

Code 672

Artificial Intelligence Assistant

NSQF LEVEL – 3



NATIONAL INSTITUTE OF OPEN SCHOOLING

(An autonomous organisation under MoE, Govt. of India)

A-24-25, Institutional Area, Sector-62, NOIDA-201309 (U.P.)

Website: www.nios.ac.in, Toll Free No. 18001809393

Printed on 70/80 GSM NIOS Water Mark Paper

© **National Institute of Open Schooling**

September, 2026 (..... copies)

Published by the Secretary, National Institute of Open Schooling, A-24/25, Institutional Area, Sector-62, NOIDA-201309 and
Printed at M/s

ADVISORY COMMITTEE

Chairman
NIOS, NOIDA

Director, Vocational
NIOS, NOIDA

CURRICULUM COMMITTEE

Dr. Deepak Shudhalwar
Prof. (CSE) & Head ICT
Dept. of Engineering & Technology
PSSCIVE, NCERT, Bhopal

Mr. Prakash Khanale
Professor & Head
Dept. of Compute Science
DSM College

Ms. Charu Agarwal
ACP (Deputy Director)
IT Dept. DoIT&C
Govt. of Rajasthan

Dr. Rajesh D.
Associate Prof. &
Chief Information and Security
Officer , Dept. of ICT & TD, CIET
NCERT

Dr. Arun Kumar Dubey
Associate Professor
Bharati Vidyapeeth College
of Engineering, New Delhi

Mr. Rejaul Karim Barbhuiya
Assistant Professor (Computer Science)
CIET, NCERT

Mr. Kumar Nilay
Vice Principal
NNLI, NVS Hqrs., NOIDA

Ms. Shweta Khurana
Sr. Director – Asia Pacific & Japan
Government Partnerships & Initiatives
International Government Affairs, Intel

Ms. Saloni Singhal
Sr. Program Manager
Intel Digital Readiness Programs

LESSON WRITERS

Mr. Anoop Singh Rawat
Intel AI for Youth Coach

Ms. Laisha Wadhwa
Lead ML Engineer
Lucidity Finance

Ms. Payal Patel
Program Manager
Jayso Foundation

Ms. Ambika Saxena
Intel AI for Youth Lead Coach

Mr. Kaiwalya Patil
Data Scientist at Shoden AI

EDITORS

Mr. Rejaul Karim Barbhuiya
Assistant Professor
(Computer Science)
CIET, NCERT

Mr. Anoop Singh Rawat
Intel AI for Youth Coach

Dr. Arun Kumar Dubey
Associate Professor
Bharati Vidyapeeth College
of Engineering, New Delhi

Ms. Ambika Saxena
Intel AI for Youth Lead Coach

Ms. Payal Patel
Program Manager
Jayso Foundation

Dr. Koushalya Barik,
Deputy Director
Vocational, NIOS

Dr. Madhavi Sharma
Former DD (Software)
IGNOU, New Delhi

Ms. Anchal Goswami
Former AO (ICT)
VED, NIOS

COURSE COORDINATOR

Mr. Bijay Kumar
SEO (ICT), VED
NIOS, NOIDA

GRAPHICS/DTP

M/s Sri Krishna Graphics
Delhi

How to use the Study Material

Congratulation! You have accepted the challenge to be a self-learner. NIOS is with you at every step and has developed the material in *Artificial Intelligence Assistant* with the help of a team of experts, keeping you in mind. A format supporting independent learning has been followed. If you follow the instructions given, then you will be able to get the best out of this material. The relevant icons used in the material will guide you. These icons have been explained below for your convenience.

Title: will give a clear indication of the contents within. Do read it.

Introduction: This will introduce you to the lesson linking it to the previous one.



Objectives: These are statements that explain what you are expected to learn from the lesson. The objectives will also help you to check What have you learnt after you have gone through the lesson. Do read them.



Notes: Each page carries empty space in the side margins, for you to write important points or make notes.



Intext Questions: Very short answer self check questions are asked after every section, the answers to which are given at the end of the lesson. These will help you to check your progress. Do solve them. Successful completion will allow you to decide whether to proceed further or go back and learn again.



What have you learnt: This is the summary of the main points of the lesson. It will help in recapitulation and revision. You are welcome to add your own points to it also.



Terminal Exercise: These are long and short questions that provide an opportunity to practice for a clear understanding of the whole topic.



Answers to Intext Questions: These will help you to know how correctly you have answered the questions.

A Word With You

Dear Learner,

Congratulations for having chosen the course **Artificial Intelligence Assistant**.

In today's interconnected world, having Computer knowledge is a basic requirement not only to stay connected but to solve problems efficiently. Nowadays all the services are delivered at our doorsteps because of advancement of technologies and emergence of new digital platforms. As a learner, you should acquire and understand technologies to get the benefit of these services.

The course **Artificial Intelligence Assistant** has been developed with the valuable involvement of experts from **Intel India**, including experienced lesson writers and editors keeping in view of the demand in the IT industry. The curriculum has been developed with support from **Intel India** to ensure that it aligns with current industry standards and emerging AI technologies. It is **NSQF** aligned **at Level 3**, ensuring that students gain job-ready competencies and nationally recognized skills. The **Artificial Intelligence Assistant** qualification will cover the fundamentals of Python programming and libraries like Numpy and pandas used for data analysis. The course will also cover Visualization with Matplotlib. The course will lay stress on developing programming skills by providing practical exposure to the aspiring Python developers and also introduce to the concepts of Machine Learning. Participants from any background can develop the skills needed to become an AI Assistant.

We have ensured that the course contents are presented in a simple and straightforward manner. We are confident that you will enjoy learning this course through the Open and Distance learning system. We have also prepared a practical manual which will help you to do more practice. We hope you will find this material interesting, informative and useful.

We wish you all the very best for a bright and successful future.

Course Team
National Institute of Open Schooling (NIOS)

CONTENTS

1.	Introduction to Artificial Intelligence	1
2.	Data Literacy and AI Project Cycle	16
3.	Domains of AI	49
4.	AI for Everyday Life and Workplaces	63
5.	Logic Building and Basic Programming Concepts	101
6.	Applied Data Handling and Visualization	148
7.	Introduction to Machine Learning	181
8.	No-Code and Low-Code AI Tools	191
9.	Generative AI and Open AI Tools	200
10.	AI Systems and Framework	210
11.	Natural Language Processing	222
12.	Computer Vision	254
13.	Responsible and Ethical AI	265
14.	Employability and Future Skills	277
15.	Project Cycle	295



Notes

1

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

In this lesson, you will explore the evolution of Artificial Intelligence (AI) – tracing its history, understanding the foundations of human and artificial intelligence and distinguishing AI from automation. This lesson also highlights the diverse applications of AI in everyday life and helps you build a broader perspective on the growing AI ecosystem in India.



OBJECTIVES

After going through this lesson, you will be able to:

- explain the evolution of AI;
- define Human Intelligence;
- describe and experience Artificial Intelligence and its applications;
- describe how AI has become an integral part of our lives today;
- explore different types of AI;
- describe AI ecosystem in India.

1.1 EVOLUTION OF AI

Various stages during the evolution of AI are generally classified as below:

1.1.1 The Dream Begins (Before 1950 → 1970s)

Long before computers, people wondered if machines could think like humans. This idea grew stronger when Alan Turing introduced the Turing Test in 1950 to



Notes

Introduction to Artificial Intelligence

check if a machine could act like a human. In 1956, at the Dartmouth Conference, researchers first used the term “Artificial Intelligence,” marking the birth of AI. During the 1960s and early 1970s, scientists created early AI programs that could play simple games, solve math problems, and imitate basic human thinking, leading many to believe that human-level AI was close. (See Fig. 1.1)

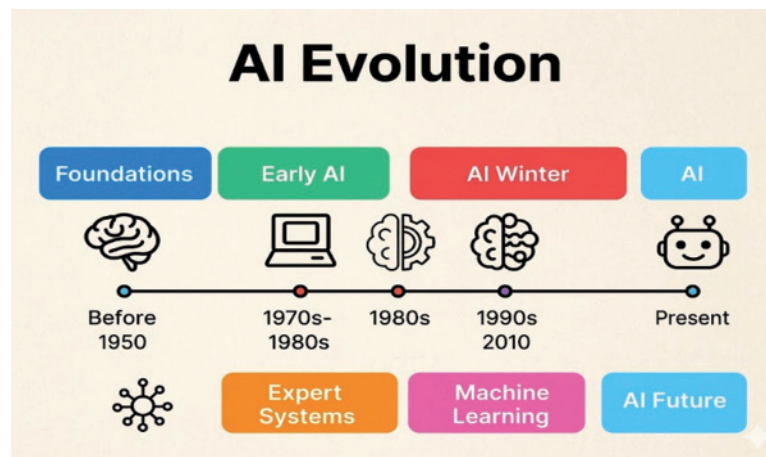


Fig. 1.1: Stages of evolution of AI

1.1.2 Winters, Comebacks, and New Hope (1970s → 2010)

When early AI systems didn't perform as well as researchers promised, interest and funding dropped sharply causing the first AI Winter – a long period where AI research stalled due to a lack of financial support and unmet expectations.

AI made a comeback in the 1980s with the invention of Expert Systems. These were computer programs designed to solve complex problems by mimicking the decision-making of a human expert, using a strict set of “if-then” rules. However, they soon became too slow, rigid, and expensive to update, leading to the Second AI Winter.

In the 1990s, a breakthrough called Machine Learning revived the field. Instead of humans having to program every single rule step-by-step, Machine Learning allowed computers to learn and improve automatically by finding patterns in large amounts of data. A big moment for this new era of AI came in 1997 when IBM's computer, Deep Blue, beat the world chess champion Garry Kasparov. In the 2000s, the rise of the internet and faster computers provided the massive amounts of data AI needed to grow, especially in speech and image recognition.

1.1.3 The Rise of Modern AI (2010 → Present → Future)

After 2012, AI grew incredibly fast as deep learning (a more advanced type of machine learning) started beating humans in tasks like image recognition. This



Notes

break throughs led to self-driving cars, voice assistants, and smarter robots. In 2016, Google’s AlphaGo won against a world champion in the complex board game Go. Soon after, Generative AI models like GPT (Generative Pre-trained Transformer) and BERT(Bidirectional Encoder Representations from Transformers) appeared. These systems don’t just analyze data; they create new content, such as writing essays, generating art, and understanding conversational language. Today, AI can make images, videos, music, and even help doctors diagnose diseases. The future is one where humans and AI work together to learn, create, and solve global problems.

Time Period	AI Era	Key Milestones & Concepts	What it meant for AI
Before 1950	Foundations	The Turing Test (1950)	The earliest ideas of machines thinking like humans take shape.
1950s – 1970s	Early AI	Dartmouth Conference (1956)	The term “Artificial Intelligence” is officially coined. Machines learn to play basic games and solve math.
1970s – 1989s	Winters & Comebacks	AI Winter, Expert Systems	Funding freezes (AI Winter) when early tech falls short. AI briefly bounces back using rule-based Expert Systems before facing a second Winter.
1990s – 2010	New Hope	Machine Learning, Deep Blue	AI is completely revived by Machine Learning (giving computers data to learn on their own). Deep Blue defeats a human chess champion (1997).
2010 – Present	Modern AI	Deep Learning, Generative AI	High-speed internet and massive data make AI powerful. Generative AI emerges, capable of creating new text, art, video, and music.



INTEXT QUESTIONS 1.1

A. Multiple Choice Questions

1. When was the term “Artificial Intelligence” first used?
- (a) 1940 (b) 1956 (c) 1975 (d) 1990



Notes

Introduction to Artificial Intelligence

2. Who introduced the Turing Test to check machine intelligence?
(a) Elon Musk (b) Geoffrey Hinton
(c) Alan Turing (d) Andrew Ng
3. What do we call the period when AI research slowed down due to lack of progress and funding?
(a) AI Growth (b) AI Winter
(c) AI Summer (d) AI Revolution
4. Which decade saw the rise of Expert Systems?
(a) 1960s (b) 1980s (c) 2000s (d) 2020s
5. Which breakthrough year marked the Deep Learning Revolution?
(a) 2001 (b) 2012 (c) 1990 (d) 2023

B. True or False

1. State True or False: AI started as a formal field of study in 1956.
2. State True or False: Expert systems could learn from data automatically.
3. State True or False: Modern Generative AI can create text, images, and videos.

1.2 UNDERSTANDING HUMAN INTELLIGENCE

Human intelligence is the natural ability of people to learn from their experiences, understand the world around them, solve problems, and make decisions. It helps us think clearly, use our memory, observe with our senses, and reason through situations. Because of this, we can adapt, try new ideas, and respond to new challenges in smart ways.

Here are some of the traits that make us intelligent –

Learning	Memory	Problem-solving	Reasoning
Creativity	Adaptability	Communication	Emotions & Empathy

1.3 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is made of two terms- Artificial and Intelligence. The term ‘Artificial’ does not need any explanation as it is self-explanatory. Something that does not exist in nature and is man-made and does not occur naturally.

Introduction to Artificial Intelligence

Just as Human Intelligence is our mental ability to make decisions, solve problems, and learn new things, Artificial intelligence (AI) is a machine's ability to make decisions, solve problems, and learn. Artificial Intelligence (AI) refers to the simulation of human intelligence in machines, enabling them to perform tasks that typically require human intelligence. So,

Artificial Intelligence (AI) is when machines or computers are designed to think and learn in ways that are similar to humans. AI helps them understand information, solve problems, make decisions, or do tasks that usually need human intelligence – like recognizing pictures, answering questions, or recommending what to watch.

More explanation required

(Some AI systems are designed to accept inputs from human and the feedback and based on it improve their performance each time. In nut shell, we can observe the following major components in any AI system as below:

Learn →

Adapt →

Plan →

Make decisions →

In fact AI is a broad field which uses several techniques and approaches to create intelligent systems. Few systems are trained to do certain tasks while others are designed to learn and adjust to new situations /data so that they can perform a variety of tasks and do decision making.)

1.4 APPLICATIONS OF AI

Now that you understand what AI is, let's look at where we see it in real life. AI is not just in big research labs – it is all around us, quietly helping us every day. Here are some simple examples that you can easily relate to:

In Mobile Phones

- **Face Unlock:** Computer vision AI analyzes key facial features and patterns, securely unlocking your device in real-time.
- **Keyboard Suggestions:** Prediction models analyze your typing history and common language patterns, suggesting the most likely next word.



Notes



Notes

In Entertainment

- **YouTube & Netflix Recommendations:** Machine learning algorithms analyze your past viewing habits, comparing them to other users' preferences to suggest content you'll likely enjoy.
- **Music Apps:** Recommendation engines learn your listening tastes over time, curating personalized playlists and suggesting new songs and artists.

In Education

- **Learning Apps:** AI adaptive learning systems track your progress, identifying areas of strength and weakness to dynamically suggest practice questions tailored to your skill level.
- **Doubt-solving Chatbots:** Natural language processing (NLP) chatbots interpret and respond to student questions immediately, offering 24/7 basic academic support.

In Everyday Travel

- **Google Maps:** AI traffic prediction models analyze real-time and historical data to optimize routes, suggesting the fastest path while dynamically adjusting for congestion.
- **Ride Apps:** Optimization algorithms instantly match available drivers with nearby passengers, while pricing models calculate fares based on demand and traffic.

In Banking & Online Payments

- **Fraud Detection:** Anomaly detection systems continuously monitor transaction patterns for unusual activity, instantly flagging potentially fraudulent purchases to protect your account.
- **Chat Support:** NLP-powered chatbots efficiently handle routine customer service queries, freeing up human agents for more complex issues.

In Shopping

- **Online Store Recommendations:** AI recommendation systems analyze your browsing and purchase history, personalizing your shopping experience by suggesting relevant products.
- **Virtual Try-ons:** Computer vision technology accurately overlays virtual items onto your image, allowing you to see how products look on you before purchasing.



In Healthcare

- **Disease Detection:** Computer vision and deep learning models analyze medical images (like X-rays or MRIs) to identify subtle patterns, assisting doctors in early disease detection.
- **Health Apps:** Algorithms process data from wearable devices to learn your activity patterns, providing personalized health insights and goal tracking.

In Homes

- **Smart Speakers:** Speech recognition and NLP technologies transcribe and interpret your spoken commands, enabling hands-free control of your devices.
- **Smart Appliances:** Sensor data integrated with AI logic allows appliances to learn your habits and automatically adjust settings like lighting or temperature based on perceived presence.



INTEXT QUESTIONS 1.2

1. What is artificial intelligence (AI)?
 - (a) The study of natural intelligence exhibited by humans and animals.
 - (b) A branch of computer science that deals with programming languages
 - (c) The simulation of human intelligence processes by machines, especially computer systems
 - (d) None of the above
2. Which of the following is an example of an AI-driven solution in everyday life?
 - (a) Weather forecasting
 - (b) Using a Google Assistant
 - (c) Using Bar-code scanner
 - (d) All of the above
3. What is the primary goal of AI-driven solutions in healthcare?
 - (a) To replace doctors and nurses
 - (b) To improve patient outcomes and streamline healthcare processes
 - (c) To increase healthcare costs
 - (d) To make healthcare less accessible



Notes

4. What do recommendation system use AI for?
 - (a) Identifying potential risks in financial markets.
 - (b) Recommending products or content based on user preferences.
 - (c) Analysing medical images for disease diagnosis.
 - (d) Controlling autonomous vehicles
5. What kind of information does Google Maps provide when giving directions?
 - (a) Name of the nearest restaurant.
 - (b) Traffic data and shortest travel path.
 - (c) The weather forecast.
 - (d) The nearest gas station
6. Which of the following statements about AI-driven solutions is true?
 - (a) AI solutions can only perform tasks that humans are already capable of doing.
 - (b) AI solutions can analyze large datasets and identify patterns that may not be obvious to humans.
 - (c) AI solutions are always completely error free and do not require human supervision.
 - (d) AI solutions cannot adapt or learn from new data or experiences

1.5 TYPES OF AI

We have learned that AI is when machines act smart – like understanding, learning, and making decisions. But AI has different parts within it and they are –

- **Artificial Intelligence** is the overall idea of making machines act smart. If a computer or machine can understand, decide, or solve a problem the way a human would, it is using AI.

All intelligent applications (whether primitive or complex) fall under the term AI.

- **Machine Learning** is a subset of AI. Here, machines learn from data instead of being given step-by-step instructions. Just like you learn from examples, a machine also improves its performance as it sees more data.

Introduction to Artificial Intelligence

An example is YouTube recommending videos based on what you watch.

- **Deep Learning** is a special type of Machine Learning. It uses computer models called neural networks, which are inspired by how the human brain works. Deep Learning helps machines handle very complex tasks, like recognising faces, driving cars, or understanding speech.

Some examples are a phone identifying objects in a photo, or Google Translate converting languages instantly.

- **Generative AI** is a newer branch of AI where machines don't just understand or learn – they create new things. Generative AI uses patterns learned from huge amounts of data and then creates something original. It can generate text, audio, video, images, and code.

Some examples are ChatGPT writing a story, an app creating a cartoon from your words, or AI making a new tune.



INTEXT QUESTIONS 1.3

Match the Following:

- | | |
|----------------------------|--|
| 1. Artificial Intelligence | A. Machines that create new content like text, images, or music. |
| 2. Machine Learning | B. The big field where machines act smart. |
| 3. Deep Learning | C. A part of AI where machines learn from data. |
| 4. Generative AI | D. A deeper, more advanced type of ML using neural networks. |

1.6 AI ECOSYSTEM IN INDIA

Artificial Intelligence, or AI, is growing quickly in India and becoming a part of our daily lives – whether we see it or not. From checking weather updates to getting instant answers from chatbots, AI is quietly supporting our routines. India's AI ecosystem includes government initiatives, research labs, startups, big companies, and talented students who are learning and building AI solutions. Together, they are helping India use AI to solve real problems in agriculture, health, education, and public services.

1.6.1 Brief History of AI in India

India's interest in AI started mainly in universities during the 1980s and 1990s. Institutes like IISc Bengaluru, ISI Kolkata, and IITs began teaching and doing



Notes



research on early AI topics such as pattern recognition and simple expert systems. However, at that time, computers were slow and data was limited, so AI stayed inside classrooms and research labs.

Things changed in the 2000s when India's IT industry exploded. Companies like TCS, Infosys, and Wipro built strong software and data teams, which became the foundation for modern AI. The wider use of the internet, smartphones, and digital services helped India produce huge amounts of data – something AI needs to learn.

The real turning point came after 2015, when the Digital India mission accelerated online payments, e-governance, and digital identity (Aadhaar). In 2018, NITI Aayog launched the National AI Strategy – #AIforAll, making AI a national priority for development.

1.6.2 India's Progress in AI

Today, India is one of the fastest-growing AI markets in the world. The government, industry, and education system are working together to build a strong AI ecosystem.

1.6.3 Government Support

The government is moving AI from a niche technology to core national infrastructure. A major turning point was the Cabinet's approval of the comprehensive **IndiaAI Mission** in March 2024. It was backed by a massive budget outlay of over 10,370 crore guided by the vision of "*Making AI in India and Making AI Work for India*". The mission focuses on democratizing technology. For example, the government has successfully onboarded over 38,000 Graphics Processing Units (GPUs) to provide startups and researchers with high-speed, subsidized computing power. To build future-ready talent, AI-based subjects are now heavily integrated into school curriculums.

1.6.4 Research and Innovation

Institutions are developing AI models specifically tailored for Indian needs, particularly in medical diagnosis, agriculture, and robotics. A massive focus is currently on "Sovereign AI" – building indigenous Large Multimodal Models (LMMs) that understand India's diverse culture and multilingual data. A prime example is **Bhashini (BHASHaInterface for India)**, a government-led language translation engine developed by MeitY. Bhashini recently surpassed 4 billion language inferences, supporting voice and text (i.e. real time translation, voice



recognition, text to speech) in 22 official Indian regional languages to ensure that AI breaks down digital and literacy barriers for citizens in rural areas. BharatGPT is another indigenous large language model (LLM) and conversational AI platform alternate to ChatGPT.

1.6.5 Growing Startups

India has emerged as a global powerhouse for AI entrepreneurship. According to recent NASSCOM reports, the Indian tech startup ecosystem has surged past 32,000 companies. The Generative AI sector specifically is experiencing explosive momentum, registering a 3.6X growth in a single year to rank among the top global economies for GenAI investments. Startups are building solutions for farming, finance, education, and e-commerce. While Bengaluru, Delhi-NCR, Mumbai, and Hyderabad remain major AI hubs, over 40% of new tech startups are now emerging from Tier-2 and Tier-3 cities, proving that AI innovation is spreading across the entire country.

1.7 IMPACT OF AI ACROSS INDUSTRIES

The impact of AI in various sectors can be experienced as below.

1.7.1 Agriculture

Farmers often struggle with unpredictable weather, pests, and crop diseases. AI tools from startups like CropIn and Fasal help farmers by analysing weather patterns and crop images. For example, if a disease is spreading in tomato crops, the app alerts farmers so they can act early. Additionally, the government's AI chatbot, Kisan e-Mitra, helps farmers get instant voice responses about agricultural schemes in their local languages. Drones powered by computer vision are also being used for "precision farming" – scanning fields from above to spray fertilizers and pesticides only exactly where they are needed, reducing waste and cost.

1.7.2 Healthcare

AI tools help doctors detect diseases faster. Few companies use AI to screen for early breast cancer, while Qure.ai reads X-rays quickly to identify lung infections. During COVID-19, AI models helped predict hotspots and guide hospital planning. Beyond diagnostics, hospitals are using predictive AI algorithms to calculate a patient's risk of heart disease based on their lifestyle and medical history. Organizations like Wadhvani AI are also deploying tools that allow frontline health workers to track and manage diseases like Tuberculosis in rural areas using just a basic smartphone.



Notes

1.7.3 Education

Platforms such as DIKSHA and Bhashini use AI to personalize learning and break down language barriers. If a student is weak in fractions, the app gives easier explanations and practice questions. Teachers also receive AI-generated reports to understand students' needs better. Few other platforms use AI-driven 3D simulations to explain complex science concepts, while apps use computer vision – allowing students to simply click a photo of a math problem to instantly receive an AI-generated video solution.

1.7.4 Smart Cities and Governance

Cities like Delhi, Bengaluru, and Hyderabad use AI cameras to monitor traffic, detect violations, and manage congestion. AI also helps state governments detect fraud in land records or predict water leakages. In the public transport sector, Indian Railways is implementing AI for the predictive maintenance of tracks and trains, as well as tracking crowd density at major stations. Furthermore, cities like Indore are using AI-based route optimization for smart waste collection trucks, which reduces fuel consumption and keeps the city cleaner.

1.7.5 Finance and Banking

Banks use AI to prevent fraud and help customers 24/7. For example, SBI's AI chatbot SIA answers questions instantly. AI models also help decide if a person qualifies for a loan by analysing financial patterns. Similarly, virtual assistants of few other banks handle millions of customers queries daily, from checking balances to paying bills. Behind the scenes, payment gateways like Razorpay use advanced machine learning models to analyze thousands of transactions per second, instantly blocking fraudulent payments before the money leaves your account.

