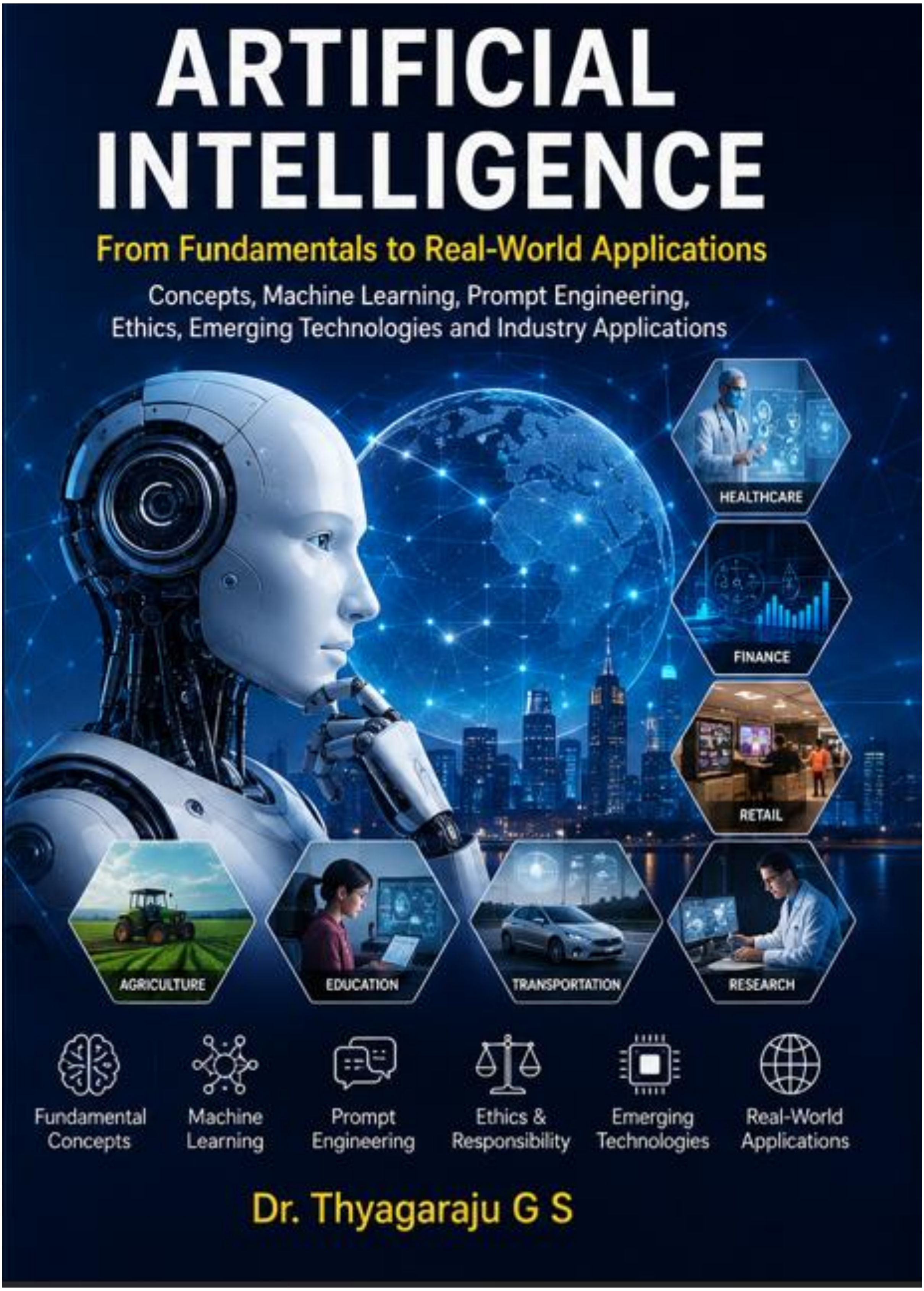

ARTIFICIAL INTELLIGENCE
From Fundamentals to Real-World Applications
Concepts, Machine Learning, Prompt Engineering, Ethics, Emerging Technologies and Industry Applications
HEALTHCARE
FINANCE
RETAIL
AGRICULTURE
EDUCATION
TRANSPORTATION
RESEARCH
Fundamental Concepts
Machine Learning
Prompt Engineering
Ethics & Responsibility
Emerging Technologies
Real-World Applications
Dr. Thyagaraju G S

Contents

Chapter 8 : Learning Resources

About the Author

Dr. Thyagaraju G. S is a Professor and Head of the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire, Karnataka, India. He has more than 21 years of teaching experience in Computer Science, along with extensive experience in Artificial Intelligence, Machine Learning, Context-Aware Computing, Data Science, Internet of Things and emerging intelligent technologies.

He holds a B.Sc., M.Sc., M.Tech. in Computer Science and Technology, and Ph.D. in Computer Science and Engineering from institutions in Karnataka. His doctoral research focused on “Context Aware Computing for Ubiquitous Computing Applications”, with Artificial Intelligence as the research domain. His research involved designing algorithms and developing context-sensitive recommendation systems for applications such as mobile phones and interactive television using approaches including rule-based systems, Bayesian networks, rough set theory and decision tables.

His academic and professional development has also included specialized learning in Artificial Intelligence, Data Science, Machine Learning, Python, Deep Learning, Quantum Computing, Quantum Algorithms and Cryptography. His certifications and advanced learning include programs from NPTEL–IIT, IBM Cognitive Class, Coursera, SWAYAM and Udemy, covering areas such as Machine Learning, Data Science, Python, Quantum Computing, Deep Learning and Agentic Artificial Intelligence.

Dr. Thyagaraju has actively contributed to AI and emerging-technology education as a resource person, workshop instructor, technical speaker and faculty-development-program expert. He has conducted and delivered sessions on topics including *Python for Machine Learning, Machine Learning using Python, Introduction to AI, Linear Algebra for AI, Data Science and Machine Learning, Design Thinking for AI and ML, Artificial Intelligence and Blockchain, Data Analytics using Python, Quantum Computing, Generative AI and AI-based research*. He has also served as a session chair at the 2025 IEEE International Conference on Emerging Trends in Computing and Communication (ETCOM).

His research interests have evolved from Context-Aware Computing and Contextual Artificial Intelligence toward emerging areas such as Quantum Artificial Intelligence-based Agents (Quantum Minds). His research and publications cover a broad range of AI-related applications, including context-aware recommendation systems, healthcare AI, agriculture, IoT, smart homes, privacy and security, deep learning, computer vision, autonomous systems, generative AI, AI agents and intelligent educational applications. His curriculum vitae records 66 research publications, including international journal and conference publications, with several appearing in Scopus-indexed and Web of Science-indexed venues.

His recent research activities demonstrate his continued engagement with contemporary AI. These include work on AI-based disease mapping for Arecanut, diabetic retinopathy detection using AI and deep learning, AI-powered career guidance, deepfake detection, AI agents for business applications, handwritten answer-script evaluation, and AI-powered yoga posture detection and personalized wellness insights.

He has also contributed to applied AI and interdisciplinary projects. Projects carried out under his guidance include machine-learning-based disease mapping, a deep-neural-network-based self-driving toy car, and crop-field monitoring systems. His patents include innovations related to IoT, Machine Learning and AI for healthcare data management, an AI-based coconut scraper, and an AI and cloud-based public security camera device.

Dr. Thyagaraju is also an author of books in the areas of Context-Aware Computing, C Programming and Data Structures, and Python Programming, and has contributed a book chapter on the role of Generative AI in the Internet of Things.

Alongside teaching and research, he has undertaken academic leadership responsibilities as

Professor and Head of the Department of Computer Science and Engineering and has served in academic curriculum-related roles, including as a VTU-nominated Board of Studies member. He also serves as a reviewer for international journals, including Elsevier journals in Artificial Intelligence and Applied Soft Computing.

With a strong combination of teaching, research, academic leadership, practical experimentation and lifelong learning, Dr. Thyagaraju brings a multidisciplinary perspective to Artificial Intelligence. His work reflects a continuing interest in connecting fundamental AI concepts with emerging technologies and meaningful real-world applications.

This book, *Artificial Intelligence: From Fundamentals to Real-World Applications*, represents this approach—building AI understanding progressively from its foundations and machine intelligence to Machine Learning, Prompt Engineering, Ethics, Emerging Technologies, Robotics and Industry Applications. It is intended to serve as a practical and conceptual learning resource for students, educators, researchers, professionals and anyone interested in understanding the rapidly evolving world of Artificial Intelligence.

“The journey of Artificial Intelligence is not only about making machines intelligent; it is about expanding human capability through responsible, contextual and meaningful intelligence.”

Preface

Artificial Intelligence (AI) has evolved from a subject of scientific research into one of the most influential technologies of the modern digital era. From intelligent search engines and recommendation systems to generative AI, autonomous vehicles, smart healthcare, robotics and scientific discovery, AI is increasingly becoming part of everyday life. Its influence extends across education, business, healthcare, finance, agriculture, transportation, manufacturing, research and many other fields.

The rapid development of AI has also changed the way we understand intelligence, learning and human–machine interaction. Modern AI systems are no longer limited to following predefined instructions. They can learn from data, recognize patterns, generate content, make predictions, interact using natural language and assist humans in solving complex problems. As a result, understanding AI has become important not only for computer science professionals but also for students, educators, researchers, entrepreneurs and professionals from diverse disciplines.

The book ***Artificial Intelligence: From Fundamentals to Real-World Applications***, with the subtitle ***Concepts, Machine Learning, Prompt Engineering, Ethics, Emerging Technologies and Industry Applications***, has been developed to provide a structured and accessible introduction to this rapidly evolving field. The book is designed especially for beginners and learners who want to understand both the **fundamental concepts of AI and its practical applications**.

The book begins with the foundations of Artificial Intelligence. **Chapter 1** introduces the meaning, working principles, advantages, limitations, history and evolution of AI. It also explains different types of AI and distinguishes AI from related concepts such as augmented intelligence and cognitive computing. The chapter concludes with an introduction to machine learning and deep learning, providing the foundation for subsequent chapters.

Chapter 2 explores **Machine Intelligence and Knowledge Representation**. It introduces the concept of intelligence from both human and machine perspectives and discusses agents, environments, search techniques and knowledge representation. Uninformed and informed search, heuristic search, Best-First Search and knowledge-based agents are introduced to help readers understand how intelligent systems represent knowledge and solve problems.

One of the most important developments in modern AI is the ability of humans to communicate with AI systems using natural language. **Chapter 3**, therefore, focuses on **Prompt Engineering for AI**. It introduces the evolution and importance of prompt engineering, different types of prompts, instruction prompting, zero-shot, one-shot and few-shot prompting, self-consistency and prompts for creative thinking and effective writing. The chapter helps readers understand prompt engineering as an important form of **human–AI communication**.

Chapter 4 introduces the fundamentals of **Machine Learning**. It covers important AI and ML techniques, machine learning models, regression, classification, clustering, Naïve Bayes, neural networks and Support Vector Machines. The emphasis is on developing a conceptual understanding of how machines learn patterns from data and use those patterns for prediction and decision-making.

AI is not only a technological discipline; it also has significant social, ethical and organizational implications. **Chapter 5**, titled **Trends and Ethical Concerns in AI**, addresses these dimensions. It discusses ethical challenges and risks, AI as a Service, recent AI trends, expert systems, the Internet of Things, and Artificial Intelligence of Things (AIoT). These topics help readers understand how AI is becoming increasingly integrated with cloud computing, connected devices and intelligent systems.

Chapter 6 moves from concepts to **Robotics and Applied AI**. It examines robotics as an application of AI, AI-powered drones, and emerging approaches such as no-code and low-code AI platforms. These technologies demonstrate how AI capabilities can increasingly be developed and deployed without requiring every user to become an expert programmer or machine learning specialist.

The final chapter, **Chapter 7**, explores **AI Across Industries and Applications**. It demonstrates the practical impact of AI in healthcare, finance, retail, agriculture, education and transportation. It also discusses AI in experimentation and multidisciplinary research, showing how AI can support scientific discovery, data analysis, prediction, optimization and collaboration across different disciplines.

A key theme running throughout this book is that **AI should be understood as a tool for augmenting human intelligence rather than merely replacing human effort**. AI can process enormous quantities of data, identify patterns, automate repetitive activities and provide recommendations at remarkable speed. However, human judgement, creativity, ethics, contextual understanding and responsibility remain essential.

The book also recognizes that AI brings challenges alongside opportunities. Issues such as **bias, privacy, security, transparency, explainability, reliability, responsible use and the changing nature of employment** must be considered when developing and applying AI systems. Understanding these concerns is essential for creating AI technologies that are trustworthy, inclusive and beneficial to society.

The organization of the book follows a gradual learning journey:

Understand AI → Understand Machine Intelligence → Communicate with AI → Learn from Data → Understand Ethics and Emerging Technologies → Apply AI → Explore Real-World Impact

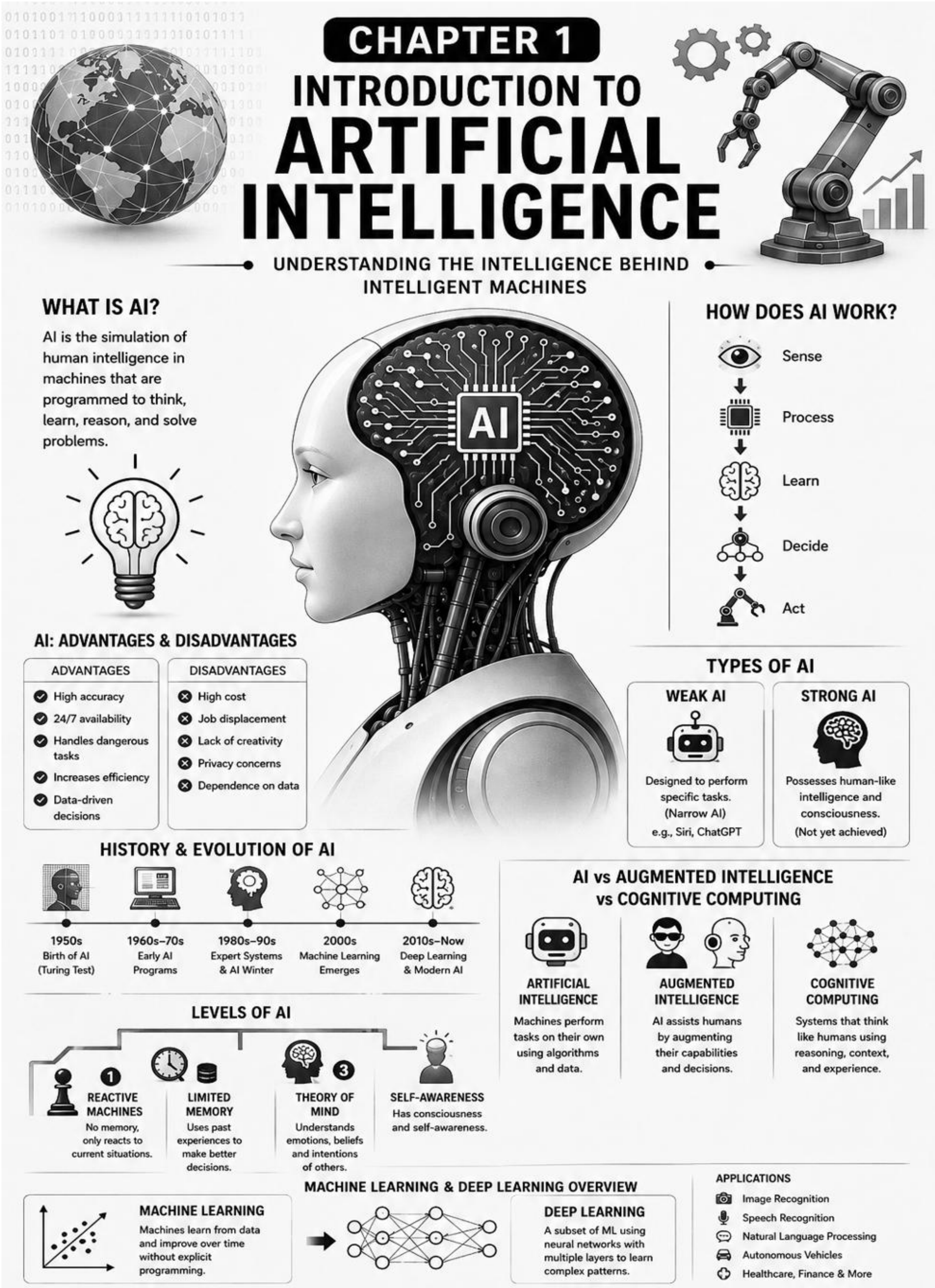
The intention is not to make the reader an expert in every AI technology after reading the book. Rather, the objective is to provide a **strong conceptual foundation, practical awareness and curiosity for further exploration**. Examples, applications and explanations are included to connect theoretical concepts with real-world situations.

As AI continues to evolve rapidly, some tools, platforms and applications will inevitably change. Therefore, this book emphasizes **fundamental concepts and transferable understanding** that can remain useful even as individual AI technologies evolve.

I hope this book serves as a useful learning companion for **students, teachers, researchers, professionals and anyone interested in understanding Artificial Intelligence**. More importantly, I hope it encourages readers not merely to use AI, but to think critically about **how AI works, where it can be applied, what its limitations are, and how it can be used responsibly**.

The future of AI will not be shaped by technology alone. It will be shaped by the people who design it, teach it, research it, deploy it and use it.

AI is not simply about making machines intelligent; it is about using intelligence responsibly to create meaningful solutions for humanity.


CHAPTER 1
INTRODUCTION TO
ARTIFICIAL
INTELLIGENCE
UNDERSTANDING THE INTELLIGENCE BEHIND INTELLIGENT MACHINES
WHAT IS AI?
AI is the simulation of human intelligence in machines that are programmed to think, learn, reason, and solve problems.
AI
HOW DOES AI WORK?
Sense
Process
Learn
Decide
Act
AI: ADVANTAGES & DISADVANTAGES
ADVANTAGES
High accuracy
24/7 availability
Handles dangerous tasks
Increases efficiency
Data-driven decisions
DISADVANTAGES
High cost
Job displacement
Lack of creativity
Privacy concerns
Dependence on data
TYPES OF AI
WEAK AI
Designed to perform specific tasks. (Narrow AI) e.g., Siri, ChatGPT
STRONG AI
Possesses human-like intelligence and consciousness. (Not yet achieved)
HISTORY & EVOLUTION OF AI
1950s Birth of AI (Turing Test)
1960s–70s Early AI Programs
1980s–90s Expert Systems & AI Winter
2000s Machine Learning Emerges
2010s–Now Deep Learning & Modern AI
AI vs AUGMENTED INTELLIGENCE vs COGNITIVE COMPUTING
ARTIFICIAL INTELLIGENCE
Machines perform tasks on their own using algorithms and data.
AUGMENTED INTELLIGENCE
AI assists humans by augmenting their capabilities and decisions.
COGNITIVE COMPUTING
Systems that think like humans using reasoning, context, and experience.
LEVELS OF AI
1 REACTIVE MACHINES
No memory, only reacts to current situations.
LIMITED MEMORY
Uses past experiences to make better decisions.
3 THEORY OF MIND
Understands emotions, beliefs and intentions of others.
SELF-AWARENESS
Has consciousness and self-awareness.
MACHINE LEARNING & DEEP LEARNING OVERVIEW
MACHINE LEARNING
Machines learn from data and improve over time without explicit programming.
DEEP LEARNING
A subset of ML using neural networks with multiple layers to learn complex patterns.
APPLICATIONS
Image Recognition
Speech Recognition
Natural Language Processing
Autonomous Vehicles
Healthcare, Finance & More

CHAPTER 1
INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Building intelligent agents that perceive, learn, reason and act to achieve goals.

1.1 WHAT IS ARTIFICIAL INTELLIGENCE?

- Branch of computer science that creates machines or systems capable of performing tasks that normally require human intelligence.
- Makes intelligent agents that perceive their environment, learn from data, reason, and take actions to achieve goals.
- Unlike traditional software, AI adapts, reasons and solves problems in changing situations.
- The term "Artificial Intelligence" was first coined by **John McCarthy** as the science and engineering of making intelligent machines.
- AI is both:
  - **A Science** *(to study and understand intelligence)*
  - **An Engineering Practice** *(to build machines that can act intelligently)*

1.2 HOW DOES AI WORK?

- Works through an intelligent agent that interacts with an environment in a continuous loop.

1. **Perceive (Sense)** – Collects data from sensors or input sources.
2. **Process** – Interprets and analyses data using algorithms and models.
3. **Learn** – Improves from experience or data using ML/DL algorithms.
4. **Decide** – Chooses the best action to achieve goals.
5. **Act** – Performs action in the physical or digital world.

Feedback from Environment

Environment – The world in which the agent operates and which is affected by its actions.

1.3 ADVANTAGES AND DISADVANTAGES OF AI

Advantages

- High accuracy
- 24/7 availability
- Handles dangerous tasks
- Increases efficiency and productivity
- Data-driven decisions
- Solves complex problems
- Handles large-scale data quickly

Disadvantages

- High cost
- May lead to job displacement
- Lack of creativity
- Lack of emotions and common sense
- Privacy and security concerns
- Dependence on data
- Potential for misuse
- No moral judgment

1.4 HISTORY AND EVOLUTION OF AI

1950s	Birth of AI – Turing Test, early AI ideas.
1960s–70s	Early AI – Symbolic AI, rule-based systems, expert systems.
1980s–90s	AI Winter – Limited funding and progress.
2000s	Machine Learning – Data-driven approaches.
2010s–Now	Deep Learning – Neural networks, big data, cloud computing revolutionize AI.
Future	More human-like AI, automation, autonomous systems.

