasting trends and outcomes (e.g., weather, stock prices).

5.4 Creativity

Generating new ideas, designs, or solutions (e.g., AI art, product design).

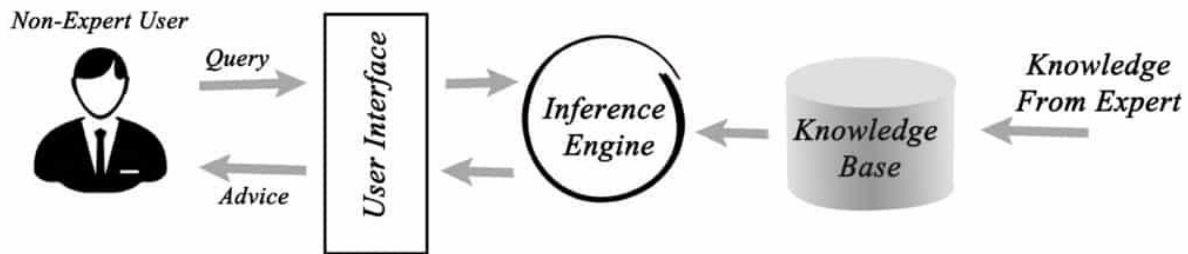
AI task classification helps in understanding the diverse applications and capabilities of artificial intelligence across various domains. This classification also provides insights into the development and deployment of AI technologies for solving real-world problems.

INTRODUCTION TO EXPERT SYSTEMS IN ARTIFICIAL INTELLIGENCE

Expert systems in *Artificial Intelligence* are a prominent domain for research in AI. It was initially introduced by researchers at Stanford University and was developed to solve complex problems in a particular domain. This blog on Expert Systems in Artificial Intelligence will cover the following topics.

An Expert system is a domain in which *Artificial Intelligence* stimulates the behavior and judgment of a human or an organization containing experts. It acquires relevant knowledge from its knowledge base and interprets it as per the user's problem. The data in the knowledge base is essentially added by humans who are experts in a particular domain. However, the software is used by non-experts to gain information. It is used in various medical diagnoses, accounting, coding, gaming, and more areas.

Expert System



Breaking down an expert system essentially is AI software that uses knowledge stored in a knowledge base to solve problems. This usually requires a human expert; thus, it aims at preserving human expert knowledge in its knowledge base. Hence, expert systems are computer applications developed to solve complex problems in a particular domain at an extraordinary level of human intelligence and expertise.

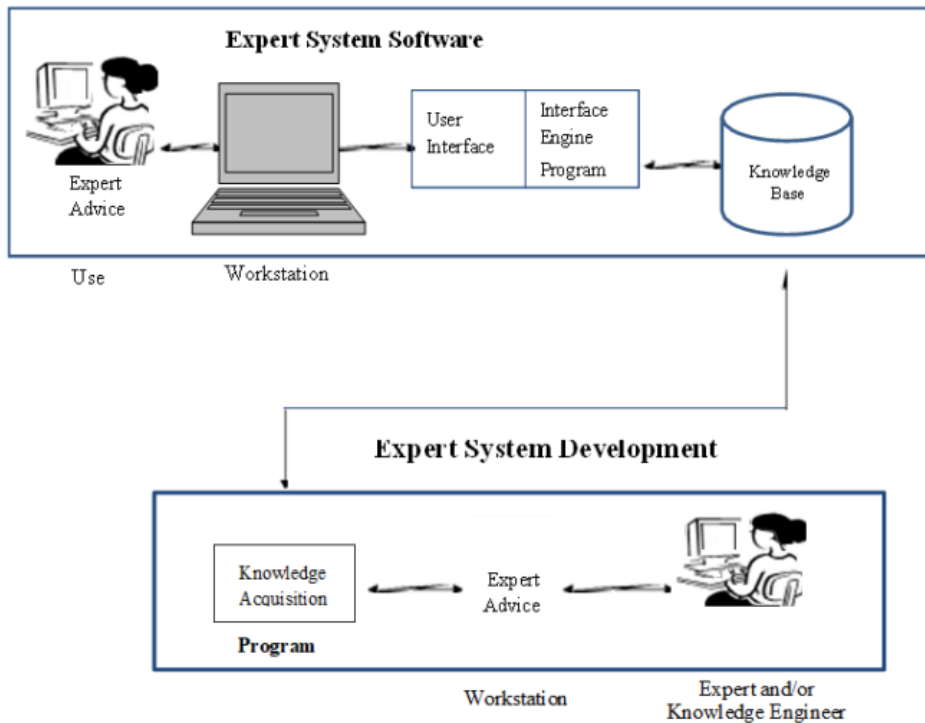
Formally, an expert system is defined as a knowledge-based computer program that exhibits, within a specific domain, a degree of expertise in problem solving that is comparable to that of a human expert. This problem method uses a knowledge base, which is carefully formulated on the basis of expert judgment, intuition, and experience.

Expert systems are computer programs or AI systems designed to emulate the decision-making capabilities of a human expert in a specific domain or field. These systems are built on a foundation of knowledge and rules and are used to provide advice, solve problems, make decisions, or assist with tasks in a particular area of expertise. Here are some basic concepts associated with expert systems:

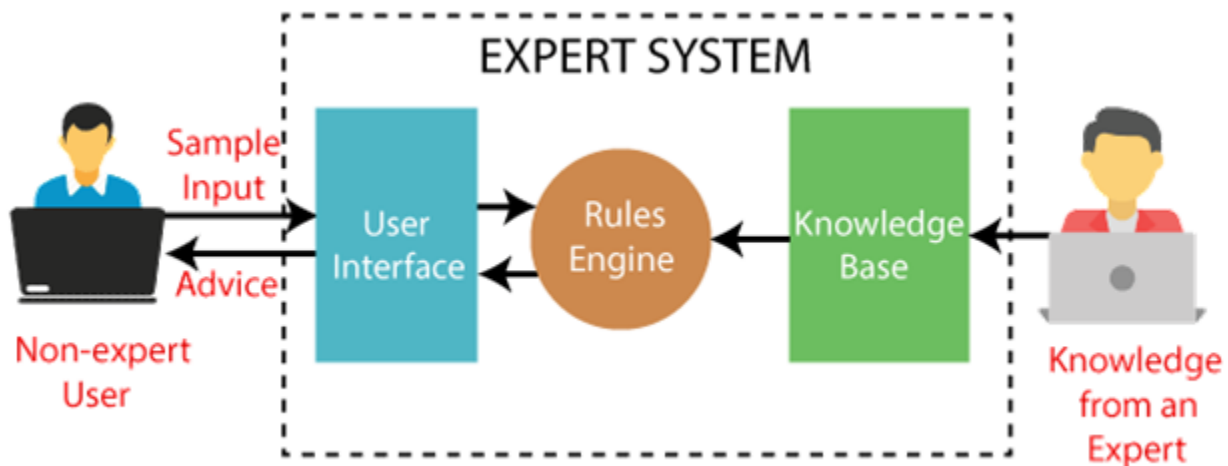
Components of an Expert System

The components of an expert system include a knowledge base and software modules that perform knowledge inference in the knowledge base and communicate answers to the user's questions. Below are the following components:

EXPERT SYSTEM



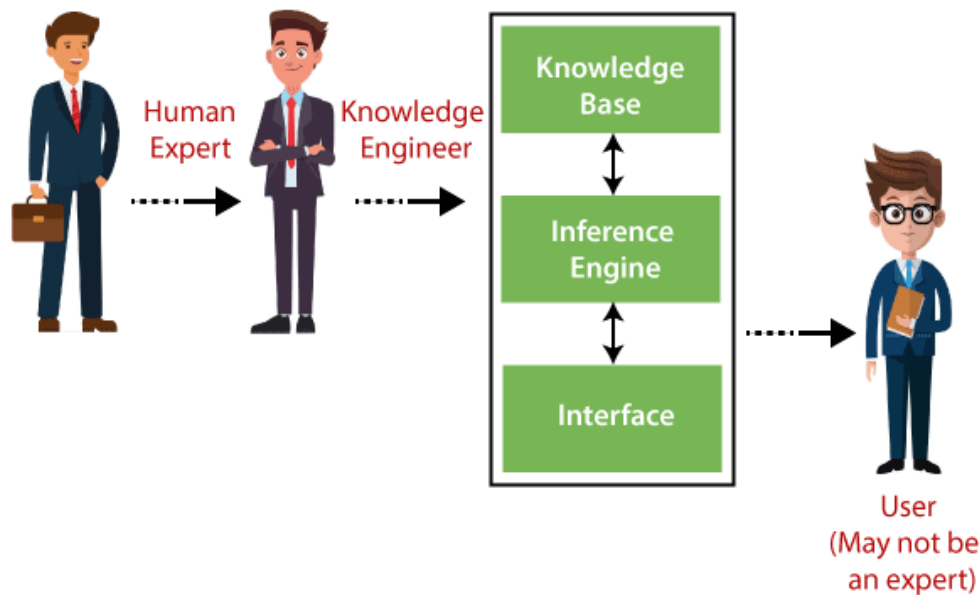
Below is the block diagram that represents the working of an expert system:



Note: It is important to remember that an expert system is not used to replace the human experts; instead, it is used to assist the human in making a complex decision. These systems do not have human capabilities of thinking and work on the basis of the knowledge base of the particular domain.

An expert system mainly consists of three components:

- **User Interface**
- **Inference Engine**
- **Knowledge Base**



1. User Interface

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, ***it is an interface that helps a non-expert user to communicate with the expert system to find a solution.***

User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence.

It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms –

- Natural language displayed on screen.
- Verbal narrations in natural language.
- Listing of rule numbers displayed on the screen.

The user interface makes it easy to trace the credibility of the deductions.

Requirements of Efficient ES User Interface

- It should help users to accomplish their goals in shortest possible way.
- It should be designed to work for user's existing or desired work practices.
- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

2. Inference Engine (Rules of Engine)

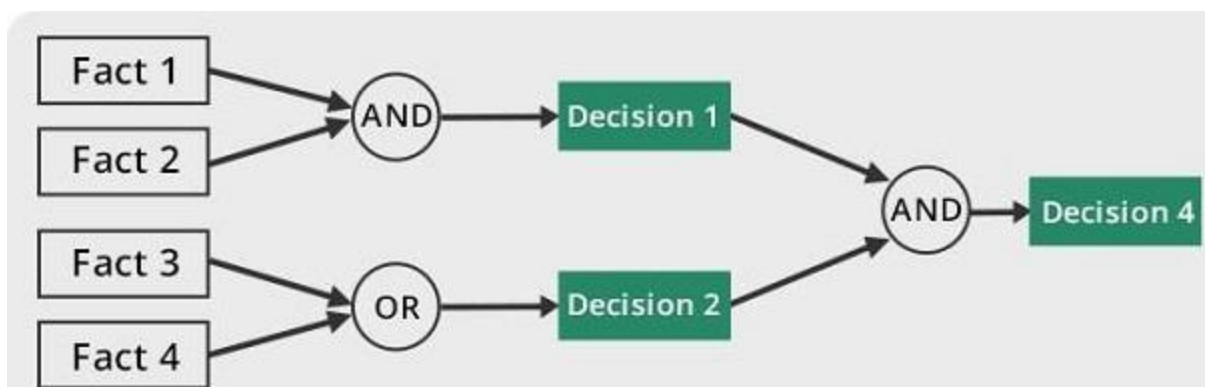
- The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion

or deduce new information. It helps in deriving an error-free solution of queries asked by the user.

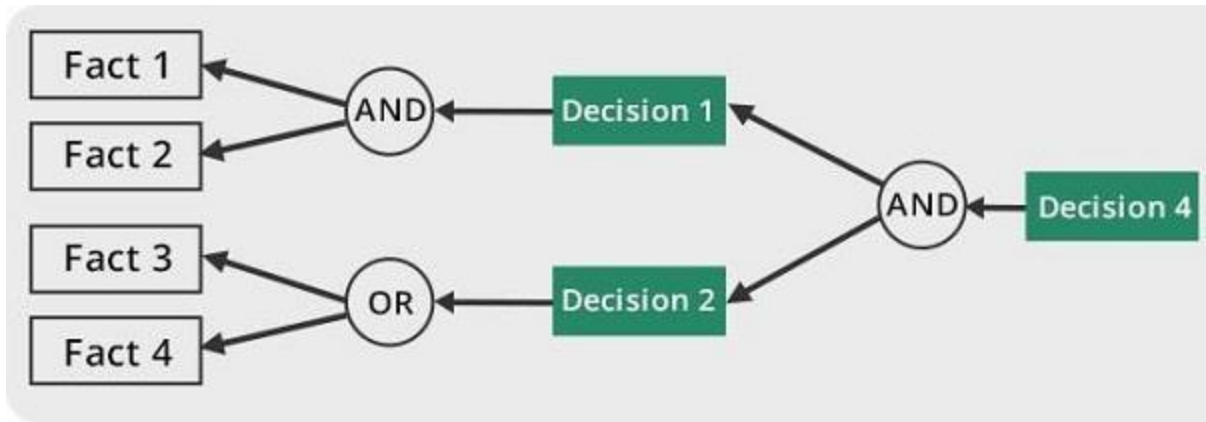
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.
- There are two types of inference engine:
- **Deterministic Inference engine:** The conclusions drawn from this type of inference engine are assumed to be true. It is based on *facts* and *rules*.
- **Probabilistic Inference engine:** This type of inference engine contains uncertainty in conclusions, and based on the probability.

Inference engine uses the below modes to derive the solutions:

- **Forward Chaining:** - It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.
- It is a strategy of an expert system to answer the question, “What can happen next?”
- Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution.
- This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.