1.8 INDIA'S ROAD AHEAD IN AI

India's goal is to build AI solutions that are affordable, inclusive, and available in all major Indian languages. The future includes:

- Making AI tools for rural areas, especially in agriculture and health
- Training more students and teachers in AI
- Supporting startups building India-focused AI models
- Building safe and responsible AI that protects privacy

Introduction to Artificial Intelligence

- Creating Indigenous AI models that understand Indian languages, cultures, and datasets with strong digital infrastructure (like UPI, Aadhaar, and DigiLocker).
- India is well-positioned to become a global leader in accessible and socially impactful AI.



WHAT HAVE YOU LEARNT

In this lesson, you have come to know the various stages of evolution of AI over a period of time. Also, you have learnt the concept of AI and their applications in various fields.

- AI began in 1956 and has evolved through breakthroughs, setbacks, and major innovations like Deep Learning and Generative AI.
- Human intelligence helps us learn, think, adapt, and solve problems using memory, senses, and reasoning.
- Artificial Intelligence (AI) is when machines act smart – like understanding, learning, and making decisions.
- Types of AI: Machine Learning learns from data, Deep Learning uses neural networks, and Generative AI creates new content.
- AI is everywhere – phones, education, travel, banking, shopping, healthcare, and homes.
- India's AI ecosystem is rapidly growing with government support, startups, research institutions, and Digital India initiatives.
- AI impacts major industries like agriculture, healthcare, finance, education, and smart cities, and India aims to build safe, inclusive, multilingual AI for the future.



TERMINAL QUESTIONS

Short-Answer Questions:

1. What is Artificial Intelligence?
2. What is Machine Learning?



Notes



Notes

Introduction to Artificial Intelligence

3. What makes human intelligence unique?
4. What is Generative AI?
5. How does AI help in healthcare?
6. What does the term AI ecosystem mean in the Indian context?
7. Why was the year 2018 important for AI in India?
8. Name any two sectors where AI is widely used in India.
9. What role do startups play in India's AI growth?

Long-Answer Questions

1. Explain the evolution of AI from the 1950s to the present.
2. How is human intelligence different from artificial intelligence?
3. Describe any four applications of AI in everyday life with examples.
4. What are Machine Learning, Deep Learning, and Generative AI? Explain with examples.
5. Explain the AI ecosystem in India and how it is growing.

Skill Based Questions

1. A phone identifies your face to unlock. Which type of AI is being used and why?
2. If an app creates a cartoon from your description, which branch of AI does this involve?
3. How can AI help improve education for students in India?
4. Why do you think AI is becoming important in industries like healthcare and agriculture?

Activity: Is it AI or Just a Normal Machine?

Objective: Let's test what we've learned! Remember, AI learns and makes decisions, while a normal machine just follows strict rules.

Instructions: Read the everyday situations below. Put a tick if you think it uses Artificial Intelligence, and a cross if it is just a regular automated machine.

1. A toaster pops up your bread after exactly 2 minutes. (Yes/No)



2. YouTube suggesting a comedy video because you watched one yesterday. (Yes/No)
3. A basic calculator solving 54×23 . (Yes/No)
4. Your phone's keyboard guessing the next word you want to type. (Yes/No)
5. An automatic sliding door opening when you step on the sensor mat. (Yes/No)
6. A smart speaker (like Alexa) understands your voice and answers your question. (Yes/No)



ANSWERS TO INTEXT QUESTIONS

1.1

- A. 1. (b) 2. (c) 3. (b) 4. (b) 5. (b)
- B. 1. False 2. False 3. True

1.2

1. (b) 2. (d) 3. (b) 4. (b) 5. (b) 6. (b)

1.3

1. B 2. C 3. D 4. A

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- define Artificial Intelligence and how it differs from Human Intelligence
- usage of AI applications in everyday life
- describe various types of AI
- explore the AI ecosystem in India



DATA LITERACY AND AI PROJECT CYCLE

In the previous lesson, you have learnt an overview of Artificial Intelligence. As data is an integral part in AI systems, in this lesson, you will be introduced to Data and its fundamentals – what it is, how it is collected, ways to interpret and process data, its importance and relevance. The AI systems require data in large quantities so that the AI model can be enabled to understand and learn data patterns. It enables AI models to make meaningful decisions and make predictions. You will also explore various AI domains – Statistical Data, Computer Vision, and Natural Language Processing based on the type of data used. Then, you will be introduced to the AI Project Cycle framework that is widely used for the development of AI solutions. With the help of the AI Project Cycle, you will be able to gain a better understanding of how to collect and use data, how to select the correct AI model, and how to train an AI model for solution development.



OBJECTIVES

After learning through this lesson, you will be able to:

- define data literacy and explain its importance with a real-world example;
- relate to the impact created by data literacy in everyday life;
- develop awareness about personal data, data privacy and data security;
- know the concept of Data Processing;
- explain the concept of an AI project cycle and its significance in completing AI projects effectively;
- identify the different stages involved in the AI project cycle and their key tasks.



2.1 INTRODUCTION TO DATA AND DATA LITERACY

In our daily life, we see and use information all the time – numbers, text, images, sounds, and even videos. In the world of computers and Artificial Intelligence, all this information is called Data.

2.1.1 What is Data?

Data is any raw information that is collected, stored, or used to understand something or decide.

It can be simple facts, observations, or details about a person, object, event, or activity. Examples of Data can be –

- Your name, age, and class
- Your marks in different subjects
- A photo taken on your phone
- A recording of your voice message
- The temperature outside
- How many steps you walked today

Data is the building block of information. It helps both humans and machines in understanding patterns, making decisions, learning new information, and solving problems.

2.1.2 Data Literacy

Data literacy means knowing critically about what the data shows, get useful information from the data, and know-how to apply the appropriate data for specific purposes. The following diagram helps us in understanding the true meaning of Data Literacy as shown in Fig. 2.1.

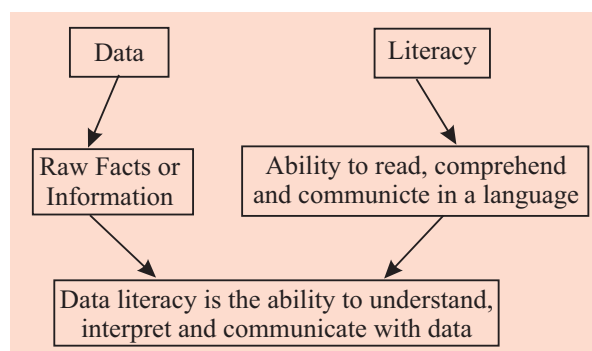


Fig. 2.1: Data Literacy



Notes

Let us look at the Data Pyramid to understand how raw information gets processed to become meaningful:

There are four levels in a Data Pyramid, and they are:

- A. **Data:** Information at the bottom level is available in raw format. In this form, data is not very useful.
- B. **Information:** This is the second bottom level in the pyramid. Here, data is processed in a way that helps us get meaningful information about the world – answering basic questions like who, what, when and where.
- C. **Knowledge:** The third level of this pyramid leads to knowledge of how things happen in the world.
- D. **Wisdom:** The top most level of the Pyramid has information in the best processed format, helping us understand why things happen in a particular way.

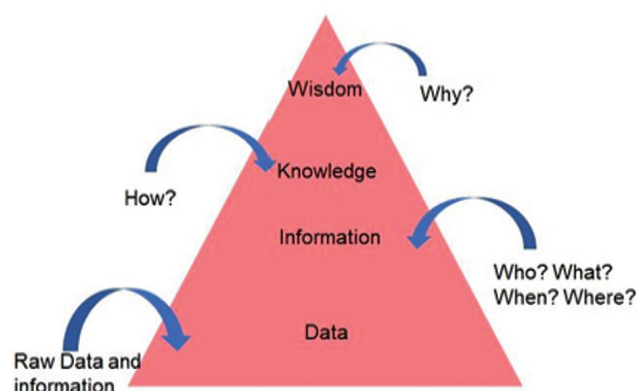


Fig. 2.2: Data Pyramid

2.1.3 Importance of Data

In today's world, Artificial Intelligence plays a major role in everything around us – mobile phones, shopping apps, healthcare, education and even transportation. But AI cannot work on its own. It needs something very important to learn and make decisions. That “something” is data.

How does AI become intelligent?

AI becomes intelligent by learning from the data it is given. Just like students learn from examples and practice, AI studies large amounts of information – such as images, Text, audio/video, or numbers – to find patterns and understand how things work. It is trained using special algorithms that guide it to make predictions, check if they are correct, and improve when it makes mistakes.

Over time, with more training and more data, the AI system becomes better at recognising objects, understanding language, solving problems, or making decisions. Modern techniques like deep learning allow AI to learn very complex patterns,

making it capable of tasks such as face recognition, language translation, and even creating new content. In simple words, AI becomes intelligent by learning, improving, and practising – just the way humans do.

Why is Data important for AI?

Data helps AI learn patterns in data that further helps AI models in making meaningful decisions and predictions. Once an AI model learns the patterns exhibited in the data it helps in making better decisions. AI uses your data to understand your likes and preferences. For example, Online stores use your search data to recommend products you might like. Some more examples in real -life where AI model learns from data are:

- (a) Doctors use AI tools to find patterns in medical images (like X-rays).
- (b) Farmers use AI to detect early signs of crop disease from image data.
- (c) YouTube suggests videos based on what you watch.
- (d) Music apps create playlists based on your listening habits.

AI grows stronger with continuous data.

Therefore, in the 21st century, it is widely accepted that “Data is the new Gold”.



INTEXT QUESTIONS 2.1

Multiple Choice Questions:

1. What is data?
 - (a) Only numbers
 - (b) Raw information collected for a purpose
 - (c) Finished reports
 - (d) Only words
2. Which of the following can be data?
 - (a) Text
 - (b) Images
 - (c) Audio
 - (d) All of the above



Notes



Notes

Data Literacy and AI Project Cycle

3. Data literacy means the ability to:
 - (a) Cook food using data
 - (b) Read, understand, and use data
 - (c) Decorate charts
 - (d) Speak different languages
4. Which level of the Data Pyramid helps us understand 'Why' things happen in a certain way?
 - (a) Data
 - (b) Information
 - (c) Knowledge
 - (d) Wisdom
5. Why is Data important?
 - (a) It confuses people
 - (b) It helps us understand patterns and make decisions
 - (c) It is only used for decorations
 - (d) It makes computers slow
6. What helps AI learn patterns from examples?
 - (a) Algorithms
 - (b) magic
 - (c) Luck
 - (d) batteries
7. More data helps AI in?
 - (a) Becoming worse
 - (b) Forget information
 - (c) Make better predictions
 - (d) Stop learning
8. Which of the following is FALSE?
 - (a) Data helps AI spot patterns
 - (b) Data helps AI make better decisions
 - (c) Data makes AI personalised
 - (d) Data is the new Silver

2.2 CATEGORIZING DATA

Data exist in many forms such as Structured data i.e. numbers in spreadsheet and unstructured data i.e. images, audio/video. To help computers and AI understand the world, we organise data into different types. Data can be broadly classified into two types –

- Quantitative Data
- Qualitative Data

Let us look at the difference between Qualitative and Quantitative data formats –



Quantitative Data	Qualitative Data
- Comprises of quantifiable information – something that can be counted - Quantitative Data can be depicted through numerical or alpha-numeric values - For example – Exam scores, attendance, temperature, and more.	- Comprises of features or traits – something that defines the qualities of a dataset - Qualitative Data can be depicted through words or phrases - For example – Image features, Emotions, and more.

Now, based on the type of information available, data can exist in various forms, namely –

1. Numerical Data – 1, 13.45, -0.234...
2. Categorical Data – Good, Bad, Sad...
3. Sensor Readings – 0, 1...
4. Image Data – Cats, Dogs, Birds...
5. Video Data – Dance performance, Yoga poses...
6. Textual Data – Novels, Books, Articles...
7. Audio Data – Songs, Noises, Sounds...



INTEXT QUESTIONS 2.2

Match the Following:

- | | |
|----------------------|--------------------------------------|
| 1. Quantitative Data | (a) 1, 13.45, -0.234... |
| | (b) Good, Bad, Sad... |
| | (c) 0, 1... |
| 2. Qualitative Data | (d) Dance performance, Yoga poses... |
| | (e) Songs, Noises, Sounds... |
| | (f) Novels, Books, Articles... |

2.3 DATA COLLECTION

Data collection or acquisition is the process of gathering the right kind of data so that AI systems can learn effectively. In modern AI, data can come from many sources, and sometimes it needs to be improved or even created. Now, the data can



Notes

be collected through various sources and the process of collecting or acquiring data involves searching for datasets that may be suitable for training AI models. There are 3 ways to acquire/collect data as shown in Fig. 2.3.

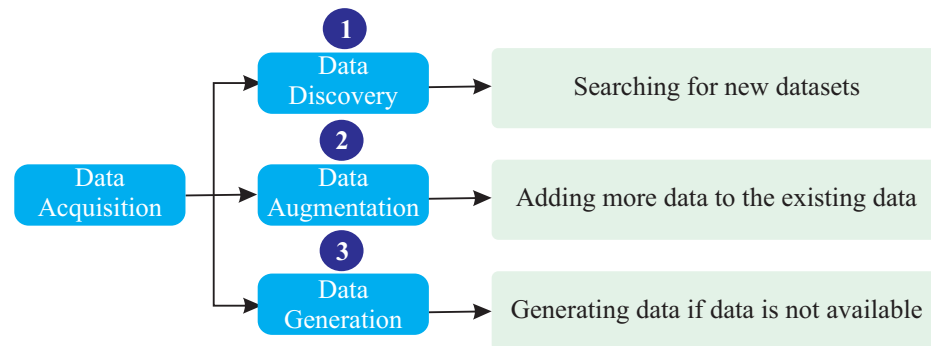


Fig. 2.3: Data Collection Methods

1. **Data Discovery:** process of finding and collecting existing data from different sources. In this step, we look for useful data that is already available and can help solve the problem. For example - Finding an online dataset of animal images to train an AI model that identifies cats and dogs.
2. **Data Augmentation:** improving or increasing the amount of data by making small changes to existing data. This step is used when we have limited data and need more examples to train AI systems better. Common augmentation methods are:
 - (a) Flipping or rotating images
 - (b) Changing brightness or colour
 - (c) Adding small noise to audio
 - (d) Rewriting text in different ways
3. **Data Generation:** creation of completely new data using tools or AI techniques. This is helpful when the required data does not exist or is difficult to collect. Data can be generated by:
 - (a) Using AI models to generate text or images
 - (b) Simulations (like creating fake traffic scenes for self-driving car training)
 - (c) Synthetic data generators

For example, using an AI tool to generate new images of handwritten numbers for training a handwriting recognition model.



Based on where the data comes from, it is classified into Primary Data and Secondary Data. Understanding the difference between these two helps us choose the right information for our purpose.

1. **Primary Data:** Primary data is the data collected directly by the researcher for the first time. It is original, fresh, and collected specifically for the problem or project at hand. It can be collected through:

- Surveys and questionnaires
- Interviews
- Observations
- Experiments
- Taking photos or recordings yourself
- Fingerprints

Examples of primary data can be:

- A student counting how many students prefer online classes
- Recording your own voice samples for a speech-recognition project
- A teacher collecting feedback directly from students

2. **Secondary Data:** Secondary data is the data that has already been collected, used, or published by someone else. Researchers use it when they do not need to collect new data or when the information they need already exists. It can be collected through:

- Books and textbooks
- Websites and articles
- Government reports
- Research papers
- Online databases
- Newspapers and magazines

Examples of secondary data can be:

- Using population data from a government website
- Checking weather information from the IMD website
- Reading research papers to study disease trends



Notes

Data Literacy and AI Project Cycle



Fig. 2.4: Example of sources of secondary data

Best practices for Data collection

To make sure the data collected is useful, correct, and reliable, we must follow certain best practices. These help ensure that the information we gather can be trusted and used effectively for analysis or training AI systems.

Let us look at the difference between good data and bad data:

Good Data	Bad Data
- Information is well structured - It is accurate - It is consistent - It is clearly presented - Contains information that is relevant to our needs	- Information is scattered - Contains a lot of incorrect values - Contains missing or duplicate values - It is poorly presented - Contains information which is not relevant to our needs

Data with high usability will be easier to work with an AI model. But the all-important question is - How do we decide the usability of data? There are three major factors that decide the usability of data:

1. Structure

- It provides uniformity to the data
- It makes data more presentable, and easier to understand
- Good structure increases the usability of data
- Example: Numbers arranged in ascending order



Notes

Purchase ID	Last name	First name	Birthday	Country
1	Davidson	Michael	04/03/1986	United States
2	Vito	Jim	09/01/1994	United Kingdom
3	Johnson	Tom	23/08/1972	France
4	Lewis	Peter	18/10/1979	Germany
5	Koenig	Edward	13/05/1983	Argentina
6	Preston	Jack	16/06/1991	United States
7	Smith	David	11/03/1965	Canada
8	Brown	Luis	03/09/1997	Australia
9	Miller	Thomas	07/01/1980	Germany

Muskan lives in Delhi. She was born on 04/10/1996.

Rekha lives in Uttar Pradesh. She was born on 09/01/1990.

Atharv lives in Odisha. He was born on 22/04/2000.

Spreadsheet Good structure

Data is stored in a sheet with the details of each individual stored according to a set of rules.

Text document Poor structure

Data is stored in a text document with no set of organizing rules.

Fig. 2.5: Structured and Unstructured Data

2. Cleanliness

- (a) It provides consistency to the data
- (b) Clean data is more useable and increases performance of AI models
- (c) Clean data:
 - (i) Has no missing values
 - (ii) Has no incorrect, duplicate or corrupt data



Fig. 2.6: Removal of Duplicate Values

3. Accuracy

- (a) Accuracy measures how true the data is
- (b) Accurate data contains actual, real-world values
- (c) It provides reliability to the data



Notes

- (d) Accurate data increases the accuracy of AI models
- (e) Accurate data has values closer to the real values or has no error

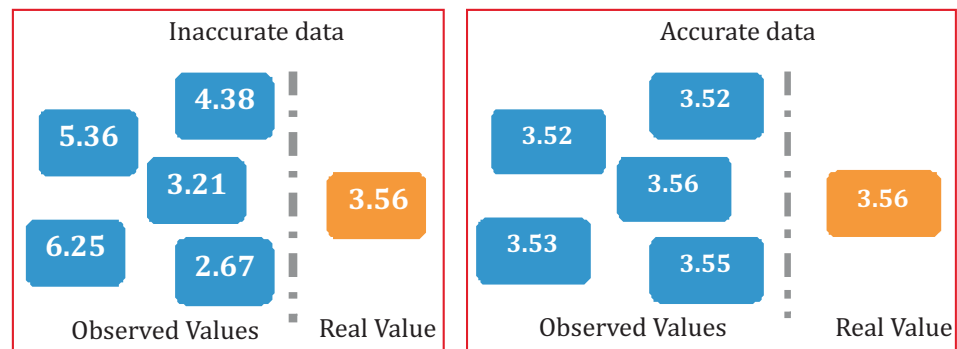


Fig. 2.7: Data Accuracy



INTEXT QUESTIONS 2.3

Multiple Choice Questions:

1. Which of the following can be a source of Data Discovery?
 - (a) Public Datasets
 - (b) Government Websites
 - (c) Research Reports
 - (d) All of the above
2. What does Data Augmentation do?
 - (a) Reduces the amount of data
 - (b) Improves or increases data by making small changes
 - (c) Deletes duplicate data
 - (d) Creates backup copies
3. Which of the following is an example of Data Generation?
 - (a) AI creating new handwritten digit images
 - (b) Students filling out a survey
 - (c) Using a thermometer to record temperature
 - (d) Copying an already existing chart



4. Which of the following examples show primary data collection?
 - (a) Reading a report on rainfall
 - (b) Filling out your own survey about study habits
 - (c) Using an online article
 - (d) Using weather data from a website
5. Why is accuracy important in Data Collection?
 - (a) It makes the data heavier
 - (b) It ensures correct results and conclusions
 - (c) It makes the data sound nicer
 - (d) It is only for decoration

2.4 FEATURES OF DATA

Data features are the important pieces of information or characteristics that help an AI or machine learning model understand a problem and make predictions. They describe the data in a meaningful way so that the computer knows what each part represents.

Examples

- In a dataset about students, features could be: age, marks, attendance, or favourite subject.
- In a dataset of fruits, features could be: colour, size, weight, or shape.
- For a house price prediction model, features might include: number of rooms, location, area, and price.

Broadly, Data Features can be classified as:

- **Independent features:** pieces of information that do not depend on other values. They are the inputs we give to the AI model so it can learn patterns and make predictions.

For example, to guess the price of a house, independent features may be: size of the house, number of rooms, location.

- **Dependent features:** values that depend on the independent features. It is the output or result the AI model tries to predict.

For example, in house price prediction, the dependent feature is the selling price.



Notes

2.5 DATA PROCESSING AND INTERPRETATION

Data Processing and Interpretation are two important steps that help us turn raw data into meaningful information. After collecting data, we cannot use it directly because it may be unorganized, incomplete, or difficult to understand. Processing and interpreting the data helps us make sense of it and draw useful conclusions.

Data processing is the step-by-step method of organizing, cleaning, and preparing data so that it becomes easy to analyse. Key steps in Data processing are –

1. **Data Cleaning:** removal of mistakes, missing values, and duplicate entries, Outliers
2. **Data Organising:** arranging data in a proper format, such as tables, charts, spreadsheets, and more.
3. **Data Transformation:** changing the data values to a useful form. This may include converting text to numbers, grouping values, creating new features, and more
4. **Data Analysis:** Studying the data to analyse patterns, trends, and relationships using calculations, graphs, features, and more

Data interpretation means understanding the results obtained after processing the data. It helps us make conclusions, decisions, and predictions. Under data interpretation, we:

- Read charts, tables, and graphs
- Look for patterns or trends
- Compare different pieces of data
- Write conclusions or explanations

For example, if data shows that students who study more hours score better marks, we interpret that more study time leads to better performance.

Importance of Data Processing and Interpretation

- They help remove errors.
- They make data organized and meaningful.
- They allow us to make correct conclusions.
- They help AI systems learn accurately and make better predictions.



2.6 DATA LABELLING

Data labelling is an important step in preparing data for Artificial Intelligence and Machine Learning. It helps AI understand what each piece of data represents so it can learn correctly.

What is Data Labelling?

Data labelling means adding meaningful tags or names to raw data so that an AI system can understand it. These labels tell the AI what the data is about.

AI models learn by looking at examples. Labels help the model make connections between the data and the correct answer.

- Without labels → the AI doesn't know what the data means.
- With labels → the AI learns patterns and makes accurate predictions.

Let's look at some examples of Data Labelling for different types of Data:

1. **Image Labelling:** Adding labels to images
Example: A photo of a dog is labelled "dog".
2. **Audio Labelling:** Marking parts of a sound recording
Example: Labelling speech recordings with the words being spoken.
3. **Video Labelling:** Tagging actions in a video
Example: Labels can be "person crossing the road," "car moving," "dog sitting."
4. **Text Labelling:** Tagging words or sentences with meaning
Example: Labelling emails as "spam" or "not spam".

AI studies thousands of labelled examples and learns to recognise patterns. For example: If an AI sees many labelled photos of cats, it learns what a cat looks like. If it sees labelled medical images, it can learn to detect diseases.

Good labelling = Better AI learning.



INTEXT QUESTIONS 2.4

Fill in the Blanks

1. In a fruit's dataset, colour and weight can be examples of
2. Features that describe data are also called of the data.



Notes

Data Literacy and AI Project Cycle

3. Data interpretation means understanding the of the processed data.
4. Removing mistakes and fixing wrong entries in data is called data
5. Labelling images with “cat” or “dog” helps AI learn to objects.
6. Without proper labels, an AI model cannot learn or make accurate

2.7 AI PROJECT CYCLE

What is a Project?

A project is a planned set of activities done to achieve a specific goal or solve a particular problem. It has a clear purpose, a beginning, an end, and a series of steps that must be followed in an organised way. Projects can be small or big, simple or complex. What makes them special is that they are done with a goal in mind and require planning, teamwork, time, and effort.