1.5 TYPES OF AI: WEAK AI vs STRONG AI

Weak AI (Narrow AI)

- Designed to perform a specific task.
- Limited to a narrow domain.
- Examples: Siri, Alexa, ChatGPT, self-driving cars.
- Most AI systems today belong to this category.

Strong AI (General AI)

- Possesses human-like intelligence.
- Can understand, learn, reason and perform any intellectual task.
- Not yet achieved.
- A future goal in AI research.

1.6 LEVELS OF AI

Reactive Machines	Limited Memory	Theory of Mind	Self-Awareness
Only reacts to current input. No memory.	Uses past data to make decisions.	Understands human emotions, beliefs and intentions.	Has self-awareness and consciousness.
Example: IBM Deep Blue (chess)	**Example:** Self-driving cars	**Example:** Not yet achieved	**Example:** Not yet achieved (future goal)

1.7 AI vs AUGMENTED INTELLIGENCE vs COGNITIVE COMPUTING

Artificial Intelligence (AI)

- Machines perform tasks on their own.
- Focuses on automation and problem solving.
- Learns from data and makes decisions.
- Examples: self-driving cars, image recognition.

Augmented Intelligence (AIg)

- AI assists humans in decision-making.
- Enhances human intelligence and supports better decisions.
- Humans remain in control.
- Examples: AI in healthcare assisting doctors, smart tutoring systems.

Cognitive Computing (CC)

- Systems that think like humans.
- Understands natural language, context and complex data.
- Aims to interact naturally like humans.
- Examples: IBM Watson, virtual assistants, legal research systems.

1.8 MACHINE LEARNING AND DEEP LEARNING OVERVIEW

Machine Learning (ML)

- Machines learn from data and improve over time without being explicitly programmed.
- Uses algorithms to find patterns and make predictions.
- Examples: email spam detection, recommendation systems, fraud detection.

Deep Learning (DL)

- A subset of ML using deep neural networks.
- Learns complex patterns from large volumes of unstructured data.
- Requires high computing power (GPUs/TPUs).
- Examples: image recognition, speech recognition, NLP, chatbots, autonomous vehicles.

6A. REAL-WORLD APPLICATIONS OF ML

- Email spam detection – classify emails
- Predictive analytics in finance – predict trends
- Recommendation systems – suggest products
- Market basket analysis – discover item patterns
- Anomaly detection – detect unusual patterns
- House price prediction – estimate property prices
- Credit scoring – evaluate credit risk

6B. REAL-WORLD APPLICATIONS OF DL

- Image recognition – recognize objects
- Image classification – classify images
- Speech recognition – convert speech to text
- Natural language processing – understand and generate human-like text
- Self-driving cars – perception, object detection
- Medical image analysis – analyze X-rays, CT scans, MRIs
- Video understanding – analyze video content
- Chatbots / Generative AI – generate human-like text

7. KEY TAKEAWAYS

- ML enables machines to learn from data and improve without being explicitly programned.
- DL is a powerful subset of ML using deep neural networks to learn complex patterns.
- AI, AIg and CC have different purposes but complement each other.
- Together, they are transforming industries and improving human life.

SUMMARY

AI is about creating intelligent agents that perceive, learn, reason and act. It is transforming the world through automation, data analysis and smart decision-making.

MORE DATA + BETTER ALGORITHMS + MORE COMPUTING POWER = A SMARTER, BRIGHTER TOMORROW

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

1. WHAT IS ARTIFICIAL INTELLIGENCE?

AI is a branch of computer science that creates machines or systems capable of performing tasks that normally require human intelligence.

- Involves making intelligent agents – machines that can look at their surroundings, process information, learn from experience, and take actions to reach a goal.
- Unlike traditional software, AI systems can adapt, reason and solve problems in changing situations.
- The term 'Artificial Intelligence' was first explained by **John McCarthy (1956)** as the science and engineering of making intelligent machines.
- Therefore, AI is both:
  – **A Science** (to study and understand intelligence)
  – **An Engineering Practice** (to build machines that can act intelligently)

HUMAN INTELLIGENCE | **INTELLIGENT AGENTS** | **LEARN** | **REASON & SOLVE PROBLEMS**

2. HOW DOES AI WORK?

AI works through an intelligent agent that interacts with its environment in a continuous loop.

1. Perception (Sense) Observes the environment through sensors or input data and collects raw information.	**Example:** Self-driving car uses cameras, radar and sensors to collect data about the road, traffic and pedestrians.
2. Processing Information Processes collected data to understand the surroundings, identify patterns, and extract relevant signals for decision-making.	**Example:** Traffic light is red, green or yellow. AI detects if a pedestrian is crossing and identifies objects in the scene.
3. Learning Improves decision-making over time by learning from past experiences or large datasets. Uses techniques like Neural Networks for self-improvement.	**Example:** Learns from thousands of driving scenarios to recognize patterns and improve behaviors.
4. Action Takes appropriate actions in the physical or digital world to achieve goals.	**Example:** The car slows down when the light turns red or changes lanes to avoid an obstacle.
Environment The external world that generates new situations and is affected by the agent's actions.	**Example:** Road, traffic, pedestrians, weather and other vehicles.

Feedback from Environment

3. ADVANTAGES AND DISADVANTAGES OF AI

ADVANTAGES

- **High Accuracy:** Can process large amounts of data quickly and reduce human errors.
- **24/7 Availability:** Work continuously without fatigue or breaks.
- **Handles Dangerous Tasks:** Can perform risky or hazardous tasks (e.g., mining, space exploration, disaster response).
- **Increases Efficiency:** Automates repetitive tasks and completes work faster, saving time and resources.
- **Data-driven Decisions:** Analyses data and finds patterns to help in better decision-making.
- **Handles Large-scale Data Quickly:** Can analyze huge datasets at speeds beyond human capability.

DISADVANTAGES

- **High Cost:** Development, training and maintaining AI systems require significant resources and investment.
- **Job Displacement:** Automation may replace certain human jobs, leading to unemployment in some sectors.
- **Lack of Creativity:** AI follows patterns in data but cannot think outside the data to create truly original ideas.
- **No Emotions or Empathy:** AI lacks human feelings, understanding and emotional intelligence.
- **Privacy Concerns:** AI systems collect and use large amounts of personal data, raising security risks.
- **Dependence on Data:** Performance depends on the quality and quantity of data; biased data can lead to biased results.

REAL-LIFE EXAMPLES

- Self-driving cars
- Voice assistants (Siri, Alexa, Google Assistant)
- Recommendation systems (Netflix, YouTube, Amazon)
- Fraud detection in banking
- Medical diagnosis systems
- AI in agriculture, industry and scientific research

WHY AI IS IMPORTANT?

- Helps organize and analyze large amounts of data.
- Enhances decision making and problem solving.
- Enables automation and efficiency.
- Supports innovation and new discoveries.
- Bridges the gap between data and intelligent action.

INTELLIGENT MACHINES → REAL-WORLD SOLUTIONS → A BETTER TOMORROW

HISTORY AND EVOLUTION OF AI

A JOURNEY FROM IDEAS TO INTELLIGENT MACHINES

OVERVIEW

- The history of AI is a journey of ideas, breakthroughs, challenges and continuous progress.
- It has evolved from simple rule-based systems to intelligent machines that can learn, reason and act like humans in many ways.
- AI continues to transform the world and solve complex problems.

1. 1950s: BIRTH OF AI – TURING TEST & EARLY IDEAS

- Alan Turing (1950) proposed the idea of thinking machines.
- Introduced the famous Turing Test.
- Early researchers explored the possibility of creating intelligent machines.
- Laid the foundation for AI research.

2. 1960s–70s: EARLY AI – SYMBOLIC AI, RULE-BASED SYSTEMS, EXPERT SYSTEMS

- AI programmed using logical puzzles and mathematical problem solving.
- Use of symbolic reasoning and rules.
- Development of early expert systems that captured human expertise.
- Examples: MYCIN, DENDRAL.

3. 1980s–90s: AI WINTER – LIMITED FUNDING AND PROGRESS

- High expectations but limited results.
- Lack of powerful hardware and large data.
- AI faced reduced funding and slower research progress.

4. 2000s: MACHINE LEARNING – DATA-DRIVEN APPROACHES

- Shift from rule-based systems to learning from data.
- Development of algorithms that learn automatically.
- Growth of data, better computers and new techniques boosted AI performance.

5. 2010s–NOW: DEEP LEARNING – NEURAL NETWORKS & BIG DATA

- Deep learning using multi-layer neural networks revolutionized AI.
- Big data and powerful hardware (GPUs) led to huge advancements.
- Breakthroughs in image recognition, speech recognition, NLP and more.
- AI is now part of everyday life.

6. TRANSFORMERS, FOUNDATION MODELS & GENERATIVE AI (Mid 2010s–Now)

- Transformers changed the way AI understands language and content.
- Large pre-trained models (e.g., GPT, BERT, DALL-E) trained on massive data.
- Generative AI can create text, images, code, music and more.
- Foundation models enable a wide range of applications.

7. MULTIMODAL AI & LARGE LANGUAGE MODELS (Recent Years)

- AI can understand and work with text, images, audio, video and code together.
- Large Language Models (LLMs) show human-like capabilities.
- Used in chatbots, virtual assistants, education, healthcare, business and more.

8. FUTURE DIRECTIONS OF AI

- AI for **Science**: faster discovery in medicine, materials and climate research.
- **Agentic AI**: autonomous systems that can plan, act and learn.
- **Responsible AI**: focus on fairness, safety, privacy, ethics and trust.
- Quantum computing, neuromorphic computing and Human–AI collaboration to solve global challenges.

IMPACT OF AI EVOLUTION

- **Healthcare** – better diagnosis and personalized treatment
- **Transportation** – autonomous vehicles and smart mobility
- **Industry** – automation, robotics and smart manufacturing
- **Education** – personalized learning and smart classrooms
- **Business & Finance** – fraud detection, trading and analytics

ADVANTAGES

- ✓ Solves complex problems
- ✓ Automates repetitive tasks
- ✓ Increases efficiency and productivity
- ✓ Learns from data and improves over time
- ✓ Enables new innovations and services
- ✓ Works at large scale
- ✓ Transforms industries and improves quality of life

DISADVANTAGES

- ✗ High development and computational cost
- ✗ Requires large amounts of data
- ✗ May lead to job displacement
- ✗ Bias in data can lead to unfair results
- ✗ Lacks common sense and human emotions
- ✗ Raises privacy and security concerns
- ✗ Can be misused (e.g., misinformation, malicious use)

OVERALL GOAL

Create intelligent systems that can perceive, reason, learn and act to improve life and solve real-world problems.

IN SUMMARY

AI has evolved from simple rule-based programs to intelligent systems that learn, reason and create. With continuous advancements in data, algorithms and computing power, AI will play a vital role in building a smarter, safer and more sustainable future.

IDEAS → ALGORITHMS → DATA → INTELLIGENCE → A BETTER TOMORROW

OVERVIEW

- AI systems are designed to perform tasks with varying levels of intelligence.
- AI can be classified into two major types based on their capabilities.
- It can also be understood in terms of levels of development.

CLASSIFICATION OF AI

Based on Capabilities (Type of AI)

- Narrow (Weak) AI
- General (Strong) AI
- Super AI

Based on Development (Type of Machines)

- Reactive Machines
- Limited Memory
- Theory of Mind
- Self-Awareness

TYPES OF AI

From Narrow to Super Intelligence

1. NARROW (WEAK) AI

Designed to perform a specific task

- Designed to perform a specific or narrow range of tasks.
- Also called Weak AI.
- Cannot perform beyond its trained scope.
- Operates based on predefined rules or learned patterns.
- Most AI systems today belong to this category.

Key Points

- Focused on perticular tasks.
- Does not possess self-awareness or understanding.
- Performs well within a limited domain.

Examples

- Voice Assistants (Siri, Alexa, Google Assistant)
- Recommendation Systems (Netflix, YouTube)
- Self-driving Cars (Task: Navigation & Driving)
- Spam Filters in Email
- Fraud Detection in Banking
- Image Recognition Systems

2. GENERAL (STRONG) AI

Possesses human-like intelligence

- Possesses human-like intelligence across a wide range of tasks.
- Also called Strong AI.
- Can understand, learn, reason and apply knowledge in different domains like humans.
- Not yet achieved.
- Would have general reasoning, understanding, learning and adaptation.
- Can transfer knowledge from one domain to another.

Key Points

- Not yet achieved.
- Would have human-level intelligence.
- Can learn, reason, solve novel problems and adapt to new situations.

Examples

- AI that can adapt to any domain environment like humans
- AI virtual assistant with human-like learning ability across all subjects

3. SUPER AI

Intelligence that surpasses humans

- Intelligence that surpasses human intelligence in almost every possible way.
- Able to perform any intellectual task better than humans in every field.
- Purely a future concept, not yet real.
- Could innovate, create, and solve problems beyond human capability.
- May lead to massive positive or negative impacts on humanity.

Key Points

- Purely a future concept.
- Surpasses human intelligence.
- Can solve global challenges (e.g., climate change, disease, energy).
- May create new technologies beyond human imagination.

Examples

- AI that can solve global challenges like climate change, disease and energy
- A virtually conscious and self-improving intelligence

LEVELS OF AI (Based on Development)

1. REACTIVE MACHINES

- Only reacts to current input.
- No ability to learn from past experiences.
- Example: IBM Deep Blue (chess program)

→

2. LIMITED MEMORY

- Uses past experiences to make better decisions.
- Can store limited data for short-term use.
- Example: Self-driving cars, recommendation systems

→

3. THEORY OF MIND

- Understands human emotions, beliefs and intentions.
- Can interact socially like humans.
- Not yet achieved (under research).

→

4. SELF-AWARENESS

- Has self-awareness and consciousness.
- Can make independent decisions.
- Not yet achieved (future possibility).

QUICK COMPARISON

Type / Level	Performs Specific Tasks	Human-like Intelligence	Surpasses Human Ability	Status
Narrow AI	Yes	No	No	Achieved
General AI	Yes	Yes	No	Not yet achieved
Super AI	Yes	Yes	Yes	Not yet achieved (Future concept)

KEY TAKEAWAY

- AI ranges from narrow systems that perform simple tasks to intelligent systems that can perceive, learn, reason, adapt and act like humans.
- The levels of AI show the path of development from simple reaction to self-awareness.
- AI has the potential to transform the world and solve real-world problems.

GOAL OF AI

To build intelligent systems that can perceive, learn, reason, adapt and act to solve real-world problems and improve life.

FROM TODAY'S NARROW AI → TO TOMORROW'S SUPER AI → A SMARTER, SAFER, BETTER WORLD

AI vs AUGMENTED INTELLIGENCE (AIg) vs COGNITIVE COMPUTING (CC)

Three different approaches to intelligence and decision-making

ARTIFICIAL INTELLIGENCE (AI)

Machines perform tasks on their own

OVERVIEW

- AI systems perform tasks that normally require human intelligence.
- They are designed to automate tasks and solve problems using algorithms and data.
- All three aim to solve problems and improve outcomes, but work in different ways and serve different goals.

MAIN FOCUS

Automation of tasks that normally require human intelligence.

ROLE OF HUMAN

Humans design, train and deploy AI systems with minimal human involvement during operation.

KEY CAPABILITIES

- Learn from data
- Recognize patterns
- Make predictions
- Automate decisions and actions
- Improve over time

EXAMPLES

- Self-driving cars
- Voice assistants (Siri, Alexa)
- Recommendation systems (Netflix, YouTube)
- Fraud detection
- Medical image analysis

AUGMENTED INTELLIGENCE (AIg)

AI assists humans in decision-making

OVERVIEW

- AI assists humans in decision-making by augmenting human capabilities.
- It enhances human intelligence and supports better decisions.
- Humans remain in control, using AI as a tool to provide insights, assistance and suggestions for decisions.

MAIN FOCUS

Enhancing human intelligence and supporting better decisions.

ROLE OF HUMAN

Humans remain in control.
AI provides insights, suggestions and assistance for humans to decide.

KEY CAPABILITIES

- Understand data and context
- Provide insights and explanations
- Support decisions in real-time
- Enable human + machine collaboration

EXAMPLES

- AI in healthcare assisting doctors
- Smart tutoring systems
- Intelligent dashboards for business decisions
- Email suggestion tools
- Human-AI collaboration systems

COGNITIVE COMPUTING (CC)

Systems that think like humans

OVERVIEW

- Cognitive Computing systems think like humans.
- They use advanced AI technologies (e.g., NLP, machine learning) to understand complex, unstructured data.
- Designed to learn from context and experience, reason, and interact naturally like humans.

MAIN FOCUS

Understanding, reasoning and learning from complex, unstructured data like humans.

ROLE OF HUMAN

Humans collaborate with the system. The system can understand natural language, learn from questions and interact continuously.

KEY CAPABILITIES

- Natural language understanding
- Context-aware reasoning
- Handles unstructured data (text, speech, images, etc.)
- Learns continuously from interactions
- Explains reasoning and interacts like humans

EXAMPLES

- IBM Watson
- Virtual assistants with natural conversation
- Document understanding systems
- AI for legal research
- Customer support chatbots

KEY DIFFERENCES

Aspect	AI	AIg	CC
Purpose	Automate tasks	Augment human intelligence	Understand and reason like humans
Human Role	Minimal involvement	Humans in control	Human-system collaboration
Data Type	Structured data	Structured + some unstructured	Unstructured data (text, speech, images)
Decision Making	By the system	Supports human decisions	Learns, reasons and interacts
Learning Approach	From data patterns	From data and human feedback	From data, context and experience
Examples	Self-driving cars, fraud detection	AI in healthcare, smart tutoring	IBM Watson, chatbots, document understanding

IN SHORT

- **AI** works for us (automates tasks).
- **AIg** works **with** us (assists and augments).
- **CC thinks with** us (understands, reasons and interacts).

SUMMARY

- **AI** focuses on automation, performing tasks independently.
- Augmented Intelligence focuses on empowering humans by providing intelligent support.
- Cognitive Computing focuses on understanding, reasoning and learning like humans using context and experience.

OVERALL GOAL

To build intelligent systems that can perceive, learn, reason, adapt and act to solve real-world problems and improve life.

INTELLIGENT TECHNOLOGIES + HUMAN INTELLIGENCE = A SMARTER TOMORROW

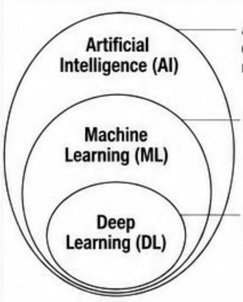
MACHINE LEARNING (ML) & DEEP LEARNING (DL) OVERVIEW

1. UNDERSTANDING MACHINE LEARNING (ML)

- Machine Learning is a subset of Artificial Intelligence.
- Enables systems to learn from data, identify patterns and make decisions without being explicitly programmed.
- Learns from experience (data) and improves over time.
- Uses statistical models and algorithms to learn patterns and relationships.
- Can be applied to various real-world problems.

"Learn from data
Adapt to new situations
Improve over time"

5. RELATIONSHIP: AI → ML → DL



AI is the broad field of creating intelligent machines.

ML is a subset of AI that enables systems to learn from data.

DL is a subset of ML that uses deep neural networks.

DL ⊂ ML ⊂ AI

4. MACHINE LEARNING & DEEP LEARNING OVERVIEW

Aspect	Machine Learning (ML)	Deep Learning (DL)
Definition	Learns patterns from data and makes predictions	Subset of ML using deep neural networks to learn complex patterns
Data Requirement	Works well with small to medium datasets	Needs large amounts of data
Feature Engineering	Requires manual feature engineering	Automatically learns features
Algorithms	Linear Regression, Decision Trees, SVM, KNN, Naive Bayes	Neural Networks, CNN, RNN, Transformers
Computing Power	Can run on normal computers	Requires high computing power (GPUs/TPUs)
Examples	Spam filter, Recommendation system, Credit scoring	Image recognition, Speech recognition, Self-driving cars

2. TYPES OF MACHINE LEARNING

2.1 SUPERVISED LEARNING

- Trains on labeled data (input with correct output).
- Learns a mapping from input to output.
- Used for prediction and classification.

Examples:
- Email spam detection
- Image classification
- House price prediction
- Medical diagnosis

2.2 UNSUPERVISED LEARNING

- Trains on unlabeled data.
- Finds hidden patterns or structures in data.
- Used for clustering and association.

Examples:
- Customer segmentation
- Market basket analysis
- Anomaly detection
- Topic modeling

2.3 REINFORCEMENT LEARNING

- Learns by interacting with an environment.
- Gets rewards for correct actions and penalties for wrong actions.
- Learns a strategy to maximize cumulative reward.