- **Backward Chaining:** - It is a backward reasoning method that starts from the goal and works backward to prove the known facts. With this strategy, an expert system finds out the answer to the question, “Why this happened?” On the basis of what has already happened, the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.



3. Knowledge Base

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.
- It is similar to a database that contains information and rules of a particular domain or subject.
- One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

It contains domain-specific and high-quality knowledge. Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.

The data is collection of facts. The information is organized as data and facts about the task domain. Data, information, and past experience combined together are termed as knowledge.

Components of Knowledge Base

- **Factual Knowledge:** The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.
- **Heuristic Knowledge:** This knowledge is based on practice, the ability to guess, evaluation, and experiences.

Knowledge Representation: It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

Knowledge Acquisitions: It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base.

Development of Expert System

Here, we will explain the working of an expert system by taking an example of MYCIN ES (*MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections*).

Below are some steps to build an MYCIN:

- Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection; provide information about the causes, symptoms, and other knowledge in that domain.
- The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.
- The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.
- Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.
- In the end, it will provide a response to the patient by using the user interface.

Participants in the development of Expert System

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Below are some popular examples of the Expert System:

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.
- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.
- **PXDES:** It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.
- **CaDeT:** The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

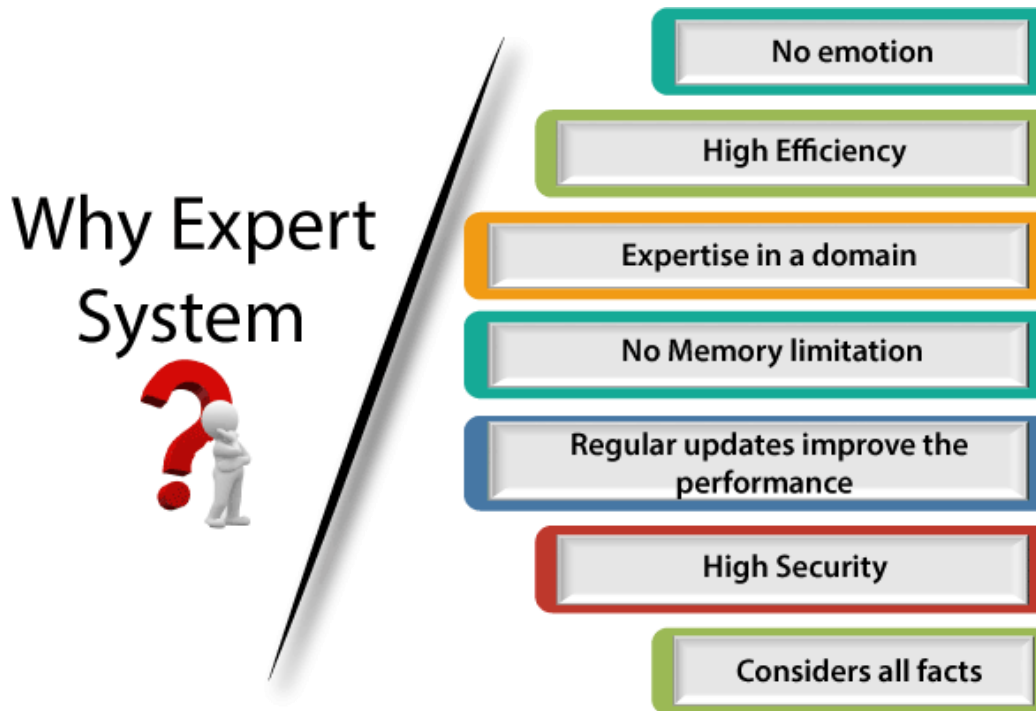
Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.

- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Why Expert System?

Before using any technology, we must have an idea about why to use that technology and hence the same for the ES. Although we have human experts in every field, then what is the need to develop a computer-based system. So below are the points that are describing the need of the ES:



1. **No memory Limitations:** It can store as much data as required and can memorize it at the time of its application. But for human experts, there are some limitations to memorize all things at every time.
2. **High Efficiency:** If the knowledge base is updated with the correct knowledge, then it provides a highly efficient output, which may not be possible for a human.
3. **Expertise in a domain:** There are lots of human experts in each domain, and they all have different skills, different experiences, and different skills, so it is not easy to get a final output for the query. But if we put the knowledge gained from human experts into the expert system, then it provides an efficient output by mixing all the facts and knowledge
4. **Not affected by emotions:** These systems are not affected by human emotions such as fatigue, anger, depression, anxiety, etc.. Hence the performance remains constant.
5. **High security:** These systems provide high security to resolve any query.
6. **Considers all the facts:** To respond to any query, it checks and considers all the available facts and provides the result accordingly. But it is possible that a human expert may not consider some facts due to any reason.

7. **Regular updates improve the performance:** If there is an issue in the result provided by the expert systems, we can improve the performance of the system by updating the knowledge base.

Capabilities of the Expert System

Below are some capabilities of an Expert System:

- **Advising:** It is capable of advising the human being for the query of any domain from the particular ES.
- **Provide decision-making capabilities:** It provides the capability of decision making in any domain, such as for making any financial decision, decisions in medical science, etc.
- **Demonstrate a device:** It is capable of demonstrating any new products such as its features, specifications, how to use that product, etc.
- **Problem-solving:** It has problem-solving capabilities.
- **Explaining a problem:** It is also capable of providing a detailed description of an input problem.
- **Interpreting the input:** It is capable of interpreting the input given by the user.
- **Predicting results:** It can be used for the prediction of a result.
- **Diagnosis:** An ES designed for the medical field is capable of diagnosing a disease without using multiple components as it already contains various inbuilt medical tools.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain**
It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.

- ***In the knowledge domain***
These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- ***In the finance domain***
In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- ***In the diagnosis and troubleshooting of devices***
In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- ***Planning and scheduling***
The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

Roles of Individuals who interact with the Expert system

- ***Domain Expert*** : The individuals who currently are experts in solving the problems; the system is intended to solve
- ***Knowledge engineer*** : The individual who encodes the expert's knowledge in a declarative form that can be used by the expert system;
- ***User***: The individual who will be consulting with the system to get advice which would have been provided by the expert.
- ***System engineer*** - the individual who builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine.

Types of Expert System Technology

ES technologies come in various levels, they are:

- ***Expert System Development Environment***: The ES development environment contains a set of hardware tools (Workstations, minicomputers, mainframes), High-level symbolic programming languages [LISt Programming (LISP) and PROgrammation en LOGique (PROLOG)], as well as large databases.
- ***Tools***: Tools, as an ES technology, assists in reducing the effort and cost involved in developing an expert system to a large extent.
- ***Shells***: A Shell is an expert system that functions without a knowledge base. It provides developers with knowledge acquisition, inference engine, user interface, and explanation facility. For example – Java Expert System Shell (JESS), Vidwan, etc.

Steps to Develop an Expert System

There are 6 steps involved in the development of an expert



Traditional Systems versus Expert Systems

A key distinction between the traditional system as opposed to the expert system is the way in which the problem-related expertise is coded. Essentially, in conventional applications, the problem expertise is encoded in both programs as well as data structures. On the other hand, in expert systems, the approach of problem-related expertise is encoded in data structures only. Moreover, the use of knowledge in expert systems is vital. However, traditional systems use data more efficiently than expert systems.

One of the biggest limitations of conventional systems is that they cannot explain a problem's conclusion. That is because these systems try to solve problems in a straightforward manner. However, expert systems can provide explanations and simplify the understanding of a particular conclusion.

Generally, an expert system uses symbolic representations to perform computations. On the contrary, conventional systems are incapable of expressing these terms. They only simplify the problems without being able to answer the "how" and "why" questions. Moreover, problem-solving tools are present in expert systems as opposed to traditional ones; hence, various problems are often entirely solved by the system's experts.

Human System Vs. Expert System

Human Experts	Expert Systems
Perishable and unpredictable in nature	Permanent and consistent in nature
Difficult to transfer and document data	Easy to transfer and document data
Human expert resources are expensive	Expert Systems are cost-effective Systems

POPULAR AI ALGORITHMS AND THEIR USES

From self-driving cars to multimodal chatbots, there's no question that artificial intelligence (AI) is making rapid progress. But behind these mystifying innovations are a set of fairly standard (and quite old) algorithms that have seen refinement and optimization over many years. If you want to better understand AI, then you'll definitely want to know about the algorithms in this article.

First, what are AI algorithms? Simply put, AI algorithms are mathematical models that enable machines to learn from data. They come in different forms, including supervised learning, unsupervised learning, and reinforcement learning (RL).

Supervised learning algorithms learn from labeled examples whereas unsupervised learning algorithms learn from unlabeled data. Labeled data is data that has been annotated with predefined target values, while unlabeled data is data that is not assigned any such values. Reinforcement learning algorithms learn by trial and error and so have become very popular in game playing (like chess and Go) and robotics.

Popular machine learning algorithms

Linear Regression:- Linear regression, also called least squares regression, is the simplest supervised machine learning algorithm for predicting numeric values. In some cases, linear regression doesn't even require an optimizer, since it is solvable in closed form. Otherwise, it is easily optimized using gradient descent (see below). The assumption of linear regression is that the objective function is linearly correlated with the independent variables. That may or may not be true for your data.

To the despair of data scientists, business analysts often blithely apply linear regression to prediction problems and then stop, without even producing scatter plots or calculating correlations to see if the underlying assumption is reasonable. Don't fall into that trap. It's not that hard to do your exploratory data analysis and then have the computer try all the reasonable machine learning algorithms to see which ones work the best. By all means, try linear regression, but treat the result as a baseline, not a final answer.

Support Vector Machines: - Support vector machines (SVMs) are a kind of parametric classification model, a geometric way of separating and classifying two label classes. In the simplest case of well-separated classes with two variables, an SVM finds the straight line that best separates the two groups of points on a plane.

In more complicated cases, the points can be projected into a higher-dimensional space and the SVM finds the plane or hyperplane that best separates the classes. The projection is called a *kernel*, and the process is called the *kernel trick*. After you reverse the projection, the resulting boundary is often nonlinear.

When there are more than two classes, SVMs are used on the classes pairwise. When classes overlap, you can add a penalty factor for points that are misclassified; this is called a soft margin.

Decision Tree: - are a non-parametric supervised learning method used for both classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Decision trees are easy to interpret and cheap to deploy, but computationally expensive to train and prone to overfitting.

Random Forest:- The random forest model produces an *ensemble* of randomized decision trees, and is used for both classification and regression. The aggregated ensemble either combines the votes modally or averages the probabilities from the decision trees. Random forest is a kind of *bagging* ensemble.

K-Means Clustering: - The k-means clustering problem attempts to divide n observations into k clusters using the Euclidean distance metric, with the objective of minimizing the variance

(sum of squares) within each cluster. It is an unsupervised method of vector quantization, and is useful for feature learning, and for providing a starting point for other algorithms.

Lloyd's algorithm (iterative cluster agglomeration with centroid updates) is the most common heuristic used to solve the problem. It is relatively efficient, but doesn't guarantee global convergence. To improve that, people often run the algorithm multiple times using random initial cluster centroids generated by the Forgy or random partition methods.

K-means assumes spherical clusters that are separable so that the mean converges towards the cluster center, and also assumes that the ordering of the data points does not matter. The clusters are expected to be of similar size, so that the assignment to the nearest cluster center is the correct assignment.

Popular Deep Learning Algorithms

There are a number of very successful and widely adopted deep learning paradigms, the most recent being the transformer architecture behind today's generative AI models.

Convolutional Neural Networks: - Convolutional Neural Networks (CNNs) are a type of deep neural network often used for machine vision. They have the desirable property of being position-independent.

The understandable summary of a convolution layer when applied to images is that it slides over the image spatially, computing dot products; each unit in the layer shares one set of weights. A *convnet* typically uses multiple convolution layers, interspersed with activation functions. CNNs can also have pooling and fully connected layers, although there is a trend toward getting rid of these types of layers.

Recurrent Neural Networks: - While convolutional neural networks do a good job of analyzing images, they don't really have a mechanism that accounts for time series and sequences, as they are strictly feed-forward networks. Recurrent neural networks (RNNs), another kind of deep neural network, explicitly include feedback loops, which effectively gives them some memory and dynamic temporal behavior and allows them to handle sequences, such as speech.

That doesn't mean that CNNs are useless for natural language processing; it does mean that RNNs can model time-based information that escapes CNNs. And it doesn't mean that RNNs can *only* process sequences. RNNs and their derivatives have a variety of application areas, including language translation, speech recognition and synthesis, robot control, time series prediction and anomaly detection, and handwriting recognition.

While in theory an ordinary RNN can carry information over an indefinite number of steps, in practice it generally can't go many steps without losing the context. One of the causes of the problem is that the gradient of the network tends to vanish over many steps, which interferes with the ability of a gradient-based optimizer such as stochastic gradient descent (SGD) to converge.

Artificial Neural Networks (ANNs): This one you've probably heard of. ANNs are inspired by the brain and are used for image and speech recognition and natural language processing. The basic idea behind ANNs is that you input data, and the network sends the data through layers of artificial neurons. Each neuron takes in information from the previous layer and calculates an output, which then gets passed on to the next layer. Deep learning uses ANNs with multiple layers and is the architecture of choice for almost every AI application today. ANNs themselves were first implemented in the 1950s.

Other types of Algorithms

Simulated Annealing Algorithm

Annealing is the process of heating and cooling a metal to change its internal structure for modifying its physical properties. When the metal cools, its new structure is seized, and the metal retains its newly obtained properties. In simulated annealing process, the temperature is kept variable.