A project cycle is the step-by-step process followed to complete a project successfully. Just like a recipe helps you prepare a dish in the correct order, a project cycle guides you through each stage of your project – from identifying the problem to presenting the outcome. The project cycle ensures that the work is

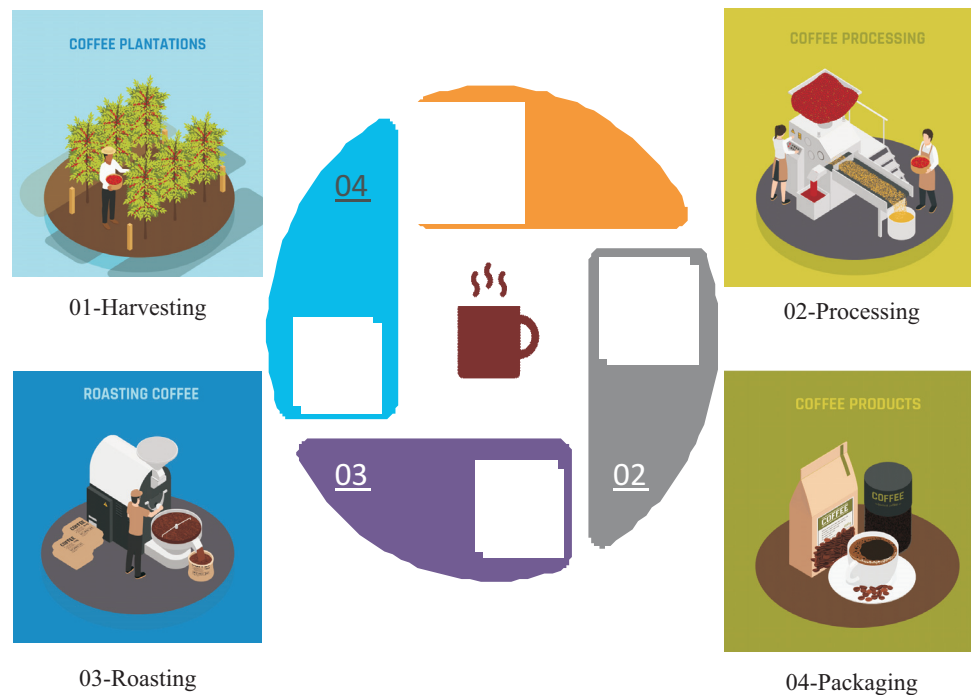


Fig. 2.8 Project Cycle for Coffee Production System



organised, goals are clear, and every step is completed properly. It also helps in planning, gathering resources, checking progress, and improving the result.

Example of a Project Cycle: Coffee Production System:

Coffee production follows a well-defined project cycle: Coffee seedlings are nurtured in fields for several years until the first harvest. – (See Fig. 2.8)

1. **Harvesting:** Ripe coffee cherries are carefully picked by hand or machine.
2. **Processing:** Different methods remove the fruit surrounding the coffee bean, impacting flavour.
3. **Roasting:** Beans are roasted to develop flavour and aroma before packaging for sale.
4. **Packaging:** Roasted coffee is then packaged in the form of coffee products.

This cycle ensures the quality and deliciousness of the coffee you enjoy!

What is an AI Project Cycle?

An AI Project Cycle is a structured series of steps used to build an Artificial Intelligence solution. Just like any project needs planning and organisation, an AI project requires a clear process to move from an idea to a working AI model. The AI Project Cycle helps students, developers, and organisations create AI systems in a systematic and effective way.

It guides you through understanding the problem, collecting data, training the model, testing it, and finally improving and presenting the solution. Each step plays an important role in making sure the AI model works correctly and solves the intended problem.

Why Do We Need an AI Project Cycle?

- It ensures the AI system is built step-by-step and not randomly.
- It helps in collecting and preparing quality data, which is essential for AI.
- It makes the project easier to manage, test, and improve.
- It helps ensure the final AI model is accurate, useful, and reliable.

Imagine applying our AI skills to explore and identify habitable planets, just like in Sci-fi movies. How could AI assist in analysing astronomical data, predicting planetary conditions, or even simulating life-supporting ecosystems on distant worlds?



Notes

Data Literacy and AI Project Cycle

We will now understand the concept of the AI Project Cycle. Imagine we want to find an 'Earth-like exoplanet'

Finding an Earth-like exoplanet

- An exoplanet is any planet beyond our solar system.
- Our aim is to find an Earth-like planet far away from Earth.
- Some examples of Earth-like planets are Gliese 667Cc and Kepler-69c.

Stages of the AI project cycle

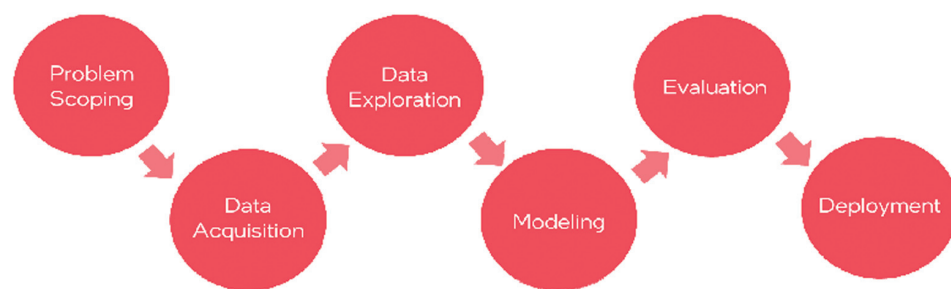


Fig. 2.9: AI Project Cycle

The AI project cycle contains 6 different stages as shown in Fig 2.8

Let's look at each stage one by one!

Stage 1, In the Problem Scoping stage we define a goal for this project.

Stage 2, we have Data Acquisition in which we collect the data that is required to train the model.

Stage 3, we have Data Exploration main objective of this stage is to visualize data (usually through Graphs/Plots) to understand trends, patterns and relationship exhibited within the data.

Stage 4, we have Modelling, the main objective of modelling is to implement algorithms and fine-tune the model which gives the best result.

Stage 5, we have Evaluation, the main objective of this stage is to test and evaluate the different models and choose the best model.

Stage 6, we have Deployment, the main objective of this stage is to deploy the solution.



2.8 PROBLEM SCOPING

Problem Scoping is the first and most important step of the AI Project Cycle. Here, we identify the problem that needs to be solved and understand why it is important. In this stage, we:

- Clearly define the problem
- Identify who is affected by it
- Understand what solution is needed
- Set clear goals and outcomes

The Problem Scoping stage helps in defining a goal for the project. In order to understand the problem better and come up with the goal of our project, we shall use the 4Ws Problem Canvas.

4Ws Problem Canvas

The 4Ws Problem Canvas is a simple and effective tool used in the Problem Scoping stage of the AI Project Cycle. It helps students clearly understand the problem they want to solve by answering four important questions: Who, What, Where, and Why. By filling out this canvas, we gain a clear picture of the problem, the people involved, and the purpose of creating an AI solution. The 4Ws Problem Canvas makes sure we focus on the right problem and create a solution that is useful.

Let us try to fill the 4Ws Problem Canvas for the extraterrestrial problem that we saw earlier:



Fig. 2.10: The 4Ws Problem Canvas

WHO

Who is having the problem?

- Who are the stakeholders?
 - o Space Organizations



Notes

Data Literacy and AI Project Cycle

- o Human Race
- What do you know about them?
 - o Various space organizations have set up telescopes on land and in space to collect data

WHAT

What is the nature of the problem?

- What is the problem?
 - o Identifying an earth like exoplanet with our existing resources
 - o Space travel has its limitations.
- How do you know it is a problem?
 - o Global Environmental Issues
 - o Research Journals, Documentaries, and news articles

WHERE

Where does the problem arise?

- What is the context/ situation in which the stakeholders experience the problem?
 - o Rise in sea levels, Melting of Polar Icecaps, Air, Soil, and Water pollution

WHY

Why do you believe it is a problem worth solving?

- What would be of key value to the stakeholders?
 - o Find an Earth-like exoplanet as close to the solar system as possible
- How would it improve their situation?
 - o Humans could populate an Earth like exoplanet in case severe crises arise on Earth in future

After getting all the insights for the 4Ws, then the details are compiled using a Problem statement template. This helps in defining or articulating the Problem statement.



Our	[stakeholders]	Who
has a problem that	[issue, problem, need]	What
when / while	[context, situation].	Where
An ideal solution would	[benefit of solution for them]	Why

Fig. 2.11: Problem statement template

Let us see this with an example as below.

Example I

Problem: Finding an exoplanet

Filling the Problem Statement Template for our exoplanet example gives us:

Our	Space Organisations, Human Race	Who
Has/Have a problem that	They need to find an Earth like exoplanet	What
When/While	Utilising our existing resources	Where
An ideal solution would be	Helping analyse astronomical data, predicting planetary conditions, or even simulating life-supporting ecosystems on distant worlds	Why

Fig. 2.12: Problem statement template of finding an exoplanet

Example 2: Managing Food Waste in the School Canteen

Let's look at another example that is much closer to home. Imagine your school canteen throws away a lot of unconsumed food every day because the cooking staff never knows exactly how many students will buy lunch or what they want to eat.

Let's use the 4Ws Problem Canvas to scope this problem:

WHO

- Who is having the problem? Canteen Staff and School Management.



Notes

- What do you know about them? The staff cooks based on rough estimates. The school wants to be eco-friendly and save money.

WHAT

- What is the nature of the problem? Large amounts of cooked food are wasted daily.
- How do you know it is a problem? Dustbins are full of leftover food at the end of the day, and the canteen is losing money.

WHERE

- Where does the problem arise? Inside the school canteen.
- What is the context/situation? During the daily lunch break when hundreds of students arrive to buy food at the same time.

WHY

- Why do you believe it is a problem worth solving? Reducing food waste helps the environment and saves money.
- How would it improve their situation? An AI model could predict exactly how many meals to cook each day based on past data (like attendance, day of the week, or weather), ensuring zero waste.

Filling the Problem Statement Template for the Canteen example:

Our	Canteen Staff and School Management	Who
Has/Have a problem that	They face the issue of excessive daily food waste and need a way to accurately predict daily food demand	What
When/While	In the school canteen during lunch hours	Where
An ideal solution would be	Predicting exactly how many meals to cook each day to reduce food wastage, save costs, and become an eco-friendly school	Why

Fig. 2.13: Problem Statement Template for the Canteen example



2.9 DATA ACQUISITION

The Data Acquisition Stage is the second step of the AI Project Cycle. After identifying the problem, the next task is to collect the right data that will help the AI model learn and make accurate predictions. Since AI learns from examples, this stage is very important – better data leads to a better AI system. In this stage, we gather all the information needed to solve the problem. The data can come from many different sources and can be of many types, such as text, numbers, images, audio, or videos. The goal is to collect enough data, correct data, and useful data.

As discussed in the previous sections of this chapter, Data can be collected through Primary and Secondary sources and can be in qualitative or quantitative form.

Importance of Data Acquisition

- AI learns from data, so the quality of data affects the quality of the model.
- Wrong or incomplete data can lead to incorrect predictions.
- A good dataset makes the AI system reliable and accurate.

For example, if you want to create an AI model that identifies different types of flowers:

- You must collect many clear images of roses, tulips, sunflowers, etc.
- This collected data becomes the training material for your AI.

2.10 DATA EXPLORATION

The Data Exploration stage is the third step of the AI Project Cycle. After collecting the required data, we need to understand it, study it, and check its quality. Data Exploration helps us know whether the collected data is useful, correct and ready for building an AI model. This stage is important because raw data is often messy – it may contain mistakes, missing values, or unwanted information. Exploring the data helps us clean it and prepare it for the next steps.

What happens in Data Exploration?

1. **Understanding the Data:** This helps us understand the structure of the dataset.

We look at the data to see:

- What kinds of data we have (numbers, images, text, etc.)
- How much data we have
- What each feature means



Notes

Data Literacy and AI Project Cycle

2. Checking Data Quality - If errors exist, we note them so they can be corrected during cleaning.

We check the data for:

- Missing Values
- Incorrect Entries
- Repeated (Duplicate) Data
- Outliers (Unusual or Extreme Values)

3. Organising the Data - This makes it easier to analyse and understand.

We arrange the data in a proper format, such as:

- Tables
- Charts
- Graphs

4. Finding Patterns and Relationships - Finding patterns helps the AI model learn more effectively.

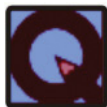
We look for patterns such as:

- Trends in numbers
- Differences between groups
- How one feature affects another

Importance of Data Exploration

- It helps us know if the data is correct and complete.
- It makes the next steps – modelling and training – much easier.
- It improves the accuracy of the final AI system.
- It avoids errors caused by bad or unclear data.

Good exploration leads to better decision-making and a better AI model.



INTEXT QUESTIONS 2.5

Short-answer questions

1. What is the purpose of the Data Exploration stage?
2. What are two things we look for when exploring data?



3. What is an outlier in a dataset?
4. Name one way we can organise data during exploration.

2.11 MODELLING

Modelling is the fourth stage of the AI Project Cycle. After collecting and exploring the data, the next step is to teach the AI model how to learn from that data. Modelling is the process where we create, train, and improve an AI model so that it can make predictions or decisions on its own. In this stage, the AI uses mathematical techniques and patterns in the data to understand how things work. Once trained, the model can recognise images, classify objects, answer questions, predict outcomes, and much more depending on the problem we are trying to solve.

Practical: Hands-on Activity - Exploring a Real-World Dataset (CSV)

In the real world, data is often stored in files called CSVs (Comma Separated Values) or spreadsheets. Let's look at a sample dataset of discovered Exoplanets and perform a manual Data Exploration exercise.

Imagine you have downloaded the following dataset. Look closely at the rows and columns:

Planet Name	Distance (Light Years)	Planet Mass (Earths)	Discovery Year	Water Detected
Kepler-22b	620	36.0	2011	Yes
Proxima Centauri b	4.2	1.27	2016	No
Gliese 667 Cc	23.6	3.8	2011	Yes
Trappist-1e	39.5	0.77	2017	NaN
Kepler-452b	1402	9999.0	2015	Yes

Your Data Exploration Task:

As an AI developer, analyse the table above and answer the following questions:

1. **Identify Features:** How many features (columns) does this dataset have?
2. **Identify Missing Data:** Do you see any missing values? (Hint: Look at Trappist-1e. "NaN" stands for Not a Number, meaning data is missing).
3. **Identify Outliers:** Is there any value that looks completely wrong or impossible compared to the others? (Hint: Look at the mass of Kepler-452b. A mass of 9999.0 Earths is likely a typing error or outlier that needs cleaning!)



Notes

What happens in the Modelling stage?

1. **Selecting the Right Model:** Different AI problems need different types of models. Choosing the right model helps the AI learn effectively.
2. **Training the Model:** Training means showing the AI many examples so it can learn patterns. The model tries to learn the relationships between the input features and the final output.
 - If it is an image model, we show labelled images.
 - If it is a language model, we give text examples.
 - If it is a prediction model, we give numerical data.
3. **Learning from Mistakes:** During training, the model may make mistakes. It then adjusts itself to reduce errors. This cycle continues repeatedly until the model becomes accurate. This is like how students improve by practising and correcting their mistakes.
4. **Testing within the modelling stage:** Even while training, we check if the model is learning well. We test it on a small part of the data to see how well it performs. If the model is not learning correctly, we may add more data, clean the data better, choose different features, or try another model.

Importance of Modelling

Modelling is the fourth stage of the AI Project cycle. In this stage an algorithm is implemented on the data, to make the model learn from data. Here, the model learns the different patterns that exists within the data and enables the model to take meaningful decisions.

2.11.1 Types of AI Models

Models help us understand and solve problems by showing how things work, like weather predictions, making decisions, or creating designs. Below is the broad classification of AI Approaches through which AI Model learns.

Rule-Based AI

A Rule-Based AI Approach is one of the simplest and earliest methods of creating artificial intelligence. In this approach, the AI system follows a set of predefined rules written by humans. These rules tell the system exactly what to do in different situations, just like instructions in a flowchart.

Rule-based AI does not learn from data. Instead, it works by checking conditions and applying rules to reach a conclusion. It behaves like an expert following a checklist. These rules usually follow an IF–THEN structure.



Rule-Based AI Example

IF X happens THEN do Y

- Example 1: IF LUNCH BREAK happens THEN do EAT LUNCH
- Example 2: IF FLOOR DIRTY happens THEN do CLEANING

Rule-based AI are NOT used at a lot of places now a days

2.11.2 How do humans take decisions?

Let's say we are making a cup of coffee for the first time, and we are not sure about how much sugar to add. So, we go through a hit and trial method, and the table here shows how it goes – (As shown in Fig. 2.14)

Time of coffee making	Number of sugar spoons added	Taste of coffee
First time	Half a spoon	Not sweet enough
Second time	One and a half spoon	Too sweet
Third time	One spoon	Right amount of sweetness
Fourth time	One spoon	Right amount of sweetness

Fig. 2.14: Example for taking humans decisions

Thus, by learning through experiences, mistakes, and feedback, we can come up with the best solution, that is – the right cup of coffee with proper amount of sugar!

Learning-Based AI

A Learning-Based Approach is a modern method used to build AI systems that can learn from data instead of relying only on fixed rules. Unlike rule-based AI, which follows human-written instructions, learning-based AI improves by studying examples, finding patterns, and making predictions on its own. This approach is also called **Machine Learning**, because the AI “learns” from experience just like humans do.

For Example, If you show 100 images of dogs and 100 images of cats to AI, it can learn to identify dogs and cats based on the shape of their eyes, ears, nose, length of their tails, body shape, etc.

Learning Based AI Approach is used very commonly these days in applications like Facial Recognition, Voice Assistants, OTT Recommendations, Self-driving cars, and more.



Notes



INTEXT QUESTIONS 2.6

Short-answer questions

1. Why is modelling important in AI?
2. What is a rule-based AI approach?
3. What is the main limitation of rule-based AI?
4. What is a learning-based approach in AI?
5. How is learning-based AI different from rule-based AI?

2.12 EVALUATION

The Evaluation Stage is the fifth step of the AI Project Cycle. After building and training the AI model in the Modeling stage, we must check how well the model performs. The evaluation stage helps us understand whether the model is accurate, reliable, and ready to be used. This step ensures that the AI does not just work on the training data, but can also perform well on new, unseen data.

What happens in the Evaluation Stage?

1. **Testing the Model:** We test the AI model using a part of the data that it has never seen before. This helps us check if the model can make correct predictions on new examples.
2. **Measuring Accuracy:** The model's performance is measured using accuracy scores or other metrics. This tells us how many predictions were correct and how many mistakes the model made. A higher accuracy means a better model.
3. **Finding errors or weak areas:** During evaluation, we identify where the model is going wrong, what types of inputs confuse the model, and which classes or categories are poorly predicted. This helps improve the model in future steps.
4. **Comparing with Expectations:** We check if the model meets the goals set during Problem Scoping. If the results are not good enough, we may need to:
 - Collect more data



- Clean the data better
- Choose better features
- Retrain the model

Importance of Evaluation

- It ensures the AI model is fair, accurate, and useful.
- It helps us detect mistakes before the model is deployed.
- It tells us if the model is ready for real-world use.
- It allows us to improve and fine-tune the AI model.

Without evaluation, we cannot trust the model's decisions.

2.13 DEPLOYMENT

The Deployment Stage is the final step of the AI Project Cycle as shown in Fig 3.0. After building, training, and evaluating the AI model, we now make it available for real-world use. Deployment means taking the AI model out of the development environment and putting it into an application, website, device, or system where people can use it. This is the stage where the AI starts helping users solve real problems.

What happens in the Deployment stage?

1. Integrating the Model into a Real System - So users can interact with it easily. The trained AI model is connected to –
 - a mobile app
 - a website
 - a computer program
 - a device or machine
2. Making the model user-friendly - The model needs to be presented in a simple and useful way. This stage focuses on making AI accessible to everyone.
3. Performance monitoring - Even after deployment, we keep checking how well the model performs in the real world. AI may face new or unexpected situations, so we must ensure it works correctly, gives accurate results, and responds quickly.



Notes

Data Literacy and AI Project Cycle

4. Updating and Improving the Model - Over time, new data becomes available. The AI model may be updated to:

- Improve accuracy
- Fix errors
- Add new features

Importance of Deployment

- It allows people to use and benefit from the AI system.
- It solves the real-world problem defined in the Problem Scoping stage.
- It shows how effective and practical the AI model is.
- It helps gather feedback for further improvement.

Deployment is not the end – it is the beginning of real-world learning.

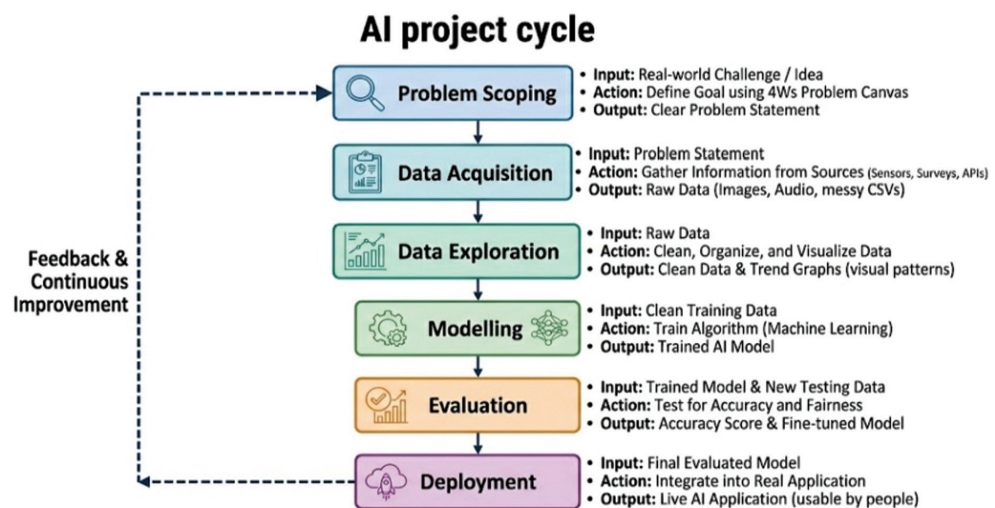


Fig. 2.15: A Project Cycle in AI



WHAT HAVE YOU LEARNT

- Data is raw information that can be numbers, text, images, or audio.
- Data Literacy means understanding and using data to make better decisions.

Data Literacy and AI Project Cycle

- The Data Pyramid shows how data becomes information, knowledge, and finally wisdom.
- Data can be collected through Data Discovery, Augmentation, and Generation.
- Primary Data is collected first-hand; Secondary Data comes from existing sources.
- Good data must be clean, accurate, and well-structured.
- Data Features describe important characteristics that help AI learn.
- Data Processing and Interpretation make raw data understandable and meaningful.
- Data Labelling adds tags so AI knows what the data represents.
- The AI Project Cycle includes Problem Scoping, Data Acquisition, Exploration, Modelling, Evaluation, and Deployment.
- Rule-Based AI follows fixed rules; Learning-Based AI learns patterns from data.



TERMINAL QUESTIONS

Short-Answer Questions

1. What is data? Give two examples.
2. Explain the meaning of Data Literacy. Why is it important?
3. What are the four levels of the Data Pyramid?
4. Differentiate between Quantitative and Qualitative data.
5. What is Data Discovery? Give one example. What is Primary Data? How is it collected?
6. Define Data Features with one example.
7. What is Data Cleaning? Why do we need it?
8. What is the purpose of the Evaluation stage in the AI Project Cycle?
9. What is a rule-based AI approach?



Notes



Notes

10. Explain Learning-Based AI with one example.
11. What happens during the Deployment stage of an AI project?

Long-Answer Questions

1. Explain the concept of Data Literacy using the Data Pyramid. How does raw data turn into wisdom?
2. Discuss the three methods of Data Acquisition: Data Discovery, Data Augmentation, and Data Generation. Give an example of each.
3. Explain the importance of Data Processing and Interpretation. Describe the key steps involved in Data Processing.
4. What is an AI Project Cycle? Describe all six stages in brief.
5. Compare Rule-Based AI and Learning-Based AI. How do they differ in the way they solve problems?
6. Problem Statement: Air pollution is a growing concern in our city, affecting the health of people and the environment. Children, especially, face challenges like breathing issues and reduced outdoor play time due to poor air quality.

Explore the problem statement using the 4Ws problem canvas and come up with the Problem Statement Template.
7. Make a list of all the data points that you think should be collected to solve the Exoplanet problem statement. Also, classify the data points as quantitative and qualitative.

Skill Based Questions

1. You are asked to build an AI model that predicts if a plant is healthy or unhealthy using leaf images. What data would you collect, and how would you ensure it is clean, accurate, and usable?
2. Imagine a situation where Google Maps suddenly starts giving incorrect route suggestions. Which stages of the AI project cycle might need to be revisited, and why?
3. You are analysing a dataset of exam marks. You notice one student has 900 marks out of 100. How would you treat this outlier, and why is it important to handle such errors?

4. If a rule-based chatbot often gives wrong answers when students ask new or confusing questions, what approach should the school use instead, and why?



ANSWERS TO INTEXT QUESTIONS

2.1

1. (b) 2. (d) 3. (b) 4. (d) 5. (b)
6. (a) 7. (c) 8. (d)

2.2

1. (a), (c) 2. (b), (d), (e), (f)

2.3

1. (d) 2. (d) 3. (a) 4. (b) 5. (b)

2.4

1. features 2. qualities 3. meaning 4. cleaning
5. classify 6. predictions

2.5

1. The purpose of Data Exploration is to study and understand the collected data, check its quality, and prepare it for the next stages of the AI project.
2. We look for – a. Patterns or trends in the data and b. Mistakes such as duplicates or missing entries.
3. An outlier is an unusual or extreme value that does not fit with the rest of the data.
4. We can organise data using tables, charts, or graphs.