Examples:
- Self-driving cars
- Game playing (AlphaGo)
- Robot control
- Recommendation systems

3. ML WORKFLOW

1. **Collect Data** (Gather and understand data)
2. **Prepare & Clean Data** (Handle missing values, preprocess data)
3. **Model & Train** (Choose algorithm and train model)
4. **Evaluate & Test** (Check performance using metrics)
5. **Deploy & Monitor** (Use in real-world and improve continuously)

Feedback (New data helps the model improve)

6A. REAL-WORLD APPLICATIONS OF ML

- Email spam detection – classify emails
- Predictive analytics in finance – predict trends
- Recommendation systems – suggest products
- Market basket analysis – discover item patterns
- Anomaly detection – detect unusual patterns
- House price prediction – estimate property prices
- Medical diagnosis – predict or classify diseases
- Demand forecasting – predict future demand
- Credit scoring – evaluate credit risk

6B. REAL-WORLD APPLICATIONS OF DL

- Image recognition – recognize objects
- Image classification – classify images
- Speech recognition – convert speech to text
- Natural language understanding (NLP) – understand and generate text
- Self-driving cars – perception, object detection
- Medical image analysis – analyze X-rays, CT scans, MRIs
- Face recognition – security and authentication
- Video understanding – analyze video content
- Chatbots / Generative AI – generate human-like text
- Complex pattern recognition – learn hierarchical patterns in large unstructured data

7. KEY TAKEAWAYS

- ML enables machines to learn from data and improve without being explicitly programmed.
- DL is a powerful subset of ML that uses deep neural networks to automatically learn complex patterns.
- There are three main types of ML: Supervised, Unsupervised and Reinforcement Learning.
- Together, ML and DL are transforming industries and improving human life.

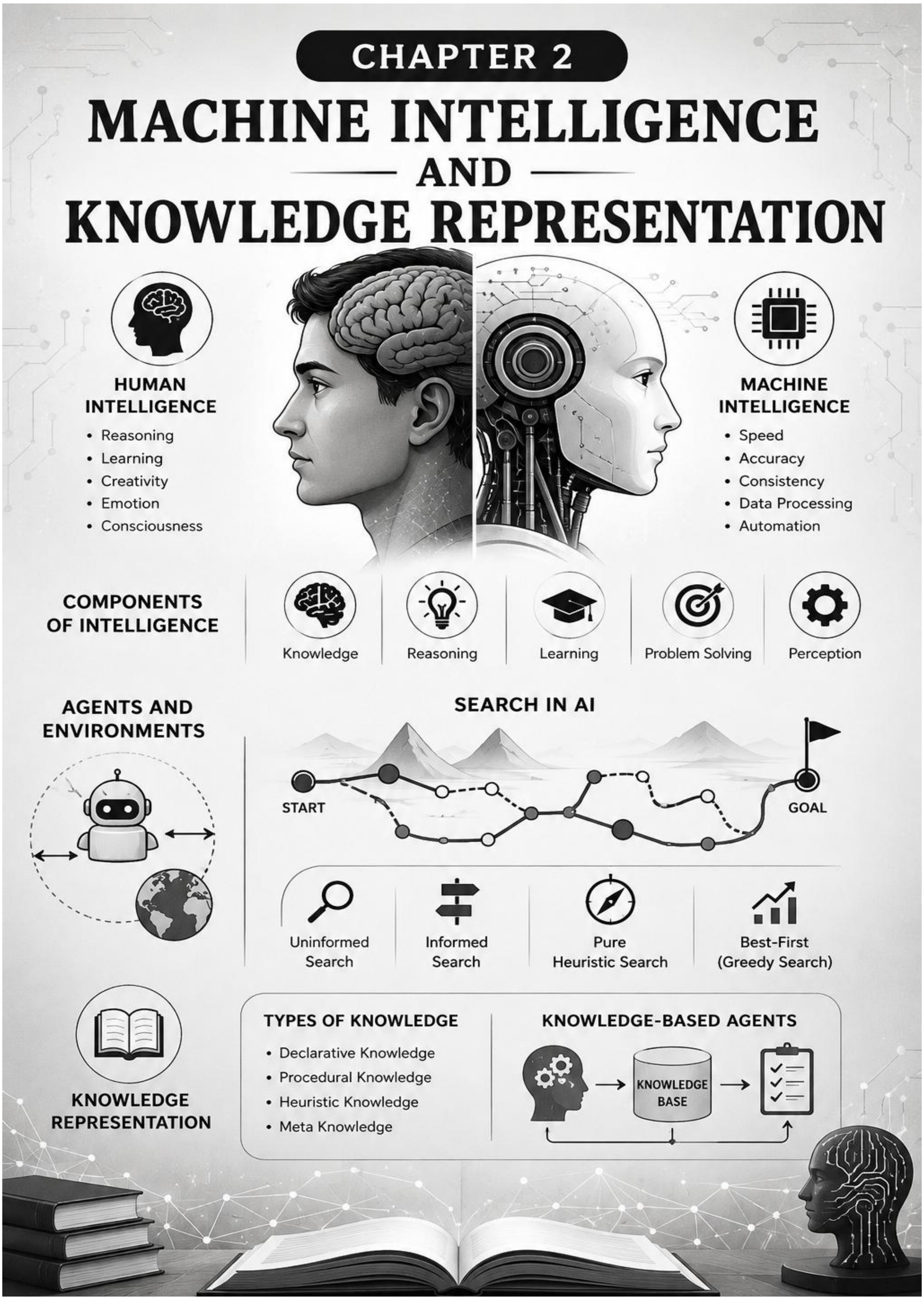
IN SUMMARY

- More data + better algorithms + more computing power = better AI systems.
- AI works for us, Augments us, CC thinks with us.

TOGETHER FOR A BETTER TOMORROW

Together, ML and DL are transforming industries and improving human life.

DATA → LEARNING → PREDICTIONS → INTELLIGENT SOLUTIONS → A BRIGHTER FUTURE


CHAPTER 2
MACHINE INTELLIGENCE
AND
KNOWLEDGE REPRESENTATION
HUMAN INTELLIGENCE
• Reasoning
• Learning
• Creativity
• Emotion
• Consciousness
MACHINE INTELLIGENCE
• Speed
• Accuracy
• Consistency
• Data Processing
• Automation
COMPONENTS OF INTELLIGENCE
Knowledge
Reasoning
Learning
Problem Solving
Perception
AGENTS AND ENVIRONMENTS
SEARCH IN AI
START
GOAL
Uninformed Search
Informed Search
Pure Heuristic Search
Best-First (Greedy Search)
KNOWLEDGE REPRESENTATION
TYPES OF KNOWLEDGE
• Declarative Knowledge
• Procedural Knowledge
• Heuristic Knowledge
• Meta Knowledge
KNOWLEDGE-BASED AGENTS
KNOWLEDGE BASE

CHAPTER 2 MACHINE INTELLIGENCE AND KNOWLEDGE REPRESENTATION

From Real-World Information to Intelligent Action

Information → Knowledge
Knowledge Base → Representation
Reasoning / Inference → Decision
Action

2.1 DEFINING INTELLIGENCE: HUMAN vs MACHINE

Human Intelligence	Machine Intelligence
• Uses experience, intuition and emotions. • Learns from environment. • Handles ambiguity. • Creative and adaptive. • Works in unstructured environments.	• Processes information using algorithms. • Fast, accurate and consistent. • Learns from data. • Scope for continuous improvement. • Works in structured and unstructured environments.

Real-World Examples

AI in Healthcare	Autonomous Vehicles	AI in Education
• Analyses medical images • Supports diagnosis • Assists treatment decisions	• Sensors perceive environment • Real-time decision making • Safe navigation and adaptability	• Personalized learning • Intelligent tutoring • Adaptive content delivery

2.2 COMPONENTS OF INTELLIGENCE

Perception	• Senses and interprets the environment • E.g., vision, speech, sensors
Learning	• Improves from experience/data • E.g., supervised, unsupervised, reinforcement learning
Reasoning & Problem Solving	• Analyses information to solve problems • E.g., search, optimization, planning
Knowledge Representation	• Organizes and stores knowledge • E.g., facts, rules, relationships
Decision-Making	• Chooses the best action based on goals • E.g., utility, probability, reasoning
Adaptability	• Adjusts to new situations • E.g., online learning, transfer learning
Creativity / Innovation	• Generates new ideas or solutions • E.g., generative AI, design, creativity

2.4 SEARCH IN AI

A. Uninformed (Blind) Search
Searches without domain knowledge.

1. Breadth-First Search (BFS)
2. Depth-First Search (DFS)
3. Uniform Cost Search (UCS)
4. Depth-Limited Search (DLS)
5. Iterative Deepening DFS (IDDFS)
6. Bidirectional Search

B. Informed (Heuristic) Search
Uses heuristic function $h(n)$.

- **Heuristic Function:** $h(n)$ – estimate of cost from n to goal.
- **Evaluation Functions:**
  - $g(n)$ – actual cost from start to n
  - $h(n)$ – heuristic cost to goal
  - $f(n) = g(n) + h(n)$ (A* Search)
- **Key Searches:** Greedy Best-First Search, A* Search
- **Goal:** Find optimal path/solution with minimum cost and effort.

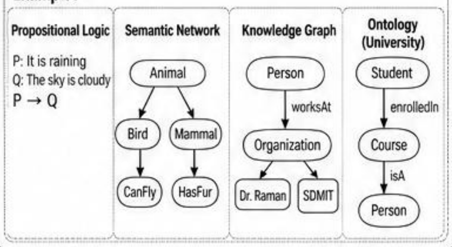
2.5 KNOWLEDGE REPRESENTATION (KR)

Purpose: Represent knowledge in a formal way so that machines can store, organize, reason with, and use it to solve problems and make decisions.

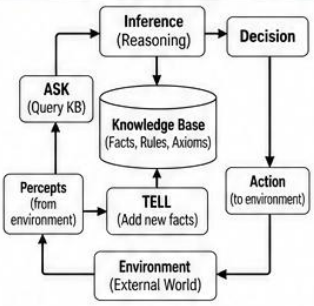
A. Knowledge Representation Structures

1. **Propositional Logic** – facts using logical symbols (P, Q, ¬, ∧, ∨)
2. **Semantic Networks** – graph-based representation
3. **Knowledge Graphs** – entities and relations in a graph
4. **Ontologies** – formal specification of a domain

Examples



C. KNOWLEDGE-BASED AGENTS



- **KB** stores facts, rules, axioms, relations.
- **TELL:** add new information to KB.
- **ASK:** query KB to retrieve/infer knowledge.
- **Inference:** derive new knowledge using rules.
- **Decision:** select best action based on goals/utility.
- **Action:** execute action in the environment.
- Cycle repeats with new percepts.

3. COMPARISON OF SEARCH METHODS

Feature	Uninformed Search	Informed Search (A*)
Uses domain knowledge	No	Yes
Exploration	Systematic	Heuristic guided
Evaluation function	–	$f(n) = g(n) + h(n)$
Finds optimal solution	Yes (if conditions)	Yes (if h is admissible)
Time complexity	High	Lower (more efficient)
Space complexity	High	Lower
Examples	BFS, DFS, UCS, DLS, IDDFS	Greedy Best-First, A* Search

4. BENEFITS OF KR

- ✓ Enables reasoning and decision-making.
- ✓ Promotes knowledge reuse and sharing.
- ✓ Improves system performance and explainability.
- ✓ Helps in building intelligent and adaptive systems.
- ✓ Bridges the gap between human knowledge and machine understanding.

⚠ CHALLENGES

- Knowledge acquisition is hard.
- Time-consuming and costly.
- Handling incomplete, uncertain, and noisy knowledge.
- Scalability with very large KBs.
- Ensuring knowledge is consistent and up-to-date.

6. EXAMPLE WALKTHROUGH

Scenario: Smart Irrigation Robot

Percept: "Soil moisture is very low."

KB before TELL:

1. Low soil moisture → Plants may require water.
2. Plants may require water → Start irrigation.
3. If soil is wet → Stop irrigation.

Steps:

1. **TELL:** Add "Soil moisture is very low."
2. **Inference:** Plants may require water.
3. **Decision:** Start irrigation.
4. **Action:** Execute irrigation.

7. KEY TAKEAWAYS

- Knowledge-based agents rely on explicit knowledge and reasoning.
- Organize, store and retrieve knowledge effectively.
- Enable intelligent decision-making and problem solving.
- Support real-world applications (e.g., robotics, healthcare, education).

8. COMMON NOTATIONS

P, Q, R	: Propositions (atomic sentences)
¬	: NOT
∧	: AND
∨	: OR
→	: Implies (if ... then)
KB	: Knowledge Base
⊨ (or |=)	: Entailment (KB entails α)

OVERALL GOAL

To build intelligent agents that perceive, reason, learn, and take effective actions to achieve goals in real-world environments.

KNOWLEDGE + REASONING + INTELLIGENT ACTION → SMARTER AI → BETTER REAL-WORLD SOLUTIONS

HUMAN INTELLIGENCE

Human intelligence is the natural, biological ability of humans to think, learn, reason, adapt, and deal with complex and uncertain situations using the brain's neural structure.

NATURE	• Rooted in the brain's complex network of neurons (≈ 86 billion neurons). • Influenced by physical and emotional state of the body and environment.
STRENGTHS	• **Creativity:** Generates novel ideas and solutions. • **Emotions:** Help in decision-making, social interactions and building relationships. • **Moral Reasoning:** Makes decisions influenced by values, culture and personal beliefs. • **Adaptability:** Adjusts effectively in diverse environments, thinks critically and shifts strategies in uncertainty.
LEARNING	• Dynamic learning using experience, intuition and social interaction. Forms mental models and applies reasoning to new situations.
LIMITATIONS	• Performance drops with tiredness or overload. • **Biases:** Cognitive biases affect judgment. • **Limited Memory:** Information is forgotten. • **Inconsistent Reasoning:** Emotions or beliefs may lead to inconsistent decisions.
EXAMPLE	• A doctor diagnosing a rare disease relies on intuition, experience and contextual reasoning. They notice subtle symptoms not fully matched in textbooks.

MACHINE INTELLIGENCE (ARTIFICIAL INTELLIGENCE)

Machine intelligence (AI) is the ability of a machine or computer system to perceive, learn, reason, solve problems, make decisions, and adapt, performing tasks like learning, problem-solving, reasoning, language understanding, and perception.

NATURE	• Works on algorithms, data and computing power. • Not biological; emerges from software and hardware. • Can be narrow (task-specific) or show generalization through advanced learning algorithms.
STRENGTHS	• **Speed:** Processes vast amounts of data at incredible speeds. • **Accuracy:** Performs tasks with high accuracy when properly trained. • **Consistency:** Does not get tired; performs tasks with the same efficiency repeatedly. • **Automation of Repetitive Task:** Increases efficiency and reduces human error in routine processes.
LEARNING	• Learns patterns from data. • Uses Machine Learning (ML), Deep Learning (DL). • Learns from large unstructured data using neural networks. • Follows predefined rules for decision-making.
LIMITATIONS	• **Lack of Creativity:** Cannot truly create original ideas. • **No Emotions:** Lacks feelings, empathy and emotional understanding. • **Dependency on Data:** Relies on quality data; biased or poor data leads to flawed results.
EXAMPLE	• An AI system analyzes millions of medical images to detect lung cancer. It quickly highlights suspicious areas, while the doctor interprets the results and makes the final decision.

DEFINING INTELLIGENCE: HUMAN vs MACHINE

"Intelligence is the ability to perceive, learn, reason, adapt, and act purposefully in an environment to achieve goals effectively."

KEY DIFFERENCES AT A GLANCE

ASPECT	HUMAN INTELLIGENCE	MACHINE INTELLIGENCE
Nature	Biological	Artificial (algorithmic)
Source	Brain & experience	Data & computing power
Strengths	Creativity, emotions, intuition, adaptability	Speed, accuracy, consistency, efficiency
Learning	From experience, intuition & reasoning	From data (ML, DL, rules)
Limitations	Tiredness, biases, limited memory	No emotions, data-dependent
Best Suited For	Complex, ambiguous, value-driven decisions	Data-intensive, repetitive, high-speed tasks

HOW THEY WORK TOGETHER

HUMAN INTELLIGENCE
Provides goals, values, judgment and creative insights.

MACHINE INTELLIGENCE
Provides speed, accuracy and data-driven outcomes.

Together, they create better decisions and outcomes.

IN ESSENCE

Machine intelligence is powerful, precise, and tireless in specific tasks, but it thrives when paired with human intelligence to solve problems at a scale more meaningful, ethical, and impactful than either could alone.

"AI is not about replacing humans, but about augmenting human potential and enabling us to solve problems at a scale and speed previously unimaginable."

- ✓ Use AI for efficiency.
- ✓ Use human intelligence for wisdom.
- ✓ Together for a better future.

HUMAN + AI → GREATER POSSIBILITIES → A BETTER TOMORROW

COMPONENTS OF INTELLIGENCE

Intelligence is not a single ability but a combination of multiple interconnected components working together.

5. DECISION-MAKING

The ability to evaluate alternatives and choose the best action based on goals, constraints or utility.

Example: Choosing the safest route in navigation considering distance, traffic and time.

6. ADAPTABILITY

The ability to adjust to new situations, changes, or unexpected conditions to perform effectively.

Example: An AI system adapting to new data, user behaviour or environmental conditions.

7. CREATIVITY / INNOVATION

The ability to generate new ideas, methods or solutions that are novel and useful.

Example: Designing a new product, discovering patterns, or creating original content.

4. KNOWLEDGE REPRESENTATION

The ability to organize, store and represent information, facts, rules and relationships in a structured way.

Example: Storing medical knowledge as rules (if–then), ontologies, or semantic networks.

These components enable an entity (human or machine) to understand its environment, learn from experience, make decisions and achieve goals effectively.

1. PERCEPTION

The ability to sense and interpret information from the environment using sensors or input data.

Example: Detecting traffic lights, recognizing voices, or reading text from an image.

3. REASONING & PROBLEM SOLVING

The ability to draw conclusions, infer relationships and solve problems to reach a solution.

Example: Diagnosing a disease by analyzing symptoms and test results.

2. LEARNING

The ability to improve performance over time by learning from data, experience or feedback.

Example: A self-driving car learns from thousands of driving experiences to handle new situations better.

HOW THESE COMPONENTS WORK TOGETHER

Perception	Learning	Reasoning & Problem Solving	Knowledge Representation	Decision-Making	Adaptability	Creativity / Innovation
Senses the environment	Learns from data/experience	Understands and analyzes information	Stores and organizes information	Chooses the best action	Adjusts to changes	Generates new ideas or solutions

Continuous feedback improves performance over time.

KEY TAKEAWAYS

- Intelligence is a combination of multiple components, not a single ability.
- These components function in a cycle and support each other.
- Strong intelligence systems balance all components effectively.
- Human intelligence excels in creativity, emotions and adaptability.
- Machine intelligence excels in speed, accuracy, consistency and data handling.
- The goal is to design systems (or humans) that integrate all components to achieve intelligent and goal-directed behaviour.
- Continuous learning and feedback lead to better performance over time.

INTELLIGENCE IN ACTION → A SMARTER TOMORROW

AGENTS AND ENVIRONMENTS IN AI

In AI, an agent is an entity that perceives its environment through sensors and acts upon it through actuators to achieve goals.

1. WHAT IS AN AGENT?

An agent is an entity that perceives its environment through sensors and acts upon it through actuators to achieve specific objectives.

Examples: Robot, Self-driving car, Software program, Game player, Industrial automation system.

2. AGENT–ENVIRONMENT INTERACTION

AGENT (Perceives & Acts) ← Actions (Actuators) — Percepts (Sensors) → **ENVIRONMENT** (Everything outside the agent)

- The agent senses (perceives) the state of the environment.
- Based on percepts and its program, it decides an action.
- The action affects the environment.
- This cycle continues to achieve the agent's objectives.

3. COMPONENTS OF AN AGENT

Sensors (Perceive the environment)	Gather information from the environment.
Agent Program (Decision-Making / Reasoning)	Decides the next action based on percepts and goals.
Actuators (Act on the environment)	Execute actions in the environment.
Goals / Performance Measure (What the agent wants to achieve)	Defines the success criteria for the agent.

4. TYPES OF AGENTS

Simple Reflex Agent	Acts only on current percept using condition-action rules. **Example:** Thermostat
Model-Based Reflex Agent	Maintains an internal state (model) to handle partially observable environments. **Example:** Robot vacuum cleaner
Goal-Based Agent	Chooses actions to achieve specific goals. **Example:** GPS navigation system
Utility-Based Agent	Chooses actions that maximize expected utility or satisfaction. **Example:** Portfolio management system
Learning Agent	Improves performance over time by learning from experience. **Example:** Email spam filter

5. EXAMPLES OF AGENTS AND ENVIRONMENTS

Agent	Environment	Goal
Self-driving Car	Roads, traffic, pedestrians, signs, weather	Reach destination safely, follow traffic rules
Industrial Robot	Factory floor, objects, assembly parts	Assemble parts accurately and quickly
Chess Program	Chess board, opponent	Win the game
Drone	Airspace, obstacles, weather	Surveillance / Delivery / Reach target location
Chatbot	User queries, conversation context	Provide helpful and relevant responses

6. TYPES OF ENVIRONMENTS

Environment Type	Description	Examples
Fully Observable	Agent has complete information about the state.	Chess, Board games
Partially Observable	Agent has incomplete information.	Self-driving car, Medical diagnosis
Deterministic	Next state is completely determined by current state and action.	Math problems, Crossword puzzle
Stochastic (Non-deterministic)	Next state is uncertain.	Weather prediction, Traffic
Static	Environment does not change while agent decides.	Crossword puzzle
Dynamic	Environment can change while agent decides.	Self-driving car, Stock market
Discrete	Finite or countable states, actions and time.	Robot navigation, Control
Continuous	Infinite or continuous states, actions or time.	Maze solving, Traffic system
Single-agent	Only one agent in the environment.	Game, Traffic control
Multi-agent	Multiple agents interact in the environment.	Soccer game, Traffic system

7. KEY TAKEAWAYS

- An agent perceives its environment through sensors and acts through actuators.
- The agent's behaviour depends on its architecture and goals.
- Different types of agents are suitable for different tasks.
- The environment can be fully or partially observable, deterministic or stochastic, static or dynamic, single-agent or multi-agent.
- The study of agents and environments forms the foundation of intelligent systems.