We initially set the temperature high and then allow it to 'cool' slowly as the algorithm proceeds. When the temperature is high, the algorithm is allowed to accept worse solutions with high frequency.

Start

- Initialize $k = 0$; $L =$ integer number of variables;
- From $i \rightarrow j$, search the performance difference Δ .
- If $\Delta \leq 0$ then accept else if $\exp(-\Delta/T(k)) > \text{random}(0,1)$ then accept;
- Repeat steps 1 and 2 for $L(k)$ steps.
- $k = k + 1$;

Repeat steps 1 through 4 till the criteria are met.

End

Fuzzy Logic Systems

Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO. Fuzzy Logic Systems (FLS) produce acceptable but definite output in response to incomplete, ambiguous, distorted, or inaccurate (fuzzy) input.

The conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human's YES or NO.

The key application areas of fuzzy logic are as given –

Automotive Systems

- Automatic Gearboxes
- Four-Wheel Steering
- Vehicle environment control

Consumer Electronic Goods

- Hi-Fi Systems
- Photocopiers
- Still and Video Cameras

- Television

Domestic Goods

- Microwave Ovens
- Refrigerators
- Toasters
- Vacuum Cleaners
- Washing Machines

Environment Control

- Air Conditioners/Dryers/Heaters
- Humidifiers

Back Propagation Algorithm

It is the training or learning algorithm. It learns by example. If you submit to the algorithm the example of what you want the network to do, it changes the network's weights so that it can produce desired output for a particular input on finishing the training.

Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks.

Bayesian Networks (BN)

These are the graphical structures used to represent the probabilistic relationship among a set of random variables. Bayesian networks are also called Belief Networks or Bayes Nets. BNs reason about uncertain domain.

In these networks, each node represents a random variable with specific propositions. For example, in a medical diagnosis domain, the node Cancer represents the proposition that a patient has cancer.

ROBOTICS

Robotics is a branch of AI, which is composed of Electrical Engineering, Mechanical Engineering, and Computer Science for designing, construction, and application of robots.

Robotics is a domain in artificial intelligence that deals with the study of creating intelligent and efficient robots. Robots are the artificial agents acting in real world environment.

Robots are aimed at manipulating the objects by perceiving, picking, moving, modifying the physical properties of object, destroying it, or to have an effect thereby freeing manpower from doing repetitive functions without getting bored, distracted, or exhausted.

Aspects of Robotics

- The robots have **mechanical construction**, form, or shape designed to accomplish a particular task.
- They have **electrical components** which power and control the machinery.
- They contain some level of **computer program** that determines what, when and how a robot does something.

Difference in Robot System and Other AI Program

Here is the difference between the two –

AI Programs	Robots
They usually operate in computer-stimulated worlds.	They operate in real physical world
The input to an AI program is in symbols and rules.	Inputs to robots is analog signal in the form of speech waveform or images
They need general purpose computers to operate on.	They need special hardware with sensors and effectors.

Robot Locomotion

Locomotion is the mechanism that makes a robot capable of moving in its environment. There are various types of locomotions –

- Legged
- Wheeled
- Combination of Legged and Wheeled Locomotion
- Tracked slip/skid

Legged Locomotion

- This type of locomotion consumes more power while demonstrating walk, jump, trot, hop, climb up or down, etc.
- It requires more number of motors to accomplish a movement. It is suited for rough as well as smooth terrain where irregular or too smooth surface makes it consume more power for a wheeled locomotion. It is little difficult to implement because of stability issues.
- It comes with the variety of one, two, four, and six legs. If a robot has multiple legs then leg coordination is necessary for locomotion.

Wheeled Locomotion

It requires fewer numbers of motors to accomplish a movement. It is little easy to implement as there are less stability issues in case of more number of wheels. It is power efficient as compared to legged locomotion.

- Standard wheel – Rotates around the wheel axle and around the contact
- Castor wheel – Rotates around the wheel axle and the offset steering joint.
- Swedish 45° and Swedish 90° wheels – Omni-wheel, rotates around the contact point, around the wheel axle, and around the rollers.
- Ball or spherical wheel – Omnidirectional wheel, technically difficult to implement.



Slip/Skid Locomotion

In this type, the vehicles use tracks as in a tank. The robot is steered by moving the tracks with different speeds in the same or opposite direction. It offers stability because of large contact area of track and ground.



Components of a Robot

Robots are constructed with the following –

- Power Supply – The robots are powered by batteries, solar power, hydraulic, or pneumatic power sources.
- Actuators – They convert energy into movement.
- Electric motors (AC/DC) – They are required for rotational movement.
- Pneumatic Air Muscles – They contract almost 40% when air is sucked in them.
- Muscle Wires – They contract by 5% when electric current is passed through them.
- Piezo Motors and Ultrasonic Motors – Best for industrial robots.
- Sensors – They provide knowledge of real time information on the task environment. Robots are equipped with vision sensors to be to compute the depth in the environment. A tactile sensor imitates the mechanical properties of touch receptors of human fingertips.

Applications of Robotics

The robotics has been instrumental in the various domains such as –

- Industries – Robots are used for handling material, cutting, welding, color coating, drilling, polishing, etc.

- Military – Autonomous robots can reach inaccessible and hazardous zones during war. A robot named *Daksh*, developed by Defense Research and Development Organization (DRDO), is in function to destroy life-threatening objects safely.
- Medicine – The robots are capable of carrying out hundreds of clinical tests simultaneously, rehabilitating permanently disabled people, and performing complex surgeries such as brain tumors.
- Exploration – The robot rock climbers used for space exploration, underwater drones used for ocean exploration are to name a few.
- Entertainment – Disney’s engineers have created hundreds of robots for movie making.

COMPUTER VISION

Computer vision is a field of study within artificial intelligence (AI) that focuses on enabling computers to Intercept and extract information from images and videos, in a manner similar to human vision. It involves developing algorithms and techniques to extract meaningful information from visual inputs and make sense of the visual world.

Computer vision enables a wide range of technological innovation. It allows self-driving cars to safely steer through streets and highways; it enables facial recognition tools to match images of people’s faces to their identities; and it enables augmented-reality applications to mix virtual objects with real-world images.

Computer vision applications are used across industries to improve the consumer experience, reduce costs, and tighten security. Manufacturers use it to spot defective products on the assembly line and prevent them from shipping to customers. Insurance adjusters use it to assess vehicle damage and reduce fraud in the claims process. Medical professionals use it to scan X-rays, MRIs, and ultrasounds to detect health problems. Banks use it to verify customers’ identities before conducting large transactions.

How computer vision works

Computer vision applications use input from sensing devices, artificial intelligence, machine learning, and deep learning to replicate the way the human vision system works. Computer vision applications run on algorithms that are trained on massive amounts of visual data or images in the cloud. They recognize patterns in this visual data and use those patterns to determine the content of other images.

Key Areas of Computer Vision

1. *Image Classification*

- Identifying what an image represents (e.g., recognizing cats, cars, etc.).
- Example: Using convolutional neural networks (CNNs) to classify objects in an image.

2. *Object Detection*

- Identifying and locating objects in an image or video.
- Example: YOLO (You Only Look Once), Faster R-CNN.

3. **Image Segmentation**

- Dividing an image into segments or regions for detailed analysis.
- Types: Semantic Segmentation (labeling pixels by object class) and Instance Segmentation (distinguishing individual objects).

4. **Face Recognition**

- Identifying or verifying individuals based on facial features.
- Example: Used in biometrics and security systems.

5. **Optical Character Recognition (OCR)**

- Converting images of text into machine-readable text.
- Example: Extracting text from scanned documents or photos of signs.

6. **Image and Video Generation**

- Generating new images or videos using techniques like Generative Adversarial Networks (GANs).
- Example: Deepfakes, artistic style transfer.

7. **Motion Analysis**

- Analyzing movement in a sequence of images or video.
- Example: Tracking the trajectory of an object in a video.

8. **3D Reconstruction**

- Creating 3D models from 2D images or videos.
- Example: Used in AR/VR and robotics.

9. **Scene Understanding**

- Interpreting the overall context or meaning of a scene.
- Example: Self-driving cars understanding road environments.

10. **Action Recognition**

- Detecting and analyzing actions in videos.
- Example: Recognizing human activities like walking, running, or jumping.

Computer Vision Examples

- **Facial recognition:** Identifying individuals through visual analysis.
- **Self-driving cars:** Using computer vision to navigate and avoid obstacles.
- **Robotic automation:** Enabling robots to perform tasks and make decisions based on visual input.
- **Medical anomaly detection:** Detecting abnormalities in medical images for improved diagnosis.
- **Sports performance analysis:** Tracking athlete movements to analyze and enhance performance.
- **Manufacturing fault detection:** Identifying defects in products during the manufacturing process.
- **Agricultural monitoring:** Monitoring crop growth, livestock health, and weather conditions through visual data.

Components of Computer Vision System

Machine vision comprises using computer vision for all industrial and non-industrial applications. While computer vision is primarily concerned with image processing on a hardware level, machine vision requires additional hardware I/O (input/output) and computer networks to transmit data generated by other process components, such as a robot arm.

Inspection of products such as microprocessors, cars, food, and pharmaceuticals is one of the most common uses of machine vision. Machine vision systems are increasingly used to solve industrial inspection issues, allowing for complete automation of the inspection process while improving accuracy and efficiency.

Devices are guided in executing their functions based on capturing and processing images by combining hardware and software. Industrial computer vision uses many similar algorithms and approaches as academic/educational and governmental/military computer vision applications, but there are some differences.

The major components of a machine vision system include the lighting, lens, image sensor, vision processing, and communications. Lighting illuminates the part to be inspected, allowing its features to stand out so the camera can see them. The lens captures the image and presents it to the sensor in the form of light. The sensor in a machine vision camera converts this light into a digital image sent to the processor for analysis.

Most machine vision hardware components, such as lighting modules, sensors, and processors, are commercial off-the-shelf (COTS). Machine vision systems can be assembled from COTS or purchased as an integrated system with all components in a single device. This post will explore various key components of a machine vision system, including lighting, lenses, vision sensor, image processing, and communications.

1. Lighting

The most important factor in achieving successful machine vision results is lighting. Machine vision systems generate images by analyzing reflected light from an object rather than itself. A lighting technique entails the placement of a light source concerning the part and the camera. An image can be enhanced by using a specific lighting technique. By silhouetting a part obscures surface details to allow measurement of its edges; for example, it eliminates some features while enhancing others.

2. Lenses

The image is captured by the lens and delivered to the camera's image sensor. The lenses' optical quality and price vary, and the captured image's quality and resolution are determined by the lens used. On most vision system cameras, there are two types of lenses: interchangeable and fixed lenses. The most common interchangeable lens mounts are C-mounts and CS-mounts. Using the right lens and extension combination will yield the best image. Autofocus, either a mechanically adjusted lens or a liquid lens that can focus on the part automatically, is typically used in a

standalone vision system with a fixed lens. Autofocus lenses usually have a fixed field of view at a given distance.

3. Image sensor

The ability of the camera to capture a properly illuminated image of the inspected object is dependent not only on the lens but also on the image sensor. To convert light (photons) to electrical signals, image sensors typically use charge-coupled devices (CCD) or complementary metal-oxide-semiconductor (CMOS) technology (electrons). The image sensor's primary function is to capture light and convert it to a digital image while maintaining a balance of noise, sensitivity, and dynamic range. The image is made up of pixels.

Low light creates dark pixels, while bright pixels are created by bright light. It's critical to ensure the camera has the correct sensor resolution for the job. The higher the resolution, the more detail and accurate measurements an image will have. Part size, inspection tolerances, and other parameters will dictate the required resolution.

4. Vision processing

Processing is the process of extracting data from a digital image, and it can happen either externally in a PC-based system or internally in a standalone vision system. The software is made up of several processing steps. The sensor is first used to obtain an image. Pre-processing may be required in some cases to optimize the image and ensure that all of the necessary features are visible. The software then locates the unique features, performs measurements, and compares them to the specification.

Finally, a decision is reached, and the outcomes are shared. While many physical components of a machine vision system (such as lighting) have similar specifications, the algorithms distinguish them. When comparing solutions, they should be at the top of the priority list. Vision software configures camera parameters, makes the pass-fail decision, communicates with the factory floor, and supports HMI development, depending on the system or application.

5. Communications

Because vision systems frequently use a variety of off-the-shelf components, these components must quickly and easily coordinate and connect to other machine elements. Typically, this is accomplished by sending discrete I/O signals or data over a serial connection to a device that logs or uses information. Discrete I/O points can be connected to a programmable logic controller (PLC), which can then use the data to control a work cell, an indicator such as a stack light, or directly to a solenoid that can trigger a reject mechanism.

A traditional RS232 serial output or Ethernet can be used to communicate data over a serial connection. Some systems use a higher-level industrial protocol, such as Ethernet/IP, which can be connected to a device such as a monitor or other operator interface to provide a process-specific operator interface for easy process monitoring and control.

Core Technologies in Computer Vision

Some of the core technologies in Computer Vision include:

1. **Machine Learning (ML)**

- Traditional ML algorithms like Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN).

2. **Deep Learning (DL)**

- CNNs for image processing, Recurrent Neural Networks (RNNs) for sequential data, and transformers for video and large-scale vision tasks.

3. **Pre-trained Models**

- Models like ResNet, VGG, MobileNet, and Vision Transformers (ViTs).