2.6

1. Modelling is important because it teaches the AI how to solve the problem. A well-trained model helps the AI give accurate and useful results.
2. A rule-based AI approach uses fixed IF–THEN rules written by humans. The AI follows these rules to solve simple and clearly defined problems.



Notes



Notes

Data Literacy and AI Project Cycle

3. It cannot learn from data and works only when rules are predefined. It cannot handle complex or changing situations.
4. A learning-based approach allows AI to learn from data instead of using fixed rules. The AI improves by studying examples and identifying patterns.
5. Learning-based AI learns from data and improves over time, while rule-based AI only follows fixed instructions and cannot learn on its own.

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- explain categorization of data and ways of data collection
- explore Stages involved in Data processing
- describe all the six stages of the AI project cycle: Problem Scoping, Data Acquisition, Data Exploration, Modelling, Evaluation, and Deployment
- define the concept of AI models



Notes

3

DOMAINS OF AI

In the previous Lesson, you have learnt how data can be managed in AI systems. In this chapter, you will be introduced to the domains of Artificial Intelligence. You will learn about how these domains form the foundation of modern AI applications and allow technology to solve real-world problems more effectively. You will also get to know about Natural Language Processing, which enables computers to understand and use human language; Computer Vision, which helps machines to interpret images, videos and Statistical data domain. It helps machine learning models to understand numeric data, analyse and helps in making well- informed decisions.



OBJECTIVES

After reading this lesson, you will be able to:

- definethe domains of AI based on the dataset used
- explain Statistical Data domain and its examples in real life
- explainComputer Vision domain and its examples in real life
- describe Natural Language Processing domain and its examples in real life

3.1 INTRODUCTION TO AI DOMAINS

Artificial Intelligence works in different ways depending on the type of problem it is trying to solve. These different areas of AI are called AI domains. Each domain focuses on a specific kind of data – such as numbers, images, or language – and uses different techniques to understand and process that data. In simple words, AI



Notes

Domains of AI

domains are categories that describe what kind of tasks an AI system is designed to do. Understanding these domains help us choose the right type of AI for the right kind of problem.

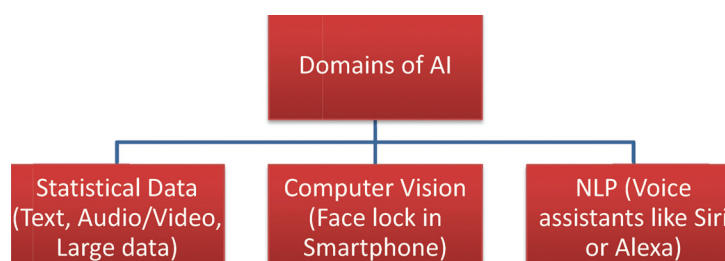


Fig. 3.1: Domains of AI

AI Domain designs the way in which an AI machine understands the world. Each domain gives AI a different type of ability – like how humans use different senses and skills. Broadly, there are 3 AI domains, and they are:

1. **Statistical Data:** Statistical data helps AI to understand and learn from numbers, patterns, and calculations. AI studies large sets of numerical data and uses mathematical models to find trends, make predictions, or identify unusual patterns. This domain caters to Numerical, Categorical, and Sensor data.

Examples: Weather Prediction, Detecting unusual transactions, Score prediction, etc.

2. **Computer Vision:** Computer Vision allows AI to see and understand images and videos, just as humans use their eyes and interpret what is in front of them. Looking at the visual information, AI can identify the features of interest, to be able to identify the similarities and differences in images. This domain caters to Image and Video data.

Examples: Face lock in Smartphones, Self-driving cars, Social Media filters, etc.

3. **Natural Language Processing:** Natural Language Processing (NLP) helps AI understand, interpret, and respond to human language – both written and spoken. NLP allows an AI machine to understand how we talk, what we mean, and respond back to us in the same manner. This domain caters to textual and audio data.

Examples: Voice assistants like Siri or Alexa, Speech to Text, Autocorrect, etc.



INTEXT QUESTIONS 3.1

1. What was the role of AI in the Rock, Paper, Scissors challenge?
2. What was the role of AI in the Semantris challenge?
3. What was the role of AI in the Quick Draw challenge?
4. Match the following:

I. Rock, Paper, Scissors challenge	A. Computer Vision
II. Semantris challenge	B. Natural Language Processing
III. Quick Draw challenge	C. Statistical Data

3.2 STATISTICAL DATA

The Statistical Data AI Domain is the area of Artificial Intelligence that works mainly with numbers, measurements, and mathematical patterns. This domain helps AI understand numerical data, analyse it and make predictions based on trends. It is useful when the problem involves quantities – such as marks, temperatures, prices, scores, or sensor readings.

3.2.1 What does Statistical Data do?

In this domain, AI learns from numerical datasets to:

- Analyse numerical data
- Identify patterns and trends
- Compare values (higher, lower, increase, decrease)
- Make predictions based on past data
- Solve problems involving quantities

This type of AI is widely used in data analysis, forecasting, and prediction tasks.

3.2.2 Importance of Statistical Data domain

- Numbers are everywhere in daily life.
- Many real-world problems depend on analysing numerical data.
- Statistical data helps AI make accurate and logical decisions.



Notes



Notes

3.2.3 Examples of Statistical Data

- Temperature readings
- Exam marks
- Population numbers
- Sales figures
- Sensor data from machines

Imagine you are trying to predict how many runs your favourite cricket team will score in their next match. You wouldn't just guess randomly; you would look at their past scores, the run rate, and the weather conditions. This is exactly how the Statistical Data domain works. It looks at historical numbers finds the hidden pattern or “trend,” and uses it to predict the future! ((like the dots on the graph, refer Fig. 3.2).

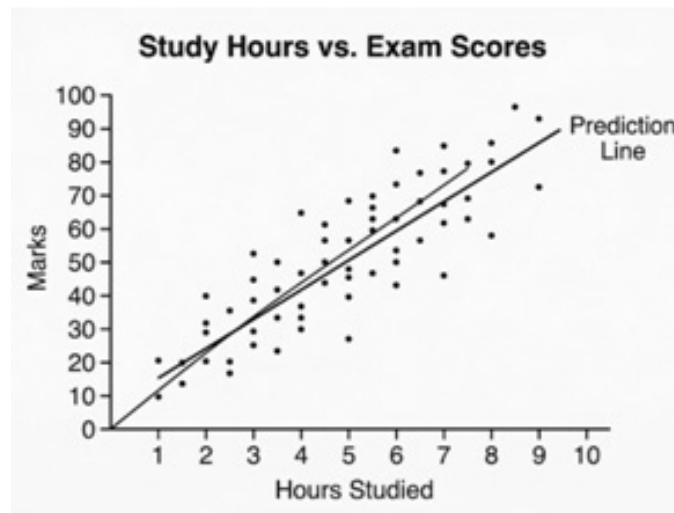


Fig. 3.2: Example of Statistical data

3.2.4 Real-world Applications of Statistical Data domain

1. **Weather Forecasting:** AI analyses temperature, humidity, and rainfall data to predict upcoming weather.
2. **Finance and Banking:** AI studies spending patterns, stock prices, or loan data to make predictions or detect fraud.
3. **Sports Analytics:** AI examines player statistics – runs scored, goals made, speed, accuracy – to judge performance.

Domains of AI

4. **Health Monitoring:** AI uses heart rate, sleep data, and steps to give fitness advice or detect health issues early.
5. **Education:** AI predicts student performance using attendance, homework records, and past marks.



INTEXT QUESTIONS 3.2

Multiple-choice questions

1. The Statistical Data Domain mainly works with:
 - (a) Images
 - (b) Human language
 - (c) Numbers and measurements
 - (d) Sounds
2. Which of the following is an example of statistical (numerical) data?
 - (a) Photo of a cat
 - (b) Temperature readings
 - (c) A storybook
 - (d) A music file
3. AI uses statistical data to:
 - (a) Understand pictures
 - (b) Analyse and predict number patterns
 - (c) Translate languages
 - (d) Play music
4. AI uses statistical data in banking to:
 - (a) Recognise faces
 - (b) Detect fraud based on spending patterns
 - (c) Translate messages
 - (d) Identify images

3.3 COMPUTER VISION

The Computer Vision domain is a major area of Artificial Intelligence that focuses on helping machines see, understand, and interpret visual information, just like humans do. Humans use their eyes and brain to recognise objects, faces, colours,



Notes



Notes

Domains of AI

shapes, movements, and scenes. In the same way, Computer Vision allows machines to make sense of images and videos.

Computer Vision teaches computers how to extract meaningful information from pictures – such as identifying what objects are present, what actions are happening, or what patterns exist in the image. This domain is especially important today because we live in a world full of cameras: mobile phones, CCTV, drones, robots, cars, and even traffic systems use vision-based AI.

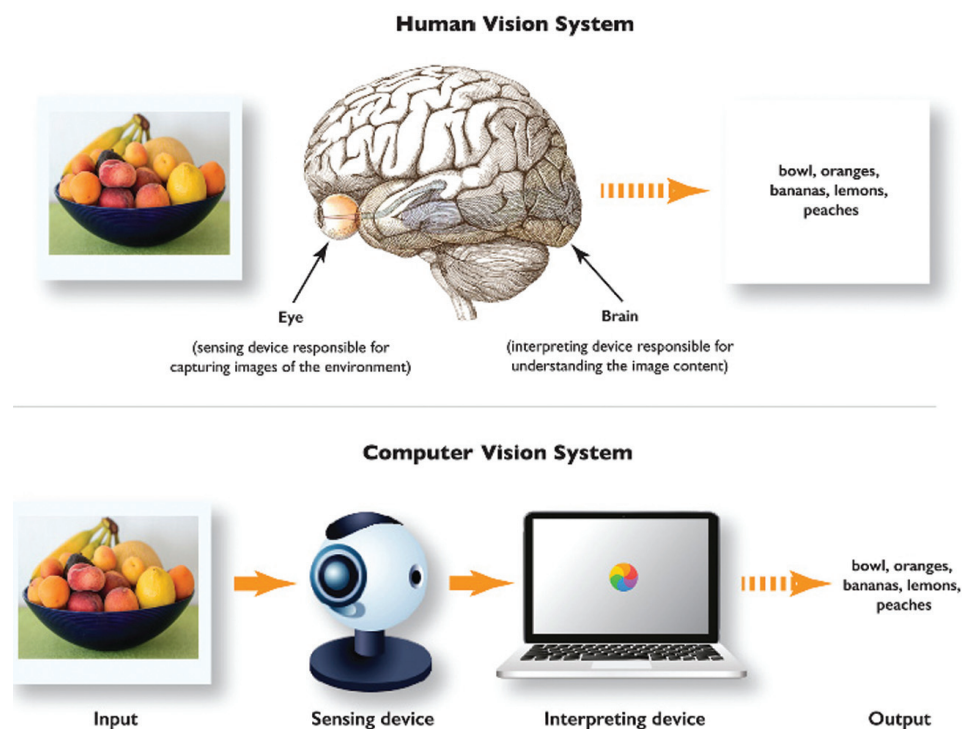


Fig. 3.3: Computer Vision

3.3.1 How does Computer Vision work?

Computer Vision follows a few simple steps:

1. **Image Input:** The system receives an image or video through a camera or file.
2. **Processing the Image:** The AI breaks the image into tiny dots called pixels and looks for patterns such as edges, shapes, colours, and textures.
3. **Feature Extraction:** Important features are identified, for example:
 - (a) The round shape of a ball



- (b) The edges of a face
 - (c) The lines on a road
4. **Prediction or Action:** The AI uses its learning to decide what the image shows:
- (a) This is a dog.
 - (b) This is a stop sign.
 - (c) The person is smiling.

Better the training data, better is the predictions. (See Fig. 3.3)

Object Identification

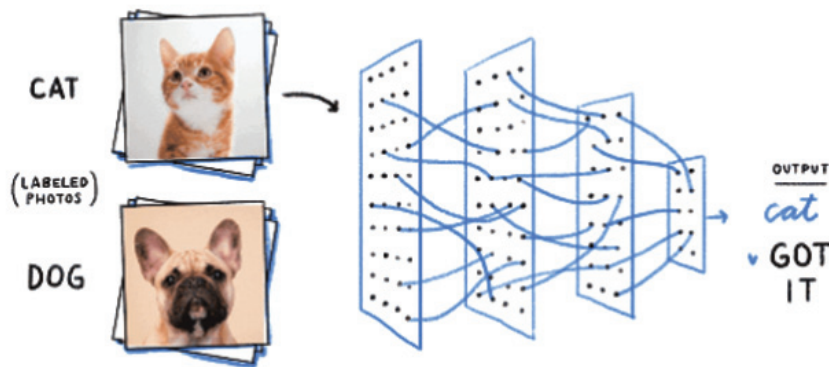


Fig. 3.4: Example of Computer Vision

A dataset having images of Cat and Dog is given as input to the model. The dataset is labelled and is balanced i.e. equal number of images both categories. The model will process the dataset and will learn the different features of Cat and Dog through the dataset. Once the model learns different features of cat and dog it is now able to identify from the random image given whether the image is cat or dog. This is Computer vision enables computers to recognize and understand image data.

3.3.2 Importance of Computer Vision (CV)

Humans cannot manually observe or analyse millions of images and videos every day, but AI can. CV helps machines:

- recognise objects
- detect people, animals, or vehicles
- understand scenes



Notes

- track movements
- analyse visual patterns

It enables machines to “see” and act on visual data just as humans would – sometimes even better and faster.

3.3.3 Examples of Computer Vision data

- Images (photos)
- Videos
- Annotated Images (images with labels)
- Medical Images
- Satellite and Drone Images
- Handwritten and Printed Text Images
- Gesture and Pose data

Labelled Data



Unlabelled Data

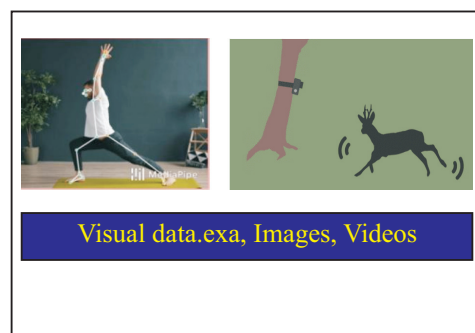


Fig. 3.5: examples of Computer Vision Data

3.3.4 Real-world applications of Computer Vision domain

1. **Face Recognition:** Used in mobile phones to unlock screens by detecting your face.
2. **Self-Driving Cars:** Cameras help cars see traffic signs, pedestrians, and lanes.
3. **Medical Imaging:** AI reads X-rays or MRI scans to identify diseases early.
4. **Security Cameras:** AI detects suspicious activities or counts people in a room.

5. **Shopping Apps:** Letting users “try on” clothes or glasses virtually.
6. **Google Lens:** Recognising plants, animals, objects, or translating text from images.
7. **Farming Technologies:** Detecting plant diseases by studying leaf images.



INTEXT QUESTIONS 3.3

Multiple-choice questions:

1. What does the Computer Vision (CV) domain mainly work with?
 - (a) Numbers
 - (b) Human Speech
 - (c) Images & Videos
 - (d) Text documents
2. Computer Vision helps AI to:
 - (a) Read and understand human language
 - (b) Analyse numerical data
 - (c) See and understand visual information
 - (d) Play games
3. Which of the following is NOT an example of CV data?
 - (a) Photos of animals
 - (b) CCTV footage
 - (c) X-ray images
 - (d) Audio recordings
4. Which sensor is the most important to collect Computer Vision related data?
 - (a) Moisture Sensor
 - (b) Thermometer
 - (c) Microphone
 - (d) Camera
5. Self-driving cars use Computer Vision to:
 - (a) Predict weather
 - (b) Measure heart rate
 - (c) See road signs and detect pedestrians
 - (d) Translate languages



Notes



3.4 NATURAL LANGUAGE PROCESSING

The Natural Language Processing (NLP) domain is a branch of Artificial Intelligence that focuses on helping computers understand, interpret, and use human language. Humans communicate using languages like English, Hindi, Tamil, or Bengali. NLP allows machines to understand these languages – just like we do – so they can read, write, listen, and respond meaningfully. NLP makes it possible for us to talk to computers, type questions, or give voice commands and the computer understands what we mean.

3.4.1 How does Natural Language Processing work?

1. AI receives text or speech input.
2. AI breaks the language into smaller parts –
 - (a) Words
 - (b) Sentences
 - (c) Grammar
 - (d) Meaning
3. AI tries to understand patterns by learning from large amounts of language data.
4. AI gives a response in the form of an answer, translation, summary, or by carrying out a task.

3.4.2 Importance of NLP

Humans and computers communicate very differently. While humans use sentences, emotions, and context, computers understand numbers and patterns. NLP acts as a bridge between the two. It converts human language into a form that computers can understand, and then converts the computer's response back into natural language for us.

NLP enables AI to:

- Understand text and speech data
- Understand sentiment express within the language data
- Facilitates the conversion to digital form from the natural form



NLP is important as it:

- Helps humans and machines communicate naturally
- Makes accessing information easier
- Saves time in typing, searching, and translating
- Supports education, business, healthcare, and daily life
- Allows AI tools to understand different languages (including Indian languages)

3.4.3 Examples of NLP data

- Text Data
- Speech or Audio Data



Fig. 3.6: Example of Natural Language Data

3.4.4 Real-world applications of NLP Domain

1. **Voice Assistants:** Siri, Alexa, Google Assistant - These tools understand spoken questions and give spoken answers.
2. **Chatbots:** Websites and apps use chatbots to answer questions instantly using text.
3. **Google Translate:** NLP helps translate one language into another.
4. **Email Spam Detection:** AI reads email content and filters unwanted or dangerous messages.
5. **Autocorrect and Predictive Text:** Your keyboard suggests the next word or corrects spellings.
6. **Sentiment Analysis** - Companies use AI to read reviews and see if people liked or disliked a product.



Notes



INTEXT QUESTION 3.4

Multiple-choice questions:

1. What does NLP mainly deal with?
 - (a) Images and videos
 - (b) Numbers and measurements
 - (c) Human language (text and speech)
 - (d) Weather data
2. Which of the following is an example of NLP?
 - (a) Unlocking a phone using face recognition
 - (b) Translating English to Hindi on Google Translate
 - (c) Detecting a tumour in an X-ray
 - (d) Counting steps on a fitness tracker
3. Which of the following is NLP data?
 - (a) A video of a cricket match
 - (b) A paragraph from a storybook
 - (c) A photo of a dog
 - (d) A medical MRI scan
4. Email spam detection works because NLP can:
 - (a) Identify images in emails
 - (b) Read the text in emails and find harmful content
 - (c) Analyse numbers only
 - (d) Translate handwriting to text
5. Why is NLP important in AI?
 - (a) It helps AI understand numbers
 - (b) It helps AI understand and use human languages
 - (c) It helps AI detect faces
 - (d) It makes AI run faster



WHAT HAVE YOU LEARNT

- The domains of AI based on the dataset used i.e. Statistical Data, Computer Vision (CV) and Natural Language Processing (NLP).
- **Statistical Data:** In the background, your map app analyzes thousands of numbers – live traffic speeds, distance, and historical travel times – to calculate and predict that your trip will take exactly 14 minutes.
- **Computer Vision:** When you pick up your phone and look at it, the front camera scans your face, extracts your facial features, and unlocks the screen.
- **Natural Language Processing (NLP):** You ask your phone, “Tell me the fastest route to school.” The phone’s smart assistant listens, translates your spoken words into text, and understands your intent.



TERMINAL QUESTIONS

Short-Answer Questions

1. What are AI domains? Give two examples.
2. What is the main purpose of Computer Vision domain?
3. Give two examples of NLP domain.
4. What does the Statistical Data domain deal with?

Long-Answer Questions

1. Explain Statistical Data domain with suitable examples.
2. Describe the Computer Vision domain and explain how it works.
3. Explain NLP domain and its importance.
4. Draw a comparison among the three AI domains.

Skill Based Questions

1. You are building an AI system to detect whether a plant is healthy or diseased. Which AI domain will you use and why?
2. A company wants to understand customer feedback from thousands of online reviews. Which domain should they use and what challenges might arise?
3. A school wants to predict student performance based on study hours, attendance, and past marks. Which domain should be applied and why?



Notes



Notes



ANSWERS TO INTEXT QUESTIONS

3.1

1. AI was trying to understand our pattern, while trying to win against us by using it.
2. AI was trying to understand human language by associating the words we input with the words that were highlighted.
3. AI was trying to identify features in our drawings and compare them with existing drawing dataset to identify the drawn object.

4. I C II B III A

3.2

1. (c) 2. (b) 3. (b) 4. (b)

3.3

1. (c) 2. (c) 3. (d) 4. (d) 5. (c)

3.4

1. (c) 2. (b) 3. (b) 4. (b) 5. (b)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- explain the three AI domains i.e. Statistical Data, Computer Vision (CV), and Natural Language Processing (NLP).
- explore how these domains are used in everyday life, from weather forecasting and smart traffic cameras to email spam filters and voice assistants.

References

1. https://cbseacademic.nic.in/web_material/Curriculum26/publication/secondary/AI_Facilitators_Handbook_IX.pdf
2. https://cbseacademic.nic.in/web_material/Curriculum26/publication/secondary/AI_Facilitators_Handbook_X.pdf



AI FOR EVERYDAY LIFE AND WORKPLACES

Artificial Intelligence (AI) is a technology that helps machines think, learn, and solve problems like humans. Today, AI is becoming a part of our everyday life and workplaces. It is used in many important areas such as governance, healthcare, agriculture, education, and finance to make work easier, faster, and more accurate. From online services provided by the government to smart tools used in schools, hospitals, and farms, AI plays an important role in improving how services reach people.

In this lesson, you will learn where and how AI is used in real life through simple examples and everyday situations. You will also explore the Sustainable Development Goals (SDGs) and understand how AI can be used as a tool to support new innovations and solutions for societal good.



OBJECTIVES

After reading this lesson, you will be able to:

- identify and explain in simple contexts where AI is used in everyday life
- explore the impact of AI in various industrial sectors like governance, healthcare, education, agriculture, and finance
- develop an understanding of the Sustainable Development Goals (SDGs)
- recognize the role of AI in supporting innovation for inclusive and sustainable development
- understand System Mapping and how it is done
- explore ways in which AI can be leveraged to improve a system or solve a problem



Notes

4.1 AI APPLICATIONS IN EVERYDAY LIFE

Artificial intelligence is no longer just a concept from science fiction; it is a practical tool that we interact with every single day, often without even realizing it. From the moment we wake up and check our phones to how we manage our finances, navigate traffic, or receive medical care, AI systems are constantly working behind the scenes.

Rather than just looking at the theory behind AI domains like Computer Vision, Natural Language Processing, and Data Analysis, it is much easier to understand them by seeing them in action. Let's explore some common, real-world examples of how AI is making our daily tasks more secure, efficient, and personalized.

1. Face Lock in Smartphones

Smartphones nowadays come with the feature of face locks in which the smartphone's owner can set up his/her face as an unlocking mechanism for it. The front camera detects and captures the face and saves its features during initiation. Next time onwards, whenever the features match, the phone is Unlocked. (See Fig. 4.1 Face Lock in Smartphones)



Fig. 4.1: Face Lock in Smartphones

2. Smart assistants

Smart assistants like Siri, Alexa, and Google Assistant use Artificial Intelligence to understand what we say, respond correctly, and perform tasks for us. They act like digital helpers who can listen, think, and reply using AI technologies – especially Natural Language Processing (NLP) and Machine Learning. (See Fig. 4.2 Example of Smart Assistant)

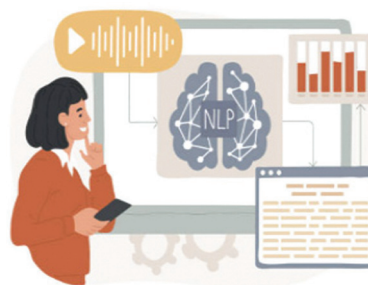


Fig. 4.2: Example of Smart Assistant

3. Fraud and Risk Detection

Finance companies were fed with bad debts and losses every year. However, they had a lot of data which used to get collected during the initial paperwork while



sanctioning loans. They decided to bring in data scientists to rescue them from losses. Over the years, banking companies learned to divide and conquer data via customer profiling, past expenditures, and other essential variables to analyse the probabilities of risk and default. Moreover, it also helped them to push their banking products based on customer's purchasing power. (See Fig. 4.3 Checks of Fraud Detection)



Fig. 4.3: Checks of Fraud Detection

4. Medical Imaging

For the last decades, computer supported medical imaging applications have been a trustworthy help for physicians. It doesn't only create and analyse images but also becomes an assistant and helps doctors with their interpretation. The application is used to read and convert 2D scan images into interactive 3D models that enable medical professionals to gain a detailed understanding of a patient's health condition. (See Fig. 4.4 Medical Imaging Application)



Fig. 4.4: Medical Imaging Application

5. Real time transaction monitoring for digital payments

UPI is one of the most widely used digital payment systems in India, supported by the Government of India and NPCI. Artificial Intelligence (AI) is used in the background to make UPI transactions safe, fast, and reliable. UPI handles millions of payments every day. AI helps monitor these transactions in real time to detect system errors, reduce transaction failures, and ensure smooth functioning of digital payments. (See Fig. 4.5 Real time transaction monitoring for digital payments)



Notes



Fig. 4.5: Digital Payments

6. Content Creation and Smart Chatbots

Generative AI (GenAI) refers to AI systems that can create entirely new content – like text, images, or music – based on simple instructions or “prompts.” Tools like ChatGPT, Google Gemini, and AI image generators are widely used today to write emails, brainstorm ideas, summarize long articles, or even create artwork. Instead of just searching for information, GenAI acts like a digital brainstorming partner, generating fresh, human-like responses and creative drafts in seconds.