8. THE AGENT–ENVIRONMENT CYCLE

1. Perceive Environment (Sensors) → 2. Process / Reason (Agent Program) → 3. Decide Action (According to Goals) → 4. Act on Environment (Actuators) → 5. Environment Changes (New State)

Repeat continuously to achieve goals.

INTELLIGENT AGENTS FOR A SMARTER WORLD

1. UNINFORMED SEARCH ALGORITHMS

Explore the search space without using any domain knowledge or heuristic information.

Characteristics:
- Use only problem definition (initial state, actions, goal test).
- Blind search – do not have any idea about distance to goal.
- May explore many nodes.
- Useful for small problems with uniform step cost.

Algorithm	Strategy	Data Structure Used	Complete	Optimal	Time Complexity	Space Complexity
Breadth-First Search (BFS)	Expand shallowest node first	Queue (FIFO)	Yes	Yes (if unit cost)	$O(b^d)$	$O(b^d)$
Depth-First Search (DFS)	Expand deepest node first	Stack (LIFO)	No	No	$O(b^m)$	$O(b^m)$
Depth-Limited Search (DLS)	DFS with fixed depth limit (l)	Stack (LIFO)	Yes (if goal within l)	No	$O(b^l)$	$O(b^l)$
Iterative Deepening Depth-First Search (IDDFS)	Repeated DLS with increasing limit	Stack (LIFO)	Yes	Yes (if unit cost)	$O(b^d)$	$O(b^d)$
Uniform Cost Search (UCS)	Expand node with lowest path cost g(n)	Priority Queue	Yes	Yes	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$

b = branching factor, d = depth of shallowest goal, m = maximum depth, C* = optimal cost, ε = minimum step cost

2. INFORMED SEARCH ALGORITHMS

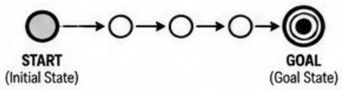
Use additional knowledge (heuristic function) to guide the search towards the goal.

Characteristics:
- Use heuristic function h(n) to estimate the cost from node n to the goal.
- Focus the search on more promising nodes.
- Generally more efficient than uninformed search.
- Heuristic must be admissible (never overestimates the true cost) for optimality.

Algorithm	Evaluation Function	Description	Complete	Optimal	Time Complexity	Space Complexity
A* Search	f(n) = g(n) + h(n)	Combines actual cost g(n) and heuristic h(n)	Yes (if h is admissible)	Yes (if h is admissible)	$O(b^d)$	$O(b^d)$
IDA* Search	f(n) = g(n) + h(n)	Uses iterative deepening A* with memory efficient search	Yes (if h is admissible)	Yes (if h is admissible)	$O(b^d)$	$O(b^d)$
RBFS (Recursive Best-First Search)	f(n) = g(n) + h(n)	Memory-bounded A* using recursion	Yes (if h is admissible)	Yes (if h is admissible)	$O(b^d)$	$O(b^d)$

Heuristic Function h(n) must be admissible (never overestimates the true cost) for optimality.

SEARCH IN AI



Search is the process of finding a sequence of actions (**a path**) from an **initial state** to a goal **state** in a state space.

A search algorithm explores the state space systematically to find a solution.

3. PURE HEURISTIC SEARCH

Uses only the heuristic function h(n) to decide the next node to expand. Ignores the actual path cost from the start.

Characteristics:
- Expands the node that appears closest to the goal (according to h).
- Does not consider the actual path cost.
- Fast but may not find the optimal path.

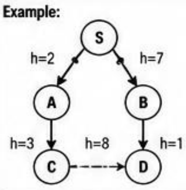
Algorithm	Evaluation Function	Complete	Optimal	Time Complexity	Space Complexity
Greedy Best-First Search	f(n) = h(n)	No	No	$O(b^m)$	$O(b^m)$

4. BEST-FIRST SEARCH (GREEDY SEARCH)

A specific type of informed search that always expands the node that looks best according to a heuristic function.

Characteristics:
- Similar to Pure Heuristic Search (Greedy Best-First Search).
- Does not consider the actual path cost.
- Fast but not optimal.

Example:



How it works:
1. Start from initial state (S) using h(n).
2. Evaluate all successors (A: h=2, B: h=7).
3. Choose the node with lowest h(n) (A).
4. Repeat until goal is reached.

5. EXAMPLES OF AGENTS AND ENVIRONMENTS

Agent	Environment	Goal	Example
Self-driving Car	Roads, traffic, pedestrians, signs, weather	Reach destination safely, follow traffic rules	Autonomous vehicle navigation
Industrial Robot	Factory floor, objects, assembly parts	Assemble parts accurately and quickly	Robotic arm in manufacturing
Chess Program	Chess board, opponent	Win the game	Chess AI (e.g., Stockfish)
Drone	Airspace, obstacles, weather	Surveillance / Delivery / Reach target location	Delivery drone
Chatbot	User queries, conversation context	Provide helpful and relevant responses	Virtual assistant (e.g., ChatGPT)

6. TYPES OF ENVIRONMENTS

Environment Type	Description	Examples
Fully Observable	Agent has complete information about the state.	Chess, Board games
Partially Observable	Agent has incomplete information.	Self-driving car, Medical diagnosis
Deterministic	Next state is completely determined by current state and action.	Math problems, Crossword puzzle
Stochastic (Non-deterministic)	Next state is uncertain.	Weather prediction, Traffic
Static	Environment does not change while agent decides.	Crossword puzzle
Dynamic	Environment can change while agent decides.	Self-driving car, Stock market
Discrete	Finite or countable states, actions and time.	Robot navigation, Control
Continuous	Infinite or continuous states, actions or time.	Maze solving, Traffic system
Single-agent	Only one agent in the environment.	Game, Traffic control
Multi-agent	Multiple agents interact in the environment.	Soccer game, Traffic system

7. KEY TAKEAWAYS

- An agent perceives its environment through sensors and acts through actuators.
- The agent's behaviour depends on its architecture and goals.
- Different types of agents are suitable for different tasks.
- The environment can be fully or partially observable, deterministic or stochastic, static or dynamic, single-agent or multi-agent.
- The study of agents and environments forms the foundation of intelligent systems.

8. THE AGENT–ENVIRONMENT CYCLE

1. Perceive Environment (Sensors) → **2. Process / Reason** (Agent Program) → **3. Decide Action** (According to Goals) → **4. Act on Environment** (Actuators) → **5. Environment Changes** (New State)

Repeat continuously to achieve goals.

SEARCH, THINK, SOLVE → TOWARDS INTELLIGENT SOLUTIONS

UNINFORMED SEARCH ALGORITHMS (BLIND SEARCH)

Uninformed search algorithms explore the state space **without** using any domain knowledge or heuristic information. They rely only on the problem definition (initial state, actions, goal test, path cost) to find a solution.

1. WHAT IS UNINFORMED SEARCH?

Techniques that find a solution by exploring the state space systematically without any idea about the distance to the goal.

Characteristics:
- Use only the problem definition.
- Explore the state space.
- Blind search – no knowledge of goal location.
- May explore many nodes.
- Guarantee completeness/optimality for some algorithms.

TERMS

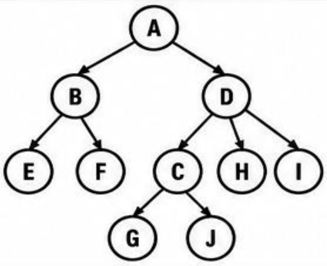
b = branching factor (maximum number of successors of any node)
d = depth of shallowest goal node
m = maximum depth of the search tree
C* = optimal cost (minimum path cost of any solution)
C = cost of a solution
O() = Asymptotic time or space complexity

2. COMPARISON OF UNINFORMED SEARCH ALGORITHMS

Algorithm	Strategy / Idea	Data Structure Used	Complete?	Optimal?	Time Complexity	Space Complexity
1. Breadth-First Search (BFS)	Explores nodes level by level (all nodes at depth *d* before d+1).	Queue (FIFO)	Yes (if *b* is finite)	Yes (if unit step cost)	$O(b^d)$	$O(b^d)$
2. Depth-First Search (DFS)	Explores as deep as possible along a branch before backtracking.	Stack (LIFO)	No (may get stuck in infinite paths)	No	$O(b^m)$	$O(b^m)$
3. Depth-Limited Search (DLS)	DFS with a fixed depth limit L.	Stack (LIFO)	Yes (if $L \geq d$)	No	$O(b^L)$	$O(b^L)$
4. Iterative Deepening Depth-First Search (IDDFS)	Performs DLS for limits 0, 1, 2, ... until the goal is found.	Stack (LIFO)	Yes	Yes (if unit step cost)	$O(b^d)$	$O(b^d)$
5. Uniform Cost Search (UCS)	Expands the node with the least path cost $g(n)$.	Priority Queue	Yes	Yes (positive step cost)	$O(b^{1+C^*/\varepsilon})$	$O(b^{1+C^*/\varepsilon})$

b = branching factor, d = depth of shallowest goal, m = maximum depth, L = depth limit, C* = optimal cost, ε = minimum step cost

3. EXAMPLE STATE SPACE TREE



BFS order (level order):
A, B, D, E, F, C, G, H, I, J

DFS order (left to right):
A, B, E, F, C, G, H, D, I, J (stops when goal is found)

DLS (L = 2):
A, B, E, F, C, G, H, D, I, J (nodes only up to depth 2)

IDDFS (0 → 1 → 2):
0: A
1: A, B, D
2: A, B, E, F, D, C, H, I, J

UCS (increasing path cost):
Expands nodes in order of increasing path cost.

4. WHEN TO USE?

- **BFS:** Shallowest solution, shortest path (unit cost), complete.
  Use when the solution is likely to be near the root.
- **DFS:** Uses very little memory, may be deep (infinite spaces).
  Use when memory is limited but known depth is acceptable.
- **DLS:** Useful when a depth limit is known. Prevents infinite search.
- **IDDFS:** Complete and optimal (unit cost). Combines the advantages of BFS and DFS.
- **UCS:** Use when path costs vary. Finds the least cost solution.

5. KEY TAKEAWAYS

- Uninformed search does not use heuristic information.
- **BFS** is complete and optimal (unit cost).
- **DFS** uses very little memory but may not find a solution.
- **DLS** is complete only if the limit is sufficient.
- **IDDFS** is complete and optimal (unit cost).
- **UCS** finds the least cost solution when step costs are positive.
- Choice of algorithm depends on the problem requirements (completeness, optimality, time and space).

6. SUCCESS MEASURE

A solution is found when a goal state is reached.
Quality depends on:
- Optimality (lowest cost).
- Completeness (finds a solution if one exists).
- Time and memory used.

GOAL OF SEARCH

Find a sequence of actions (path) from the initial state to a goal state that satisfies the goal test.

EXPLORE → SEARCH → REACH THE GOAL → SOLVE PROBLEMS

INFORMED SEARCH ALGORITHMS

Use domain knowledge (heuristics) to guide the search towards the goal.

- ✓ Smarter Search
- ✓ Better Decisions
- ✓ Efficient Solutions

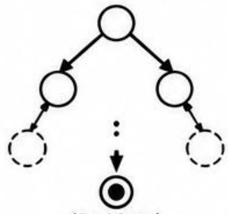
1. WHAT IS INFORMED SEARCH?

Uses heuristic information h(n) to estimate the cost from a node to the goal. Explores the most promising nodes first, making search more efficient than uninformed search.

Characteristics:

- Uses heuristic function h(n) to decide which node to expand next.
- Explores toward the goal.
- More efficient and faster than uninformed search.
- Generally achieves better performance.
- The quality of heuristic determines the performance.

START
(Initial State)



(Goal State)

Informed search algorithms use domain knowledge in the form of a heuristic function to guide the search towards the goal, making it more efficient than uninformed (blind) search.

KEY CONCEPTS

- **g(n)** : actual cost from start node to node n (path cost so far)
- **h(n)** : heuristic estimate of cost from node n to goal
- **f(n)** : evaluation function used to select the next node
- **Good heuristic:** admissible and consistent

TERMS

b	=	branching factor (maximum number of successors)
d	=	depth of the shallowest goal node
m	=	maximum depth of the search tree
C*	=	optimal cost (minimum path cost of any solution)
h(n)	=	heuristic cost estimate from node n to goal
f(n)	=	evaluation function
O()	=	asymptotic time or space complexity

2. A* SEARCH (A-STAR)

- Combines actual cost g(n) and heuristic cost h(n).
- Evaluation function: f(n) = g(n) + h(n)
- Expands the node with the lowest f(n).
- Complete and optimal with admissible and consistent heuristic.

Aspect	Details
Evaluation Function	f(n) = g(n) + h(n)
Idea	Balancing path cost so far (g) and estimated cost to goal (h)
Complete?	Yes (if h is admissible)
Optimal?	Yes (if h is admissible)
Time Complexity	Exponential (but less than memory expansion)
Space Complexity	Exponential ($O(b^{d})$)
Best Used When	Optimal solution is required

3. IDA* SEARCH (ITERATIVE DEEPENING A*)

- Memory-efficient version of A*.
- Performs iterative deepening on f(n).
- Expands nodes with f(n) ≤ threshold.
- Complete and optimal with admissible heuristic, but slower than A*.

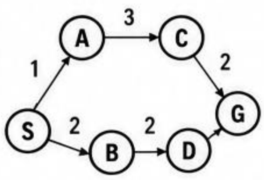
Aspect	Details
Evaluation Function	f(n) = g(n) + h(n)
Idea	Depth-first search with increasing f-limit (iterative deepening)
Complete?	Yes (if h is admissible)
Optimal?	Yes (if h is admissible)
Time Complexity	Exponential
Space Complexity	Linear (O(d))
Best Used When	Memory is limited but optimal solution is required

4. RBFS (RECURSIVE BEST-FIRST SEARCH)

- Best-first search with limited memory using backtracking.
- Expands the most promising node (lowest f).
- Keeps only the path from root to current node.
- Complete and optimal with admissible heuristic.

Aspect	Details
Evaluation Function	f(n) = g(n) + h(n)
Idea	Best-first search with memory limit (recursive backtracking)
Complete?	Yes (if h is admissible)
Optimal?	Yes (if h is admissible)
Time Complexity	Exponential (like A*)
Space Complexity	Linear (O(d))
Best Used When	Memory is limited, needs optimal solution

5. EXAMPLE: SIMPLE STATE SPACE



Heuristic Values h(n)	
Node	**h(n)**
S	4
A	3
B	2
C	2
D	1
G	0

(G is the goal)

A* Search (f = g + h)

Expansion order:
S → B → D → G

Optimal path cost:
2 + 2 + 1 = 5

COMPARISON OF INFORMED SEARCH ALGORTIHMS

Property	A* Search	IDA* Search	RBFS
Uses g(n)?	Yes	Yes	Yes
Uses h(n)?	Yes	Yes	Yes
Complete (with admissible h)	Yes	Yes	Yes
Optimal (with admissible h)	Yes	Yes	Yes
Memory Usage	High	Low (O(d))	Low (O(d))
Typical Use	When memory is sufficient	When memory is very limited	When memory is limited
Time Complexity	$O(b^{d})$	$O(b^{d})$	$O(b^{d})$
Space Complexity	$O(b^{d})$	O(d)	O(d)

WHEN TO USE?

- **A*:** When optimal solution is required and memory is available.
- **IDA*:** When memory is limited but optimal solution is needed.
- **RBFS:** When memory is limited and backtracking is preferred.

KEY TAKEAWAYS

- ✔ Informed search uses heuristic information.
- ✔ A* is complete and optimal with admissible heuristic.
- ✔ IDA* saves memory but may be slower.
- ✔ RBFS uses very little memory and backtracking.
- ✔ The better the heuristic, the better the performance.

SUCCESS MEASURE

A solution is found when a goal state is reached.
Quality depends on:

- Optimality (lowest cost)
- Completeness (finds a solution if one exists)
- Time and memory used

BETTER HEURISTICS → SMARTER SEARCH → OPTIMAL SOLUTIONS

PURE HEURISTIC SEARCH ALGORITHMS

Search algorithms that use only heuristic information h(n) to guide the search toward the goal. They ignore the actual path cost g(n) from the start state.

1. WHAT IS PURE HEURISTIC SEARCH?

Pure heuristic search relies entirely on the heuristic function h(n) (estimated cost) to guide the search toward the goal. It ignores the actual path cost g(n) and focuses only on how close a node appears to the goal.

Characteristics:

- Uses only heuristic information h(n).
- Ignores path cost g(n).
- Expands the node that looks closest to the goal.
- Fast in practice but may not always find the optimal solution.
- Can be more efficient even if the path turns out to be longer or costlier.
- Suitable for large search spaces where quick answers are acceptable.

2. EVALUATION FUNCTION

Pure heuristic search uses the heuristic function:

$$\boxed{f(n) = h(n)}$$

Where,

- **h(n)** : heuristic estimate of cost from node n to goal.
- The node with the smallest h(n) is selected for expansion.

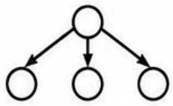
h(n) estimates how close a node is to the goal (lower value is better).

3. ALGORITHM EXAMPLE: GREEDY BEST-FIRST SEARCH (GBFS)

- A common pure heuristic search algorithm.
- Selects the node with the lowest h(n).
- Expands it and continues until the goal is reached.

How it works:

1. Start from the initial state.
2. Evaluate all successors using h(n).
3. Choose the node with the lowest h(n).
4. Expand that node.
5. Repeat until the goal is found (or no nodes left).

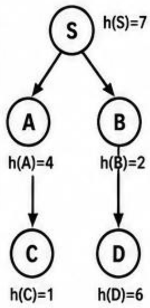


Always expand the node that appears closest to the goal (lowest h).

4. PROPERTIES

Property	
Complete?	No May get stuck in loops or infinite paths.
Optimal?	No May find a suboptimal (non-shortest) path.
Time Complexity	Depends on heuristic accuracy and branching factor (O(b^m))
Space Complexity	Depends on heuristic accuracy and branching factor (O(b^m))
Performance	Better heuristic → expands fewer nodes.

5. EXAMPLE



Step-by-Step (GBFS):

1. Start at S.
2. From S, successors → A (h=4), B (h=2). Choose B (lowest h).
3. Expand B. Successors → D (h=6). Choose D.
4. Next, expand A → C (h=1).
5. Path found: S → A → C (optimal path cost = 4, but GBFS may choose longer paths in other problems).

6. COMPARISON WITH OTHER SEARCHES

Property	Uninformed Search	Informed Search (A*)	Pure Heuristic Search (GBFS)	Best-First Search (Greedy)
Search Type	Uninformed	Informed	Pure Heuristic	Heuristic (Greedy)
Evaluation Function	–	f(n) = g(n) + h(n)	f(n) = h(n)	f(n) = h(n)
Uses g(n) (path cost)?	No	Yes	No	No
Uses h(n)?	No	Yes	Yes	Yes
Complete?	Depends	Yes*	No	No
Optimal?	Depends	Yes*	No	No
Memory Usage	Varies	High	Low	Low
Typical Use	No domain knowledge	When optimal solution is required	When fast solution is needed	When goal-directed speed is priority

* If heuristic h(n) is admissible and consistent.

7. WHEN TO USE?

- When a fast solution is needed.
- When optimality is not critical.
- In large search spaces where quick approximate answers are acceptable.
- When memory is limited.
- **Examples:** route finding, game playing, real-time planning.

8. KEY TAKEAWAYS

- Pure heuristic search uses only heuristic information h(n).
- It is fast but not complete or optimal.
- Greedy Best-First Search is the most common example.
- Good heuristics guide the search effectively.
- Performance depends on the quality of the heuristic function.
- Suitable for problems where speed is more important than optimality.

GUIDED BY HEURISTICS → FASTER SOLUTIONS → REAL-WORLD IMPACT

BEST-FIRST SEARCH (GREEDY SEARCH)

A specific type of informed search that always expands the node that looks best according to a heuristic function h(n). It is also called Greedy Best-First Search.

1. WHAT IS BEST-FIRST SEARCH?

A specific type of informed search that always expands the node that looks best according to a heuristic function h(n). It is also called Greedy Best-First Search.

Key Points:

- Uses heuristic information h(n) to guide the search.
- Selects the node that appears closest to the goal (smallest h(n)).
- Does not consider the actual path cost g(n).
- Goal-directed and fast, but may not find the optimal path.

2. EVALUATION FUNCTION

$$f(n) = h(n)$$

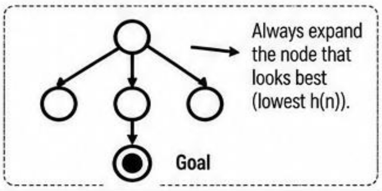
Where,

- **h(n)** = heuristic estimate of cost from node n to the goal.
- The node with the smallest h(n) is selected for expansion.

Example:
h(n) may represent the estimated distance to the goal (e.g., straight-line distance).

3. HOW IT WORKS?

1. Start from the initial state.
2. Evaluate all successors using h(n).
3. Choose the node with the lowest h(n) (most promising).
4. Expand that node.
5. Repeat until the goal is reached (or no nodes are left).