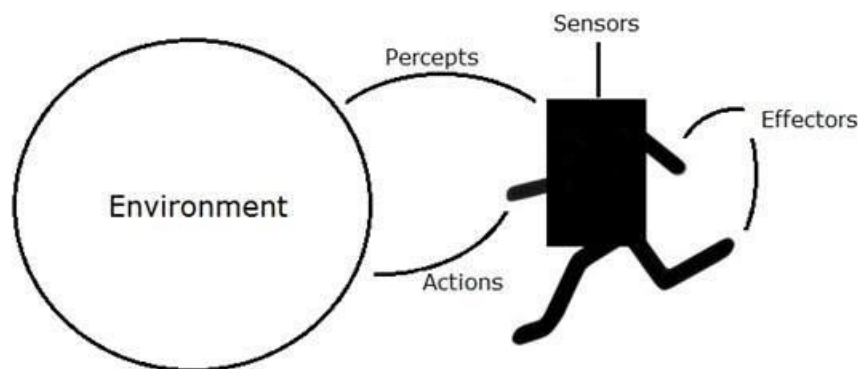
AGENTS & ENVIRONMENTS

In the field of Artificial Intelligence (AI), **agents** and **environments** are fundamental concepts that define how AI systems perceive, reason, and act. Understanding these concepts is key to designing and analyzing intelligent systems.

1. **Agent**

An **agent** is any entity that can perceive its environment through sensors and act upon it through actuators to achieve specific goals. Agents can range from simple programs to complex systems like robots or autonomous vehicles. An **agent** is anything that can perceive its environment through **sensors** and acts upon that environment through **effectors**.

- Organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effectors.
- A robotic agent replaces cameras and infrared range finders for the sensors, and various motors and actuators for effectors.
- A software agent has encoded bit strings as its programs and actions.



Characteristics of an Agent:

1. **Perception:** Gathers information from the environment via sensors (e.g., cameras, microphones, or software inputs).
2. **Decision-making:** Uses the perceived information to decide what actions to take.
3. **Action:** Executes decisions using actuators or outputs (e.g., motors, displays, or software commands).

4. **Autonomy**: Operates independently to some degree, without direct human intervention.
5. **Goal-oriented**: Designed to achieve specific objectives or optimize specific metrics.

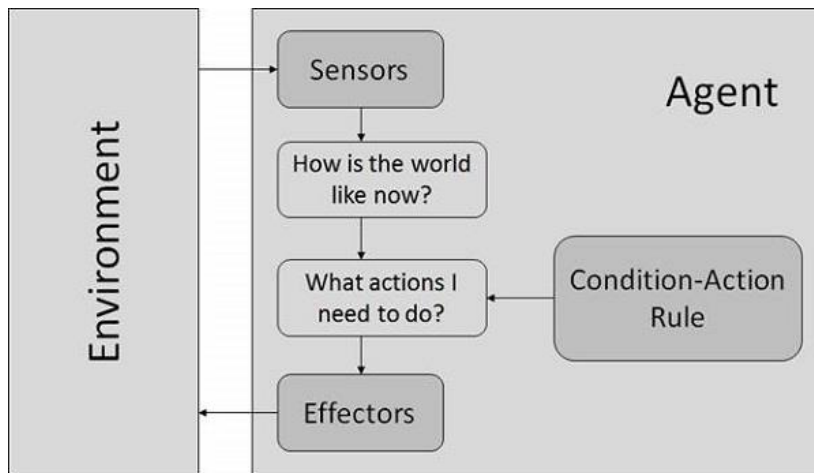
Types of Agents

1. Simple Reflex Agents

Simple Reflex Agents is a type of AI agent that selects actions based solely on the current state of the environment, ignoring its history or future consequences. These agents rely on a predefined set of condition-action rules, also called **if-then rules**, to respond to specific situations.

- Act based on current perceptions only, without considering history.
- Example: A thermostat that turns on/off based on room temperature.

Condition-Action Rule – It is a rule that maps a state (condition) to an action.



Features of Simple Reflex Agents

1. Stateless

- They do not maintain any memory of past states or actions.
- Decision-making is based entirely on the current percept (input from the environment).

2. Condition-Action Rules

- Use predefined rules to map percepts to actions.
- Example: "If the road is clear, move forward."

3. Reactive

- React immediately to percepts without any reasoning about long-term goals.

4. Efficient but Limited

- Simple reflex agents are efficient for predictable, well-defined environments.
- They struggle in complex or partially observable environments.

Architecture of a Simple Reflex Agent

The structure can be represented as:

1. **Input:** Sensors perceive the environment and generate percepts.
2. **Decision-making:** A rule-based system matches the current percept to an action.
3. **Output:** Actuators execute the selected action.

Example of a Simple Reflex Agent

1. Vacuum Cleaner Agent

A vacuum cleaner agent designed to clean two rooms (Room A and Room B).

- **Rules:**
 - If the current location is dirty, clean it.
 - If the current location is clean, move to the next room.

- **Algorithm:**

python

Copy code

```
def SimpleVacuumAgent(percept):
```

```
    location, status = percept
```

```
    if status == "Dirty":
```

```
        return "Clean"
```

```
    elif location == "A":
```

```
        return "Move to B"
```

```
    elif location == "B":
```

```
        return "Move to A"
```

- **Limitations:**
 - The agent does not remember which rooms were cleaned.
 - It may repeat actions unnecessarily.

2. Traffic Light Controller

A simple reflex agent that controls traffic lights.

- **Rules:**
 - If traffic on the main road exceeds a threshold, switch to green.
 - If no traffic is detected on the main road, switch to red.
- **Behavior:** The agent directly switches the lights based on real-time traffic sensor input.

Strengths of Simple Reflex Agents

1. **Simplicity**
 - Easy to design and implement.
 - No need for complex algorithms or memory management.
2. **Efficiency**
 - Fast decision-making because it only considers the current percept.
3. **Reliable in Predictable Environments**
 - Performs well in static, fully observable, and deterministic environments.

Limitations of Simple Reflex Agents

1. *Lack of Memory*

- Cannot handle environments where past states are important.
- Example: If a robot keeps visiting the same cleaned location, it wastes resources.

2. *Inability to Handle Partial Observability*

- Struggles when the agent cannot perceive the complete state of the environment.

3. *No Long-Term Planning*

- Cannot optimize or plan actions for future goals.
- Example: A maze-solving agent may get stuck in loops.

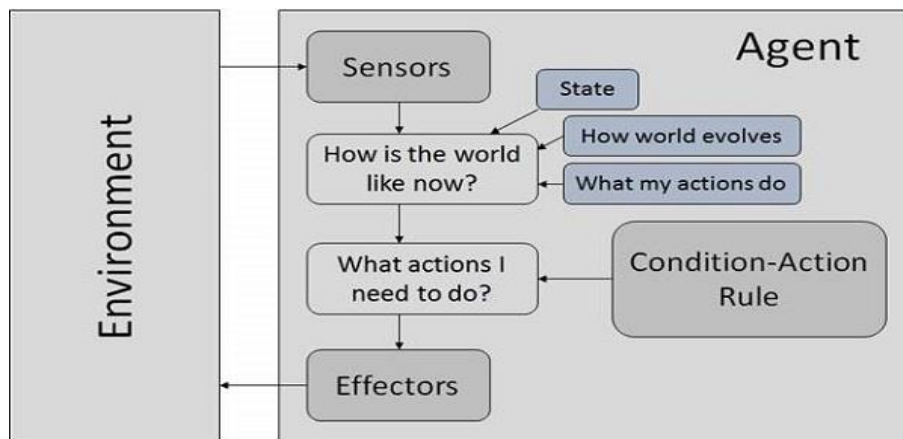
4. *Limited Adaptability*

- Can only handle scenarios for which explicit rules are predefined.
- Cannot learn new behaviors or adapt to dynamic environments.

2. Model Based Reflex Agents

A **Model-Based Reflex Agent** is an advanced type of AI agent that maintains an internal model of its environment. This model helps the agent handle situations where the environment is **partially observable**, meaning the agent cannot directly perceive all aspects of the environment at a given time.

- Maintain an internal model of the environment to handle partially observable situations.
- Example: A robot that navigates a map while avoiding obstacles.



Features of Model-Based Reflex Agents

1. *Internal State (Memory)*

- The agent uses an internal state to store information about the environment's history.
- This state is updated based on percepts and actions, allowing the agent to infer unobservable aspects.

2. *Environment Model*

- The agent has a model of how the environment works.

- The model includes rules for how actions affect the environment and how the environment evolves.
- 3. **Decision-Making**
 - Decisions are based on both the current percept and the internal state.
- 4. **Suitability**
 - Designed for *partially observable* environments.
 - Can perform well in dynamic and complex settings.

How Model-Based Reflex Agents Work

The agent uses the following components:

1. **Perception:** Receives current information about the environment through sensors.
2. **Update State:** Updates its internal state using the perception and the environment model.
3. **Decision:** Uses the updated state and condition-action rules to determine the next action.
4. **Action:** Executes the chosen action via actuators.

Architecture of a Model-Based Reflex Agent

General Structure:

1. **Input:** Percept from the environment.
2. **State Update Function:** Updates the agent's internal representation of the world.
3. **Decision Rules:** Determines action based on the internal state and percepts.
4. **Output:** Executes the action to affect the environment.

Example of a Model-Based Reflex Agent

Vacuum Cleaner Agent (Model-Based)

Enhancement over the simple reflex vacuum cleaner:

- **Environment:** Two rooms (A and B) that can be clean or dirty, and the agent cannot see the status of the other room.
- **Internal State:** Keeps track of which room was cleaned last.
- **Rules:**
 1. If the current location is dirty, clean it.
 2. If both rooms are clean, stop.
 3. If one room was just cleaned, move to the other room.

- **Algorithm:**

python

Copy code

```
def ModelBasedVacuumAgent(percept, state):
```

```
    location, status = percept
```

```
    if status == "Dirty":
```

```
        state[location] = "Clean"
```

```
        return "Clean"
```

```
    else:
```

```
        state[location] = "Clean"
```

```
if location == "A" and state["B"] == "Unknown":
    return "Move to B"
elif location == "B" and state["A"] == "Unknown":
    return "Move to A"
else:
    return "Stop"
```

Strengths of Model-Based Reflex Agents

1. **Handles Partial Observability**
 - Maintains an internal state to compensate for missing perceptual information.
2. **More Intelligent Behavior**
 - The agent can reason about unobservable parts of the environment and plan better responses.
3. **Scalable to Complex Environments**
 - Suitable for dynamic environments with evolving conditions.
4. **Adaptable**
 - By updating the internal state and using a flexible model, the agent can adapt to changes in the environment.

Limitations of Model-Based Reflex Agents

1. **Increased Complexity**
 - Requires designing and maintaining an accurate model of the environment.
2. **Computational Overhead**
 - Maintaining and updating the internal state can be computationally expensive, especially in large or dynamic environments.
3. **Dependence on the Model**
 - The agent's performance heavily depends on the accuracy of the environment model.
 - Errors in the model can lead to suboptimal or incorrect actions.

Applications of Model-Based Reflex Agents

1. **Robotics**
 - Autonomous robots that operate in partially observable environments, such as warehouses or homes.
2. **Self-Driving Cars**
 - Maintaining a model of the surrounding vehicles, pedestrians, and road conditions.
3. **Game AI**
 - Non-player characters (NPCs) maintaining a model of the game state for strategy planning.
4. **Healthcare**
 - Monitoring patient data and using models to predict health trends.
5. **Smart Assistants**

- Virtual assistants like Siri or Alexa using user history to refine responses and provide better suggestions.

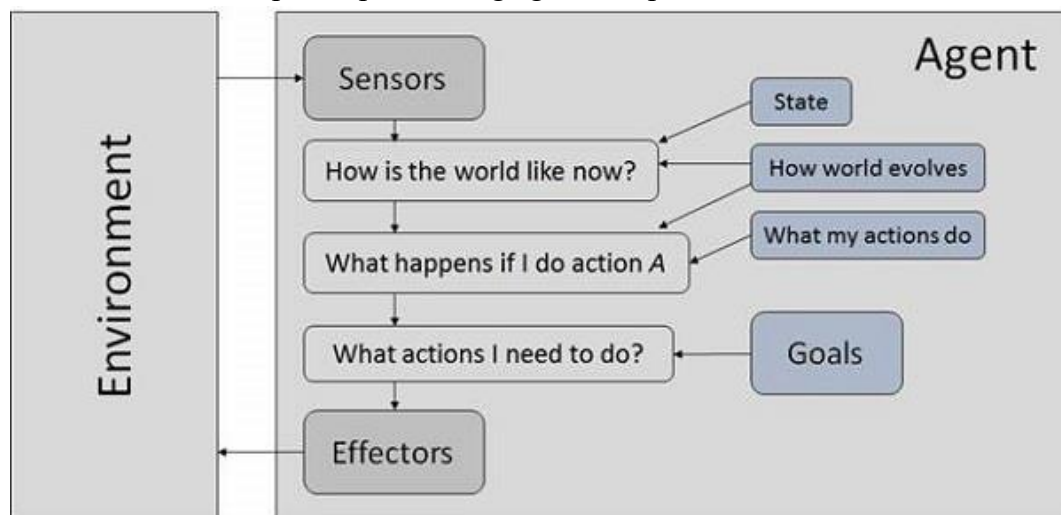
Note

- **Model-Based Reflex Agents** are a step up from simple reflex agents, as they can handle partially observable environments using an internal state and a model of the environment.
- They are particularly useful when the environment cannot be fully observed or when the agent needs to remember past actions or events to make informed decisions.
- However, their reliance on an accurate model and increased computational demands can pose challenges in implementation.

3. Goal Based Agents

Goal-Based Agents are a type of intelligent agent designed to make decisions and take actions based on specific goals. Unlike reflex agents, which act based on predefined rules or current states, goal-based agents focus on achieving objectives by considering future consequences and planning accordingly.