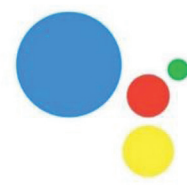
“Hey Cortana”



“Hey Alexa”

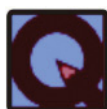


“Hey Siri”



“Hey Google”

Fig. 4.6: Smart Chatbots



INTEXT QUESTIONS 4.1

Multiple-choice questions:

1. Which AI technology is used to unlock smartphones using face recognition?
 - (a) Natural Language Processing
 - (b) Data Analysis
 - (c) Computer Vision
 - (d) Robotics



2. Voice assistants like Alexa and Google Assistant mainly use which AI technology to understand our speech?
 - (a) Computer Vision
 - (b) Machine Hardware
 - (c) Natural Language Processing
 - (d) Internet Connectivity
3. How does AI help in identifying spam emails and messages?
 - (a) By blocking the internet
 - (b) By studying patterns and words in large amounts of data
 - (c) By deleting all messages automatically
 - (d) By changing the email address
4. Navigation apps use AI mainly to do which of the following?
 - (a) Send text messages
 - (b) Play music
 - (c) Store photos
 - (d) Analyse traffic data and suggest faster routes
5. How does AI help online learning platforms or apps recommend content?
 - (a) By randomly selecting videos
 - (b) By studying user interests and learning data
 - (c) By asking every user directly
 - (d) By copying content from books

Now, let us look at the impact of AI on various industrial sectors in detail.

4.2 AI IN HEALTHCARE

Artificial Intelligence is transforming the healthcare industry by helping doctors, hospitals, and patients in smarter and faster ways. AI does not replace doctors – it works with them to improve diagnosis, treatment, and patient care. With the ability to analyse large amounts of medical data quickly, AI is becoming an important tool for saving lives and improving health services.

4.2.1 How AI is Transforming Healthcare

AI has been improving the healthcare sector in several powerful ways:

Early Disease Detection

AI plays a major role in identifying diseases at an early stage by studying medical images such as X-rays, CT scans, and MRI reports. These images contain small



Notes

patterns that may be difficult for the human eye to notice, especially in the early stages of diseases like cancer, tuberculosis, or pneumonia. AI systems are trained on thousands of similar images, allowing them to quickly recognise early warning signs.

Early detection is extremely important because it gives patients a better chance of recovery. When a disease is found early, treatment can begin sooner and is often less costly and less risky. AI helps doctors save time, act faster, and provide the right treatment before the disease becomes serious.

Faster and More Accurate Diagnosis

Doctors often need to study reports, lab test results, and medical images to diagnose illnesses. This takes time and may sometimes lead to confusion, especially when symptoms look similar across diseases. AI systems can analyse large amounts of medical data within seconds, spotting patterns that humans might miss. This makes the diagnosis process faster and more reliable.

With AI's support, doctors can confirm diagnoses more confidently and reduce chances of error. AI is especially helpful in emergencies, where quick decisions can save lives. It ensures that patients get the correct treatment at the right time.

Personalised Treatment

Every patient is unique – different age groups, lifestyles, medical histories, and body responses mean that one treatment may not work the same for everyone. AI helps create personalised treatment plans by analysing past medical records, test results, and even data from wearable devices like smartwatches. It studies what has worked for similar patients and suggests the most suitable treatment options.

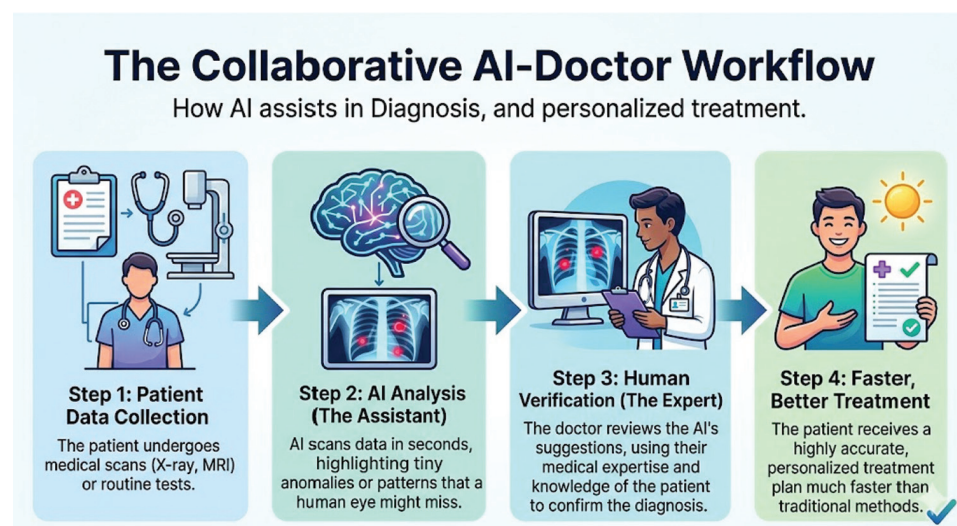


Fig. 4.7: Personalised Treatment



This personalised approach helps doctors choose medicines, dosages, and therapies that are safer and more effective for individual patients. AI also monitors patient progress and alerts doctors if changes need to be made, ensuring continuous and improved care.

Smarter Patient Care

AI is making patient care more convenient and accessible. AI-powered chatbots and mobile apps help patients track their health, ask basic medical questions, and get reminders for medicines, exercise, or follow-up appointments. This is especially helpful for elderly patients or people who live far away from hospitals.

These tools offer support 24/7, reducing the need for frequent hospital visits for small issues. Patients feel more in control of their health, and doctors can also use data from these apps to better understand a patient's condition. AI makes healthcare more organised, personalised, and patient-friendly.

Improved Hospital Management

Hospitals handle a huge amount of information every day – patient records, appointment schedules, operation theatre bookings, equipment availability, and medicine stock. Managing all of this manually is difficult and time-consuming. AI systems help hospitals organise these tasks more efficiently by automatically updating records, scheduling appointments, and tracking medical supplies.

This reduces waiting times for patients and frees up doctors and nurses to focus on treatment instead of paperwork. AI also helps predict patient load, prepare emergency plans, and ensure that important equipment is available on time. As a result, hospitals become smarter, faster, and more efficient.

Generative AI is also helping doctors with their most time-consuming task: paperwork. GenAI tools can securely listen to a doctor-patient conversation and instantly generate a well-organized clinical note or discharge summary. This frees up doctors and nurses to focus entirely on the patient rather than typing on a computer.

4.2.2 Importance of AI in Healthcare

AI is important because it helps solve challenges such as:

- large amounts of medical data that need quick analysis,
- shortage of doctors in many regions,
- late detection of serious diseases, and



Notes

- need for faster and safer treatment decisions.

By providing timely insights, AI supports healthcare workers and improves patient outcomes.

4.2.3 Advantages of AI in Healthcare

- Quick decision-making during emergencies
- Reduced human error in analysing medical images
- Better access to healthcare for people in rural areas
- Lower costs for diagnosis and treatment
- Continuous monitoring through wearable devices

These advantages make healthcare safer, faster, and more reliable.

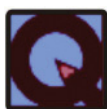
4.2.4 Real-life examples of AI in Healthcare

In India

- **Qure.ai:** Interprets chest X-rays to detect TB and lung diseases.
- **Aarogya Setu:** Helped identify COVID-19 risk levels across the country.
- **AI-based ICU Management** (Rajasthan): Helps track ventilators and patient needs.

Around the World

- **Google Health:** Detects diabetic eye disease through retinal images.
- **IBM Watson Health:** Supports cancer treatment planning.
- **Da Vinci Surgical Robot:** Assists surgeons during complex surgeries with precision.



INTEXT QUESTIONS 4.2

Multiple-choice Questions:

1. What is one major way AI helps doctors in healthcare?
(a) By replacing all doctors (b) By giving medicines automatically
(c) By building hospitals (d) By analysing medical data quickly



2. AI can study medical images such as X-rays and MRIs to:
 - (a) Change the picture colours
 - (b) Detect diseases at early stages
 - (c) Entertain patients
 - (d) Make the image larger
3. Which of the following helps AI improve hospital management?
 - (a) Playing music for patients
 - (b) Designing hospital buildings
 - (c) Growing food for patients
 - (d) Managing patient records and schedules
4. During COVID-19, AI helped the government by:
 - (a) cleaning hospitals
 - (b) Building new roads
 - (c) Teaching online classes
 - (d) Predicting hotspots and infection risks
5. What is an advantage of using AI in healthcare?
 - (a) It increases waiting time
 - (b) It slows down diagnosis
 - (c) It helps detect diseases early
 - (d) It makes hospitals more crowded

4.3 AI IN EDUCATION

Artificial Intelligence is transforming how students learn and how teachers teach. It makes learning more personalised, engaging, and accessible for everyone. From recommending practice questions to translating languages instantly, AI helps create a smarter and more supportive learning environment. In many ways, AI acts like a helpful assistant that understands students' needs and supports teachers in managing classrooms more efficiently.

4.3.1 How AI is Transforming Education

Here are some examples of how AI is impacting the education sector:

- **Personalised Learning**

AI tailors learning content according to each student's strengths and weaknesses. For example, if a student struggles with fractions but is strong



Notes

in geometry, the learning app gives easier fraction problems and more practice in that area. It adjusts the difficulty level based on the student's progress, making learning neither too easy nor too hard.

This helps students learn at their own pace instead of following a one-size-fits-all approach. Personalised learning ensures that every child gets the support they need, which boosts confidence and improves understanding. Teachers can also see detailed reports showing how each student is performing.

- **Smart Content and Digital Learning Tools**

AI can create digital textbooks, interactive quizzes, summaries, and video lessons that make learning more interesting. It can even generate practice tests automatically based on the topics a student needs to review. Tools like these help students revise better and stay engaged with the subject.

AI-powered platforms also support different learning styles – visual learners can watch videos, while others may prefer reading summaries or solving quizzes. This flexibility makes learning more enjoyable and effective for a wide range of students.

- **Smart Content and Generative AI Tools**

AI can create digital textbooks, interactive quizzes, summaries, and video lessons that make learning more interesting. Today, Generative AI is taking this a step further by allowing teachers to instantly generate customised lesson plans, or helping students summarize long chapters into easy-to-read bullet points. It can even generate creative writing prompts or practice tests automatically based on the topics a student needs to review.

- **Digital Doubt Solving and Tutoring**

AI chatbots and virtual tutors are available anytime to help students with their questions. Instead of waiting for the next class or a teacher's reply, students can ask the AI tutor and get explanations immediately. The AI can provide step-by-step solutions, examples, and even videos to clarify doubts.

This makes learning continuous and removes the hesitation some students may have when asking questions in class. It also helps revise concepts quickly before exams or tests.

- **Automated Assessments and Feedback**

AI can evaluate quizzes, assignments, and even short answers more quickly than humans can. It checks for patterns, accuracy, grammar, or problem-



solving steps. Students receive instant feedback, allowing them to correct mistakes and improve faster.

Teachers save valuable time as they no longer have to manually grade every paper. They can instead focus on designing better lessons and supporting students who need extra help.

- **Classroom and School Management**

AI tools help schools manage timetables, attendance, homework submissions, and communication with parents. For example, face recognition can mark attendance automatically, and AI systems can schedule classes more efficiently.

These tools reduce administrative work for teachers, giving them more time to focus on teaching and interacting with students. Schools also benefit from smoother operations and better monitoring of student progress.

- **Accessibility and Inclusive Education**

AI supports students with special needs by providing customised tools. Speech-to-text helps students who have difficulty writing, while text-to-speech helps students who struggle with reading. AI can also translate content into different languages, helping students who come from various backgrounds.

This ensures that learning is fair and inclusive, allowing every student – regardless of ability – to participate fully in the classroom.

4.3.2 Importance of AI in Education

AI addresses several challenges in traditional classrooms:

- Students learn at different speeds but receive the same instruction.
- Teachers often manage large class sizes and heavy workloads.
- Students may hesitate to ask questions in class.
- Learning resources may not always match a student's level.

AI helps solve these problems by providing personalised support, reducing teacher burden, and making learning more engaging and inclusive.

4.3.3 Advantages of AI in Education

- Learning becomes personalised and adaptive.
- Students receive instant feedback and support.



Notes

AI for Everyday Life and Workplaces

- Teachers save time on routine tasks like grading or attendance.
- Learning materials become more interactive and fun.
- Students with special needs get better accessibility tools.

These improvements help create a more effective, student-friendly learning environment.

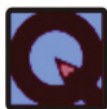
4.3.4 Real-life Examples of AI in Education

In India

- **DIKSHA** – Provides AI-powered digital learning resources for students and teachers.
- **Bhashini** – Uses AI to translate educational content into Indian languages.

Around the World

- **Google Classroom** – Uses AI to organise assignments and provide suggestions.
- **Duolingo** – Uses AI to adjust language learning lessons based on user responses.



INTEXT QUESTIONS 4.3

Multiple-choice Questions:

1. How does AI support personalised learning?
 - (a) By giving the same content to every student
 - (b) By adjusting lessons based on each student's performance
 - (c) By removing teachers from classrooms
 - (d) By making exams easier
2. What is one major advantage of AI chatbots in education?
 - (a) They help students get answers instantly
 - (b) They grade only final exams
 - (c) They replace all school staff
 - (d) They decide school fees



3. Automated assessments using AI help teachers by:
 - (a) Making more manual work
 - (b) Slowing down the grading process
 - (c) Checking assignments quickly and accurately
 - (d) Removing the need to give homework
4. Which of the following is an example of AI making learning inclusive?
 - (a) Text-to-speech for visually challenged students
 - (b) Making classrooms larger
 - (c) Giving every student the same lesson
 - (d) Removing exams
5. Which Indian platform uses AI to offer digital learning resources?
 - (a) Netflix (b) DIKSHA (c) FASTag (d) Zomato

4.4 AI IN AGRICULTURE

Agriculture is one of the most important sectors in India, and Artificial Intelligence is helping improve farming in many new and powerful ways. Farmers face challenges like unpredictable weather, pests, plant diseases, poor soil health, and limited resources. AI technologies help farmers make better decisions by analysing data from the soil, weather, crop patterns, and satellite images. This leads to healthier crops, better yields, and reduced losses.

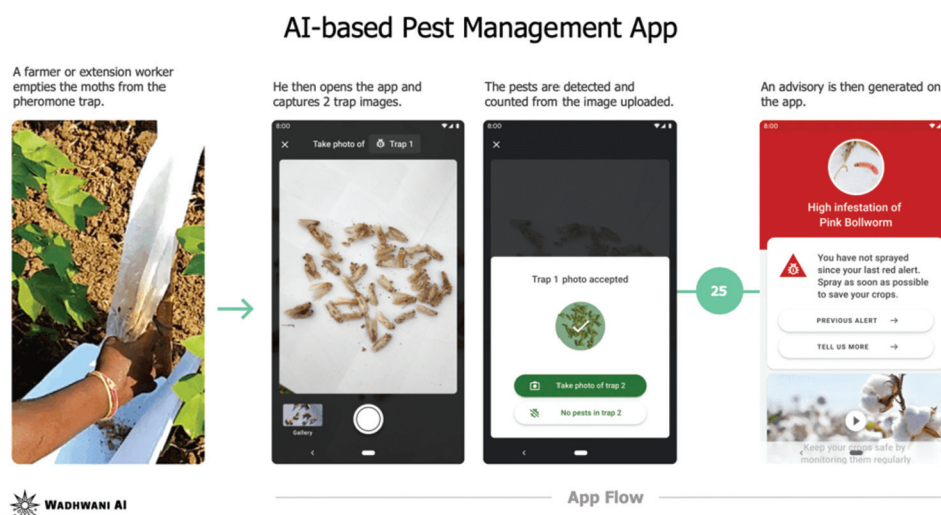


Fig. 4.8: AI in Agriculture



Notes

4.4.1 How is AI Transforming Agriculture

AI is transforming agriculture from traditional methods to smart, modern farming that saves time, money, and effort. Here's how:

- **Early Crop Disease Detection**

AI tools analyse images of crop leaves to identify signs of diseases or pests before they spread. Using mobile apps or drones, farmers can capture photos of their plants, and AI immediately highlights infected areas. This helps farmers take action quickly and prevents major damage to crops. Early detection is especially important in crops like tomatoes, cotton, and wheat, where diseases can spread very fast.

These AI systems learn from thousands of images, so they can recognise even tiny spots that may be difficult for the human eye to notice. This saves money, protects yields, and reduces the need for excessive pesticides.

- **Soil and Weather Analysis**

AI studies soil data – such as moisture, nutrients, and pH level – to help farmers understand what their soil needs. AI systems also analyse weather forecasts, rainfall predictions, temperature changes, and humidity. These insights help farmers decide the best time to plant, irrigate, or harvest crops.

By combining soil information with weather data, AI helps farmers choose the right type of seeds, the right fertilisers, and the right planting schedule. This reduces wastage and improves crop quality.

- **Smart Irrigation and Water Management**

AI-powered irrigation systems measure soil moisture levels and weather conditions to provide crops with the exact amount of water they need. This prevents overwatering or underwatering and helps conserve water, which is especially important in drought-prone areas.

Smart irrigation systems can automatically turn water supply on or off, making farming more efficient. They ensure that every drop counts, reducing costs and protecting the environment.

- **Better Yield Prediction**

AI studies past crop data, weather patterns, soil conditions, and farming practices to predict the expected crop yield. This helps farmers plan their sales, storage, and finances more effectively. Accurate yield prediction also helps governments and markets prepare for supply and demand.



Yield prediction tools can alert farmers about potential risks, such as low rainfall or pest attacks, giving them time to respond and protect their crops.

- **Automation and Smart Farming Tools**

AI-powered machines and robots help automate tasks like planting seeds, removing weeds, monitoring crop growth, and even harvesting in large farms. Drones equipped with cameras can scan fields from above, providing a quick and detailed view of the entire farm.

These tools reduce the farmer's workload, save time, and improve productivity. Automation is especially helpful for older farmers or those managing large areas of land.

4.4.2 Importance of AI in Agriculture

AI helps solve challenges that farmers face every day, such as unpredictable weather, pest attacks, poor soil health, and rising costs. It allows farmers to make decisions based on accurate data rather than guesswork. AI also makes farming more sustainable by reducing wastage of water, fertilisers, and pesticides.

In a country like India, where crores of people depend on farming, AI can significantly improve food quality, crop production, and farmers' income. It supports rural development and strengthens the entire agricultural ecosystem.

4.4.3 Advantages of AI in Agriculture

- Early detection of pests and diseases
- Higher crop yields and better-quality produce
- Efficient use of water, fertilisers, and pesticides
- Reduced cost of operations
- Better planning and risk management
- Improved income and stability for farmers

4.4.4 Real-life examples of AI in Agriculture

In India

- **CropIn:** Provides AI-based crop monitoring using Drone technology and health alerts.
- **Fasal:** Uses AI to give irrigation and advice on pesticide.



Notes

AI for Everyday Life and Workplaces

- **Microsoft and ICRISAT Project:** Predicts sowing dates and weather conditions for Indian farmers.
- **Kisan e-Mitra (Rajasthan):** AI-powered robots that assist in farming tasks.

Around the World

- **John Deere's AI tractors:** Automatically detect weeds and spray only where needed.
- **Plantix App:** Identifies plant diseases using AI image recognition.
- **IBM's Watson Decision Platform:** Helps farmers manage crops through weather and soil analytics.



INTEXT QUESTIONS 4.4

Multiple-choice Questions

1. How does AI help farmers detect crop diseases early?
 - (a) By tasting the crops
 - (b) By studying images of leaves to find signs of disease
 - (c) By planting seeds automatically
 - (d) By watering the field
2. AI-powered irrigation systems are useful because they:
 - (a) Water the fields randomly
 - (b) Use more water than needed
 - (c) Provide the right amount of water based on soil moisture
 - (d) Stop the use of fertilisers
3. Which of the following is an example of AI being used in agriculture?
 - (a) Drones scanning fields to check crop health
 - (b) A scarecrow placed in the field
 - (c) A farmer shouting to scare birds
 - (d) Manual hand-picking of weeds



4. What does AI analyse to help farmers decide the best time to plant or harvest?
 - (a) Movie reviews
 - (b) Weather and soil data
 - (c) Music playlists
 - (d) Internet speed
5. Which Indian AI tool helps analyse crop images and detect diseases?
 - (a) CropIn
 - (b) Google Maps
 - (c) WhatsApp
 - (d) Uber

4.5 AI IN FINANCE

The finance sector includes banks, digital payments, insurance companies, and stock markets. Artificial Intelligence is transforming this sector by making transactions safer, faster, and more accurate. People use digital payments and online banking every day, and AI works silently in the background to keep these systems secure and reliable.

AI also helps customers receive better financial services through personalised suggestions, fraud detection, smart chatbots, and automatic loan approvals. It reduces human errors, saves time, and supports millions of users across India in making safe and smart financial decisions.

4.5.1 How is AI transforming Finance

- **Fraud Detection and Security**

Banks handle lakhs of transactions every minute, and some may be risky or suspicious. AI analyses transaction patterns to quickly spot unusual activity – like sudden large withdrawals or payments from unexpected locations. When AI detects something abnormal, it alerts the bank or blocks the transaction instantly.

This protects customers from losing money to fraud or cyberattacks. AI also strengthens online banking security by verifying user identity through face recognition, fingerprints, or one-time passwords.

- **Smarter Digital Payments**

AI is behind many features used in digital payment apps such as UPI, Paytm, PhonePe, and Google Pay. These apps use AI to process payments quickly, detect errors, and ensure secure transactions. AI also predicts common payments (like mobile recharge or electricity bills) and reminds users at the right time.



Notes

By studying user behaviour, AI identifies patterns and prevents failed transactions. This makes digital payments smooth, reliable, and widely accessible across India.

- **Personalised Financial Services**

AI helps banks and apps suggest the right products to customers – such as savings plans, insurance policies, or investment options. It analyses user spending habits, income, and financial goals to recommend tailored solutions.

This personalised approach helps people make better financial decisions without confusion. For example, AI may suggest a budget plan for someone who overspends or recommends a fixed deposit for someone who wants to save safely.

- **Automated Loan Approvals**

Traditionally, applying for a loan took several days because banks had to manually check documents, income details, and credit history. AI now speeds up this entire process by analysing all needed information in seconds.

It checks how likely a person is to repay the loan based on their financial records and spending behaviour. This makes loan approvals faster, fairer, and more accurate – especially for students, small businesses, and first-time borrowers.

- **Chatbots and Customer Support**

A chatbot is an application software or web interface which facilitates to interact/search through text or voice. Banks and financial apps use AI chatbots to answer customer questions instantly. These chatbots can check account balances, help reset passwords, explain fees, and clarify loan details – anytime, without waiting in long queues.

AI chatbots reduce the workload on customer care teams and provide 24/7 support. They help people get quick solutions, especially in urgent situations.

4.5.2 Importance of AI in Finance

- Detects fraud and keeps digital transactions safe
- Makes banking faster and more convenient
- Helps customers manage money smartly
- Provides personalised financial advice
- Reduces paperwork and long waiting times
- Helps banks make accurate decisions



- Supports digital financial inclusion in rural areas
- Improves the overall safety and stability of the financial system

4.5.3 Advantages of AI in Finance

- Better security through fraud detection
- Faster services like instant payments and quick loan approvals
- Personalised recommendations based on user behaviour
- 24/7 customer support via chatbots
- Improved accuracy in identifying risks and errors
- Convenient digital access even for people in remote areas

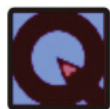
4.5.4 Real-life Examples of AI in Finance

In India

- **UPI Apps** – Use AI to detect fraud and recommend payments.
- **SBI's SIA Chatbot** – Helps customers find information instantly.
- **RBI's Fraud Monitoring Systems** – Track suspicious transactions across banks.
- **Insurance AI Tools** – Used by companies for faster claim approvals.