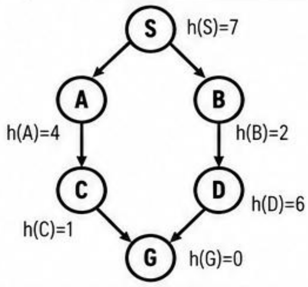


4. PROPERTIES

Complete?	No (may get stuck in loops or infinite paths)
Optimal?	No (may find a suboptimal path)
Time Complexity	O(b^m)
Space Complexity	O(b^m)
Depends on	Heuristic function h(n) only
Best Used When	Need a fast solution and optimality is not critical

b = branching factor
m = maximum depth of search tree

5. EXAMPLE



Step-by-Step (Greedy Best-First Search):

1. Start at S.
2. From S, successors → A (h=4), B (h=2). Choose B (lowest h).
3. From B, successor → D (h=6). Choose D.
4. Next, choose C (h=1).
5. Path found: S → B → C → G (not always optimal).

6. WHEN TO USE?

- When a fast solution is needed.
- When optimality is not critical.
- In large search spaces where quick approximate answers are acceptable.
- **Examples:** route planning, game playing, real-time systems.

7. ADVANTAGES AND LIMITATIONS

Advantages:

- ✔ Simple and easy to implement.
- ✔ Fast and goal-directed.
- ✔ Uses only heuristic information.

Limitations:

- ✖ Not complete (may get stuck in loops).
- ✖ Not optimal (may find a suboptimal path).
- ✖ Performance depends heavily on the quality of the heuristic function.

8. COMPARISON WITH OTHER SEARCHES

Property	Uninformed Search (BFS, DFS)	Informed Search (A*)	Greedy Best-First Search
Uses g(n)?	No	Yes	No
Uses h(n)?	No	Yes	Yes
Complete?	BFS: Yes DFS: No	Yes (if h admissible)	No
Optimal?	BFS: Yes DFS: No	Yes (if h admissible)	No
Time Complexity	Varies	Depends on h(n)	O(b^m)
Space Complexity	Varies	Depends on h(n)	O(b^m)
Typical Use	General search	Optimal solution required	Fast solution (not optimal)

TERMS

- **b** = branching factor (maximum number of successors of any node)
- **m** = maximum depth of the search tree
- **h(n)** = heuristic estimate from node n to goal
- **g(n)** = actual path cost from start to node n
- **f(n)** = evaluation function (used in A*)
- **O()** = asymptotic time or space complexity

GOAL OF SEARCH

Find a sequence of actions (path) from the initial state to the goal state as quickly as possible (not necessarily the best path).

GUIDED BY HEURISTICS → FASTER SOLUTIONS → REAL-WORLD IMPACT

KNOWLEDGE REPRESENTATION (KR)

1. WHAT IS KNOWLEDGE REPRESENTATION?

Knowledge Representation (KR) is the process of encoding information about the real world in a formal and structured way so that a machine can understand, store, retrieve, and use it to solve problems and make decisions.

Key Points:

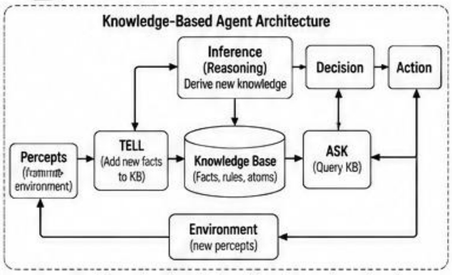
- Represents real-world knowledge in a machine-readable form.
- Organize and retrieve knowledge easily.
- Enable reasoning and inferencing.
- Support intelligent decision-making.
- Make knowledge reusable and shareable.

PURPOSE

Represent knowledge in a formal way so that machines can understand, reason with it to solve problems and make decisions.

6. KNOWLEDGE-BASED AGENTS

Agents that use a **knowledge base** (KB) and inference (reasoning) to decide actions.



- KB contains domain knowledge (facts, rules, axioms).
- **TELL**: add new information to KB.
- **ASK**: query KB to retrieve or infer knowledge.
- **Inference**: derive new facts from existing KB.
- The cycle repeats with new percepts.

2. KNOWLEDGE REPRESENTATION STRUCTURES

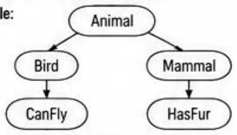
Common structures used to represent knowledge in AI systems.

2.1 Propositional Logic

- Represents facts as propositional symbols.
- Uses logical operators (¬, ∧, ∨, →).
- Suitable for simple domains.
- **Example:** P: It is raining.
  Q: The sky is cloudy.
  P → **Q** : If it is raining, then the sky is cloudy.

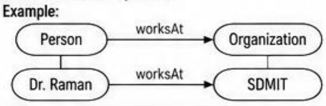
2.2 Semantic Networks

- Graphical representation of knowledge.
- Nodes represent objects/concepts.
- Links show relationships.
- Useful for hierarchical and associative knowledge.
- **Example:**



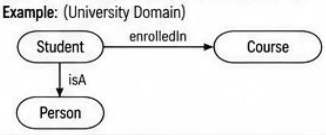
2.3 Knowledge Graphs

- Represents knowledge as entities and relations in a graph.
- Used in search engines, question answering, recommendation systems.
- **Example:**



2.4 Ontologies

- Formal specification of a domain.
- Defines concepts, relationships, and constraints.
- Used for knowledge sharing and interoperability.
- **Example:** (University Domain)



3. TYPES OF KNOWLEDGE

Different forms of knowledge used in AI.

Declarative Knowledge (What is known)	• Facts about the world. • Example: "Water boils at 100°C."
Procedural Knowledge (How to do)	• Steps, rules, or procedures to perform a task. • Example: "Steps to solve a math problem."
Heuristic Knowledge (Rules of thumb)	• Experience-based knowledge. • Useful for problem solving. • Example: "Try simpler solutions first."
Meta-Knowledge (Knowledge about knowledge)	• Knowledge about what we know and how to use it. • Example: "This rule is reliable in most cases."
Common-Sense Knowledge (Everyday knowledge)	• General world knowledge. • Example: "Birds can fly."

4. LOGICAL REASONING

Process of deriving new conclusions from given knowledge using formal rules of logic.

Inference Types:

- **Deductive Reasoning** – from general to specific.
- **Inductive Reasoning** – from specific to general.
- **Abductive Reasoning** – best explanation for observations.

Key Properties:

- **Soundness** – only true conclusions.
- **Completeness** – all true conclusions can be derived.
- **Consistency** – no contradictory facts.
- Used for automated reasoning and decision making in real-world applications.

5. KNOWLEDGE REPRESENTATION LANGUAGES (KRLs)

Formal languages used to represent knowledge in AI systems.

Language	Representation Logic	Used For
Propositional Logic	Simple facts and rules	Basic reasoning
First-Order Logic (FOL)	Objects, relations, quantifiers	Expressive knowledge representation
Rule-Based Representation	IF–THEN rules	Expert systems
Frames	Structured objects with attributes	Object-oriented knowledge
Ontological Languages (OWL)	Web semantics, ontologies	Knowledge sharing and interoperability

Examples: Prolog, CLIPS, OWL, RDF, Lisp

7. BENEFITS OF KNOWLEDGE REPRESENTATION

- Enables reasoning and decision-making.
- Promotes knowledge reuse and sharing.
- Improves system performance and explainability.
- Helps in building intelligent and undaptive systems.
- Supports complex real-world problem solving.

COMMON NOTATIONS

P, Q, R	:	Propositions (atomic sentences)
¬	:	NOT
∧	:	AND
∨	:	OR
←	:	Implies (if ... then)
KB	:	Knowledge Base
⊨	:	Entailment (KB entails α)

7. BENEFITS OF KNOWLEDGE REPRESENTATION

- Enables reasoning and decision-making.
- Promotes knowledge reuse and sharing.
- Improves system performance and explainability.
- Helps in building intelligent and adaptive systems.
- Supports complex real-world problem solving.

8. CHALLENGES

- Knowledge acquisition is hard.
- Time-consuming and costly.
- Handling incomplete, uncertain, and noisy knowledge.
- Scalability issues.
- Keeping knowledge base consistent and up to date.

GOAL OF KNOWLEDGE REPRESENTATION

To represent real-world knowledge in a formal, well-structured form that enables machines to understand, reason, and act intelligently in complex environments.

REPRESENT KNOWLEDGE → ENABLE REASONING → SOLVE REAL-WORLD PROBLEMS

1. WHAT IS A KNOWLEDGE-BASED AGENT?

A knowledge-based agent uses explicit knowledge (facts, rules, axioms) and inference (reasoning) to understand situations, decide on appropriate actions, and achieve goals.

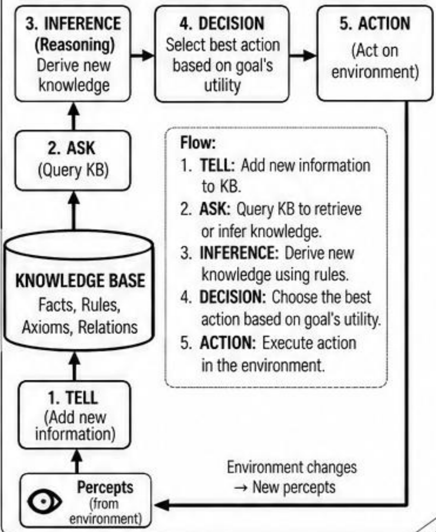
Key Points:

- Uses stored knowledge to make decisions.
- Applies reasoning (inference) to derive new knowledge.
- Selects actions appropriate for the current situation.
- Capable of handling complex, real-world problems.
- More flexible and intelligent than simple reflex agents.

Example: Smart Irrigation Robot

- KB: Low soil moisture → Plants may require water.
- Reasoning: If plants require water → Activate irrigation.
- Action: Start irrigation.

3. ARCHITECTURE / FLOW OF A KNOWLEDGE-BASED AGENT



Flow:

1. **TELL:** Add new information to KB.
2. **ASK:** Query KB to retrieve or infer knowledge.
3. **INFERENCE:** Derive new knowledge using rules.
4. **DECISION:** Choose the best action based on goal's utility.
5. **ACTION:** Execute action in the environment.

7. KNOWLEDGE REPRESENTATION LANGUAGES (KRLs)

Formal languages used to represent knowledge in AI systems.

Language	Used For
Propositional Logic	Simple facts and rules
First-Order Logic (FOL)	Objects, relations, quantifiers
Rule-Based Representation	IF–THEN rules
Frames	Structured objects with attributes
Ontological Languages (OWL)	Web semantics, ontologies

Examples: Prolog, CLIPS, OWL, RDF, Lisp

2. COMPONENTS OF A KNOWLEDGE-BASED AGENT

1. **Knowledge Base (KB)**
   Stores domain knowledge about the world (facts, rules, axioms).
2. **Knowledge Representation Language (KRL)**
   Formal language to represent knowledge (e.g., Propositional Logic, First-Order Logic, Ontologies).
3. **Axioms**
   Initial statements accepted as true and used to derive new facts.
4. **TELL Operation**
   Add new knowledge/information to the KB.
5. **ASK Operation**
   Query the KB to retrieve or infer information.
6. **Inference (Reasoning)**
   Derives new sentences from existing ones (using inference rules).
7. **Background Knowledge**
   General knowledge about the domain in which the agent operates.
8. **Agent Program**
   Takes percepts as input and returns an action using the KB, inference, and reasoning.

KNOWLEDGE-BASED AGENTS

Intelligent agents that use a knowledge base and inference (reasoning) to decide what action should be performed.

4. BENEFITS

- Can handle complex and uncertain situations.
- Supports reasoning and decision-making.
- Easy to update knowledge without changing code.
- Reusability of knowledge across different tasks.
- Better performance in complex environments.
- Helps in building intelligent and adaptive systems.

5. CHALLENGES

- Knowledge acquisition is hard.
- Time-consuming and costly.
- Handling incomplete, uncertain, and noisy knowledge.
- Scalability with very large KBs.
- Ensuring knowledge is consistent and up-to-date.

6. EXAMPLE WALKTHROUGH

Scenario: Smart Irrigation Robot
Percept: "Soil moisture is very low."
KB before TELL:

1. Low soil moisture → Plants may require water.
2. Plants may require water → Start irrigation.
3. If soil is wet → Stop irrigation.

Steps:

1. **TELL (KB update):** Add "Soil moisture is very low."
2. **Inference:** Plants may require water.
3. **Decision:** Start irrigation.
4. **Action:** Execute irrigation.

8. KEY TAKEAWAYS

- Knowledge-based agents rely on explicit knowledge and reasoning.
- TELL, ASK and inference operations are central.
- Goal-directed actions help achieve desired outcomes.
- More powerful and flexible than simple reflex agents.
- Knowledge representation language plays a crucial role.
- Widely used in expert systems, question answering, robotics, and intelligent assistants.

COMMON NOTATIONS

P, Q, R	: Propositions (atomic sentences)
$\neg$	: NOT
$\wedge$	: AND
$\vee$	: OR
$\Rightarrow$	: Implies (if ... then)
KB	: Knowledge Base
$\models$	: Entailment (KB entails α)

GOAL OF A KNOWLEDGE-BASED AGENT

Use knowledge, reasoning, and action to achieve goals in their environment.

Example Goal:
Keep plants healthy by starting irrigation when soil moisture is low.

KNOWLEDGE + REASONING + ACTION → INTELLIGENT DECISIONS → REAL-WORLD IMPACT

CHAPTER 2 MACHINE INTELLIGENCE AND KNOWLEDGE REPRESENTATION

From Real-World Information to Intelligent Action

Information → Knowledge
Knowledge Base → Representation
Reasoning / Inference → Decision
Action

2.1 DEFINING INTELLIGENCE: HUMAN vs MACHINE

Human Intelligence

- Uses experience, intuition and emotions.
- Learns from environment.
- Handles ambiguity.
- Creative and adaptive.
- Works in unstructured environments.

Machine Intelligence

- Processes information using algorithms.
- Fast, accurate and consistent.
- Learns from data.
- Scope for continuous improvement.
- Works in structured and unstructured environments.

Real-World Examples

AI in Healthcare
- Analyses medical images
- Supports diagnosis
- Assists treatment decisions

Autonomous Vehicles
- Sensors perceive environment
- Real-time decision making
- Safe navigation and adaptability

AI in Education
- Personalized learning
- Intelligent tutoring
- Adaptive content delivery

2.2 COMPONENTS OF INTELLIGENCE

Perception	• Senses and interprets the environment • E.g., vision, speech, sensors
Learning	• Improves from experience/data • E.g., supervised, unsupervised, reinforcement learning
Reasoning & Problem Solving	• Analyses information to solve problems • E.g., search, optimization, planning
Knowledge Representation	• Organizes and stores knowledge • E.g., facts, rules, relationships
Decision-Making	• Chooses the best action based on goals • E.g., utility, probability, reasoning
Adaptability	• Adjusts to new situations • E.g., online learning, transfer learning
Creativity / Innovation	• Generates new ideas or solutions • E.g., generative AI, design, creativity

2.4 SEARCH IN AI

A. Uninformed (Blind) Search
Searches without domain knowledge.
1. Breadth-First Search (BFS)
2. Depth-First Search (DFS)
3. Uniform Cost Search (UCS)
4. Depth-Limited Search (DLS)
5. Iterative Deepening DFS (IDDFS)
6. Bidirectional Search

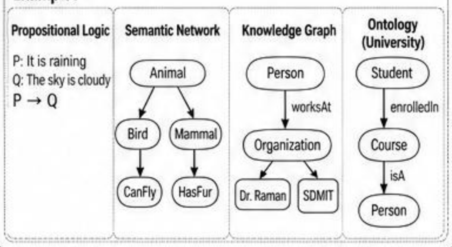
B. Informed (Heuristic) Search
Uses heuristic function $h(n)$.
- **Heuristic Function:** $h(n)$ – estimate of cost from n to goal.
- **Evaluation Functions:**
  - $g(n)$ – actual cost from start to n
  - $h(n)$ – heuristic cost to goal
  - $f(n) = g(n) + h(n)$ (A* Search)
- **Key Searches:** Greedy Best-First Search, A* Search
- **Goal:** Find optimal path/solution with minimum cost and effort.

2.5 KNOWLEDGE REPRESENTATION (KR)

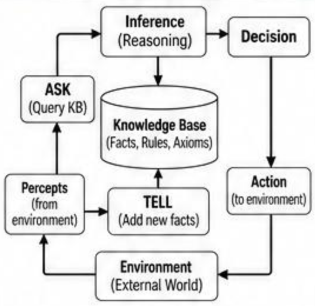
Purpose: Represent knowledge in a formal way so that machines can store, organize, reason with, and use it to solve problems and make decisions.

A. Knowledge Representation Structures
1. **Propositional Logic** – facts using logical symbols (P, Q, $\neg$, $\wedge$, $\vee$)
2. **Semantic Networks** – graph-based representation
3. **Knowledge Graphs** – entities and relations in a graph
4. **Ontologies** – formal specification of a domain

Examples



C. KNOWLEDGE-BASED AGENTS



- **KB** stores facts, rules, axioms, relations.
- **TELL:** add new information to KB.
- **ASK:** query KB to retrieve/infer knowledge.
- **Inference:** derive new knowledge using rules.
- **Decision:** select best action based on goals/utility.
- **Action:** execute action in the environment.
- Cycle repeats with new percepts.

3. COMPARISON OF SEARCH METHODS

Feature	Uninformed Search	Informed Search (A*)
Uses domain knowledge	No	Yes
Exploration	Systematic	Heuristic guided
Evaluation function	–	$f(n) = g(n) + h(n)$
Finds optimal solution	Yes (if conditions)	Yes (if h is admissible)
Time complexity	High	Lower (more efficient)
Space complexity	High	Lower
Examples	BFS, DFS, UCS, DLS, IDDFS	Greedy Best-First, A* Search

4. BENEFITS OF KR

- ✔ Enables reasoning and decision-making.
- ✔ Promotes knowledge reuse and sharing.
- ✔ Improves system performance and explainability.
- ✔ Helps in building intelligent and adaptive systems.
- ✔ Bridges the gap between human knowledge and machine understanding.

⚠ CHALLENGES

- Knowledge acquisition is hard.
- Time-consuming and costly.
- Handling incomplete, uncertain, and noisy knowledge.
- Scalability with very large KBs.
- Ensuring knowledge is consistent and up-to-date.

6. EXAMPLE WALKTHROUGH

Scenario: Smart Irrigation Robot

Percept: "Soil moisture is very low."

KB before TELL:
1. Low soil moisture → Plants may require water.
2. Plants may require water → Start irrigation.
3. If soil is wet → Stop irrigation.

Steps:
1. **TELL:** Add "Soil moisture is very low."
2. **Inference:** Plants may require water.
3. **Decision:** Start irrigation.
4. **Action:** Execute irrigation.

7. KEY TAKEAWAYS

- Knowledge-based agents rely on explicit knowledge and reasoning.
- Organize, store and retrieve knowledge effectively.
- Enable intelligent decision-making and problem solving.
- Support real-world applications (e.g., robotics, healthcare, education).

8. COMMON NOTATIONS

P, Q, R	: Propositions (atomic sentences)
$\neg$	: NOT
$\wedge$	: AND
$\vee$	: OR
→	: Implies (if ... then)
KB	: Knowledge Base
$\models$ (or |=)	: Entailment (KB entails α)

OVERALL GOAL

To build intelligent agents that perceive, reason, learn, and take effective actions to achieve goals in real-world environments.

KNOWLEDGE + REASONING + INTELLIGENT ACTION → SMARTER AI → BETTER REAL-WORLD SOLUTIONS


CHAPTER 3
PROMPT ENGINEERING FOR AI
BETTER PROMPTS. BRIGHTER POSSIBILITIES.
Ask
Design
Explore
Create
Together
IDEAS
KNOWLEDGE
SOLUTIONS
CREATIVITY
A BRIGHTER
TOMORROW
IDEA
GOAL
CONTEXT
CLARITY
BETTER PROMPTS
YOUR PROMPT
Instructions
Context
Input Data
Output Format
LARGE LANGUAGE MODEL
Understands
Reasons
Generates
Stays Aligned
MEANINGFUL RESPONSE
Useful
Relevant
Accurate
Context-aware
Creative
UNDERSTAND
REASON
GENERATE
ASSIST
EMPOWER
REAL-WORLD
IMPACT
From a Prompt to Possibilities
HUMAN
+
AI
=
GREATER
POSSIBILITIES
IDEAS
QUESTIONS
PROMPTS
KNOWLEDGE
BETTER ANSWERS
LEARN
NEW THINGS
SOLVE
REAL PROBLEMS
CREATE
ORIGINAL IDEAS
IMPROVE
PRODUCTIVITY
COLLABORATE
WITH AI
SHAPE
A BRIGHTER FUTURE
HUMAN
INTELLIGENCE
PROMPTS TODAY. A BRIGHTER TOMORROW.
AI
POSSIBILITIES

CHAPTER 3 PROMPT ENGINEERING FOR AI

Better Prompts. Brighter Possibilities.

Effective communication with AI through well designed prompts.

3.1 INTRODUCTION TO PROMPT ENGINEERING

- Prompt Engineering is the art and science of designing effective inputs (prompts) for AI systems.
- Involves understanding the task and specifying context, format, constraints and examples.
- A prompt can be a question, instruction, description or a combination.
- Goal: Get accurate, relevant, creative and context-aware responses.