- Use goals to decide actions and evaluate the desirability of outcomes.
- Example: A pathfinding agent that plans routes to a destination.



Features of Goal-Based Agents

1. Goal Orientation

- Operate with explicit goals that define desired states or outcomes.
- Goals guide the agent's actions and decision-making process.

2. Decision-Making through Search and Planning

- Evaluate possible actions by considering their impact on achieving the goal.
- Use search and planning algorithms to find the best path to the goal.

3. Flexibility

- Can adapt to new situations by recalculating paths to the goal when conditions change.

4. Suitability for Complex Environments

- Work well in environments where achieving a goal requires reasoning, planning, or sequential actions.

How Goal-Based Agents Work

Goal-based agents use the following components:

1. **Perception**
 - The agent perceives the environment through sensors to gather information.
2. **State Update**
 - The agent maintains an internal state representing the current environment.
3. **Goals**
 - Goals are desired states that the agent tries to achieve.
4. **Action Selection**
 - The agent evaluates different actions based on their ability to bring it closer to the goal.

Architecture of a Goal-Based Agent

General Workflow:

1. **Input:** Percept from the environment.
2. **State Update:** Update internal state using perception.
3. **Goal Evaluation:** Determine if the goal is achieved or if further action is needed.
4. **Action Selection:** Plan and execute actions to move toward the goal.
5. **Output:** Actuators execute the action.

Pseudocode:

python

Copy code

```
function GoalBasedAgent(percept, state, goals):
    state = UpdateState(state, percept)
    if GoalAchieved(state, goals):
        return "Stop"
    action = PlanAction(state, goals)
    return action
```

Example of a Goal-Based Agent

Autonomous Car

- **Environment:** Roads with other vehicles, traffic lights, and obstacles.
- **Goal:** Reach a destination safely and efficiently.
- **Behavior:**
 - Perceive current location, road conditions, and traffic.
 - Plan routes using maps and navigation systems.
 - Reevaluate routes in case of traffic or roadblocks.

Strengths of Goal-Based Agents

1. **Long-Term Planning**
 - Capable of reasoning about sequences of actions required to achieve a goal.
2. **Adaptability**
 - Can adjust behavior dynamically based on changes in the environment or goals.
3. **Flexibility in Decision-Making**
 - Evaluate multiple strategies and choose the one best aligned with the goal.
4. **Handles Complex Scenarios**
 - Effective in environments where achieving goals involves reasoning about future actions.

Limitations of Goal-Based Agents

1. **Computational Complexity**
 - Planning and search algorithms can be computationally expensive, especially in large or dynamic environments.
2. **Dependence on Goal Definition**
 - The agent's performance heavily depends on well-defined, achievable goals.
3. **Time-Consuming Planning**
 - Real-time environments may pose challenges due to the time needed for planning.
4. **Inability to Handle Trade-Offs**
 - Unlike utility-based agents, goal-based agents lack the ability to weigh trade-offs between competing goals.

Applications of Goal-Based Agents

1. **Robotics**
 - Robots designed for exploration, rescue missions, or manufacturing tasks.
2. **Autonomous Vehicles**
 - Cars navigating to destinations while avoiding obstacles and adhering to traffic rules.
3. **Game AI**
 - Non-player characters (NPCs) with objectives such as reaching a specific location or defeating an opponent.
4. **Logistics and Planning**
 - Systems that optimize delivery routes or manage supply chains.
5. **Virtual Assistants**
 - Assistants like Alexa or Siri performing tasks based on user-defined goals.

Note

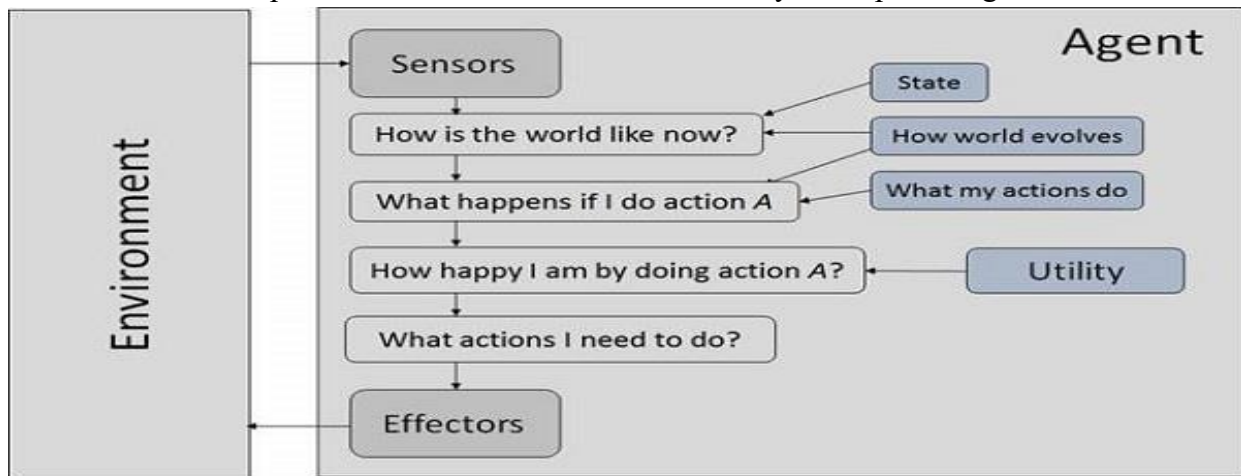
- **Goal-Based Agents** focus on achieving specific objectives by reasoning about future states and planning actions accordingly.

- They are ideal for complex, dynamic environments where achieving a goal requires sequential decision-making.
- While they offer greater flexibility and adaptability than reflex agents, they require more computational resources and careful goal definition.

4. Utility Based Agents

A **Utility-Based Agent** is an advanced type of intelligent agent that not only considers achieving goals but also evaluates the quality or desirability of different outcomes. These agents use a utility function to measure the "*degree of satisfaction*" or "*happiness*" associated with a particular state, enabling them to make decisions that maximize overall utility.

- Optimize utility functions to make trade-offs and find the best solution.
- Example: An e-commerce recommendation system optimizing user satisfaction.



Features of Utility-Based Agents

1. Utility Function

- Assigns a numerical value to each state, representing the agent's preference or satisfaction.
- The goal is to maximize this utility value.

2. Trade-Off Analysis

- Can handle situations where there are multiple conflicting goals by weighing their utilities.
- Example: Choosing between speed and safety in autonomous driving.

3. Decision-Making Under Uncertainty

- Use probabilistic reasoning to evaluate the expected utility of actions in uncertain environments.

4. Flexibility and Optimality

- Can choose among multiple strategies or goals to achieve the most desirable outcome.

How Utility-Based Agents Work

Utility-based agents work by following these steps:

1. **Perception**
 - Gather information about the environment through sensors.
2. **State Update**
 - Update the internal state based on the perception and environment model.
3. **Utility Evaluation**
 - Evaluate the utility of each possible state or action using the utility function.
4. **Action Selection**
 - Choose the action that maximizes the expected utility.

Architecture of a Utility-Based Agent

General Workflow:

1. **Input:** Percept from the environment.
2. **State Update:** Update internal representation of the environment.
3. **Utility Calculation:** Compute the utility for each possible state/action.
4. **Action Selection:** Choose the action with the highest expected utility.
5. **Output:** Execute the selected action.

Pseudocode:

python

Copy code

```
function UtilityBasedAgent(percept):
    state = UpdateState(state, percept)
    action = ChooseActionBasedOnUtility(state)
    return action
```

```
function ChooseActionBasedOnUtility(state):
    best_action = None
    max_utility = -infinity
    for action in actions:
        utility = CalculateUtility(state, action)
        if utility > max_utility:
            best_action = action
            max_utility = utility
    return best_action
```

Example of a Utility-Based Agent

1. Self-Driving Car

- **Environment:** Roads with varying traffic, speed limits, and weather conditions.
- **Utility Function:** Combines factors like safety, speed, fuel efficiency, and passenger comfort.

2. E-Commerce Recommendation System

- **Environment:** Online marketplace with users browsing products.
- **Utility Function:** Measures user satisfaction, combining factors like product relevance, price, and user reviews.
 - The agent recommends products that maximize the expected utility (e.g., likelihood of purchase).

Strengths of Utility-Based Agents

1. **Optimal Decision-Making**
 - Ensures that actions are chosen to achieve the highest possible satisfaction or desirability.
2. **Handles Trade-Offs**
 - Can balance conflicting goals by assigning appropriate weights in the utility function.
3. **Works in Uncertain Environments**
 - Uses probabilities to calculate expected utility when outcomes are not guaranteed.
4. **Adaptable and Flexible**
 - Can adapt to changes in preferences or priorities by modifying the utility function.

Limitations of Utility-Based Agents

1. **Complexity**
 - Designing an accurate utility function can be challenging.
 - Evaluating utility for all possible states/actions can be computationally expensive.
2. **Subjectivity**
 - Utility functions are subjective and depend on how the designer defines preferences.
3. **Scalability Issues**
 - In large or highly dynamic environments, the number of possible states and actions can make computation infeasible.
4. **Dependency on Accurate Models**
 - Requires an accurate model of the environment to calculate utilities effectively.

Applications of Utility-Based Agents

1. **Autonomous Systems**
 - Self-driving cars optimizing safety, fuel efficiency, and travel time.
2. **Decision Support Systems**
 - Medical diagnosis systems recommending treatments based on patient health and preferences.
3. **Robotics**
 - Robots performing tasks while balancing speed, energy consumption, and success rate.

4. *E-Commerce and Marketing*

- Personalization of recommendations to maximize customer satisfaction and revenue.

5. *Game AI*

- NPCs (non-player characters) making strategic decisions in games by maximizing utilities like health, resources, or victory chances.

Note:

- **Utility-Based Agents** focus on achieving the **most desirable outcome** by evaluating the trade-offs between multiple factors.
- They are highly effective in complex and dynamic environments where uncertainty and conflicting objectives exist.
- However, their success depends on the accuracy of the utility function and the computational resources available.

5. **Learning Agents**

A **Learning Agent** is a type of intelligent agent that can improve its performance over time by learning from its experiences and adapting to changes in the environment. This makes it one of the most versatile and advanced types of agents.

Unlike other agents that rely on predefined rules or models, a learning agent refines its behavior dynamically to optimize its performance.

- Improve their performance over time by learning from experiences.
- Example: An AI system that improves gameplay by learning strategies.

Features of Learning Agents

1. *Learning from Experience*

- Improves performance by analyzing past actions and their outcomes.

2. *Adaptability*

- Capable of adapting to changes in the environment or goals over time.

3. *Exploration and Exploitation*

- Balances exploring new actions (to learn) and exploiting learned knowledge (to act optimally).

4. *Feedback-Based*

- Uses feedback (e.g., rewards or penalties) to improve decision-making.

Structure of a Learning Agent

A learning agent consists of four key components:

1. *Learning Element*

- Responsible for making improvements by learning from data or feedback.
- Implements various learning methods like supervised, unsupervised, or reinforcement learning.

2. *Performance Element*

- Executes actions based on the learned knowledge.
- Responsible for interacting with the environment.

3. Critic

- Provides feedback to the agent by comparing the current state or action with a desired goal or standard.
- Evaluates performance and identifies areas for improvement.

4. Problem Generator

- Suggests exploratory actions to help the agent gather new information or improve its understanding.
- Encourages exploration to prevent the agent from getting stuck in suboptimal behaviors.

How Learning Agents Work

1. **Perception:** Gather data from the environment using sensors.
2. **Learning:** Use the learning element to process data and improve internal models.
3. **Action:** Execute actions based on the performance element's decision.
4. **Feedback:** Use feedback from the critic to assess performance.
5. **Exploration:** Use the problem generator to explore new possibilities.

Types of Learning in Agents

1. Supervised Learning

- The agent is provided with labeled examples and learns a mapping between inputs and desired outputs.
- Example: Training a classifier to recognize images of cats and dogs.

2. Unsupervised Learning

- The agent is provided with unlabeled data and discovers patterns or clusters in the data.
- Example: Grouping customers based on purchasing behavior.

3. Reinforcement Learning (RL)

- The agent learns by interacting with the environment and receiving rewards or penalties based on its actions.
- Example: A robot learning to navigate a maze by maximizing cumulative rewards.

4. Semi-Supervised Learning

- A combination of supervised and unsupervised learning, where the agent has access to both labeled and unlabeled data.

5. Online Learning

- The agent updates its knowledge continuously as new data arrives.
- Example: A spam filter that adapts to new types of spam emails.

Example of a Learning Agent

1. Chess-Playing Agent

- **Goal:** Improve its ability to play chess over time.
- **Learning Element:** Learns strategies by analyzing past games and moves.
- **Performance Element:** Uses learned strategies to make decisions during a game.

- **Critic:** Evaluates the agent's performance by comparing outcomes of games (win/loss).
- **Problem Generator:** Encourages the agent to explore novel moves to discover new strategies.

2. Autonomous Driving Agent

- **Goal:** Safely and efficiently navigate roads.
- **Learning Element:** Learns driving patterns, road conditions, and traffic rules.
- **Performance Element:** Drives based on learned behaviors.
- **Critic:** Monitors actions for violations or accidents.
- **Problem Generator:** Encourages exploration of less-traveled routes to improve navigation.

Strengths of Learning Agents

1. **Self-Improvement**
 - Continuously improves performance over time, even without human intervention.
2. **Adaptability**
 - Adjusts to changing environments or unforeseen scenarios.
3. **Scalability**
 - Capable of handling increasingly complex tasks by learning incrementally.
4. **Autonomy**
 - Reduces reliance on predefined rules or models, making it suitable for dynamic environments.