Around the World

- **Mastercard AI security:** Detects fraudulent card transactions.
- **Robinhood:** Uses AI to analyse stock market trends.
- **American Express:** Uses AI to prevent suspicious spending in real time.



INTEXT QUESTIONS 4.5

Multiple-choice Questions

1. Which of the following is an example of AI in digital payments?
 - (a) Using chalk to write account numbers
 - (b) UPI apps detecting suspicious transactions
 - (c) Counting cash manually
 - (d) Printing passbooks



Notes

2. Why is AI important in finance?
 - (a) It increases paperwork
 - (b) It slows down digital payments
 - (c) It replaces all banks
 - (d) It helps make safer and smarter financial decisions
3. How does AI personalise financial services?
 - (a) By giving the same advice to everyone
 - (b) By suggesting saving or investment plans based on user habits
 - (c) By removing all investment options
 - (d) By increasing ATM fees
4. AI-powered chatbots in banking help customers by:
 - (a) Cooking food for them
 - (b) Answering questions instantly
 - (c) Locking bank doors
 - (d) Playing music in the branch
5. How does AI help prevent fraud in banking?
 - (a) By decorating the bank
 - (b) By analysing transactions to spot unusual activity
 - (c) By closing accounts
 - (d) By reducing the number of customers

4.6 AI IN GOVERNANCE

Governance refers to how governments plan, manage, and deliver services to citizens. Artificial Intelligence is becoming an important tool that helps governments work more efficiently, make better decisions, and improve public services. With India's large population and vast geography, AI supports government departments in analysing big data, predicting future needs, and solving problems faster.

AI strengthens digital governance by making public services faster, more accurate, and more citizen friendly. It helps improve areas such as public safety, traffic management, pollution monitoring, welfare schemes, and emergency response systems. By reducing human error and automating routine work, AI allows officials to focus on solving high-priority challenges.



4.6.1 How is AI transforming Governance

Smarter Public Services and Decision Making

Governments collect massive amounts of data on healthcare, education, transportation, pollution, and welfare schemes. AI studies this data to help officials make informed decisions. For example, AI can identify which districts need more schools, where water shortages may occur, or which communities require extra healthcare support. This helps the government plan resources more efficiently.

AI also helps monitor the performance of government programs by analysing feedback and finding areas that need improvement. This leads to better service delivery, reduced delays, and more transparency in governance. Citizens benefit from faster and more reliable public services.

Improved Public Safety and Law Enforcement

AI-powered surveillance systems help monitor crowded areas, detect unusual activities, and support police in maintaining public safety. For example, smart CCTV cameras can recognise number plates, track traffic violations, or spot abandoned objects. AI tools can also predict crime hotspots by analysing past crime data, helping police plan patrols more effectively.

In emergencies – such as natural disasters, fires, or accidents – AI helps identify affected areas quickly and supports faster rescue operations. These tools save lives and make cities safer for everyone.

Better Traffic and Urban Management

City traffic control centres use AI to study live traffic camera feeds and adjust signal timings to reduce congestion. AI also predicts busy hours, helps design safer roads, and recommends alternate routes. This improves travel time, reduces pollution, and makes transportation systems more efficient.

Urban governance also benefits from AI in areas like waste management, water distribution, and electricity usage. By studying usage patterns, AI helps cities manage resources and prevent shortages.

Efficient Delivery of Welfare Schemes

Governments run many welfare schemes for students, farmers, senior citizens, and low-income households. AI helps identify eligible beneficiaries, detect duplicate or fake applications, and ensure that benefits reach the right people on time. This improves fairness and reduces corruption.

AI also helps track scheme performance by analysing who is benefiting and what improvements are needed. This ensures transparency and better utilisation of public funds.



Notes

Automating Routine Government Work

Government offices manage huge amounts of paperwork – applications, certificates, permissions, and reports. AI-powered systems can automatically read documents, sort data, and respond to basic citizen queries. Chatbots on government websites help people check application status, ask questions, and access services quickly.

Automation reduces the workload on government staff, speeds up processes, and improves accuracy in official work.

4.6.2 Importance of AI in Governance

- Helps governments analyse large amounts of public data
- Improves speed and efficiency of public services
- Reduces corruption and increases transparency
- Enhances safety through surveillance and early warnings
- Supports faster disaster response and relief planning
- Helps monitor and evaluate welfare schemes
- Automates repetitive tasks, saving time and resources
- Supports evidence-based decision-making

4.6.3 Advantages of AI in Governance

- Smarter cities with better traffic, water, and waste management
- Safer communities through improved surveillance
- Faster access to government services
- Accurate identification of people who need welfare support
- Better planning through data-driven policies
- Reduced manual errors in government systems

4.6.4 Real-life Examples of AI in Governance

In India

- **FASTag Traffic Monitoring:** Cameras automatically read number plates and track toll payments.
- **Delhi & Hyderabad Smart CCTV Systems:** Detect traffic violations and monitor public safety.



- **Aadhaar Authentication:** AI helps verify identities quickly and securely.
- **DigiLocker:** Uses AI to help citizens access important documents easily.
- **Rajasthan's AI-Based Crime Mapping:** Predicts crime hotspots for better policing.
- **UMANGApp:** AI-enabled platform to access multiple government services.

Around the World

- **Singapore Smart Nation:** Uses AI for traffic control, waste management, and public safety.
- **UK's NHS Data Systems:** Use AI for predicting disease outbreaks.
- **USA Smart Policing Tools:** Analyse crime patterns to support law enforcement.



INTEXT QUESTIONS 4.6

Multiple-choice questions

1. What is one major way AI helps in governance?
 - (a) By building shopping malls
 - (b) By helping governments analyse large amounts of public data
 - (c) By teaching subjects in schools
 - (d) By replacing all government officers
2. Which of the following is an example of AI improving public services?
 - (a) AI predicting crime hotspots
 - (b) AI painting public buildings
 - (c) AI delivering newspapers
 - (d) AI choosing names for streets
3. How does AI help during natural disasters?
 - (a) By stopping the rain
 - (b) By predicting affected areas and planning faster rescue operations
 - (c) By moving people on its own
 - (d) By building new cities



Notes

4. Which of the following is an example of AI improving urban management?
 - (a) AI checking movie tickets
 - (b) AI adjusting traffic signals to reduce congestion
 - (c) AI closing all city roads
 - (d) AI planting trees
5. Digi Locker is an example of AI helping citizens by:
 - (a) Storing important documents digitally and securely
 - (b) Printing certificates
 - (c) Taking school attendance
 - (d) Doing online shopping

4.7 SUSTAINABILITY

Natural Resources

We use energy to move cars on the road, bake a cake in the oven, light our homes, and for a lot of other things! Natural resources are the raw materials and sources of energy – land, air, water, minerals, fossil fuels, etc. These resources provide us with energy in different forms- Light, Heat, Motion, Sound.

Activity 1

Circle objects that burn fuel, and rectangle around the things that need people power to move in Fig. 4.9.



Fig. 4.9: Example of Sustainability



Activity 2

Which of the following objects provides us with light energy in Fig. 4.10?

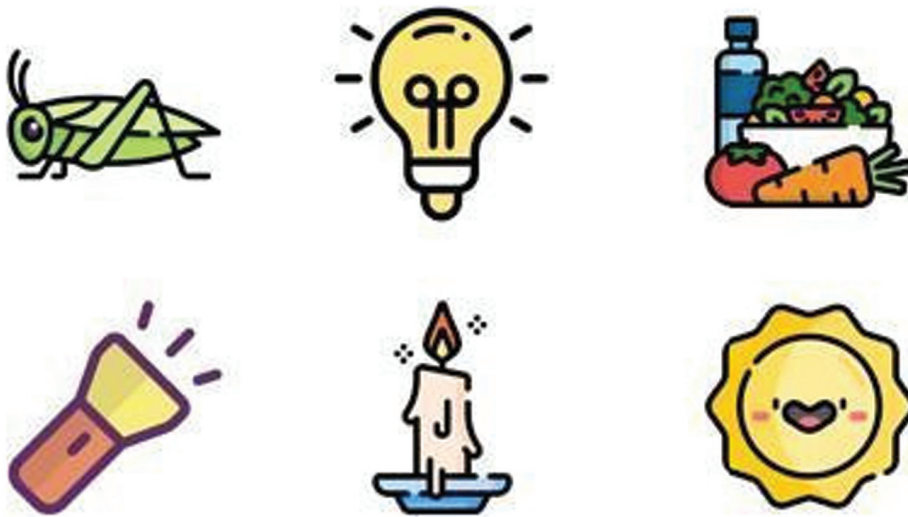


Fig. 4.10: Example of Light

Answer: Following are the objects that provide us with light energy



Our natural resources are the key to sustainability.

Sustainability

The word “Sustainability” comes from the word “Sustain.” Do you know what “Sustain” means? It means to maintain, support, withstand for a long period. Thus, sustainability means, maintaining the world we live in. The key idea is – we must act responsibly so that the natural resources on our planet can support future generations to come. Let’s understand what is meant by using resources responsibly.

Imagine that you want to make a sandwich! To make your sandwich you need – bread, your favourite jam & butter. When you enter the kitchen, you realize that there is no jam! Your elder sister comes in and says the jam got finished when she



Notes

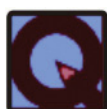
had made her sandwich earlier. She further adds that if she knew that you would be making sandwiches later, she would have left some jam for you.

The concept of saving resources for future generations is similar. Our Earth has limited resources. The land, air, water, and food, that provide us with energy are not unlimited. If we use all of them or pollute them, coming generations might not be able to use them. Sustainability tries to solve this problem. Sustainability encourages everyone around us to use resources in a responsible manner by sharing resources equally and not being wasteful.

Society basically refers to people around us. To be truly sustainable, we need everyone in our society to follow sustainability. Imagine if some segments of society are not sustainable –

- Can we truly achieve full sustainability?
- No, that would not be possible

Hence, we need to think of sustainability from a societal point of view.



INTEXT QUESTIONS 4.7

Let's Think

Answer the following questions:

- Do you think you live a sustainable life?
- How can you become more sustainable?
- Which natural resources around you are threatened by wastage or pollution?

Note: Students will relate real life scenarios around them and answer the above questions.

4.8 SUSTAINABLE DEVELOPMENT GOALS (SDGS)

Implementation of the SDGs is a shared responsibility among governments, the private sector, civil society, and individuals. Progress and success towards achieving the SDGs are measured through various indicators and targets set for each goal.

Developed countries are expected to provide financial resources, technology transfer, and capacity-building support to developing countries to help them achieve the SDGs.



Individuals can contribute to the achievement of the SDGs by raising awareness, adopting sustainable lifestyles, volunteering, advocating for policy changes, and supporting organizations working towards the goals.

To address these global challenges, the United Nations established 17 distinct yet interconnected Sustainable Development Goals. Each goal has a series of targets and indicators to track progress until 2030. (See Fig. 4.11)



Fig. 4.11: Sustainability Development Goals

Who is responsible for implementing the SDGs?

Implementation of the SDGs is a shared responsibility among governments, the private sector, civil society, and individuals. Progress and success towards achieving the SDGs are measured through various indicators and targets set for each goal.

Developed countries are expected to provide financial resources, technology transfer, and capacity-building support to developing countries to help them achieve the SDGs.

Individuals can contribute to the achievement of the SDGs by raising awareness, adopting sustainable lifestyles, volunteering, advocating for policy changes, and supporting organizations working towards the goals.

The 17 SDG Goals are:

Descriptive List of the 17 SDGs:

- GOAL 1: No Poverty:** End poverty in all its forms everywhere. This involves ensuring all people have access to basic resources, services, and opportunities.



Notes

2. **GOAL 2: Zero Hunger:** End hunger, achieve food security and improved nutrition, and promote sustainable agriculture. This means making sure everyone can afford nutritious food.
3. **GOAL 3: Good Health and Well-being:** Ensure healthy lives and promote well-being for all at all ages. Focuses on topics like reducing child mortality, fighting diseases, and promoting mental health.
4. **GOAL 4: Quality Education:** Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all. Aims to provide all children and adults with access to learning.
5. **GOAL 6: Clean Water and Sanitation:** Ensure availability and sustainable management of water and sanitation for all. Focuses on safe drinking water and effective sanitation systems.
6. **GOAL 5: Gender Equality:** Achieve gender equality and empower all women and girls. Works towards ending discrimination and violence against females.
7. **GOAL 7: Affordable and Clean Energy:** Ensure access to affordable, reliable, sustainable, and modern energy for all. Promotes the use of clean energy sources like solar and wind power.
8. **GOAL 8: Decent Work and Economic Growth:** Promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all. Focuses on job creation and fair working conditions.
9. **GOAL 9: Industry, Innovation and Infrastructure:** Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation. Relates to building strong factories, connecting places, and encouraging new ideas.
10. **GOAL 10: Reduced Inequalities:** Reduce inequality within and among countries. Works to close gaps in income, opportunity, and privilege based on gender, age, race, or ability.
11. **GOAL 11: Sustainable Cities and Communities:** Make cities and human settlements inclusive, safe, resilient, and sustainable. Aims for green spaces, good transportation, and affordable housing.
12. **GOAL 12: Responsible Consumption and Production:** Ensure sustainable consumption and production patterns. Focuses on reducing waste, using eco-friendly materials, and recycling.



13. **GOAL 13: Climate Action:** Take urgent action to combat climate change and its impacts. Involves global efforts to reduce carbon emissions and adapt to a changing climate.
14. **GOAL 14: Life Below Water:** Conserve and sustainably use the oceans, seas, and marine resources for sustainable development. Aims to reduce pollution and overfishing.
15. **GOAL 15: Life on Land:** Protect, restore, and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss. Protects animals and plants on land.
16. **GOAL 16: Peace, Justice and Strong Institutions:** Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all ages. Promotes peace, fairness, and reliable systems.
17. **GOAL 17: Partnerships for the Goals:** Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development. Emphasizes collaboration between countries, companies, and people.

Let's look more closely at the relationship between these goals. While they may seem independent, the "SDG wedding cake" (See Fig. 4.10) concept, as shown in the diagram on the right, visualizes their deep interconnectedness. This model shows how progress in one goal often depends on and supports progress in others. It groups the goals into three broad, layered categories from the bottom up:

- **Biosphere** (e.g., Goals 6, 13, 14, 15): The parts of Earth where life exists, providing the fundamental resources.
- **Society** (e.g., Goals 1, 2, 3, 4, 5, 7, 11, 16): A group of people living together, who rely on the Biosphere.
- **Economy** (e.g., Goals 8, 9, 10, 12, 17): The production, consumption of goods and services, and management of resources, which must serve the society.

This concept states that to achieve a sustainable world we must do the following: "The economy must serve the society within the resources provided by our Earth (Biosphere)" (See Fig. 4.12)



Notes

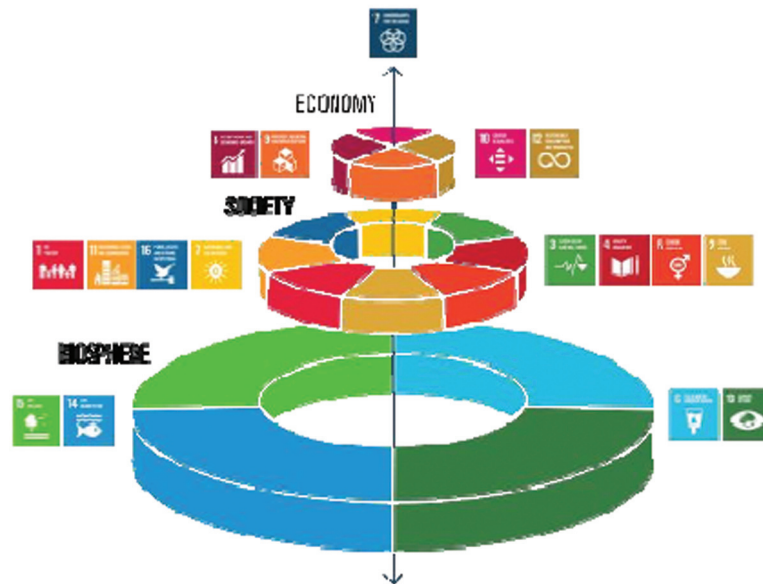


Fig. 4.12: Example of Sustainability



INTEXT QUESTIONS 4.8

SDG Quiz

In this activity, you will be assessing your knowledge on Sustainable Development Goals.

Follow the steps below to get started:

1. Visit the link <https://www.globalgoals.org/quiz/>
2. Step 2: Enter your name and click on the Submit button
3. Your quiz begins! Swipe left or right to choose your answer and continue till you finish the quiz.

4.9 SYSTEMS THINKING AND SYSTEM MAP

Systems thinking

In this section, we will see that a lot of problems which appear to be straightforward are not so straightforward. Let us understand this using the following example

Problem: Large outbreak of Malaria in Borneo in 1950s

The people of Borneo reached out to WHO (World Health Organization) for help. WHO sent in pesticides (DDT) to control the outbreak of malaria. This did control spread of malaria for a while However, the following happened as well:



- Outbreak of Plague
- Roofs of huts breaking down!

You must be wondering; how can this happen! Watch the video to know more – <https://www.youtube.com/watch?v=17BP9n6g1F0>

A lot of times, straightforward solutions can lead to unintended consequences as we saw in the last example. To avoid such situations, we can use a style of thinking known as **Systems Thinking**.

Systems Thinking is a holistic approach to understanding and addressing complex problems. Rather than focusing on individual parts in isolation, systems thinking considers how various elements interact within a larger system to produce outcomes. It emphasizes the interconnectedness of components and recognizes that changes in one part can have ripple effects throughout the entire system.

- A system is simply a group of interconnected things.
- Systems behave differently from their individual components.
- Systems are represented visually using Systems Map.
- This simply means that its components interact and affect each other. This feature of systems is known as interconnectedness.
- Example: Water Cycle, School, the Digestive System, food chains

Basics of System Map

System Maps are useful when there are multiple factors affecting an issue - we call these factors, “elements.” In a systems map, all elements are connected by relationships - we represent these by “arrowhead lines.” (See Fig. 4.13).

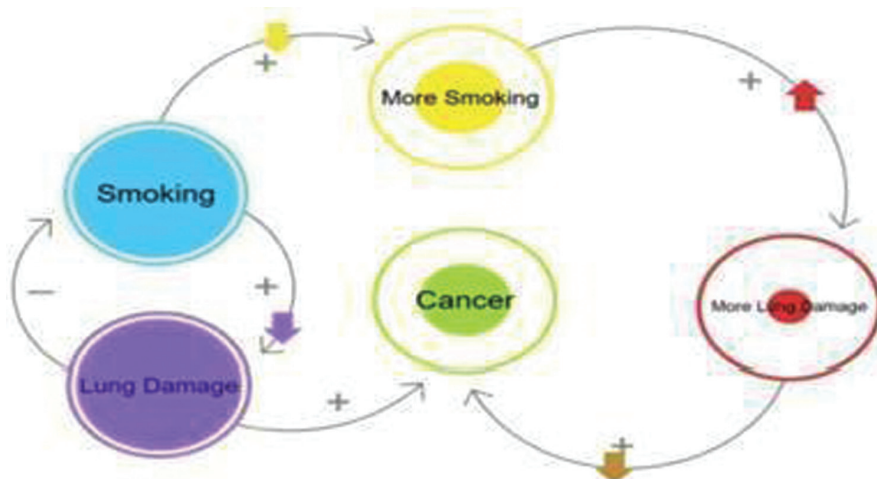


Fig. 4.13: Relationships among elements in System Map



Notes

AI for Everyday Life and Workplaces

The loops indicate the direction and degree of feedback. Changing elements or changing relationships between elements changes the system outcome. Positive Relationships (direct relation) are represented by a “+” sign.

e.g., An increase in element “X” leads to an increase in element “Y”

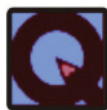
Negative Relationships (inverse relation) are represented by a “-” sign.

e.g., An increase in element “X” leads to a decrease in element “Y”

In a system, every element is interconnected. In a systems map, we try to represent that relationship using arrows. Within a system map, we will identify loops. These loops are important because they represent a specific chain of causes and effects. A system typically has several chains of causes and effects. You may notice that some arrows are longer than others. A longer arrow represents a longer time for a change to happen. We also call this a time delay.

To change the outcome of a system, as a change maker, we have two options - change the elements in a system or change the relationships between elements. It is usually more effective to change the relationship between elements in a system.

Systems Mapping Provide participants with a complex system (e.g., a local ecosystem, a transportation network, a business organization). Have them create a visual representation of the system using a systems map. This exercise helps to identify the various components of the system and understand how they are interconnected. The system map given in the above figure demonstrates the relationship between smoking, lung damage, and cancer.



INTEXT QUESTIONS 4.9

Make your own System Map

In this activity, you will learn to make a System Map. Follow the steps below to get started:

1. Visit the link <https://ncase.me/loopy/>
2. Click on the option ‘Make a model from scratch’
3. You can see the examples and click on the ‘How to’ button to understand how it works.
4. Now, create a system map for the following problems and play its simulation to see how elements impact each other – (See Fig. 4.14).



NOW PLAY WITH AN EXAMPLE:



Fig. 4.14: Example of Creating System Map

- Create a system map depicting the relation between the number of units of electricity consumed in a house and the electricity bill.
- Create a system map depicting the factors which affect the drying of wet clothes.
- Create a system map showing water cycle

Using AI as leverage in a systems map

Leverage represents our best opportunity to affect a change in the system. The leverage we can use depends on our skills and ability to influence. For example, as youth, we are unlikely to effect changes on large issues like war, climate change etc., but we can influence it. However, with AI skills (particularly in getting AI to predict), we might be able to help people in charge take informed decisions.

Let's look at a problem and then try to apply Systems Thinking to it.

What are Corals and Coral Reefs?

Let's look at video to learn what are corals and coral reefs - <https://www.youtube.com/watch?v=ZiULxLLP32s>

Corals are tiny animals which do not have a backbone (invertebrate). Corals secrete a material called Calcium Carbonate, clusters of which form the bulk of coral reefs. Coral Reefs are essentially skeletons of large colonies of corals present on the outside. Even though coral reefs, cover less than 1% of the ocean floor, they are home to more than 25% of marine creatures.

What is Coral Bleaching?

- The exoskeletons of corals form the bulk of coral reefs
- Coral reefs also include tiny plants known as algae
- The reefs are sensitive to changes in temperature and pollution



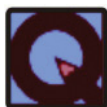
Notes

AI for Everyday Life and Workplaces

- When temperatures rise or pollution increases, the algae expel themselves from the reefs causing the coral to turn white

This process is called coral bleaching.

Let's try to understand the effect of pollution & temperature change on coral reefs through an activity.



INTEXT QUESTIONS 4.10

Do-It-Yourself Activity

Follow the steps given below to get started with this activity –

PART A

1. Go to the website – <https://climatekids.nasa.gov/coral-bleaching/>
2. Press the '+' or '-' buttons to increase or decrease the water temperature or pollution levels
3. You can also click on 'Start Storm' to see what happens when a storm occurs
4. Notice how different variations in the factors stated above affect the coral reefs

Note that the symbiotic relationships are relationships between two different organisms who stand to benefit by staying together. In the case of coral reefs, the algae get a home in the reefs, whereas corals receive nutrients from the algae. Corals can survive a bleaching event, but they are under more stress and are susceptible to diseases.

PART B

1. Draw a Systems Map on /
2. Scroll down and select the 'Make a model from scratch option' (as shown in the image attached beside)
3. List down the important elements that should be part of system map (The physical structure of the reef, water temperature, Extreme Weather Events etc)
4. Based on how you think a particular component is related to another, draw a positive feedback loop or a negative feedback loop

- Remember, a component could affect one or more than one component
- Figure out where you could use AI as a leverage! Compare the system map drawn with the one below.