3.2 EVOLUTION OF PROMPT ENGINEERING

- Early stage – simple queries and natural language.
- Growing use of LLMs – need for better prompts.
- From basic to advanced techniques (e.g., few-shot, chain-of-thought).
- Development of prompt design frameworks and best practices.
- Now a key skill for effective Human–AI interaction.

3.3 TYPES OF PROMPTS

- **Instruction Prompts** – give clear instructions.
- **Question Prompts** – ask specific questions.
- **Descriptive Prompts** – provide context and description.
- **Comparative Prompts** – compare options or ideas.
- **Chain-of-Thought Prompts** – encourage step-by-step reasoning.
- **Role-based Prompts** – assign a specific role (e.g., teacher, expert).
- **Few-Shot Prompts** – provide a few examples.
- **Zero-Shot Prompts** – no examples, just the task.
- **Multi-Modal Prompts** – include text, image, audio, etc.
- **And others** – depending on the task and goal.

3.4 HOW PROMPT ENGINEERING WORKS

- LLM processes the prompt, understands intent and context.
- Uses training data and internal patterns to generate a response.
- Examples in the prompt guide the output (few-shot).
- Iterative refinement of the prompt improves the quality of the output.

3.5 ROLE IN HUMAN–AI COMMUNICATION

- Acts as a bridge between human intent and AI understanding.
- Helps express needs clearly and effectively.
- Reduces ambiguity and improves response quality.
- Enables collaborative and productive interaction.

3.6 ADVANTAGES AND CHALLENGES

Advantages

- ✓ Better and more relevant outputs
- ✓ Improves accuracy and usability
- ✓ Saves time and effort
- ✓ Enables creative and innovative solutions
- ✓ Useful across multiple domains

Challenges

- ✕ Requires practice and skill
- ✕ Needs domain knowledge
- ✕ Sensitive to prompt wording
- ✕ Can produce biased or incorrect outputs
- ✕ Keeping up with rapidly evolving models

3.7 FUTURE OF LLM COMMUNICATION

- More natural and intuitive interaction.
- Multimodal prompts (text, image, audio, video).
- Personalized and adaptive responses.
- Integration with domain-specific tools and agents.
- More reliable, safe and controllable AI communication.

3.8 PROMPT TECHNIQUE

- Provide clear and specific instructions.
- Use the right context.
- Specify output format (e.g., bullet points, table, summary, etc.).
- Add examples (few-shot).
- Set constraints and tone.
- Ask for step-by-step reasoning.
- **Use iteration** – refine the prompt.
- **Example:** "Explain machine learning in simple words with two examples."

3.9 COMMON PROMPTING STRATEGIES

Zero-Shot Prompting

- No examples are provided.
- Model performs the task directly based on the instruction.
- Useful for simple and well-defined tasks.
- **Example:** "Classify the sentiment of this review: 'The food was great!'"

Few-Shot Prompting

- Provide one or a few examples before the task.
- Helps the model understand the pattern and expected output format.
- Useful for complex tasks.
- **Example:** "Review: Good service → Positive
  Review: Poor quality → Negative
  Review: Excellent! → ?"

Self-Consistency Prompting

- Generates multiple responses and selects the most consistent answer.
- Improves reliability and accuracy.
- Useful for reasoning tasks.
- **Example:** "Solve this math problem in three different ways and choose the most common answer."

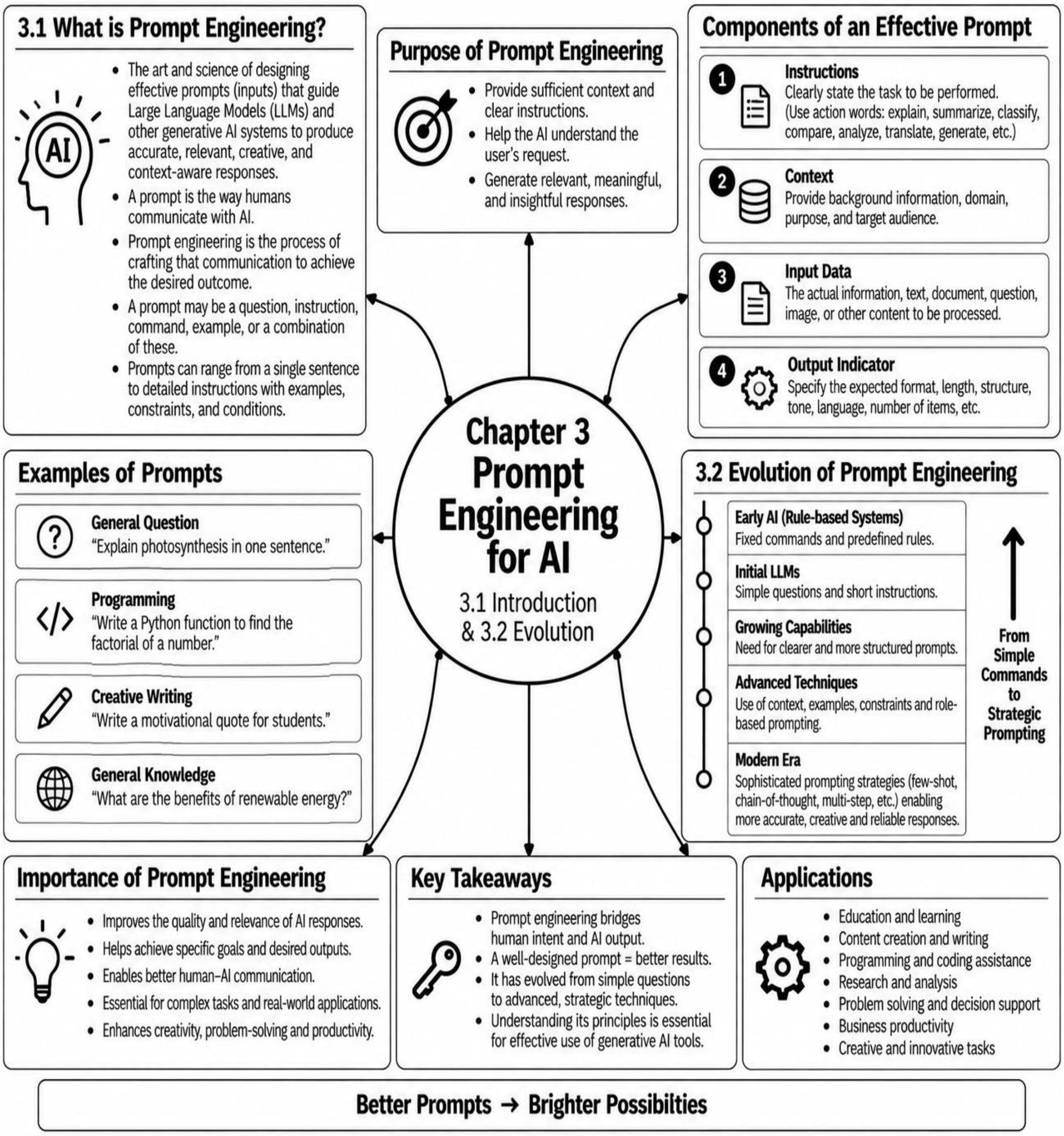
Prompts for Creative Thinking

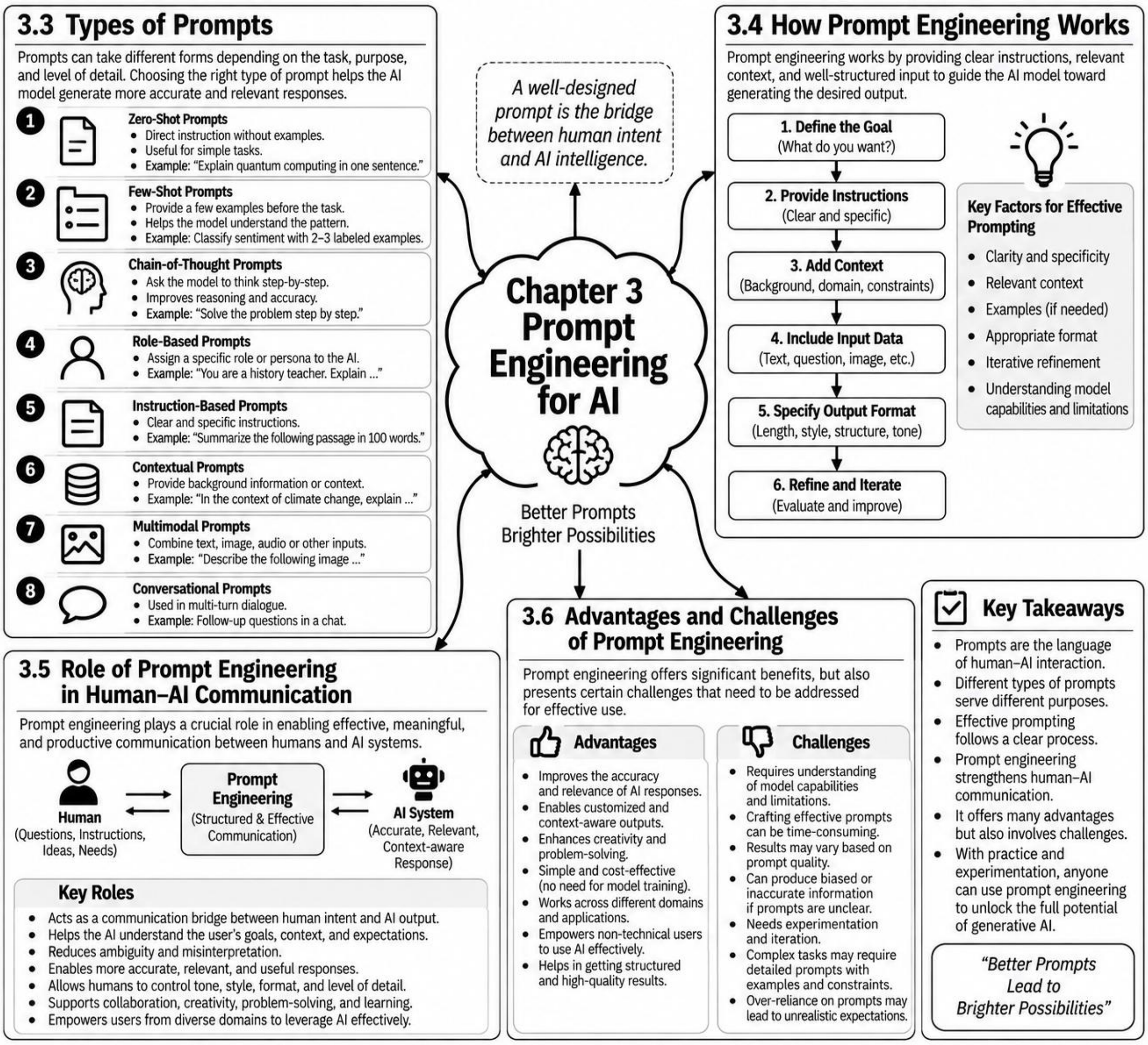
- Designed to generate creative and original ideas.
- Use open-ended instructions.
- Can include style, tone, perspective or constraints.
- **Example:** "Write a short story about a journey to Mars from the perspective of a child."

Prompts for Effective Writing

- Help in creating clear, structured and high-quality content.
- Can specify audience, tone, format and length.
- Useful for reports, emails, blogs, etc.
- **Example:** "Write a 200-word blog on the benefits of renewable energy for college students."

WELL-DESIGNED PROMPTS UNLOCK THE TRUE POTENTIAL OF AI.


3.1 What is Prompt Engineering?
The art and science of designing effective prompts (inputs) that guide Large Language Models (LLMs) and other generative AI systems to produce accurate, relevant, creative, and context-aware responses.
A prompt is the way humans communicate with AI.
Prompt engineering is the process of crafting that communication to achieve the desired outcome.
A prompt may be a question, instruction, command, example, or a combination of these.
Prompts can range from a single sentence to detailed instructions with examples, constraints, and conditions.
AI
Purpose of Prompt Engineering
Provide sufficient context and clear instructions.
Help the AI understand the user's request.
Generate relevant, meaningful, and insightful responses.
Components of an Effective Prompt
1 Instructions
Clearly state the task to be performed. (Use action words: explain, summarize, classify, compare, analyze, translate, generate, etc.)
2 Context
Provide background information, domain, purpose, and target audience.
3 Input Data
The actual information, text, document, question, image, or other content to be processed.
4 Output Indicator
Specify the expected format, length, structure, tone, language, number of items, etc.
Chapter 3
Prompt Engineering for AI
3.1 Introduction & 3.2 Evolution
Examples of Prompts
General Question
"Explain photosynthesis in one sentence."
Programming
"Write a Python function to find the factorial of a number."
Creative Writing
"Write a motivational quote for students."
General Knowledge
"What are the benefits of renewable energy?"
3.2 Evolution of Prompt Engineering
Early AI (Rule-based Systems)
Fixed commands and predefined rules.
Initial LLMs
Simple questions and short instructions.
Growing Capabilities
Need for clearer and more structured prompts.
Advanced Techniques
Use of context, examples, constraints and role-based prompting.
Modern Era
Sophisticated prompting strategies (few-shot, chain-of-thought, multi-step, etc.) enabling more accurate, creative and reliable responses.
From Simple Commands to Strategic Prompting
Importance of Prompt Engineering
Improves the quality and relevance of AI responses.
Helps achieve specific goals and desired outputs.
Enables better human–AI communication.
Essential for complex tasks and real-world applications.
Enhances creativity, problem-solving and productivity.
Key Takeaways
Prompt engineering bridges human intent and AI output.
A well-designed prompt = better results.
It has evolved from simple questions to advanced, strategic techniques.
Understanding its principles is essential for effective use of generative AI tools.
Applications
Education and learning
Content creation and writing
Programming and coding assistance
Research and analysis
Problem solving and decision support
Business productivity
Creative and innovative tasks
Better Prompts → Brighter Possibilties


3.3 Types of Prompts
Prompts can take different forms depending on the task, purpose, and level of detail. Choosing the right type of prompt helps the AI model generate more accurate and relevant responses.
1 Zero-Shot Prompts
• Direct instruction without examples.
• Useful for simple tasks.
• Example: "Explain quantum computing in one sentence."
2 Few-Shot Prompts
• Provide a few examples before the task.
• Helps the model understand the pattern.
• Example: Classify sentiment with 2–3 labeled examples.
3 Chain-of-Thought Prompts
• Ask the model to think step-by-step.
• Improves reasoning and accuracy.
• Example: "Solve the problem step by step."
4 Role-Based Prompts
• Assign a specific role or persona to the AI.
• Example: "You are a history teacher. Explain ..."
5 Instruction-Based Prompts
• Clear and specific instructions.
• Example: "Summarize the following passage in 100 words."
6 Contextual Prompts
• Provide background information or context.
• Example: "In the context of climate change, explain ..."
7 Multimodal Prompts
• Combine text, image, audio or other inputs.
• Example: "Describe the following image ..."
8 Conversational Prompts
• Used in multi-turn dialogue.
• Example: Follow-up questions in a chat.
A well-designed prompt is the bridge between human intent and AI intelligence.
Chapter 3 Prompt Engineering for AI
Better Prompts Brighter Possibilities
3.4 How Prompt Engineering Works
Prompt engineering works by providing clear instructions, relevant context, and well-structured input to guide the AI model toward generating the desired output.
1. Define the Goal (What do you want?)
2. Provide Instructions (Clear and specific)
3. Add Context (Background, domain, constraints)
4. Include Input Data (Text, question, image, etc.)
5. Specify Output Format (Length, style, structure, tone)
6. Refine and Iterate (Evaluate and improve)
Key Factors for Effective Prompting
• Clarity and specificity
• Relevant context
• Examples (if needed)
• Appropriate format
• Iterative refinement
• Understanding model capabilities and limitations
3.5 Role of Prompt Engineering in Human–AI Communication
Prompt engineering plays a crucial role in enabling effective, meaningful, and productive communication between humans and AI systems.
Human (Questions, Instructions, Ideas, Needs)
Prompt Engineering (Structured & Effective Communication)
AI System (Accurate, Relevant, Context-aware Response)
Key Roles
• Acts as a communication bridge between human intent and AI output.
• Helps the AI understand the user's goals, context, and expectations.
• Reduces ambiguity and misinterpretation.
• Enables more accurate, relevant, and useful responses.
• Allows humans to control tone, style, format, and level of detail.
• Supports collaboration, creativity, problem-solving, and learning.
• Empowers users from diverse domains to leverage AI effectively.
3.6 Advantages and Challenges of Prompt Engineering
Prompt engineering offers significant benefits, but also presents certain challenges that need to be addressed for effective use.
Advantages
• Improves the accuracy and relevance of AI responses.
• Enables customized and context-aware outputs.
• Enhances creativity and problem-solving.
• Simple and cost-effective (no need for model training).
• Works across different domains and applications.
• Empowers non-technical users to use AI effectively.
• Helps in getting structured and high-quality results.
Challenges
• Requires understanding of model capabilities and limitations.
• Crafting effective prompts can be time-consuming.
• Results may vary based on prompt quality.
• Can produce biased or inaccurate information if prompts are unclear.
• Needs experimentation and iteration.
• Complex tasks may require detailed prompts with examples and constraints.
• Over-reliance on prompts may lead to unrealistic expectations.
Key Takeaways
• Prompts are the language of human–AI interaction.
• Different types of prompts serve different purposes.
• Effective prompting follows a clear process.
• Prompt engineering strengthens human–AI communication.
• It offers many advantages but also involves challenges.
• With practice and experimentation, anyone can use prompt engineering to unlock the full potential of generative AI.
"Better Prompts Lead to Brighter Possibilities"

3.12 Prompts for Creative Thinking

Creative thinking prompts help generate new ideas, explore different perspectives, solve problems in innovative ways, and encourage imagination. They guide the AI to think beyond conventional answers and produce original and engaging responses.

Common Types of Creative Prompts

- **Idea Generation Prompts** – Generate new ideas, possibilities, or alternatives.
- **What If Prompts** – Explore hypothetical scenarios.
- **Role-Play / Perspective Prompts** – Think from a different person's perspective.
- **Analogy Prompts** – Use analogies to discover new insights.
- **Problem-Solving Prompts** – Find creative solutions to challenges.
- **Combination Prompts** – Combine multiple ideas or concepts.
- **Creative Constraints Prompts** – Generate ideas within specific limits.

Example Prompts

- Suggest 10 innovative ideas to reduce plastic waste.
- What if humans could live on Mars? Describe a possible day in the life.
- Look at this problem from the perspective of a child.
- Use an analogy to explain the internet.
- Combine AI and traditional education to create new learning opportunities.
- Generate a story with the constraint that it must be set in a single room.

Benefits

- Encourages originality and innovative thinking
- Helps explore multiple possibilities
- Supports brainstorming and ideation
- Useful for product design, research, business strategy, etc.
- Helps overcome mental blocks

Imagination turns questions into new possibilities.

Chapter 3 Prompt Engineering for AI

Better Prompts
Brighter Possibilities

"The right prompt doesn't just get answers, it sparks ideas and brings stories to life."

Create Ideas
Communicate Clearly
Build a Better Tomorrow

3.13 Prompts for Effective Writing

Effective writing prompts help the AI generate clear, well-structured, coherent, and engaging content for various purposes. They can support writing tasks such as drafting, summarizing, rewriting, editing, and adapting content for different audiences and styles.

Common Types of Writing Prompts

- **Content Generation Prompts** – Create articles, essays, reports, blog posts, etc.
- **Summarization Prompts** – Summarize long content into key points.
- **Rewriting / Paraphrasing Prompts** – Rewrite text in different words or style.
- **Tone and Style Prompts** – Adjust tone (formal, informal, professional, etc.).
- **Audience-Specific Prompts** – Write for a specific audience (students, professionals, etc.).
- **Editing and Improvement Prompts** – Improve grammar, clarity, and structure.
- **Translation Prompts** – Translate text into another language.