Limitations of Learning Agents

1. **Data Dependency**
 - Requires significant amounts of data to learn effectively.
2. **Time and Resource Intensive**
 - Training and learning processes can be computationally expensive and time-consuming.
3. **Risk of Overfitting**
 - May perform well on training data but poorly on new or unseen data.
4. **Unintended Consequences**
 - Learning agents might develop behaviors that optimize the wrong objectives if the feedback or goals are poorly designed.

Applications of Learning Agents

1. **Robotics**
 - Robots learning to perform tasks like assembly, cleaning, or exploration.
2. **Healthcare**
 - AI systems diagnosing diseases and recommending treatments based on medical records and data.
3. **Finance**
 - Agents predicting stock prices, detecting fraud, or optimizing investment portfolios.

4. **Gaming**

- NPCs in video games learning to adapt to player strategies.

5. **Natural Language Processing (NLP)**

- Virtual assistants improving their understanding of user queries over time.

Note:

- **Learning Agents** are the most advanced type of agent, capable of improving their behavior and adapting to new environments over time.
- They are highly versatile and can be applied to a wide range of tasks, from robotics to recommendation systems.
- However, their effectiveness depends on the quality of the data, feedback, and learning algorithms used.

2. Environment

An **environment** is the external context or surroundings within which an agent operates. The environment provides input to the agent and responds to the agent's actions.

Characteristics of Environments:

1. **Perceptibility:**

- **Fully Observable:** The agent has access to the complete state of the environment.
- **Partially Observable:** The agent only perceives part of the environment.

2. **Determinism:**

- **Deterministic:** Actions have predictable outcomes.
- **Stochastic:** Outcomes are uncertain, even with the same action.

3. **Dynamics:**

- **Static:** The environment remains unchanged unless acted upon.
- **Dynamic:** The environment changes independently of the agent.

4. **Discrete vs. Continuous:**

- **Discrete:** Limited set of states or actions (e.g., a chessboard).
- **Continuous:** Infinite states or actions (e.g., a self-driving car's road environment).

5. **Multi-Agent vs. Single-Agent:**

- **Single-Agent:** Only one agent interacts with the environment.
- **Multi-Agent:** Multiple agents interact, often competitively or cooperatively.

Agent-Environment Interaction

The interaction between an agent and its environment is often modeled as a feedback loop:

1. **Perception:** The agent perceives the environment via its sensors.
2. **Decision-making:** The agent uses a decision-making mechanism (rules, logic, or algorithms).
3. **Action:** The agent acts on the environment using its actuators.
4. **Feedback:** The environment changes in response to the action, completing the loop.

This cycle repeats as the agent strives to achieve its objectives.

Agent Design

Designing an agent involves:

1. **Defining the Environment:** Understanding its characteristics (e.g., deterministic vs. stochastic).
2. **Developing a Perception Mechanism:** Sensors or data streams to gather inputs.
3. **Implementing Decision Logic:** Algorithms, machine learning models, or rules to determine actions.
4. **Executing Actions:** Actuators or outputs that impact the environment.

INTRODUCTION TO MACHINE LEARNING

Machine learning (ML) is a type of artificial intelligence (AI) that allows computers to learn without being explicitly programmed. This article explores the concept of machine learning, providing various definitions and discussing its applications.

Machine learning (ML) is a type of Artificial Intelligence (AI) that allows computers to learn and make decisions without being explicitly programmed. It involves feeding data into algorithms that can then identify patterns and make predictions on new data.

Machine learning is used in a wide variety of applications, including image and speech recognition, natural language processing, and recommender systems. Machine learning is able to learn, train from data and solve/predict complex solutions which cannot be done with traditional programming. It enables us with better decision making and solve complex business problems in optimized time. Machine learning has applications in various fields, like Healthcare, finance, educations, sports and more.

Reasons why Machine learning has become essential in every field:-

1. Solving Complex Business Problems:

It is too complex to tackle problems like Image recognition, Natural language processing, disease diagnose etc. with Traditional programming. Machine learning can handle such problems by learning from examples or making predictions, rather than following some rigid rules.

2. Handling Large Volumes of Data:

Expansion of Internet and users is producing massive amount of data. Machine Learning can process these data effectively and analyze, predict useful insights from them.

- For example, ML can analyze millions of everyday transactions to detect any fraud activity in real time.
- Social platforms like Facebook, Instagram use ML to analyze billions of post, like and share to predict next recommendation in your feed.

3. Automate Repetitive Tasks:

With Machine Learning, we can automate time-consuming and repetitive tasks, with better accuracy.

- Gmail uses ML to filter out Spam emails and ensure your Index stay clean and spam free. Using traditional programming or handling these manually will only make the system error-prone.
- Customer Support chatbots can use ML to solve frequent occurring problems like Checking order status, Password reset etc.
- Big organizations can use ML to process large amount of data (like Invoices etc) to extract historical and current key insights.

4. *Personalized User Experience:*

All social-media, OTT and E-commerce platforms uses Machine learning to recommend better feed based on user preference or interest.

- Netflix recommends movies and TV shows based on what you've watched
- E-commerce platforms suggesting products you are likely to buy.

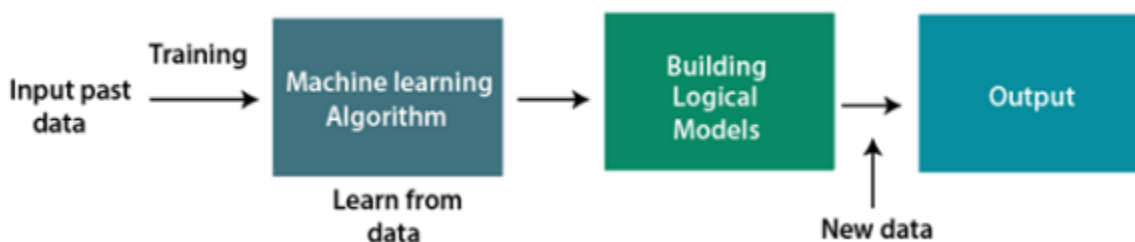
5. *Self-Improvement in Performance:*

ML models are able to improve themselves based on more data, like user-behavior and feedback. For example,

- **Voice Assistants (Siri, Alexa, Google Assistant)** – Voice assistants continuously improve as they process millions of voice inputs. They adapt to user preferences, understand regional accents better, and handle ambiguous queries more effectively.
- **Search Engines (Google, Bing)** – Search engines analyze user behavior to refine their ranking algorithms.
- **Self-driving Cars** – Self-driving cars use data from millions of miles driven (both in simulations and real-world scenarios) to enhance their decision-making.

How does Machine Learning work

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically. It is a data-driven technology.

- Machine learning is much similar to data mining as it also deals with the huge amount of the data. There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning.

There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning.

- Supervised learning involves training a model on labeled data, while unsupervised learning involves training a model on unlabeled data.
- Reinforcement learning involves training a model through trial and error.
- Machine learning is used in a wide variety of applications, including image and speech recognition, natural language processing, and recommender systems

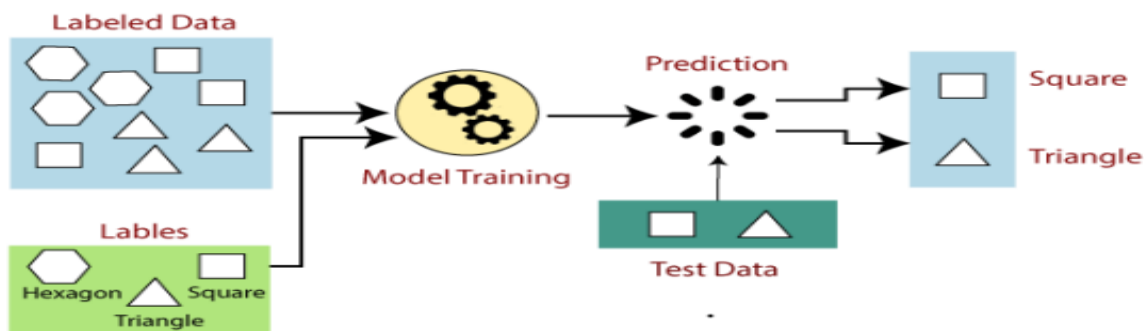
Classification of Machine Learning

Machine learning implementations are classified into four major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:

1. Supervised learning:

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. The given data is labeled. Both *classification* and *regression* problems are supervised learning problems.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. The working of Supervised learning can be easily understood by the below example and diagram:

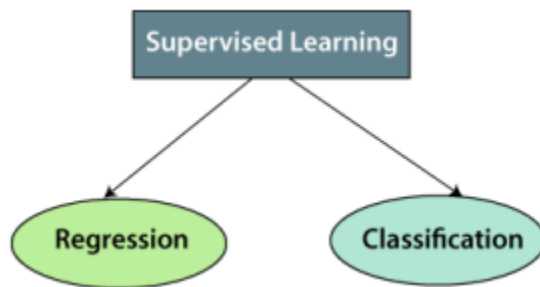


Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square. o If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has six equal sides then it will be labelled as hexagon. Now, after training, we test our model using the test set, and the task of the model is to identify the shape. The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output. Steps Involved in Supervised Learning:
 - o First Determine the type of training dataset
 - o Collect/Gather the labelled training data.
 - o Split the training dataset into training dataset, test dataset, and validation dataset.
 - o Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
 - o Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc. o Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
 - o Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:



- **Regression algorithms** are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.
- **Classification models**, on the other hand, predict discrete categories, such as spam or not spam, or whether a customer will default on a loan.

To assign data points to predefined categories or classes. Examples:

- o **Logistic Regression:** Predicting the probability of a binary outcome, such as whether an email is spam or not.

- *Support Vector Machines (SVM)*: Effective for both linear and non-linear classification tasks.
- *Decision Trees*: A hierarchical structure that makes decisions based on feature values.
- *Random Forests*: An ensemble method that combines multiple decision trees to improve accuracy.
- *Naive Bayes*: A probabilistic classifier based on Bayes' theorem.
- *K-Nearest Neighbors (KNN)*: Classifies a data point based on the majority class of its nearest neighbors.
- *Neural Networks*: Complex models with interconnected layers, particularly useful for intricate classification problems.

2. Unsupervised learning:

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. In unsupervised learning algorithms, classification or categorization is not included in the observations

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

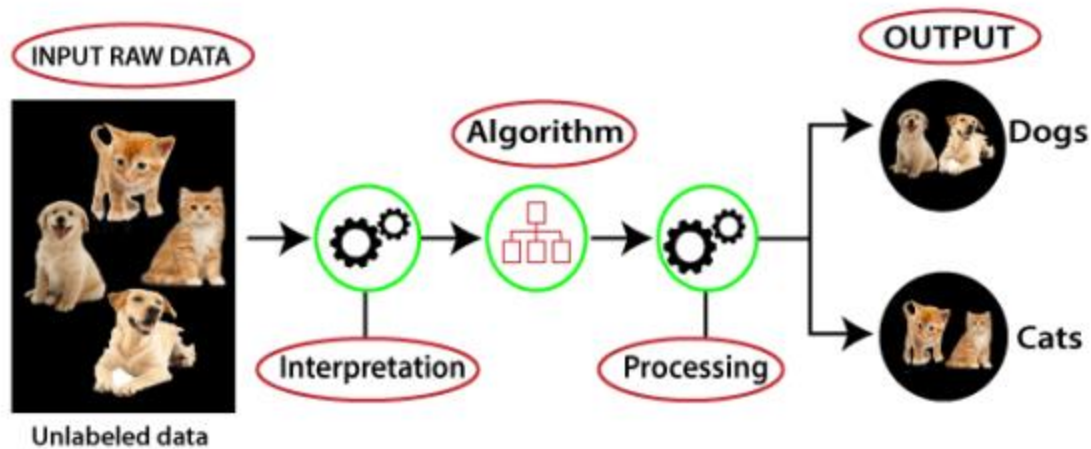
Below are some main reasons which describe the importance of Unsupervised Learning: o Unsupervised learning is helpful for finding useful insights from the data.

Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.

- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

Working of Unsupervised

Learning Working of unsupervised learning can be understood by the below diagram:

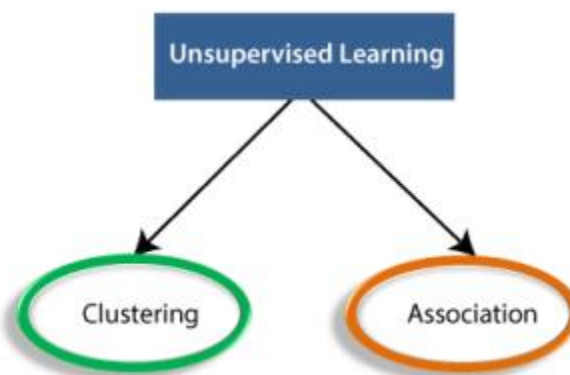


Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



- **Clustering:** Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

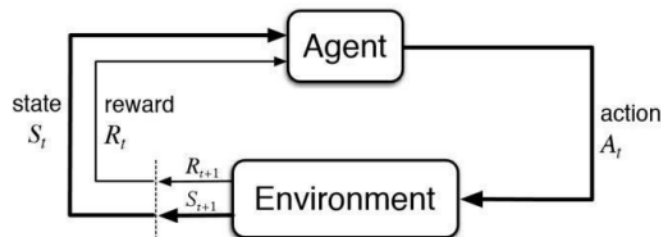
- **Association:** An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

3. Reinforcement learning:

Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards.