Notes

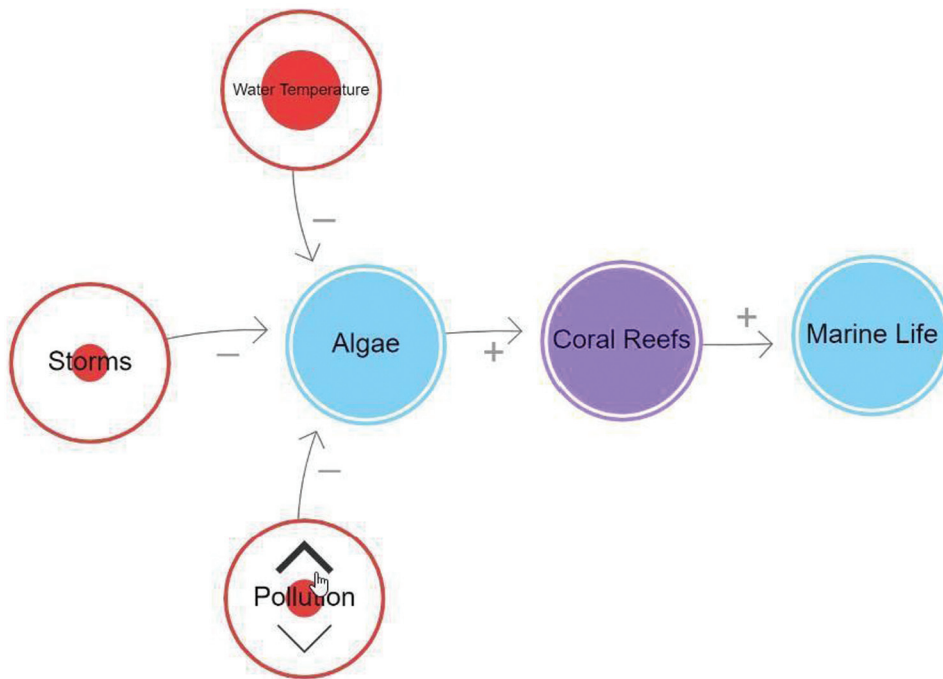


Fig. 4.15: Example of System Map in Coral Bleaching

Can you see how the relationship between different components has been accurately represented?

- Coral Reefs
- Algae
- Pollution
- Water Temperature
- Storms
- Marine Life

4.10 IMPACT OF CHANGE IN REAL-LIFE SITUATIONS

An interactive machine learning experiment by Sey Min that visualises what we might lose and what will remain with the rising global temperatures. She has



Notes

AI for Everyday Life and Workplaces

trained a machine learning model on thousands of Google Search images, clustering by the type of species. As you increase the temperature, discover what happens to 62 different animal species - and reveal the new Anthropocene species or waste, we leave.

Let's experiment with the Climate Change Impact Filter by Sey Min in collaboration with Google Arts & Culture Lab on the link - <https://experiments.withgoogle.com/climate-impact-filter>

Activity Guidelines

1. Visit the link <https://experiments.withgoogle.com/climate-impact-filter> and click on START
2. Go through the tutorial for the experiment to understand how to work.
3. Choose any of the options given in the header row. Let's assume you choose Plant
4. Now, under Plant, choose one of the options given. Let's assume you choose Barley
5. As you move the temperature bar on the right up and down, you will notice how the change in temperature affects the species. In some cases, you may also imagine what will happen if the species do not exist.
6. Keep trying the experiment for various species and temperature effects on them.



WHAT HAVE YOU LEARNT

- AI is used in everyday life and in important sectors such as:
 - AI in Healthcare: AI helps doctors detect diseases early and provide better, faster treatment.
 - AI in Education: AI makes learning personalised and supports students with instant help.
 - AI in Agriculture: AI helps farmers improve crop health, save resources and increase yield.
 - AI in Finance: AI keeps digital payments safe and makes banking quicker and smarter.
 - AI in Governance: AI improves public services, safety, and decision-making for citizens.



- The Sustainable Development Goals (SDGs) and their importance for society and the environment.
- Sustainability encourages everyone around us to use resources in a responsible manner by sharing resources equally and not being wasteful.
- Systems Thinking is a holistic approach to understand and address complex problems.
- Various elements of a system interact within a system and System Maps help visualise the dependency and relationships among them
- Rising global temperatures affect biodiversity, focusing on the potential loss of species and ecosystems.



TERMINAL QUESTIONS

Short-Answer Questions

1. How does AI help in early disease detection?
2. What is personalised learning in AI-powered education?
3. How does AI improve digital payments like UPI?
4. What is a System Map?
5. What is 'leverage' in systems thinking?
6. Name two advantages of AI in governance.
7. Give one way AI helps farmers.

Long-Answer Questions

1. How can AI increase farm productivity and sustainability?
2. Discuss ethical or practical challenges of using AI in public services.
3. What is systems thinking and how can AI be used as leverage in a systems map?
4. Explain how AI helps prevent fraud in finance and supports financial inclusion.
5. Describe three ways AI makes classroom learning better.



Notes

Skill Based Questions

1. Design a simple AI solution to reduce food waste in your school canteen.
2. Critically evaluate: 'AI will replace teachers and doctors in the future.' Do you agree?
3. A small town has poor access to healthcare. Propose a low-cost AI intervention to improve services and explain implementation steps.



ANSWERS TO INTEXT QUESTIONS

4.1

1. (c)
2. (c)
3. (b)
4. (d)
5. (b)

4.2

1. (d)
2. (b)
3. (d)
4. (d)
5. (c)

4.3

1. (b)
2. (a)
3. (c)
4. (a)
5. (b)

4.4

1. (b)
2. (c)
3. (a)
4. (b)
5. (a)

4.5

1. (b)
2. (d)
3. (b)
4. (b)
5. (b)

4.6

1. (b)
2. (a)
3. (b)
4. (b)
5. (a)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- use AI applications in everyday life i.e. healthcare, agriculture, education, finance, governance
- explain sustainability and sustainable goals for society and the environment
- describe System Thinking and System Map



LOGIC BUILDING AND BASIC PROGRAMMING CONCEPTS

In the age of Artificial Intelligence, the ability to solve problems systematically and translate those solutions into computer instructions is paramount. This lesson serves as the critical bridge, taking the essential problem-scoping and logical thinking skills developed in the AI curriculum and applying them to formal programming. Before writing complex AI models, a strong foundation in logic and basic programming constructs (like loops and data types) is essential. This chapter will equip you with the fundamental tools - algorithms, flowcharts, and basic Python syntax to start creating structured, executable code.



OBJECTIVES

After learning this lesson, you will be able to:

- define an algorithm and explain its key characteristics;
- design sequential, conditional, and iterative logic using pseudocode;
- represent algorithms graphically using standard flowchart symbols;
- identify and apply basic Python syntax, data types (int, float, string, boolean), and variables;
- utilize conditional statements (if-elif-else) and iterative statements (for, while) to control program flow;
- develop and execute simple Python programs to solve real-world logical problems.



Notes

5.1 INTRODUCTION

In today's digital world, computers, smartphones, and even simple devices like calculators and smartwatches work because they follow a set of instructions. These instructions are created by humans using a process called programming. But before we write any program, we first need to learn how to think logically and how to break a problem into smaller steps. This skill is known as logic building.

Logic building is the foundation of all programming. It helps us organise our thoughts, understand a problem clearly, and plan the exact steps needed to solve it. Just like solving a puzzle or preparing a recipe, programming also requires a clear sequence of actions. If we think in a confused or random way, the computer cannot produce the correct result. That is why developing strong logical thinking is the first step toward learning how to code.

5.2 LOGIC, LOGIC DEVELOPMENT, AND REASONING

Before we learn how to write algorithms or programs, we must first learn how to think clearly and step-by-step. Computers follow instructions exactly as we give them, so our thinking needs to be organised, logical, and error-free. This is where logic, logic development, and reasoning become important.

5.2.1 What is Logic?

Logic is the ability to think in a clear, sensible, and orderly manner. It helps us understand:

- What is true and what is not
- Why something happens
- What the correct next step should be

For example, if the bus is late, then you might decide to call a cab. Your decision is based on logical thinking. In programming, logic ensures that every step of a solution follows a proper sequence.

5.2.2 What is Logic Development?

Logic development means improving our ability to solve problems by breaking them into smaller, manageable steps. It includes skills such as:

- Identifying the problem
- Finding patterns



- Planning steps in the correct order
- Checking if the steps make sense

Good logic development helps us create strong algorithms and write correct programs. It is similar to planning a recipe before cooking – if the order is wrong, the final result will not turn out well.

5.2.3 What is Reasoning?

Reasoning is the ability to analyse information, understand relationships, and make correct conclusions. It helps us answer questions like:

- Why did this happen?
- What should I do next?
- Does this solution make sense?

There are two main types of reasoning:

- Logical reasoning – solving problems using rules and patterns
- Analytical reasoning – examining information and understanding how ideas connect

For example, if all squares have four equal sides ! and a shape has four equal sides→ then we reason that the shape might be a square.

Reasoning helps us judge whether our algorithm or program will work correctly.

5.2.4 Importance of Logic and Reasoning in Programming

Logic and reasoning are essential because they help us:

1. **Understand problems clearly:** Before writing a program, we must know exactly what needs to be solved.
2. **Write correct algorithms:** Algorithms must follow the right sequence. Logic ensures no step is missed or misplaced.
3. **Debug** (find and fix errors): When a program does not work, reasoning helps identify what went wrong and how to correct it.
4. **Think in a step-by-step way:** Computers follow steps in order, so we must think sequentially too.
5. **Build confidence in problem-solving:** Strong logic makes programming easier, smoother, and more enjoyable.



Notes

5.2.5 Logic in Real-life

Logic is not used only in programming – it is used everywhere:

- Planning a school project
- Solving maths problems
- Following a recipe
- Making a timetable
- Deciding the shortest route to school

Whenever we think clearly and choose the best steps, we are using logic.



INTEXT QUESTIONS 5.1

Fill in the blanks:

1. is the ability to think clearly and understand the correct order of steps in a solution.
2. helps us analyse information and make correct conclusions.
3. When we break a big problem into smaller steps, we are using thinking.
4. Logical reasoning often uses and rules to reach a conclusion.
5. Logic, reasoning, and logic development form the of programming.

5.3 ALGORITHM DESIGN AND PSEUDOCODE

5.3.1 Concept of an Algorithm

A program is a set of unambiguous steps required to solve a specific problem. To write a logical step-by-step method to solve the identified problem is called an algorithm, in other words, an algorithm is a procedure for solving problems. To solve a mathematical or computer problem, this is the first step of the procedure. An algorithm includes calculations, reasoning, and data processing. Algorithms can be presented by natural languages, pseudocode, flowcharts, etc.

An algorithm is based on Input → Process → Output



To understand an algorithm better, let us take an example of calculating average.

Basic steps to calculate average are:

- Input a list of numbers,
- calculate the sum,
- count the numbers,
- divide the sum by the count,
- output the result

Characteristics of a good algorithm:

- **Finite:** An algorithm must terminate after a finite number of steps. It cannot run infinitely.
- **Definite:** Each step must be precisely and clearly defined. Ambiguity is not allowed.
- **Effective:** Every operation must be simple enough to be executed practically.
- **Input-based:** It must take zero or more external inputs.
- **Output-oriented:** It must produce one or more outputs, which are the desired results.

Algorithms are an integral part of our lives.

5.3.2 Introduction to Pseudocode

Pseudocode is a simplified, English-like language used to represent the steps of an algorithm without using the strict syntax of any specific programming language. It is a tool for developers to plan the logic before writing the actual code, making the design clear and easily understood by anyone familiar with programming.

Some of the standard Pseudocode keywords that are used to structure the logic are START, INPUT, PROCESS, IF-THEN-ELSE, WHILE, FOR, OUTPUT, and END.

For Example, let us look at a Pseudocode for checking if a number is positive, negative, or zero:

START	● Starting the Pseudocode
INPUT Number X	● Get the number input in X variable
IF $X > 0$ THEN	● Check if entered number X is greater than 0
OUTPUT "The number is positive."	● If the condition above becomes true, then output is Positive



Notes

Logic Building and Basic Programming Concepts

ELSE IF $X < 0$ THEN

- If the previous condition is not True, check again if X is smaller than 0

OUTPUT “The number is negative.”

- If the condition above becomes true, then output is negative

ELSE

- This gets executed if neither of the conditions above are true

OUTPUT “The number is zero.”

- If the condition above becomes true, then output is zero

END IF

- End the ‘If condition’ section

END

- Ending the Pseudocode



INTEXT QUESTIONS 5.2

Multiple-choice Questions:

1. An algorithm is defined as:
 - (a) A machine that solves problems
 - (b) A set of unambiguous steps to solve a problem
 - (c) A long computer program
 - (d) A type of software
2. Which of the following is NOT a characteristic of a good algorithm?
 - (a) It should be finite
 - (b) It should be definite
 - (c) It should be confusing
 - (d) It should give output
3. Which of the following BEST describes pseudocode?
 - (a) Code written in a specific programming language
 - (b) A simplified English-like way to write algorithm steps
 - (c) Drawings used to show program flow
 - (d) Mathematical equations

5.4 FLOWCHARTS AND VISUAL LOGIC

5.4.1 What is a Flowchart?

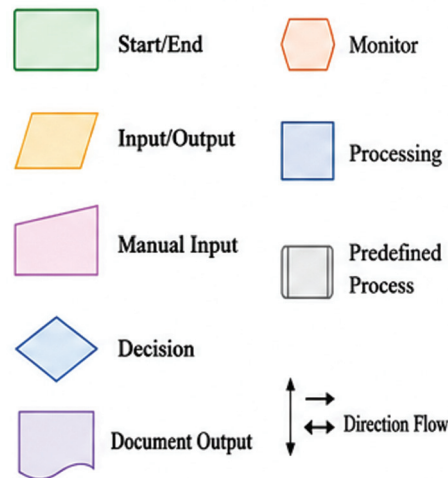
A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows in order to demonstrate a process or a program. (See Fig. 5.1) While computers work with numbers at ease, humans need visual representations to understand the information well and communicate



it effectively. Thus, flowcharts are used to break a process into smaller parts and elaborate it using visual representations.

Standard Flowchart Symbols

- Terminal (Start/End)
- Input/Output (Parallelogram)
- Process (Rectangle)
- Decision (Diamond)
- Flow Lines (Arrows)



The graphics above represent different parts of a flowchart. The process in a flowchart can be expressed through boxes and arrows with different sizes and colours. In a flowchart, we can easily highlight a certain element and the relationships between each part. Let us look as to how we can use flowcharts to represent algorithms.

Fig. 5.1: Flowchart Symbols

5.4.2 Drawing Basic Flowcharts

How to Use Flowcharts to Represent Algorithms

Now that we have the definitions of algorithm and flowchart, how do we use a flowchart to represent an algorithm? Let us look at given examples and see how they work. (See Fig. 5.2)

Example 1: Print 1 to 20

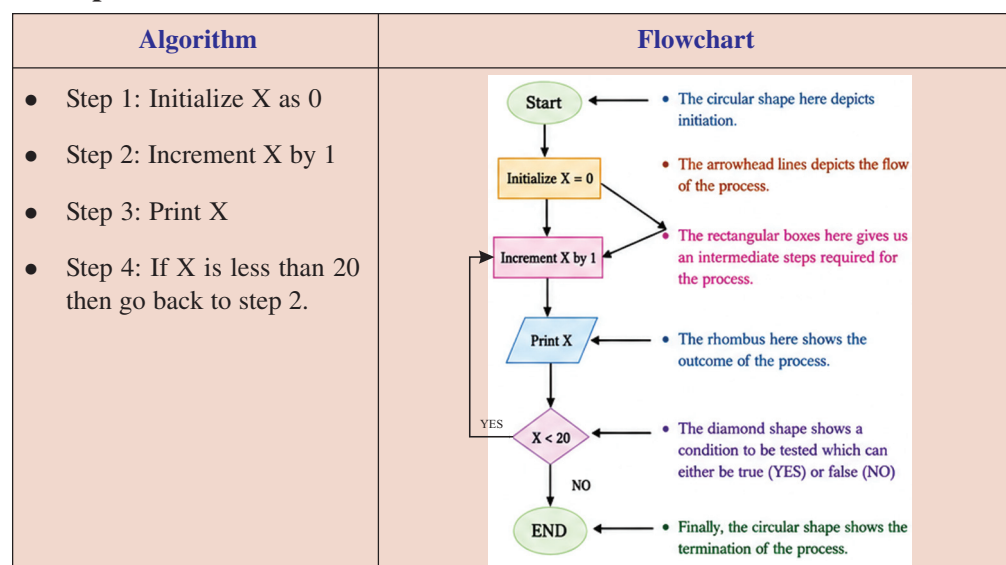


Fig. 5.2: Flowchart for Printing Numbers from 1 to 20



Notes

Example 2: Convert Temperature from Fahrenheit (F) to Celsius (C) (See Fig. 5.3)

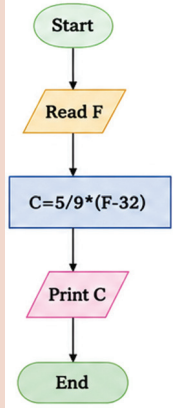
Algorithm	Flowchart
- Step 1: Read temperature in Fahrenheit - Step 2: Calculate temperature with formula $C = 5/9 * (F - 32)$ - Step 3: Print C	 <pre>graph TD; Start([Start]) --> ReadF[/Read F/]; ReadF --> Process["C = 5/9 * (F - 32)"]; Process --> PrintC[/Print C/]; PrintC --> End([End]);</pre>

Fig. 5.3: Flowchart for converting temperature

5.5 PYTHON OVERVIEW AND BASIC CONSTRUCTS

5.5.1 Programming Language

In the previous section, you learnt about the different methodologies for programming. A programming language is a language that specifies a set of instructions that can be used to produce various kinds of output. In simple words, a programming language is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. Though there are many different programming languages such as BASIC, Pascal, C, C++, Java, Haskell, Ruby, Python, etc. we will study Python in this lesson.

5.5.2 What is a Program?

A program is a step-by-step collection of instructions written in a language that a computer can understand. It tells the computer –

- What task to perform
- How to perform it
- In what order to perform each step

Computers cannot think or make decisions on their own. They only follow the instructions given to them. For example:

- When a calculator app adds two numbers → a program is running.



- When a game character jumps → a program is deciding how high and when.
- When a smart assistant answers your question → a program processes your voice and gives a response.

Just like an algorithm act as the plan and a flowchart act as the diagram, a program is the final working version of those ideas written in a computer-friendly form.

5.5.3 What is Programming?

Programming is the process of designing and writing programs. It includes:

- Understanding the problem
- Breaking it into smaller logical steps (algorithm)
- Representing the steps using a flowchart (optional but helpful)
- Writing the steps in a programming language
- Checking and correcting errors
- Running the program to get the final output

Programming is basically teaching the computer how to solve a problem in a logical way.

5.5.4 Need of Programming

Computers are powerful, but they don't automatically know what to do. Programming helps to

- automate tasks (like marking attendance or calculating bills)
- build apps and websites
- create games and animations
- control robots and AI systems
- solve real-life problems using logic and creativity

Almost every digital tool we use – messaging apps, music players, online learning platforms, search engines – works because someone programmed it.

There are several benefits of learning how to code, and they are

- Improves logical thinking
- Builds problem-solving skills



Notes

Logic Building and Basic Programming Concepts

- Encourages creativity
- Helps develop analytical thinking
- Opens doors to building apps, games, AI tools, and more

Programming teaches students how to think, not just how to write code.

Computers do not understand human languages; they only understand machine language, which is made up of binary codes (0s and 1s). However, writing complex instructions using only 0s and 1s is incredibly difficult, slow, and error-prone for humans. To make programming easier, we use special high-level languages that look more like English, such as Python, Scratch, JavaScript, C++, and more. These languages help us write instructions in a form that is easy for humans to understand. A program written in these languages is later converted by a translator into 0s and 1s so the computer can execute it. Before getting into the details of Python, let's first understand what Python is and why it is so popular.

5.5.5 What is Python?

Python is a high-level programming language that is used all over the world to create software, websites, games, apps, and even Artificial Intelligence (AI) systems. It is one of the most popular programming languages today because it is easy to read, easy to write, and easy to learn, especially for beginners.

Python uses simple English-like words, which makes it less complicated than many other programming languages. Those words in Python are referred to as Syntax.

5.5.6 Why Python for AI?

Artificial intelligence is a breakthrough technology. You can see so many applications around you. If you as an individual can also plan to develop an AI application, you will require to know a programming language. There are various programming languages like Lisp, Prolog, C++, Java and Python, which can be used for developing AI Systems. Out of these, Python gains a maximum popularity because of the following reasons:

- **Easy to learn, read and maintain:** Python has few keywords, simple structures and a clearly defined syntax. A program written in Python is fairly easy to maintain.
- **A broad standard library:** Python has a huge bunch of libraries with plenty of built-in functions to solve a variety of problems.
- **Interactive mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.



- **Portability and Compatibility:** Python can run on a wide variety of operating systems and hardware platforms.
- **Extensible:** We can add low-level modules to Python interpreter. These modules enable programs to customize their tools to be more efficient.
- **Databases and Scalable:** Python provides interfaces to all major open source and commercial databases along with a better structure and support for large programs than Shell Scripting.

5.5.7 Applications of Python Programming

Python is powerful and can be used for many purposes such as:

- Web application development
- Business applications
- Games and 3D graphics
- Software Development
- Desktop and GUI applications
- App development
- Robotics
- Data analysis and Machine Learning
- Automation of daily tasks



INTEXT QUESTIONS 5.3

Multiple-choice questions

1. What is the purpose of a flowchart?
 - (a) To play games on a computer
 - (b) To write code in a programming language
 - (c) To visually represent the steps of an algorithm
 - (d) To test internet speed
2. Which symbol in a flowchart is used to show the start or end of a process?
 - (a) Diamond
 - (b) Rectangle
 - (c) Oval
 - (d) Parallelogram



Notes

3. A program is best described as:
 - (a) A random set of computer files
 - (b) A set of step-by-step instructions for a computer
 - (c) A drawing made on a computer
 - (d) A computer virus
4. Why do we need programming?
 - (a) To slow the computer down
 - (b) Because computers understand human languages
 - (c) To automate tasks and solve problems using computers
 - (d) To increase electricity usage
5. Python is widely used for:
 - (a) Calculations
 - (b) Artificial Intelligence and data science
 - (c) Cleaning Software
 - (d) Repairing machines

5.6 GETTING STARTED WITH PYTHON

Python is a cross-platform programming language, meaning, it runs on multiple platforms like Windows, MacOS, Linux and has even been ported to the Java and .NET virtual machines. To write and run Python program, we need to have Python interpreter installed in our computer.

5.6.1 Downloading and Setting up Python for use

Follow the steps given below to download and setup up a Python Compiler and Interpreter on your system:

- Download Python from python.org using link www.python.org/downloads
- Select appropriate download link as per Operating System [Windows 32 Bit/ 64 Bit, Apple iOS, Linux]

If you are on Linux/Ubuntu, use the following commands:



```
Update package list
sudo apt update

# Install Python 3
sudo apt install python3

# Verify installation
python3 --version
```

- For Windows 64 Bit OS, follow the required steps as shown in Fig. 5.4.

5.6.2 Python IDLE installation

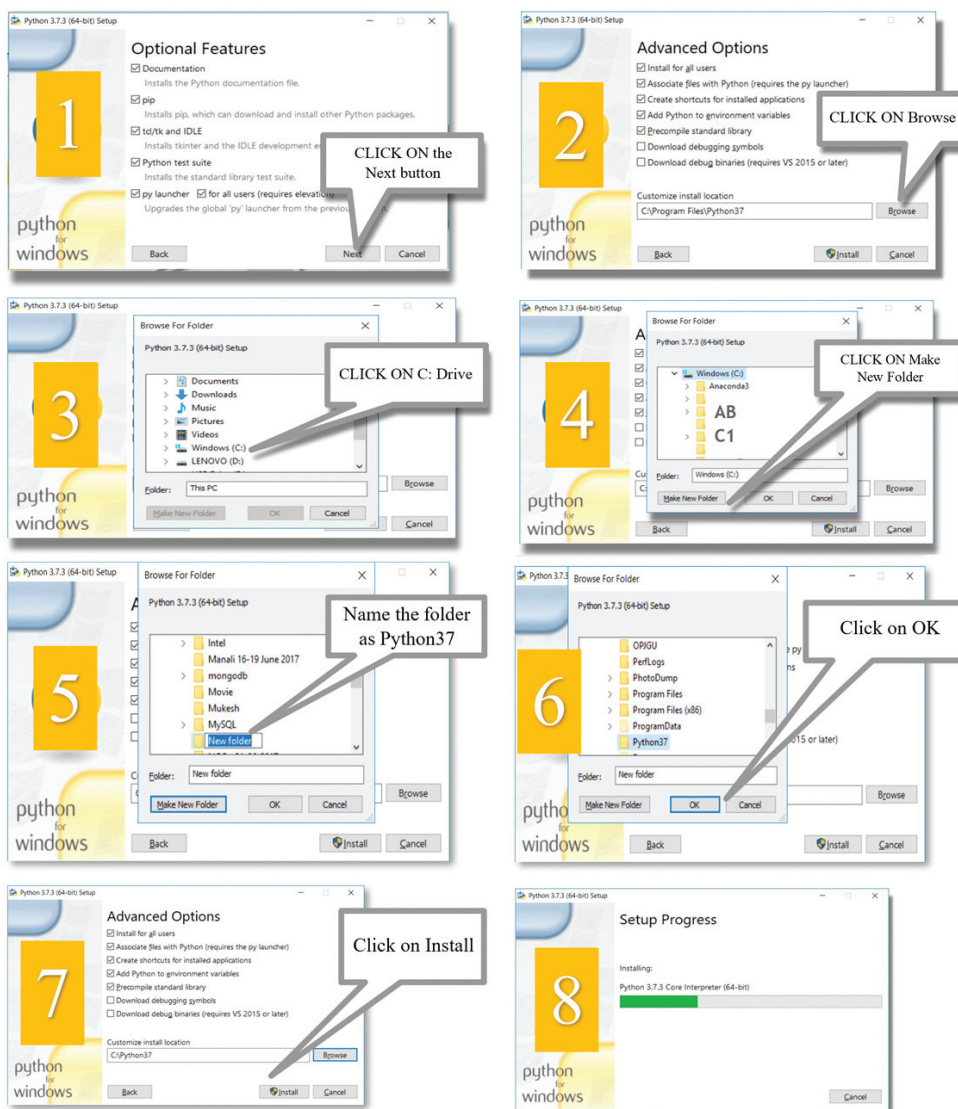


Fig. 5.4: Steps for installation of Python on Windows



Notes

5.6.3 Run in the Integrated Development Environment (IDE)

When we install Python, an IDE named IDLE is also installed. We can use it to run Python on our computer. IDLE (GUI integrated) is the standard, most popular Python development environment. IDLE is an acronym of Integrated Development and Learning Environment. It lets one edit, run, browse, and debug Python Programs from a single interface. This environment makes it easy to write programs. Python shell can be used in two ways, viz., interactive mode and script mode. Where Interactive Mode, as the name suggests, allows us to interact with OS; script mode lets us create and edit Python source file.