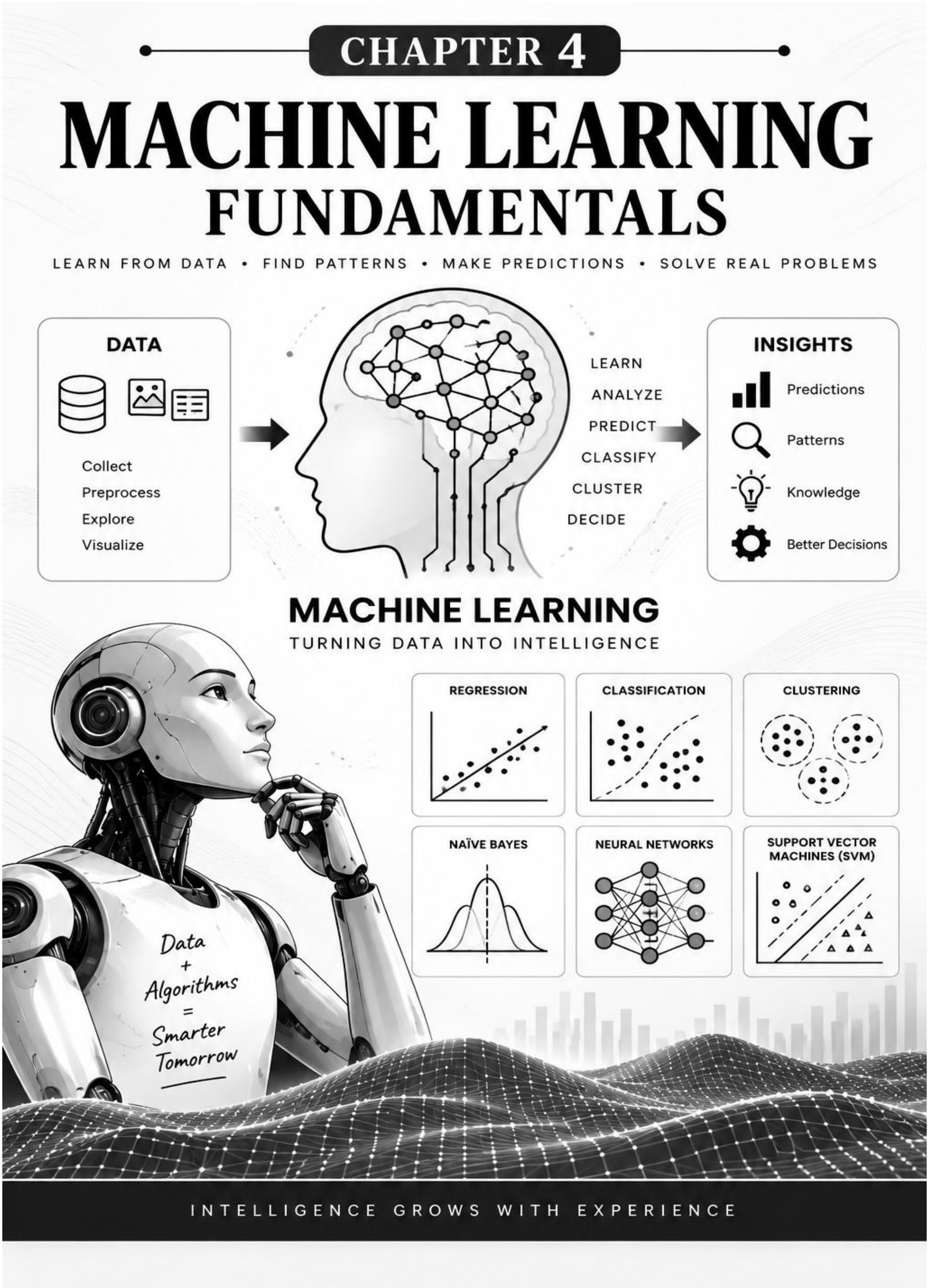
Example Prompts

- Write a 300-word blog post on the importance of clean energy.
- Summarize the following article in five key points.
- Rewrite this paragraph in a more professional tone.
- Explain climate change to school students in simple language.
- Improve the clarity and grammar of the following text.
- Translate the following text into Hindi.

Benefits

- Saves time and effort in writing
- Produces high-quality and well-structured content
- Helps adapt content for different audiences
- Improves clarity, grammar, and coherence
- Useful for academic, professional, and creative writing
- Supports multilingual communication

Better Prompts → Creative Ideas → Powerful Writing → A Brighter Future


CHAPTER 4
MACHINE LEARNING
FUNDAMENTALS
LEARN FROM DATA • FIND PATTERNS • MAKE PREDICTIONS • SOLVE REAL PROBLEMS
DATA
Collect
Preprocess
Explore
Visualize
LEARN
ANALYZE
PREDICT
CLASSIFY
CLUSTER
DECIDE
INSIGHTS
Predictions
Patterns
Knowledge
Better Decisions
MACHINE LEARNING
TURNING DATA INTO INTELLIGENCE
REGRESSION
CLASSIFICATION
CLUSTERING
NAÏVE BAYES
NEURAL NETWORKS
SUPPORT VECTOR MACHINES (SVM)
Data + Algorithms = Smarter Tomorrow
INTELLIGENCE GROWS WITH EXPERIENCE

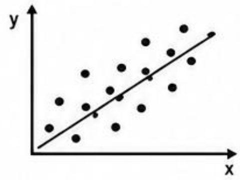
4.3 REGRESSION ANALYSIS IN ML

Predicts continuous numerical values (e.g., price, sales, temperature).

Common Algorithms:
- Linear Regression
- Polynomial Regression
- Ridge & Lasso Regression

Evaluation Metrics:
- MAE, MSE, RMSE, R^2

- **Applications:** House price prediction, sales forecasting, weather prediction.

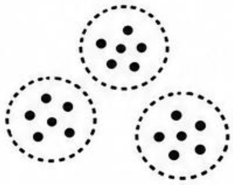


4.5 CLUSTERING TECHNIQUES

Groups similar data points without using labels (unsupervised learning).

Major Techniques:
- K-Means Clustering
- Hierarchical Clustering
- DBSCAN (Density-Based)
- Mean Shift Clustering
- Gaussian Mixture Models (GMM)

Applications:
Customer segmentation, image segmentation, market analysis.



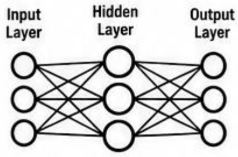
4.7 NEURAL NETWORKS IN AI

Inspired by the human brain, used for complex pattern learning.

Structure:
- Input Layer
- Hidden Layer(s)
- Output Layer

Key Features:
- Learns complex patterns
- Handles large and unstructured data
- Used in deep learning

Applications:
- Image recognition
- Speech recognition
- Natural language processing

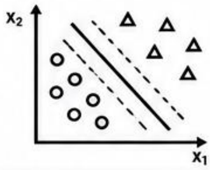


4.8 SUPPORT VECTOR MACHINES (SVM)

Finds the optimal hyperplane to separate classes.

Key Features:
- Effective for high-dimensional data
- Works well for non-linear problems (using kernels)
- Robust to overfitting

Applications:
- Text classification (spam detection)
- Image recognition
- Bioinformatics (disease prediction)



4.2 MACHINE LEARNING MODELS

Learn from data to make predictions or decisions.

Steps:
1. Collect data
2. Preprocess data
3. Train the model
4. Evaluate and tune
5. Deploy and monitor

Types of ML:
- Supervised Learning (labeled data)
- Unsupervised Learning (unlabeled data)
- Reinforcement Learning (rewards/penalties)

Applications:
- Spam detection
- Fraud detection
- Medical diagnosis
- Predictive analytics

Chapter 4
MACHINE LEARNING FUNDAMENTALS

Learn from Data
Find Patterns
Make Predictions
Solve Real Problems

KEY TAKEAWAYS

- Uses data, algorithms and computing power to learn from data.
- Enables machines to make predictions and decisions automatically.
- Different techniques solve different types of problems.
- Choice of algorithm depends on data, problem type and domain.
- Real-world applications span across many industries.
- ML techniques have limitations such as bias, data quality and interpretability.
- Continuous learning and innovation are improving ML models.

Data + Algorithms + Experience = Smarter Tomorrow

4.1 AI TECHNIQUES AND ML OVERVIEW

Artificial Intelligence (AI) enables machines to perform tasks that normally require human intelligence (e.g., learning, reasoning, perception).

Major AI Techniques:
- Machine Learning (ML)
- Deep Learning (DL)
- Natural Language Processing (NLP)
- Computer Vision (CV)
- Robotics
- Expert Systems

Applications:
- Autonomous vehicles
- Virtual assistants
- Medical diagnosis
- Fraud detection
- Smart manufacturing

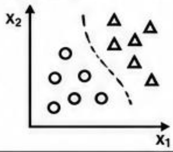
4.4 CLASSIFICATION TECHNIQUES

Predicts categorical class labels (e.g., spam/not spam, disease/healthy).

Common Algorithms:
- Logistic Regression
- Decision Trees
- Random Forest
- K-Nearest Neighbours (KNN)
- Support Vector Machine (SVM)
- Naive Bayes
- Neural Networks
- Gradient Boosting (XGBoost)

Evaluation Metrics:
Accuracy, Precision, Recall, F1-Score

Applications:
- Spam detection
- Sentiment analysis
- Image classification
- Disease prediction
- Customer churn prediction



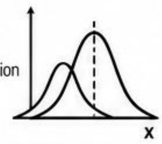
4.6 NAIVE BAYES CLASSIFICATION

A probabilistic classifier based on Bayes' Theorem with an assumption of feature independence.

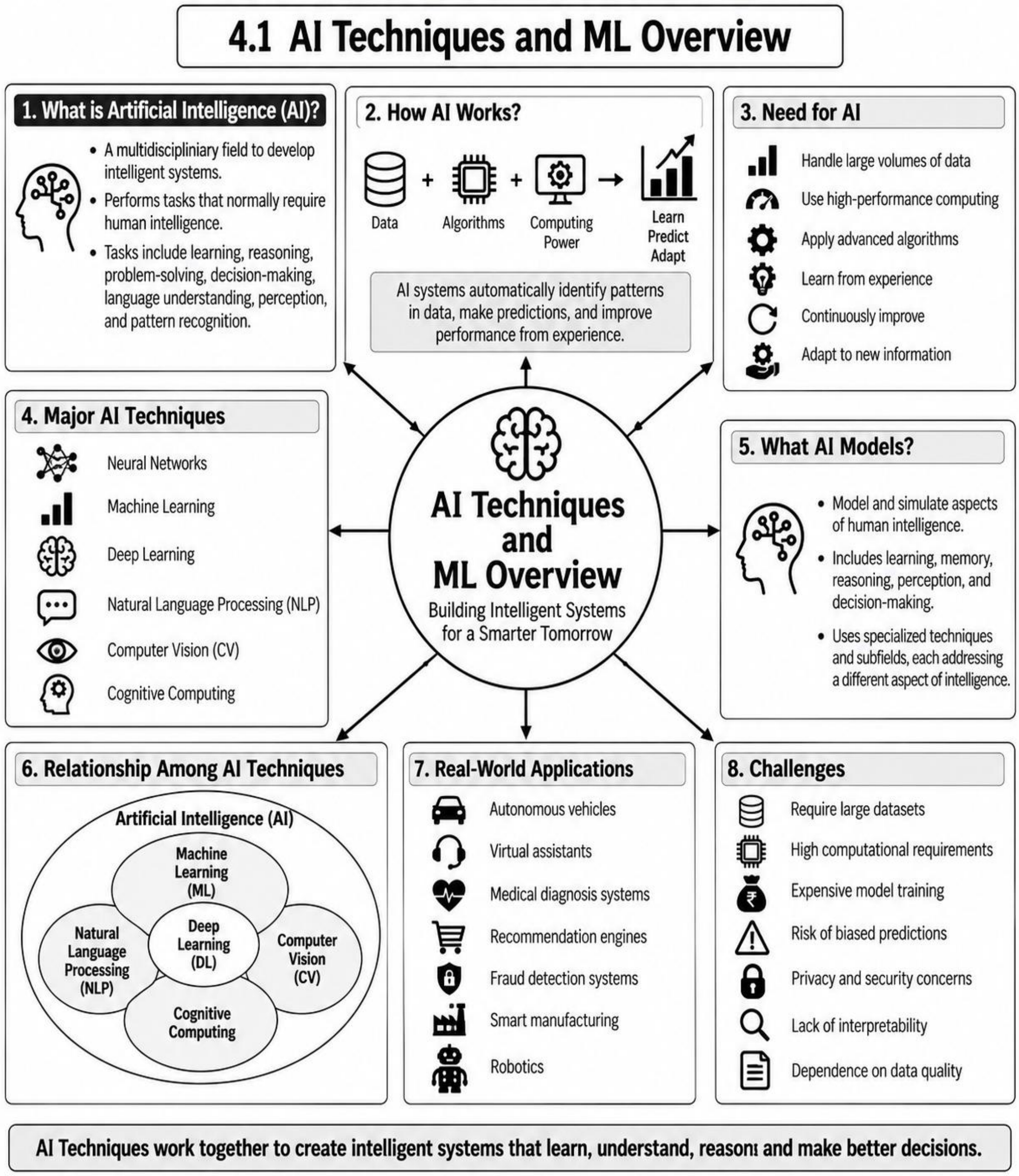
Common Types:
- Gaussian Naive Bayes (continuous data)
- Multinomial Naive Bayes (text data)
- Bernoulli Naive Bayes (binary data)

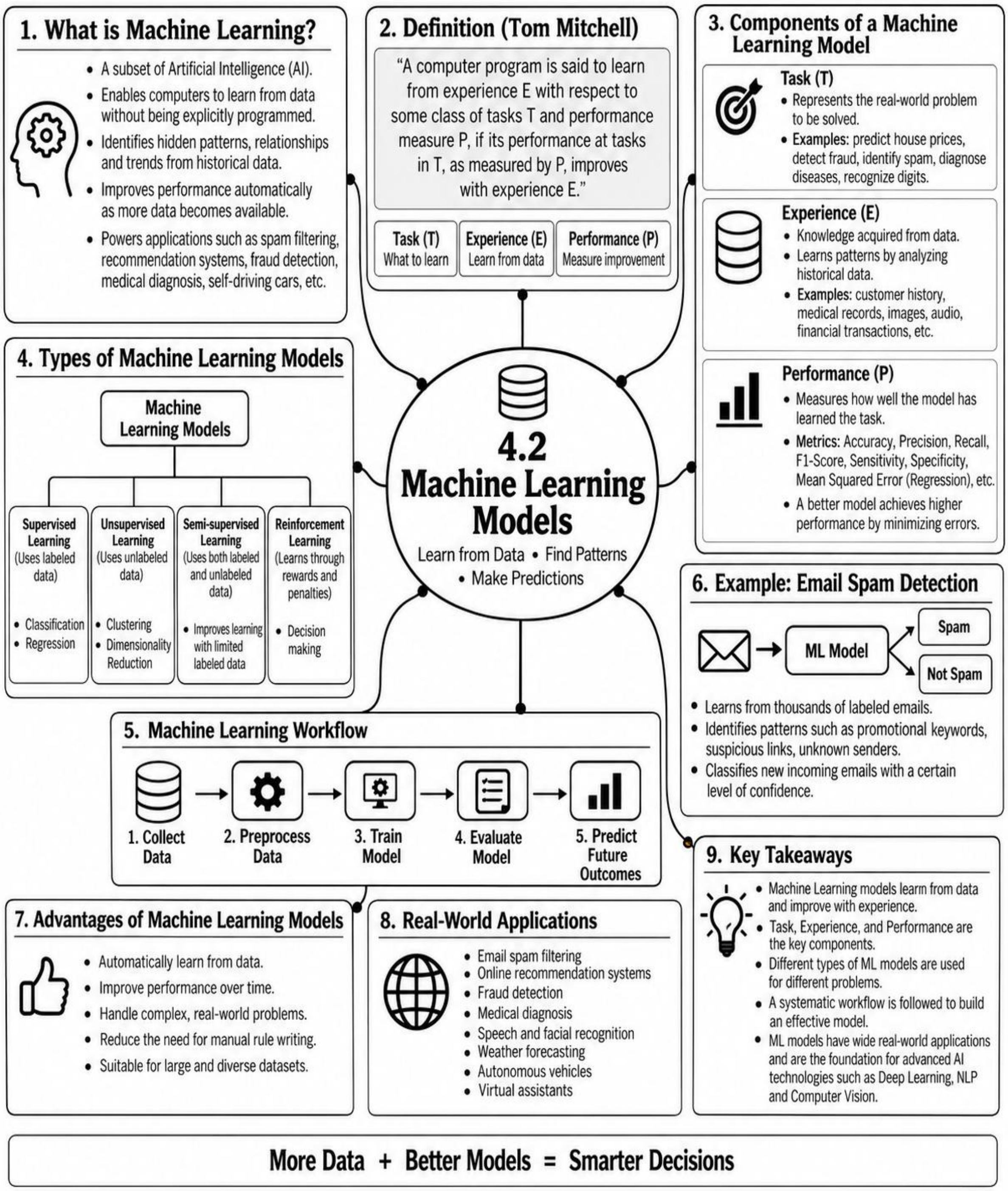
Advantages:
- Simple and fast
- Works well for high-dimensional data
- Effective for text classification

Applications:
- Spam filtering
- Sentiment analysis
- Document categorization
- Disease prediction



DATA + ALGORITHMS + EXPERIENCE = A SMARTER TOMORROW


4.1 AI Techniques and ML Overview
1. What is Artificial Intelligence (AI)?
• A multidiscipliniary field to develop intelligent systems.
• Performs tasks that normally require human intelligence.
• Tasks include learning, reasoning, problem-solving, decision-making, language understanding, perception, and pattern recognition.
2. How AI Works?
Data
+
Algorithms
+
Computing Power
→
Learn Predict Adapt
AI systems automatically identify patterns in data, make predictions, and improve performance from experience.
3. Need for AI
Handle large volumes of data
Use high-performance computing
Apply advanced algorithms
Learn from experience
Continuously improve
Adapt to new information
4. Major AI Techniques
Neural Networks
Machine Learning
Deep Learning
Natural Language Processing (NLP)
Computer Vision (CV)
Cognitive Computing
AI Techniques and ML Overview
Building Intelligent Systems for a Smarter Tomorrow
5. What AI Models?
• Model and simulate aspects of human intelligence.
• Includes learning, memory, reasoning, perception, and decision-making.
• Uses specialized techniques and subfields, each addressing a different aspect of intelligence.
6. Relationship Among AI Techniques
Artificial Intelligence (AI)
Machine Learning (ML)
Natural Language Processing (NLP)
Deep Learning (DL)
Computer Vision (CV)
Cognitive Computing
7. Real-World Applications
Autonomous vehicles
Virtual assistants
Medical diagnosis systems
Recommendation engines
Fraud detection systems
Smart manufacturing
Robotics
8. Challenges
Require large datasets
High computational requirements
Expensive model training
Risk of biased predictions
Privacy and security concerns
Lack of interpretability
Dependence on data quality
AI Techniques work together to create intelligent systems that learn, understand, reason and make better decisions.


1. What is Machine Learning?
• A subset of Artificial Intelligence (AI).
• Enables computers to learn from data without being explicitly programmed.
• Identifies hidden patterns, relationships and trends from historical data.
• Improves performance automatically as more data becomes available.
• Powers applications such as spam filtering, recommendation systems, fraud detection, medical diagnosis, self-driving cars, etc.
2. Definition (Tom Mitchell)
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."
Task (T) What to learn
Experience (E) Learn from data
Performance (P) Measure improvement
3. Components of a Machine Learning Model
Task (T)
• Represents the real-world problem to be solved.
• Examples: predict house prices, detect fraud, identify spam, diagnose diseases, recognize digits.
Experience (E)
• Knowledge acquired from data.
• Learns patterns by analyzing historical data.
• Examples: customer history, medical records, images, audio, financial transactions, etc.
Performance (P)
• Measures how well the model has learned the task.
• Metrics: Accuracy, Precision, Recall, F1-Score, Sensitivity, Specificity, Mean Squared Error (Regression), etc.
• A better model achieves higher performance by minimizing errors.
4.2
Machine Learning Models
Learn from Data • Find Patterns • Make Predictions
4. Types of Machine Learning Models
Machine Learning Models
Supervised Learning (Uses labeled data)
• Classification
• Regression
Unsupervised Learning (Uses unlabeled data)
• Clustering
• Dimensionality Reduction
Semi-supervised Learning (Uses both labeled and unlabeled data)
• Improves learning with limited labeled data
Reinforcement Learning (Learns through rewards and penalties)
• Decision making
5. Machine Learning Workflow
1. Collect Data
2. Preprocess Data
3. Train Model
4. Evaluate Model
5. Predict Future Outcomes
6. Example: Email Spam Detection
ML Model
Spam
Not Spam
• Learns from thousands of labeled emails.
• Identifies patterns such as promotional keywords, suspicious links, unknown senders.
• Classifies new incoming emails with a certain level of confidence.
7. Advantages of Machine Learning Models
• Automatically learn from data.
• Improve performance over time.
• Handle complex, real-world problems.
• Reduce the need for manual rule writing.
• Suitable for large and diverse datasets.
8. Real-World Applications
• Email spam filtering
• Online recommendation systems
• Fraud detection
• Medical diagnosis
• Speech and facial recognition
• Weather forecasting
• Autonomous vehicles
• Virtual assistants
9. Key Takeaways
• Machine Learning models learn from data and improve with experience.
• Task, Experience, and Performance are the key components.
• Different types of ML models are used for different problems.
• A systematic workflow is followed to build an effective model.
• ML models have wide real-world applications and are the foundation for advanced AI technologies such as Deep Learning, NLP and Computer Vision.
More Data + Better Models = Smarter Decisions

4.3 Regression Analysis in ML

1. What is Regression?

- A supervised machine learning technique used to predict a continuous numerical value.
- Learns the relationship between input variables (features) and an output variable (target) from historical data.
- Used to estimate values, identify trends, and support data-driven decision making.
- **Examples:** house prices, sales, temperature, rainfall, stock prices, etc.