A learner is not told what actions to take as in most forms of machine learning but instead must discover which actions yield the most reward by trying them. For example — Consider teaching a dog a new trick: we cannot tell him what to do, what not to do, but we can reward/punish it if it does the right/wrong thing.

When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.



A learner is not told what actions to take as in most forms of machine learning but instead must discover which actions yield the most reward by trying them. For example — Consider teaching a dog a new trick: we cannot tell him what to do, what not to do, but we can reward/punish it if it does the right/wrong thing.

Reinforcement learning (RL) is based on rewarding desired behaviors or punishing undesired ones. Instead of one input producing one output, the algorithm produces a variety of outputs and is trained to select the right one based on certain variables – Gartner.

It is a type of machine learning technique where a computer agent learns to perform a task through repeated trial and error interactions with a dynamic environment. This learning approach enables the agent to make a series of decisions that maximize a reward metric for the task without human intervention and without being explicitly programmed to achieve the task

4. Semi-supervised learning:

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

Semi-supervised learning is an approach to machine learning that combines small labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning and supervised learning.

Categorizing based on Required Output

Another categorization of machine-learning tasks arises when one considers the desired output of a machine-learned system:

1. **Classification:** When inputs are divided into two or more classes, the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are “spam” and “not spam”.
2. **Regression:** This is also a supervised problem, A case when the outputs are continuous rather than discrete.
3. **Clustering:** When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

Examples of Machine Learning in Action

Machine learning is woven into the fabric of our daily lives. Here are some examples to illustrate its diverse applications.

Supervised Learning

- **Filtering Your Inbox:** Spam filters use machine learning to analyze emails and identify spam based on past patterns. They learn from emails you mark as spam and not spam, becoming more accurate over time.
- **Recommending Your Next Purchase:** E-commerce platforms and streaming services use machine learning to analyze your purchase history and viewing habits. This allows them to recommend products and shows you're more likely to enjoy.
- **Smart Reply in Emails:** Machine learning powers features like “Smart Reply” in Gmail, suggesting short responses based on the content of the email.

Unsupervised Learning

- **Grouping Customers:** Machine learning can analyze customer data (purchase history, demographics) to identify customer segments with similar characteristics. This helps businesses tailor marketing campaigns and product offerings.
- **Anomaly Detection:** Financial institutions use machine learning to detect unusual spending patterns on your credit card, potentially indicating fraudulent activity.
- **Image Classification in Photos:** Facial recognition in photos on social media platforms is powered by machine learning algorithms trained on vast amounts of labeled data.

Beyond Categories

- **Self-Driving Cars:** These rely on reinforcement learning, a type of machine learning where algorithms learn through trial and error in a simulated environment.
- **Medical Diagnosis:** Machine learning algorithms can analyze medical images (X-rays, MRIs) to identify abnormalities and aid doctors in diagnosis.

CONCEPT OF DEEP LEARNING

Deep Learning is computer software that mimics the network of neurons in a brain. It is a subset of machine learning based on artificial neural networks with representation learning. It is called deep learning because it makes use of deep neural networks. This learning can be supervised, semi-supervised or unsupervised.

In deep learning, models use different layers to learn and discover insights from the data. Some popular applications of deep learning are self-driving cars, language translation, natural language processing, etc. Some popular deep learning models are:

- Convolutional Neural Network
- Recurrent Neural Network
- Autoencoders
- Classic Neural Networks, etc.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep learning vs. machine learning

If deep learning is a subset of machine learning, how do they differ? Deep learning distinguishes itself from classical machine learning by the type of data that it works with and the methods in which it learns.

Machine learning algorithms leverage structured, labeled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. This doesn't necessarily mean that it doesn't use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format.

Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let's say that we had a set of photos of different pets, and we wanted to categorize by "cat", "dog", "hamster", et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

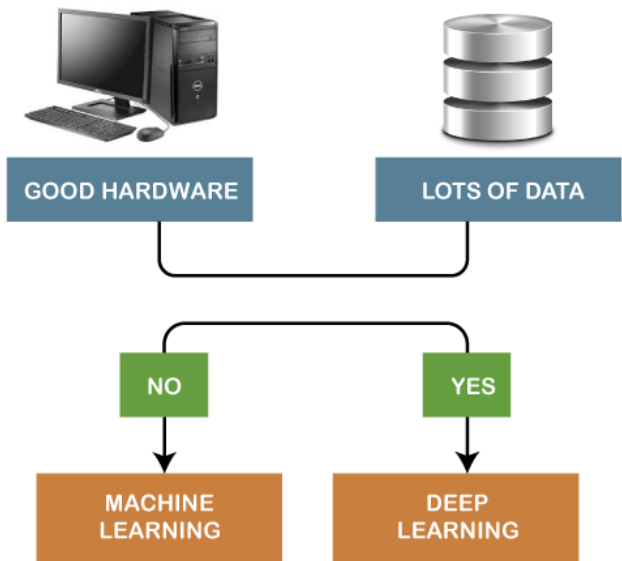
Then, through the processes of gradient descent and back propagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions about a new photo of an animal with increased precision.

Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn't require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

Parameter	Machine Learning	Deep Learning
Data Dependency	Although machine learning depends on the huge amount of data, it can work with a smaller amount of data.	Deep Learning algorithms highly depend on a large amount of data, so we need to feed a large amount of data for good performance.
Execution time	Machine learning algorithm takes less time to train the model than deep learning, but it takes a long-time duration to test the model.	Deep Learning takes a long execution time to train the model, but less time to test the model.
Hardware Dependencies	Since machine learning models do not need much amount of data, so they can work on low-end machines.	The deep learning model needs a huge amount of data to work efficiently, so they need GPU's and hence the high-end machine.
Feature Engineering	Machine learning models need a step of feature extraction by the expert, and then it proceeds further.	Deep learning is the enhanced version of machine learning, so it does not need to develop the feature extractor for each problem; instead, it tries to learn high-level features from the data on its own.
Problem-solving approach	To solve a given problem, the traditional ML model breaks the problem in sub-parts, and after	The problem-solving approach of a deep learning model is different from the traditional ML model, as it takes input for a given problem,

Parameter	Machine Learning	Deep Learning
	solving each part, produces the final result.	and produce the end result. Hence it follows the end-to-end approach.
Interpretation of result	The interpretation of the result for a given problem is easy. As when we work with machine learning, we can interpret the result easily, it means why this result occur, what was the process.	The interpretation of the result for a given problem is very difficult. As when we work with the deep learning model, we may get a better result for a given problem than the machine learning model, but we cannot find why this particular outcome occurred, and the reasoning.
Type of data	Machine learning models mostly require data in a structured form.	Deep Learning models can work with structured and unstructured data both as they rely on the layers of the Artificial neural network.
Suitable for	Machine learning models are suitable for solving simple or bit-complex problems.	Deep learning models are suitable for solving complex problems.

As we have seen the brief introduction of ML and DL with some comparisons, now why and which one needs to be chosen to solve a particular problem. So, it can be understood by the given flowchart:



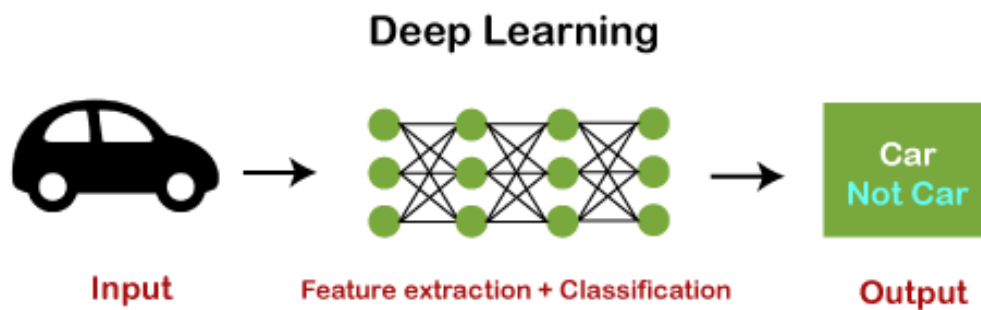
Hence, if you have lots of data and high hardware capabilities, go with deep learning. But if you don't have any of them, choose the ML model to solve your problem.

In conclusion, we can say that deep learning is machine learning with more capabilities and a different working approach. And selecting any of them to solve a particular problem is depending on the amount of data and complexity of the problem.

How Deep Learning Works

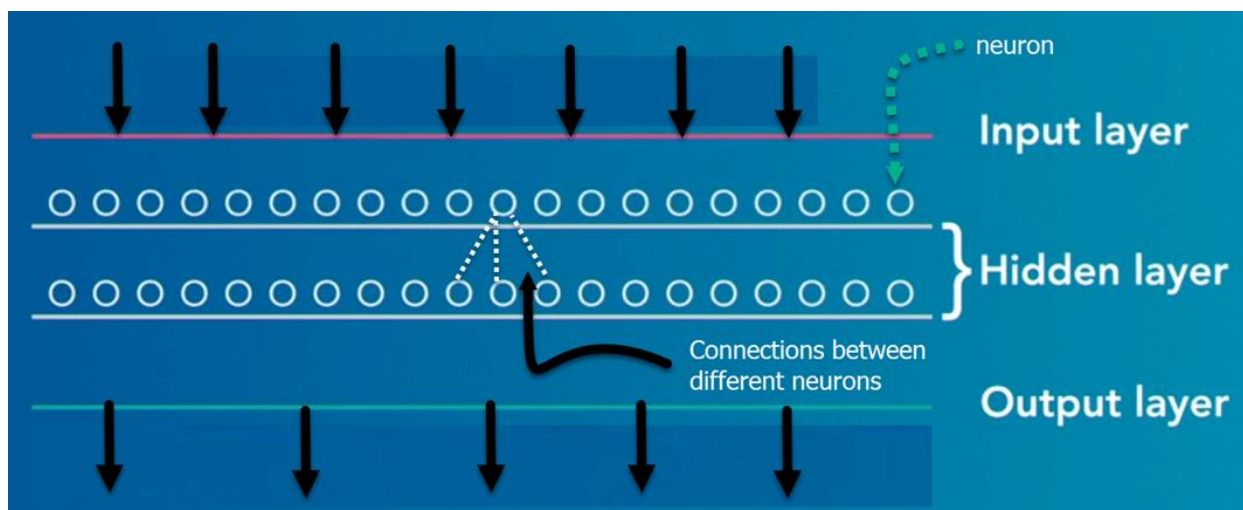
We can understand the working of deep learning with the same example of identifying cat vs. dog. The deep learning model takes the images as the input and feed it directly to the algorithms without requiring any manual feature extraction step. The images pass to the different layers of the artificial neural network and predict the final output.

Consider the below image:



Deep learning algorithms are constructed with connected layers.

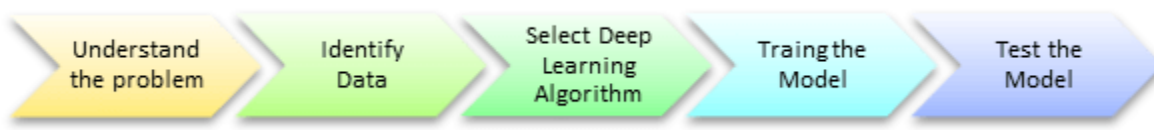
- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.



Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receives the layer above it. The strength of the signal given the neuron in the next layer depends on the weight, bias and activation function. The network consumes large amounts of input data and operates them through multiple layers; the network can learn increasingly complex features of the data at each layer.

Deep learning Process

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.



Deep learning Process

To grasp the idea of deep learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word ‘cat.’ As his parents are concerned about his education, they keep telling him ‘Yes, that is a cat’ or ‘No, that is not a cat.’ The infant persists in pointing objects but becomes more accurate with ‘cats.’ The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind.

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with two layers.

The learning occurs in two phases:

First Phase: The first phase consists of applying a nonlinear transformation of the input and creates a statistical model as output.

Second Phase: The second phase aims at improving the model with a mathematical method known as derivative.

The neural network repeats these two phases hundreds to thousands of times until it has reached a tolerable level of accuracy. The repeat of this two-phase is called *iteration*.

To give a Deep learning example, take a look at the motion below, the model is trying to learn how to dance. After 10 minutes of training, the model does not know how to dance, and it looks like a scribble.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,

- *Convolutional neural networks (CNNs)*, used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition.
- *Recurrent neural network (RNNs)* are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

Classification of Neural Networks

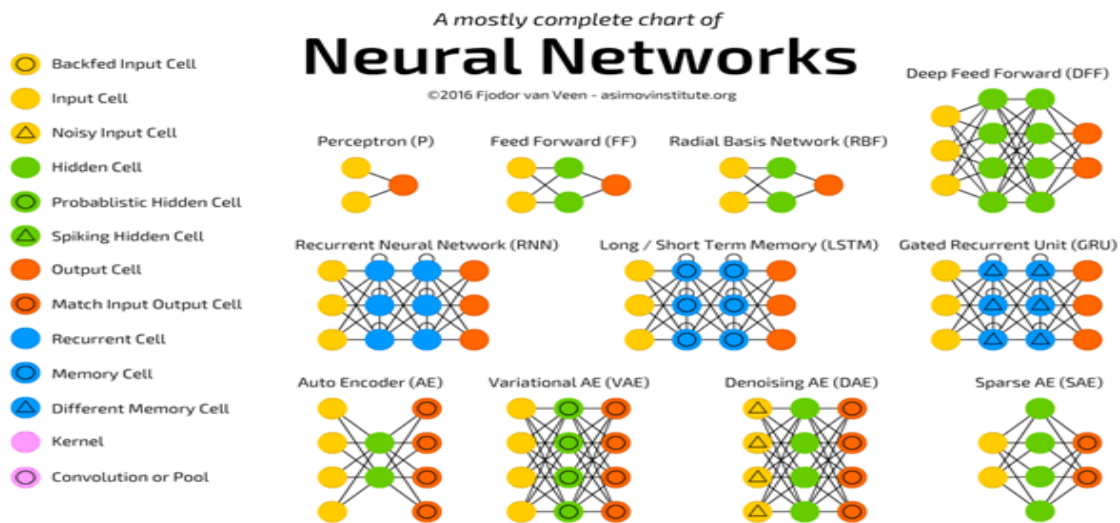
Shallow neural network: The Shallow neural network has only one hidden layer between the input and output.

Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

Nowadays, deep learning is used in many ways like a driverless car, mobile phone, Google Search Engine, Fraud detection, TV, and so on.

Types of Deep Learning Networks

Now in this Deep Neural network tutorial, we will learn about types of Deep Learning Networks:



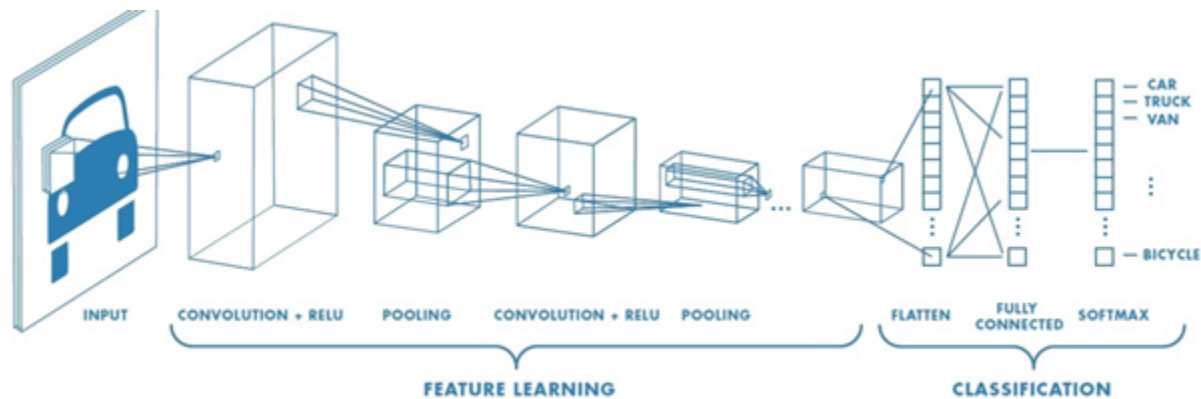
Types of Deep Learning Networks

Feed-forward Neural Networks

With this type of architecture, information flows in only one direction, forward. It means, the information's flows starts at the input layer, goes to the "hidden" layers, and end at the output layer. The network does not have a loop. Information stops at the output layers.

Convolutional Neural Networks (CNN)

CNN is a multi-layered neural network with a unique architecture designed to extract increasingly complex features of the data at each layer to determine the output. CNN's are well suited for perceptual tasks.



Convolutional Neural Network

CNN is mostly used when there is an unstructured data set (e.g., images) and the practitioners need to extract information from it.

For instance, if the task is to predict an image caption:

- The CNN receives an image of let's say a cat, this image, in computer term, is a collection of the pixel. Generally, one layer for the greyscale picture and three layers for a color picture.
- During the feature learning (i.e., hidden layers), the network will identify unique features, for instance, the tail of the cat, the ear, etc.
- When the network thoroughly learned how to recognize a picture, it can provide a probability for each image it knows. The label with the highest probability will become the prediction of the network.

MODEL PERFORMANCE MEASURE

Most model-performance measures are based on the comparison of the model's predictions with the (known) values of the dependent variable in a dataset. For an ideal model, the predictions and the dependent-variable values should be equal. In practice, it is never the case, and we want to quantify the disagreement.

The measures may be applied for several purposes, including:

- *Model evaluation*: we may want to know how good the model is, i.e., how reliable are the model's predictions (how frequent and how large errors may we expect)?;
- *Model comparison*: we may want to compare two or more models in order to choose between them;

- *Out-of-sample and out-of-time comparisons*: we may want to check a model's performance when applied to new data to evaluate if performance has not worsened.

Depending on the nature of the dependent variable (continuous, binary, categorical, count, etc.), different model-performance measures may be used. Moreover, the list of useful measures is growing as new applications emerge. here, we will discuss only a selected set of measures, some of which are used in dataset-level exploration techniques.

Confusion Matrix in Machine Learning

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an ***error matrix***. Some features of Confusion matrix are given below:

- For the 2 prediction classes of classifiers, the matrix is of 2*2 tables, for 3 classes, it is 3*3 tables, and so on.
- The matrix is divided into two dimensions that are ***predicted values*** and ***actual values*** along with the total number of predictions.
- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
- It looks like the below table:

n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

The above table has the following cases:

- ***True Negative***: Model has given prediction No, and the real or actual value was also No.
- ***True Positive***: The model has predicted yes, and the actual value was also true.
- ***False Negative***: The model has predicted no, but the actual value was Yes, it is also called as ***Type-II error***.
- ***False Positive***: The model has predicted Yes, but the actual value was No. It is also called a ***Type-I error***.

Need for Confusion Matrix in Machine learning

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either *type-I* or *type-II error*.

- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

Example: We can understand the confusion matrix using an example.

Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. So, the confusion matrix for this is given as:

n = 100	Actual: No	Actual: Yes	
Predicted: No	TN: 65	FP: 3	68
Predicted: Yes	FN: 8	TP: 24	32
	73	27	

From the above example, we can conclude that:

- The table is given for the two-class classifier, which has two predictions "Yes" and "NO." Here, Yes defines that patient has the disease, and No defines that patient does not has that disease.
- The classifier has made a total of **100 predictions**. Out of 100 predictions, **89 are true predictions**, and **11 are incorrect predictions**.
- The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.

Calculations using Confusion Matrix:

We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are given below:

- **Classification Accuracy:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

- **Misclassification rate:** It is also termed as Error rate, and it defines how often the model gives the wrong predictions. The value of error rate can be calculated as the number of incorrect predictions to all number of the predictions made by the classifier. The formula is given below:

$$\text{Error rate} = \frac{FP+FN}{TP+FP+FN+TN}$$

- **Precision:** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:

$$\text{Precision} = \frac{TP}{TP+FP}$$

- **Recall:** It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.

$$\text{Recall} = \frac{TP}{TP+FN}$$

- **F-measure:** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

$$\text{F-measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

From the confusion matrix, Sensitivity and Specificity is evaluated using the following equations also.

- **Sensitivity** (true positive rate) refers to the probability of a positive test, conditioned on truly being positive.
- **Specificity** (true negative rate) refers to the probability of a negative test, conditioned on truly being negative

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Cross-Validation in Machine Learning

Cross-validation is a technique for validating the model efficiency by training it on the subset of input data and testing on previously unseen subset of the input data. *We can also say that it is a technique to check how a statistical model generalizes to an independent dataset.*

In machine learning, there is always the need to test the stability of the model. It means based only on the training dataset; we can't fit our model on the training dataset. For this purpose, we reserve

a particular sample of the dataset, which was not part of the training dataset. After that, we test our model on that sample before deployment, and this complete process comes under cross-validation. This is something different from the general train-test split.

Hence the basic steps of cross-validations are:

- Reserve a subset of the dataset as a validation set.
- Provide the training to the model using the training dataset.
- Now, evaluate model performance using the validation set. If the model performs well with the validation set, perform the further step, else check for the issues.

Methods used for Cross-Validation

There are some common methods that are used for cross-validation. These methods are given below:

1. Validation Set Approach
2. Leave-P-out cross-validation
3. Leave one out cross-validation
4. K-fold cross-validation
5. Stratified k-fold cross-validation
6. Holdout Method

Validation Set Approach

We divide our input dataset into a training set and test or validation set in the validation set approach. Both the subsets are given 50% of the dataset. But it has one of the big disadvantages that we are just using a 50% dataset to train our model, so the model may miss out to capture important information of the dataset. It also tends to give the underfitted model.

Leave-P-out cross-validation

In this approach, the p datasets are left out of the training data. It means, if there are total n datapoints in the original input dataset, then $n-p$ data points will be used as the training dataset and the p data points as the validation set. This complete process is repeated for all the samples, and the average error is calculated to know the effectiveness of the model.

There is a disadvantage of this technique; that is, it can be computationally difficult for the large p .

Leave one out cross-validation

This method is similar to the leave- p -out cross-validation, but instead of p , we need to take 1 dataset out of training. It means, in this approach, for each learning set, only one datapoint is reserved, and the remaining dataset is used to train the model. This process repeats for each datapoint. Hence for n samples, we get n different training set and n test set. It has the following features:

- In this approach, the bias is minimum as all the data points are used.
- The process is executed for n times; hence execution time is high.
- This approach leads to high variation in testing the effectiveness of the model as we iteratively check against one data point.

K-Fold Cross-Validation

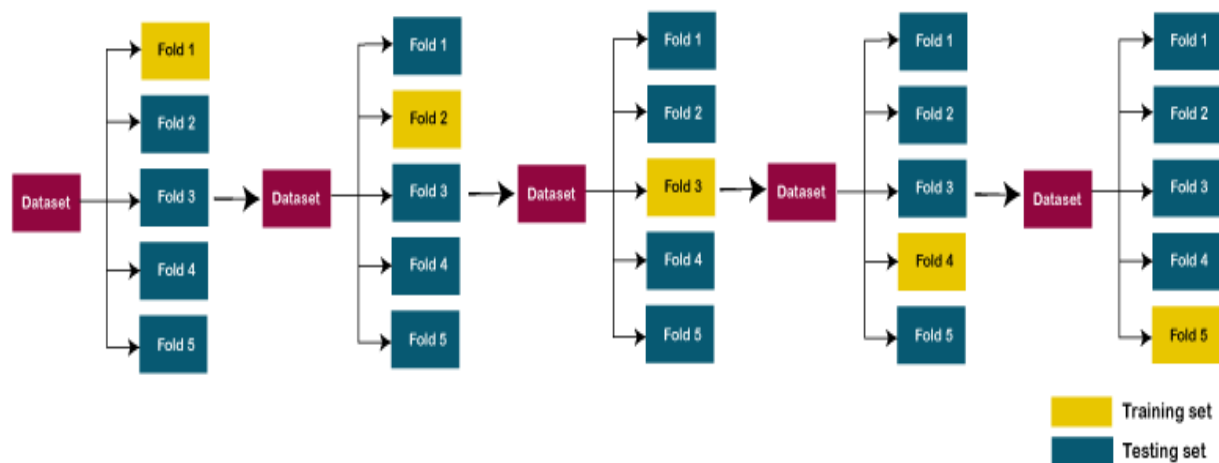
K-fold cross-validation approach divides the input dataset into K groups of samples of equal sizes. These samples are called **folds**. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

The steps for k-fold cross-validation are:

- Split the input dataset into K groups
- For each group:
 - Take one group as the reserve or test data set.
 - Use remaining groups as the training dataset
 - Fit the model on the training set and evaluate the performance of the model using the test set.

Let's take an example of 5-folds cross-validation. So, the dataset is grouped into 5 folds. On 1st iteration, the first fold is reserved for test the model and rest are used to train the model. On 2nd iteration, the second fold is used to test the model, and rest is used to train the model. This process will continue until each fold is not used for the test fold.

Consider the below diagram:



Stratified k-fold cross-validation

This technique is similar to k-fold cross-validation with some little changes. This approach works on stratification concept, it is a process of rearranging the data to ensure that each fold or group is a good representative of the complete dataset. To deal with the bias and variance, it is one of the best approaches.

It can be understood with an example of housing prices, such that the price of some houses can be much high than other houses. To tackle such situations, a stratified k-fold cross-validation technique is useful.

Holdout Method

This method is the simplest cross-validation technique among all. In this method, we need to remove a subset of the training data and use it to get prediction results by training it on the rest part of the dataset.

The error that occurs in this process tells how well our model will perform with the unknown dataset. Although this approach is simple to perform, it still faces the issue of high variance, and it also produces misleading results sometimes.

Comparison of Cross-validation to train/test split in Machine Learning

- **Train/test split**: The input data is divided into two parts that are training set and test set on a ratio of 70:30, 80:20, etc. It provides a high variance, which is one of the biggest disadvantages.
 - **Training Data**: The training data is used to train the model, and the dependent variable is known.
 - **Test Data**: The test data is used to make the predictions from the model that is already trained on the training data. This has the same features as training data but not the part of that.

Cross-Validation dataset

It is used to overcome the disadvantage of train/test split by splitting the dataset into groups of train/test splits, and averaging the result. It can be used if we want to optimize our model that has been trained on the training dataset for the best performance. It is more efficient as compared to train/test split as every observation is used for the training and testing both.

Limitations of Cross-Validation

There are some limitations of the cross-validation technique, which are given below:

- For the ideal conditions, it provides the optimum output. But for the inconsistent data, it may produce a drastic result. So, it is one of the big disadvantages of cross-validation, as there is no certainty of the type of data in machine learning.
- In predictive modeling, the data evolves over a period, due to which, it may face the differences between the training set and validation sets. Such as if we create a model for the prediction of stock market values, and the data is trained on the previous 5 years stock values, but the realistic future values for the next 5 years may drastically different, so it is difficult to expect the correct output for such situations.

Applications of Cross-Validation

- This technique can be used to compare the performance of different predictive modeling methods.
 - It has great scope in the medical research field.
 - It can also be used for the meta-analysis, as it is already being used by the data scientists in the field of medical statistics.
-