2. Key Difference

Feature	Regression	Classification
Output	Continuous numerical value	Discrete class label
Goal	Estimate quantities	Assign categories
Example	House Price Prediction	Spam Detection
Output Example	₹75 Lakhs	Spam
Evaluation Metrics	MAE, MSE, RMSE, R^2 Score	Accuracy, Precision, Recall, F1-Score
Common Algorithms	Linear Regression, Random Forest Regression	Decision Tree, SVM, Naïve Bayes

3. How Regression Works?

Input Data (features X) → Historical data with input features (e.g., area, bedrooms)

↓

Regression Model (f(x)) → Learns the relationship between X and Y

↓

Continuous Output (predicted value ŷ) → Predicts numerical value for new data (e.g., house price)

Regression Analysis in ML

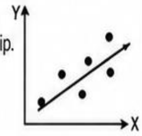
Learn from Data • Predict Continuous Values
Support Better Decisions

4. Types of Regression

a) Simple Linear Regression

- One independent variable and one dependent variable.
- Models a linear relationship.

$$Y = \beta_0 + \beta_1 X + \varepsilon$$



b) Multiple Linear Regression

- Two or more independent variables.
- Models the combined influence of multiple predictors.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

5. Regression Model Components

Task (T): Real-world problem to be solved (e.g., predict house prices) → **Experience (E)**: Knowledge gained from data (e.g., historical records) → **Performance (P)**: Measure how well the model works (e.g., MAE, MSE, R^2 Score)

6. Working Process (Steps)

1. **Collect Historical Data** (Gather input and output data)
2. **Preprocess Data** (Clean, handle missing values, feature selection)
3. **Train the Model** (Learn the relationship between X and Y)
4. **Evaluate the Model** (Use appropriate metrics)
5. **Predict** (Estimate output for new, unseen data)

7. Popular Regression Algorithms

- Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- Elastic Net Regression
- Decision Tree Regression
- Random Forest Regression
- Support Vector Regression (SVR)
- Gradient Boosting Regression
- XGBoost Regression

8. Evaluation Metrics

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE}$$

- R^2 Score (Coefficient of Determination)

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

9. Real-Time Examples

- **House Price Prediction** (Predict price based on area, location, etc.)
- **Sales Forecasting** (Predict future sales)
- **Weather Prediction** (Estimate temperature, rainfall, etc.)
- **Stock Price Prediction** (Forecast stock prices)
- **Student Performance** (Predict marks based on study hours, attendance, etc.)

10. Limitations

- Requires high-quality historical data.
- Sensitive to noisy data and outliers.
- Linear regression assumes linear relationships.
- Performance decreases when important variables are missing.
- Complex nonlinear relationships require advanced techniques.

Regression turns data into insights, enabling better predictions for a smarter future.

4.4 Classification Techniques

1. What is Classification?

- A supervised machine learning technique.
- Predicts discrete class labels (categories) for a given input based on its features.
- Learns from labeled training data where the correct class label is known.
- Used in many real-world applications such as spam detection, disease diagnosis, fraud detection, etc.

2. Classification vs Regression

Feature	Classification	Regression
Output	Discrete class label (e.g., Yes/No)	Continuous numerical value (e.g., price)
Goal	Assign categories	Estimate quantities
Example	Spam Detection	House Price Prediction
Output Example	Spam / Not Spam	₹75 Lakhs

3. Types of Classification

Binary Classification

- Two possible classes.
- **Examples:** Yes/No, Spam/Not Spam, Fraud/Not Fraud, Pass/Fail.

Multiclass Classification

- More than two mutually exclusive classes.
- **Example:** Identify a fruit as Apple, Mango, Orange, or Banana.

Multilabel Classification

- A single instance may belong to more than one class.
- **Example:** A movie may belong to both "Action" and "Adventure".

Classification Techniques

Assign Data to Predefined Categories

4. How Does Classification Work?

Labeled Training Data (Input features + Class labels)

↓

Learning Algorithm (Find patterns in data)

↓

Trained Model

↓

New / Unseen Data (Input features)

↓

Predicted Class Label (with confidence / probability)

5. Common Classification Algorithms

- K-Nearest Neighbours (KNN)
- Logistic Regression
- Naïve Bayes Classifier
- Decision Tree Classifier
- Random Forest Classifier
- Support Vector Machine (SVM)
- Gradient Boosting Classifier
- XGBoost Classifier
- Artificial Neural Networks (ANN)
- Deep Learning Classifiers

6. Evaluation Metrics

- **Accuracy** – Overall correntcess
- **Precision** – Correct positive predictions
- **Recall** – Ability to find all positives
- **F1-Score** – Balance between precision and recall
- **Specificity** – Correct negative predictions
- **ROC-AUC** – Overall performance

7. Real-World Applications

- Email spam filtering
- Medical diagnosis
- Fraud detection
- Customer segmentation
- Image classification
- Sentiment analysis
- Document categorization
- Intrusion detection

8. Key Points

- Learns from labeled data to classify new instances.
- Suitable for binary, multiclass and multilabel problems.
- Many algorithms are available, each with its own strengths and limitations.
- Performance depends on data quality, feature selection and choice of algorithm.
- Widely used in real-world AI applications to automate decision-making.

From Data to Decisions – Classification Makes It Possible!

4.5 Clustering Techniques

1. What is Clustering?

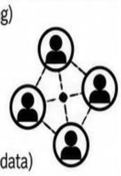
- An unsupervised machine learning technique.
- Groups similar data points into clusters without predefined labels.
- Identifies natural patterns and structures in data.
- Useful for exploratory data analysis and discovering hidden relationships.
- Widely used in customer segmentation, image grouping, anomaly detection, etc.

2. Key Characteristics

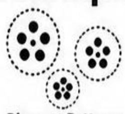
- Unsupervised learning (no labels).
- Groups similar data points.
- Discovers natural patterns in data.
- Number of clusters may or may not be specified (depends on algorithm).
- Can handle large and complex datasets.
- Used for segmentation, pattern discovery and outlier detection.

3. Applications

- Customer segmentation (e.g., marketing)
- Image segmentation
- Document grouping / topic modeling
- Gene and biological data analysis
- Anomaly / outlier detection
- Social network analysis
- Geographical data analysis (e.g., GPS data)
- Market basket analysis
- Recommender systems (user grouping)



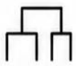
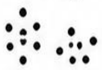
Clustering Techniques



Discover Patterns in Unlabeled Data

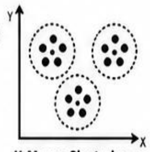
4. Types of Clustering Techniques

Clustering Techniques

Partitional Clustering	Hierarchical Clustering	Density-Based Clustering
		
Divides data into K clusters (e.g., K-Means)	Builds a hierarchy of clusters (e.g., Agglomerative)	Groups based on density (e.g., DBSCAN)

5. Partitional Clustering (K-Means)

- Divides data into K clusters.
- K (number of clusters) must be specified before clustering.
- Works well for spherical clusters.
- Simple and efficient.

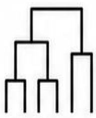


K-Means Clustering

Advantages	Limitations
• Simple and fast. • Works well for large datasets. • Easy to implement.	• Must specify K in advance. • Assumes spherical clusters. • Sensitive to outliers.

6. Hierarchical Clustering

- Builds a hierarchy (tree of clusters).
- Two approaches:
  - Agglomerative (bottom-up)
  - Divisive (top-down)
- No need to specify the number of clusters initially.
- Visualized using a dendrogram.

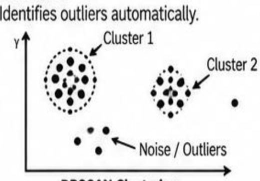


Dendrogram

Advantages	Limitations
• Easy to visualize. • No need to specify K. • Produces stable results. • Reveals relationships among clusters.	• Computationally expensive. • Slow for very large datasets. • Sensitive to noisy data. • Once merged/split, process cannot be reversed.

7. Density-Based Clustering

- Groups data points based on density.
- Finds clusters of arbitrary shapes.
- Does not require the number of clusters in advance.
- Identifies outliers automatically.



DBSCAN Clustering

Advantages	Limitations
• Detects arbitrary shapes. • No need to specify K. • Handles noise effectively. • Identifies outliers.	• Difficult to choose parameters (e.g., ε, MinPts). • Performance decreases in high-dimensional data. • Struggles with clusters of varying densities.

8. Comparison of Clustering Techniques

Feature	Partitional (K-Means)	Hierarchical	Density-Based (DBSCAN)
Basic Idea	Divide into K clusters	Build hierarchy	Group by density
Number of Clusters	Specify K	Not required (initially)	Determined automatically
Cluster Shape	Spherical	Any structure	Arbitrary shapes
Noise Handling	Poor	Moderate	Excellent
Outlier Detection	Poor	Limited	Excellent
Scalability	High	Moderate	Moderate
Computational Cost	Low	High	Moderate
Deterministic	No	Yes	Yes
Popular Algorithms	K-Means, K-Medoids	Agglomerative, Divisive	DBSCAN, OPTICS, HDBSCAN
Typical Applications	Customer segmentation	Gene analysis	GPS data, Fraud detection

Clustering helps us find hidden patterns and groups in data, enabling better insights and decision-making.

4.6 Naïve Bayes Classification

A Simple Probabilistic Approach for Effective Classification

1. What is Naïve Bayes Classification?

- A probabilistic classification algorithm based on Bayes' Theorem.
- Assumes that the features (attributes) are conditionally independent of each other given the class label.
- Simple, fast, and effective, especially for high-dimensional data.
- Widely used in text classification, spam detection, sentiment analysis, etc.

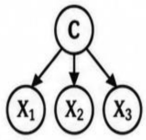
2. Bayes' Theorem

$$P(C \mid X) = \frac{P(X \mid C)\, P(C)}{P(X)}$$

- $P(C \mid X)$: Posterior probability (probability of class C given features X)
- $P(X \mid C)$: Likelihood (probability of features X given class C)
- $P(C)$: Prior probability of class C
- $P(X)$: Evidence (probability of features X)

3. Naïve Bayes Assumption

- Assumes that all features are conditionally independent given the class label.
- This assumption is called the "naïve" assumption.
- Even though the assumption is simple, it works well in many real-world problems.



4. Types of Naïve Bayes Classifier

- **Gaussian Naïve Bayes**
  – Used for continuous numeric data (assumes normal distribution).
- **Multinomial Naïve Bayes**
  – Used for discrete count data (e.g., word frequencies in text).
- **Bernoulli Naïve Bayes**
  – Used for binary data (e.g., presence or absence of words).

5. Working Process

Collect and preprocess data
↓
Calculate prior probabilities P(C)
↓
Calculate likelihood P(X | C)
↓
Use Bayes' theorem to compute P(C | X)
↓
Assign the class with highest probability

6. Example: Email Spam Detection

Input Email: "Free lottery win now"
Classes: Spam / Not Spam
Features: words, sender info, links, etc.
Output: Predicted Class = Spam (based on highest probability)

7. Applications

- Email spam detection
- Text classification
- Sentiment analysis
- Medical diagnosis
- Document categorization
- Fraud detection
- Recommendation systems

8. Advantages

- Simple and easy to implement.
- Fast training and prediction.
- Works well with high-dimensional data.
- Requires less training data.
- Performs well in text classification.

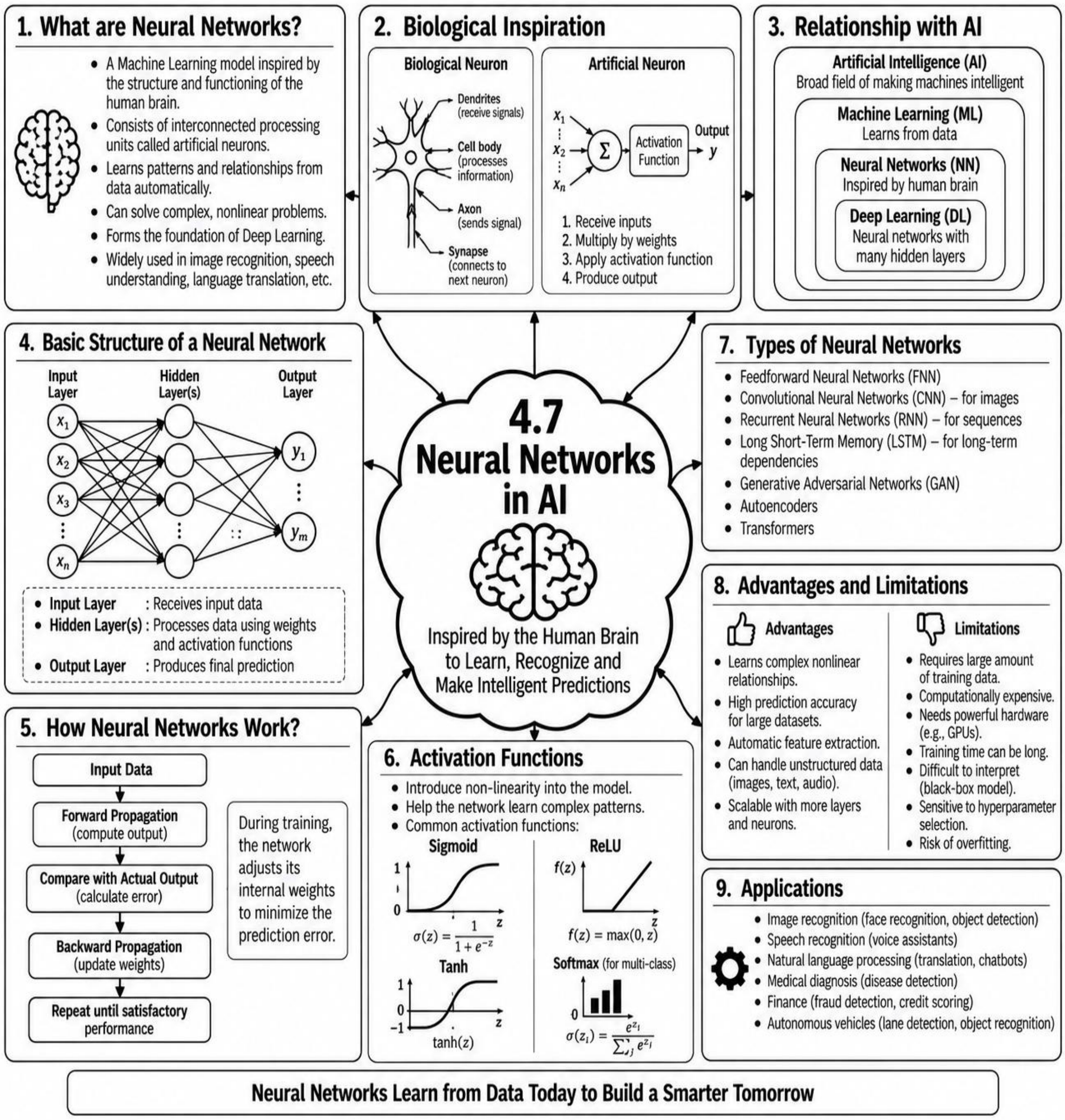
9. Limitations

- Assumes feature independence (which may not hold in real-world data).
- Performance may decrease when features are highly correlated.
- Not suitable for modeling complex relationships between features.
- Less accurate compared to more advanced algorithms in some cases.

10. Key Takeaways

- Naïve Bayes is a simple yet powerful probabilistic classifier.
- Based on Bayes' theorem with independence assumption.
- Available in Gaussian, Multinomial and Bernoulli variants.
- Widely used in real-world applications, especially text classification.
- Works well with small and high-dimensional datasets.

Simple Probabilities → Powerful Predictions → Real-World Impact


1. What are Neural Networks?
• A Machine Learning model inspired by the structure and functioning of the human brain.
• Consists of interconnected processing units called artificial neurons.
• Learns patterns and relationships from data automatically.
• Can solve complex, nonlinear problems.
• Forms the foundation of Deep Learning.
• Widely used in image recognition, speech understanding, language translation, etc.
2. Biological Inspiration
Biological Neuron
Dendrites (receive signals)
Cell body (processes information)
Axon (sends signal)
Synapse (connects to next neuron)
Artificial Neuron
x_1
x_2
x_n
Σ
Activation Function
Output y
1. Receive inputs
2. Multiply by weights
3. Apply activation function
4. Produce output
3. Relationship with AI
Artificial Intelligence (AI)
Broad field of making machines intelligent
Machine Learning (ML)
Learns from data
Neural Networks (NN)
Inspired by human brain
Deep Learning (DL)
Neural networks with many hidden layers
4. Basic Structure of a Neural Network
Input Layer
Hidden Layer(s)
Output Layer
x_1
x_2
x_3
x_n
y_1
y_m
• Input Layer : Receives input data
• Hidden Layer(s) : Processes data using weights and activation functions
• Output Layer : Produces final prediction
4.7
Neural Networks in AI
Inspired by the Human Brain to Learn, Recognize and Make Intelligent Predictions
7. Types of Neural Networks
• Feedforward Neural Networks (FNN)
• Convolutional Neural Networks (CNN) – for images
• Recurrent Neural Networks (RNN) – for sequences
• Long Short-Term Memory (LSTM) – for long-term dependencies
• Generative Adversarial Networks (GAN)
• Autoencoders
• Transformers
8. Advantages and Limitations
Advantages
• Learns complex nonlinear relationships.
• High prediction accuracy for large datasets.
• Automatic feature extraction.
• Can handle unstructured data (images, text, audio).
• Scalable with more layers and neurons.
Limitations
• Requires large amount of training data.
• Computationally expensive.
• Needs powerful hardware (e.g., GPUs).
• Training time can be long.
• Difficult to interpret (black-box model).
• Sensitive to hyperparameter selection.
• Risk of overfitting.
5. How Neural Networks Work?
Input Data
Forward Propagation (compute output)
Compare with Actual Output (calculate error)
Backward Propagation (update weights)
Repeat until satisfactory performance
During training, the network adjusts its internal weights to minimize the prediction error.
6. Activation Functions
• Introduce non-linearity into the model.
• Help the network learn complex patterns.
• Common activation functions:
Sigmoid
$\sigma(z) = \frac{1}{1+e^{-z}}$
ReLU
$f(z)$
$f(z) = \max(0, z)$
Tanh
$\tanh(z)$
Softmax (for multi-class)
$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$
9. Applications
• Image recognition (face recognition, object detection)
• Speech recognition (voice assistants)
• Natural language processing (translation, chatbots)
• Medical diagnosis (disease detection)
• Finance (fraud detection, credit scoring)
• Autonomous vehicles (lane detection, object recognition)
Neural Networks Learn from Data Today to Build a Smarter Tomorrow

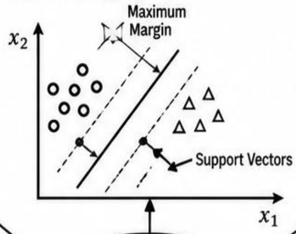
4.8 Support Vector Machines (SVM)

1. What is SVM?

- A supervised machine learning algorithm used for classification (and also regression – SVR).
- Finds the optimal hyperplane that separates different classes with the maximum margin.
- Works well in high-dimensional spaces.
- Effective for both linear and nonlinear data (using kernel trick).
- Widely used in real-world applications such as text classification, image classification, medical diagnosis, etc.

2. Key Idea – Optimal Hyperplane

- A hyperplane is a decision boundary that separates classes.
- SVM chooses the hyperplane with the maximum margin (maximum distance from nearest data points).
- Only a few training points, called support vectors, lie on the margin and determine the hyperplane.



3. Why SVM?

- Effective in high-dimensional spaces.
- Works well with small and medium-sized datasets.
- Uses only support vectors (memory efficient).
- Can model nonlinear relationships using kernels.
- Good generalization ability.

4. Types of SVM

- **Support Vector Classification (SVC)**
  – Used for classification problems (discrete class labels).
- **Support Vector Regression (SVR)**
  – Used for regression problems (continuous output).

Support Vector Machines (SVM)

Find the Best Boundary for Better Decisions

5. Kernel Trick (for Nonlinear Data)

- Transforms data into a higher-dimensional feature space where classes become linearly separable.
- Common kernel functions:
  - **Linear Kernel** : $K(x, x') = x \cdot x'$
  - **Polynomial Kernel** : $K(x, x') = (\gamma x \cdot x' + r)^d$
  - **RBF (Gaussian) Kernel** : $K(x, x') = \exp(-\gamma \|x - x'\|^2)$
  - **Sigmoid Kernel** : $K(x, x') = \tanh(\gamma x \cdot x' + r)$

6. Important Parameters

- **C (Cost Parameter)**
  - Controls the trade-off between maximizing the margin and minimizing classification errors.
  - Large C → low training error, higher risk of overfitting.
  - Small C → wider margin, better generalization.
- **Gamma (γ)**
  - Defines the influence of a single training example (for RBF kernel).
  - Small γ → smooth decision boundary, less overfitting.
  - Large γ → complex decision boundary, higher risk of overfitting.

7. How SVM Works? (Steps)

1. Prepare the training dataset (labeled data).
2. Represent data in feature space.
3. Find the optimal hyperplane (maximum margin).
4. Apply kernel function (if data is nonlinear).
5. Train the model using support vectors.
6. Predict the class label for new, unseen data.

8. Data Standardization

- SVM is a distance-based algorithm.
- Features should be standardized (same scale).
- Common techniques:
  - Z-Score Standardization (mean = 0, std = 1)
  - Min-Max Normalization (scale to [0,1] or [−1,1])

9. Advantages and Limitations

Advantages	Limitations
Works well in high-dimensional data.	Sensitive to parameter selection (C, γ, kernel).
Effective with small/medium datasets.	Computationally expensive for very large datasets.
Uses only support vectors (memory efficient).	Requires data standardization.
Good generalization ability.	Not suitable for noisy data with overlapping classes.
Can handle nonlinear data (using kernels).	

10. Real-World Applications

- Text classification (email spam)
- Image classification
- Medical diagnosis
- Fraud detection
- Face recognition
- Bioinformatics (gene classification)
- Document categorization

11. Key Takeaways

- Finds the best boundary with maximum margin.
- Uses kernel trick for nonlinear problems.
- Proper parameter tuning and data standardization are essential.
- Widely used in real-world classification and regression tasks.

SVM: Simple Idea – Powerful Results → Works Across Real-World Problems