Interactive Mode

You can see this example, Python IDLE Shell account has `>>>` as Python prompt, where simple mathematical expressions and single line Python commands can be written and can be executed simply by pressing enter. The first expression `3+10` written on the first Python prompt shows 13 as output in the next line. The second expression `2+4*10` written on the second Python prompt shows 42 as output in the next line. The third statement `print ("Hello Learner")` written on the third Python prompt shows Hello Learner as output in the next line. The fourth statement `print ("Result:", 40+5*100)` written on the fourth Python prompt shows Result: as output in the next line.

```
Python 3.11.5 Shell
File Edit Shell Debug Options Window Help
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:3
13:38:19) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>> 3+10
13
>>> 2+4*10
42
>>> print ("Hello Learner")
Hello Learner
>>> print ("Result:", 40+5*100)
Result: 540
>>> |
```

Script Mode

In script mode, we type Python program in a file and then use the interpreter to execute the content from the file. Working in interactive mode is convenient for beginners and for testing small pieces of code, as we can test them immediately. But for coding more than few lines, we should always save our code so that we may modify and reuse the code.



Note: Result produced by Interpreter in both the modes, viz., Interactive and script mode is the same.

Python Script/Program: Python statements written in a particular sequence to solve a problem is known as Python Script/Program.

To write a Python script/program, we need to open a new file - **File → New File**, type a sequence of Python statements for solving a problem, save it with a meaningful name – **File → Save**, and finally run the program to view its output.

Let us type our first program:

Code explanation

- Line 1 in this code starting with # is a comment line, which means the line is non- executable and it is only for the programmer's reference.

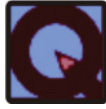
```
CBSE1.py - C:/Python37/Lib/idlelib/CBSE1.py (3.7.3)
File Edit Format Run Options Window Help
#Python Program 1
print("Hello")
Name="Sam"
print("Learner ", end="")
print(Name)
A=50
B=300
print(A, " Times ", B, " is ", A*B)
```

- Line 2 will **display** Hello
- Line 3 will **assign** a string value "Sam" to a variable called 'Name'
- Line 4 will **display** Learner and will allow the next output to get displayed in the same line
- Line 5 will **display** Sam in the same line as 'Learner' in the output
- Line 6 will **assign** an integer 50 to a variable 'A'
- Line 7 will **assign** an integer 300 to a variable 'B'
- Line 8 will **display** 50 times 300 is 15000
- So, the complete output of the Python Code will be:

Hello

Learner Sam

50 times 300 is 15000

**INTEXT QUESTIONS 5.4****Fill in the Blanks:**

1. Python can be downloaded from the official website
2. After installing Python, the default editor that comes with it is called
3. The full form of IDE is
4. A Python file is also known as a Python
5. The window where we type and run code directly in IDLE is called the Python

5.7 PYTHON STATEMENTS AND COMMENTS

In this section we will learn about Python statements, why indentation is important and how to use comments in programming.

5.7.1 Python Statement

Instructions written in the source code for execution are called statements. There are different types of statements in the Python programming language like Assignment statement, Conditional statement, Looping statements etc. These help the user to get the required output. For example, `n = 50` is an assignment statement.

Multi-line statement

In Python, end of a statement is marked by a newline character. However, Statements in Python can be extended to one or more lines using parentheses `()`, braces `{ }`, square brackets `[]`, semi-colon `;`, continuation character slash `\`. When we need to do long calculations and cannot fit these statements into one line, we can make use of these characters.

Examples are:

Table 5.1: Statements in Python

Type of Multi-line Statements	Usage
Using Continuation Character (<code>/</code>)	<pre>s = 1 + 2 + 3 + \ 4 + 5 + 6 + \ 7 + 8 + 9</pre>



Using Parentheses ()	<code>n = ("The only way to \n" "learn to programming language is \n" "by writing code.")</code>
Using Square Brackets []	<code>footballer = ['MESSI', 'NEYMAR', 'SUAREZ']</code>
Using braces { }	<code>x = {1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9}</code>
Using Semicolons (;)	<code>flag = 2; ropes = 3; pole = 4</code>

5.7.2 Python Comments

A comment is text that doesn't affect the outcome of a code, it is just a piece of text to let someone know what you have done in a program or what is being done in a block of code. In Python, we use the hash (#) symbol to start writing a comment.

For example:

```
python
# This is a comment
```

5.7.3 Python Keywords and Identifiers

When we write a program in Python, we use different words and names to give instructions to the computer. These words fall into two important categories: keywords and identifiers. Understanding these is essential before writing any program.

Keywords

Keywords are special words in Python that have a fixed meaning and purpose. They are reserved by the programming language, which means we cannot use them for anything else – such as naming variables, functions, or objects. Python uses keywords to define the structure of a program. Here is the list of all the keywords in Python:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def',  
'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',  
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with',  
'yield']
```



Notes

Identifiers

Identifiers are the names we give to things in our program, such as variables, functions, or classes. They help us label data or code so we can use it later. These names are chosen by the programmer and can be anything meaningful.

Rules for writing identifiers

Python has simple rules for creating valid identifiers:

- Identifiers can contain letters, numbers, and underscores (_).

Example: student_name, rollno1

- They must not start with a number.

Example: age1 is correct, 1age is not

- They cannot contain spaces or special characters.

Example: student_name is correct, student name is not, and marks% is also not correct

- Keywords cannot be used as identifiers.

Example: for = 5 is incorrect as 'for' is a keyword

- Identifiers are case-sensitive.

Age, age, and AGE are all different identifiers.

5.7.4 Variables and Datatypes

When we write programs, we often need to store information so the computer can use it later. For example, a program may need to store a student's marks, a person's name, or the price of a product. To store such information, Python uses variables and data types.

5.7.4.1 Introduction to Variables

A variable is a name that we give to a value stored in the computer's memory. Think of a variable as a container or a labelled box that holds some information. For example:

- Age = 14
- Here, Age is a variable storing the number 14
- Name = "Python"
- Here, Name is a variable storing the text *Python*

Variables allow us to store values, change them when needed, and use them again in our program.



There are some rules that one must follow while naming variables, and they are –

- They cannot start with a number.
Example: age1 is correct, 1age is not.
- They can contain letters, digits, and underscores.
Example: student_name, count1, total_marks
- They cannot contain spaces or symbols like @, #, %, etc.
- They cannot use keywords as names.
- Python is case-sensitive – Name, name, and NAME are different variables.

Here are a few tasks that are commonly done with Variables:

Table 5.2: Using Variables in Python

Task	Sample Code	Output
Assigning a value to a variable	Website = "xyz.com" print(Website)	xyz.com
Changing value of a variable	Website = "xyz.com" print(Website) Website = "abc.com" print(Website)	xyz.com abc.com
Assigning different values to different variables	a,b,c = 5, 3.2, print(a) print(b) print(c)	5 3.2
Assigning same value to different variable	x = y = z = "Same" print(x) print(y) print(z)	Same Same Same

5.7.4.2 Introduction to Datatypes

When we write a program, we work with different kinds of information – numbers, text, true/false values, decimals, etc. To help the computer understand what kind of information we are using, Python classifies data into types. These are called data types.

A data type tells Python:

- What kind of value is stored



Notes

- How much memory it needs
- What operations can be performed on it

For example:

- You can add two numbers
- You cannot add a number to a sentence
- You can join two strings together
- You can compare two Booleans

Let us look at some of the commonly used data types in Python:

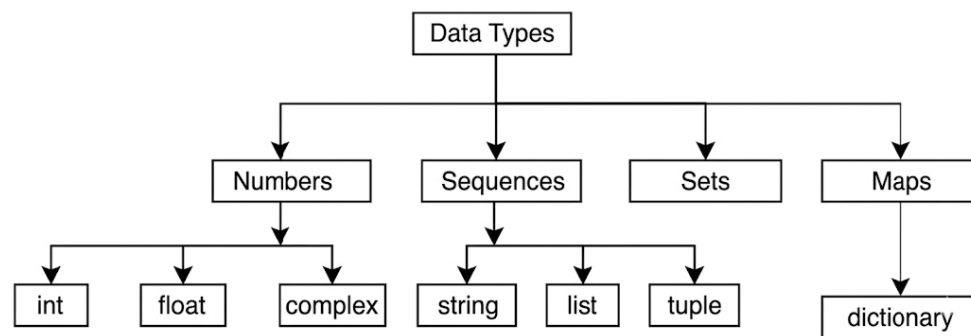


Fig. 5.5: Data Types

1. **Integer (int):** whole numbers as they do not have decimals. Integers allow operations like addition, subtraction, multiplication, and division.

They can be:

- Positive (10, 45, 200)
- Negative (-5, -100)
- Zero (0)

Some uses of integers are:

- Counting students in class
- Calculating score in a game
- Tracking number of items in stock

2. **Float:** numbers with decimal points. Floats can be used in mathematical calculations just like integers.



For example:

- Height = 153.3
- Price = 39.99

Some uses of float numbers are:

- Measuring height, weight, distances
- Handling prices in shopping apps
- Calculating averages or percentages

3. **String (str)** – any text written inside quotes. Strings must be written inside single ‘ ‘ or double “ “ quotes.

A string can contain:

- Letters - “Aarav”
- Numbers written as text - “2025”
- Symbols - “@home!”
- Spaces - “AI for Youth”

Some uses of string datatype are –

- Names, sentences, addresses
 - Chat messages
 - Labels and titles in programs

Strings can be joined (concatenated) using + symbol. For example, the python statement `print(“Hello “ + “World”)`, will give output as **Hello World**.

NOTE: Numbers written as strings are considered as text only and can not be used for mathematical operations without changing to a numerical datatype.

4. **Boolean (bool):** It is a data type that has only two values: **True** or **False**. It is used to represent conditions and make decisions in a program

For example:

- `is_raining = True`
- `is_student = False`



Notes

Some uses of Boolean datatype are:

- Checking login success
- Verifying conditions in if-else statements
- Comparing numbers ($5 > 3$ gives True)

5. **None** type: means No value or empty value. It is used when a variable exists but doesn't have a value yet. This tells the program that result has not been assigned any data.

For example:

- Result = None

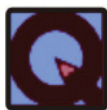
5.7.4.3 Importance of Datatypes

Understanding data types helps you write correct and efficient programs. Data types allow Python to:

- Store values correctly
- Perform correct operations
- Prevent mistakes
- Manage memory properly

For example:

- You **cannot** multiply a string by a string.
- You **can** multiply a number by a string ("hi" * 3 ! "hihihi").
- You **cannot** compare a number with text.



INTEXT QUESTIONS 5.5

Short-answer questions:

1. What do we call a single line of instruction in Python?
2. What does Python ignore while executing code?
3. What are reserved words in Python called?
4. Python identifiers cannot begin with a
5. What symbol is used to assign values to variables?
6. What data type stores text?
7. Which datatype represents "no data" Python?



5.8 PYTHON OPERATORS

Operators are special symbols which represent computation. They are applied on operand(s), which can be values or variables. Same operators can behave differently on different data types. Operators when applied on operands form an expression. Value and variables when used with operator are known as operands.

Python provides six different types of operators: These are:

- Arithmetic operator
- Assignment operator
- Membership operator
- Relational Operator
- Logical operator
- Identity operator

Let us learn about them

5.8.1 Arithmetic Operators

These operators perform arithmetic operations on operands. The list of arithmetic operators is given in Table: 5.3.

Table: 5.3: Arithmetic Operators

Operator	Meaning	Expression	Result
+	Addition	10 + 20	30
-	Subtraction	30 - 10	20
*	Multiplication	30 * 100	300
/	Division	30 / 10	20.0
		1 / 2	0.5
//	Integer Division	25 // 10	2
		1 // 2	0
%	Remainder	25 % 10	5
**	Raised to power	3 ** 2	9

5.8.2 Comparison operators

Comparison operators are symbols in Python that are used to compare two values. They check whether values are equal, greater, smaller, or not equal, and they always give a Boolean result – either True or False. We use comparison operators in conditions, such as in if-else statements, to help the computer make decisions. List of comparison operators is as follows:



Notes

Table 5.4: Comparison Operators

Operator	Meaning	Expression	Result
>	Greater Than	20 > 10	True
		15 > 25	False
<	Less Than	20 < 45	True
		20 < 10	False
==	Equal To	5 == 5	True
		5 == 6	False
!=	Not Equal to	67 != 45	True
		35 != 35	False
>=	Greater than or Equal to	45 >= 45	True
		23 >= 34	False
<=	Less than or equal to	13 <= 24	True
		13 <= 12	False

5.8.3 Logical operators

Logical operators are special operators in Python that are used to combine two or more conditions and decide whether the overall result is True or False. They help the computer make smarter decisions in programs – especially when using if-else statements. Python has 3 logical operators, and they are:

Table 5.5: Logical Operators

Operator	Meaning	Expression	Result
And	• Returns True only if both conditions are True.	(5 > 2) and (10 > 3)	True
	• If any condition is False → result is False.	(5 < 2) and (10 > 3)	False
Or	• Returns True if any one of the conditions is True.	(5 > 10) or (3 == 3)	True
	• Only False when all conditions are False.	(5 > 10) or (3 > 3)	False
Not	• Reverses the result of a condition.	not (5 > 2)	True
	• True becomes False; False becomes True.	not (5==5)	False



5.8.4 Assignment operators

Assignment operators are symbols in Python that are used to assign values to variables. They store data in a variable so we can use it later in the program. The basic assignment operator is the equal to '=' sign. It stores value in a variable.

Compound assignment operators

Python also allows us to update (change) a variable's value in a shorter way. These are called compound assignment operators. Some of them are:

Table: 5.6: Assignment Operators

Operator	Expression	Equivalent to
+=	X +=5	X = X + 5
-=	X -= 5	X = X - 5
*=	X *= 5	X = X * 5
/=	X /= 5	X = X / 5

5.8.5 Identity Operators

These operators are used either to check the type of data contained in a variable or to compare the memory location of two variables. The list of identity operators are given in Table 5.7.

Table 5.7: Identity Operators

Operator	Description	Example	Explanation
is	Returns true if the two variables on either side of the operator point towards the same memory location otherwise it returns False. Note: The memory location of a variable can be determined by the function- id()	<pre>>>> a=10 >>> b=20 >>> id(a) 1452001344 >>> id(b) 1452001504 >>> a is b False >>> b=a >>> id(a) 1452001344 >>> id(b) 1452001344 >>> a is b True</pre>	Here a and b are two variables with values 10 and 20 respectively. As you can see the memory locations of two variables are different thus the statement '<i>a is b</i>' returns False. Now when the statement <i>b=a</i> is executed both <i>a</i> and <i>b</i> point to the same memory location. Thus, the statement '<i>a is b</i>' returns True.



Notes

Logic Building and Basic Programming Concepts

	It is also used to check the type of data contained in a variable.	>>> type(a) is int True	As a contains int type data thus the statement <i>type(a) is int</i> returns True.
is not	Returns true if the two variables on either side of the operator does not point towards the same memory location otherwise it returns false.	>>> a=10 >>> b=20 >>> a is not b True	It returns true as a and b are not pointing towards the same memory location.

5.8.6 Membership Operators

These are used to check whether a specified value/variable is present in a given sequence such as strings, lists, or tuples. The list of membership operators are given in Table

Table 5.8: Membership Operators

Operator	Description	Example
in	Returns true if the specified value/variable is present in a given sequence such as strings, lists, or tuples otherwise it returns false.	>>> str="I am learning Python" >>> "Python" in str True >>> "learN" in str False
not in	Returns true if the specified value/variable is NOT present in a given sequence such as strings, lists, or tuples otherwise it returns false.	>>> l1=[10,20,30] >>> 10 not in l1 False >>> 50 not in l1 True

5.9 PYTHON INPUT AND OUTPUT

A program usually takes an input from any source (user, file, code, etc.), processes the information based on the logic built in the code, and give an output based on the given logic. For example:

Input ☐ **Name** and **Age** is "Amar" and 5;
 Process ☐ Check if Amar's age is less than 4;
 Output ☐ "Amar can go to school".



5.9.1 Python Output Using print() function

We use the print() function to output data to the standard output device (screen). We can also output data to a file. An example is given below.

```
a = "Hello World!"
```

```
print(a)
```

The output of the above code will be:

Hello World!

Some of the examples are:

Table 5.9: Output Using Print() Function

Example Code	Sample Output
<code>a = 20b = 10print(a + b)</code>	30
<code>print(15 + 35)</code>	50
<code>print("My name is Kabir")</code>	My name is Kabir
<code>a = "Tarun"print("My name is :",a)</code>	My name is : Tarun
<code>x = 1.3print("x =", x)</code>	x = 1.3
<code>m = 6print(f'I have {m} apples')</code>	I have 6 apples

5.9.2 User input

In all the examples till now, we have been using the calculations on known values (constants). Now let us learn to take user's input in the program. In python, input() function is used for the same purpose.

Syntax

input ([prompt])

where prompt is a message you wish to display on the screen. It is optional. During the execution, input() shows the prompt to the user and waits for the user to input a value from the keyboard. When the user enters the value from the keyboard, input() returns that value which is usually stored in a variable. For example

Python Code:

```
>>>rno = input ('Enter your roll number')
```

Output:

Enter your roll number



Notes

5.10 TYPE CONVERSION

The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion. Python has two types of type conversion:

- Implicit Type Conversion
- Explicit Type Conversion

5.10.1 Implicit Type Conversion

In Implicit type conversion, Python automatically converts one data type to another data type. This process doesn't need any user involvement. For example:

```
# Code to calculate the Simple Interest
principal_amount = 2000
roi = 4.5
time = 10

simple_interest = (principal_amount * roi * time)/100

print("datatype of principal amount : ", type(principal_amount))
print("datatype of rate of interest : ", type(roi))
print("value of simple interest : ", simple_interest)
print("datatype of simple interest : ", type(simple_interest))

datatype of principle amount :<class 'int'>
datatype of rate of interest :<class 'float'>

value of simple interest : 900

datatype of simple interest :<class 'float'>
```

When we run the above program, the output will be

In the above program,

- We calculate the simple interest by multiplying the variables `principal_amount`, `roi` and `time` and then divide the value by 100



- In the output we can see the datatype of principal_amount is an integer, datatype of roi is float.
- Also, we can see the simple_interest has float data type because Python always converts smaller data type(int) to larger data type(float) to avoid the loss of data.

Let us look at some more examples:

Table 5.10: Implicit Type Conversion

Example Code	Sample Output	Explanation
a = 10 b = "Hello" print(a+b)	TypeError: unsupported operand type(s) for +: 'int' and 'str'	The output shows an error which says that we cannot add integer and string variable types using implicit conversion
c = 'Ram' N = 3 print(c*N)	RamRamRam	The output shows that the string is printed 3 times when we use a multiply operator with a string.
x = True y = 10 print(x + 10)	11	The output shows that the boolean value x will be converted to integer and as it is true will be considered as 1 and then give the output.
m = False n = 23 print(n - m)	23	The output shows that the boolean value m will be converted to integer and as it is false will be considered as 0 and then give the output.

5.10.2 Explicit Type Conversion

In Explicit Type Conversion, users convert the data type of an object to required data type. We use the predefined functions like int(), float(), str(), etc to perform explicit type conversion. This type of conversion is also called typecasting because the user casts (changes) the data type of the objects.

Syntax:

(required_datatype)(expression)

Typecasting can be done by assigning the required data type function to the expression. Let us look at this example of adding a string and an integer using explicit conversion:



Notes

Logic Building and Basic Programming Concepts

```
Birth_day = 10
Birth_month = "July"
print("data type of Birth_day before type casting :", type(Birth_day))
print("data type of Birth_month : ", type(Birth_month))
Birth_day = str(Birth_day)
print("data type of Birth_day after type casting :", type(Birth_day))
Birth_date = Birth_day + Birth_month
print("birth date of the student : ", Birth_date)
print("data type of Birth_date : ", type(Birth_date))
```

When we run the above program, the output will be:

```
data type of Birth_day before type casting
:<class 'int'>
data type of Birth_month :<class 'str'>
data type of Birth_day after type casting
:<class 'str'>
birth date of the student : ' 10 July '
```

In above program,

- We add Birth_day and Birth_month variable.
- We converted Birth_day from integer(lower) to string(higher) type using str() function to perform the addition.
- We got the Birth_date value and data type to be string.

Let us look are some more examples:

Table 5.11: Explicit Type Conversion

Example Code	Sample Output	Explanation
a = 20 b = "Apples" print(str(a),b)	20 Apples	Writing str(a) will convert integer a into a string and then will add to the string b.
x = 20.3 y = 10 print(int(x) + y)	30	Writing int(x) will convert a float number to integer by just considering the integer part of the number and then perform the operation.
m = False n = 5 print(Bool(n)+ m)	1	Writing Bool() will convert the integer value to Boolean. If it is zero then it is converted to False, else to True for all other cases.



Note: There is no difference in single or double quoted string. Both representations can be used interchangeably. However, if either single or double quote is a part of the string itself, then the string must be placed in double or single quotes respectively.

5.11 LISTS AND TUPLES: DATA TYPES IN PYTHON

5.11.1 Introduction to Lists

As studied in the previous section, List is a sequence of values of any type. Values in the list are called elements / items. List is enclosed in square brackets.

For example: `a = [1,2.3,"Hello"]`

List is one of the most frequently used and very versatile data type used in Python. A number of operations can be performed on the lists, which we will study as we go forward.

How to create a list ?

In Python programming, a list is created by placing all the items (elements) inside a square bracket `[ ]`, separated by commas. It can have any number of items and they may be of different types (integer, float, string etc.).

```
#empty list empty_list = []  
  
#list of integers age = [15,12,18]  
  
#list with mixed data types student_height_weight = ["Ansh", 5.7, 60]
```

For example:

Note: A list can also have another list as an item. This is called nested lists.

How to access elements of a list ?

A list is made up of various elements which need to be individually accessed on the basis of the application it is used for. There are two ways to access an individual element of a list:

```
# nested list  
student marks = ["Aditya", "10-A", ["english",75]]
```

- List Index
- Negative Indexing



Notes

List Index

A list index is the position at which any element is present in the list. Index in the list starts from 0, so if a list has 5 elements the index will start from 0 and go on till 4. To access an element in a list we need to use index operator [].

Negative Indexing

Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item and so on.

In order to access elements using negative indexing, we can use the negative index as mentioned in the above Figure.

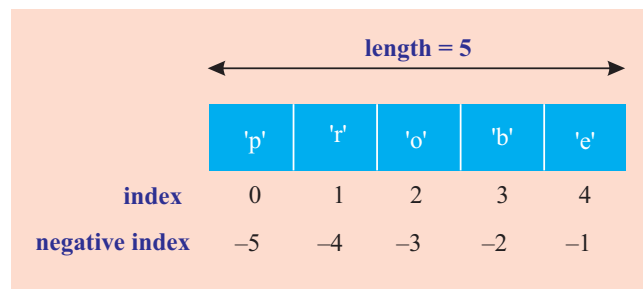


Fig. 5.6: Index in Python

Here are some samples of using List Index and Negative Index:

Table 5.12: Coding using List Index & -ve Index

Task	Sample Code	Output
Accessing Using List Index	<pre>language = ['p','y','t','h','o','n'] print(language [0]) print(language [4]) language = ['p','y','t','h','o','n'] print(my_list[4.0])</pre>	p o Y Error! Only Integer can be used for indexing
Accessing Value in a nested list	<pre>n_list = ["Happy",[2,0,1,5]] print(n_list[0][1]) print(n_list[1][3])</pre>	a 5
Accessing using Negative Index	<pre>day = ['f','r','i','d','a','y'] print(a[-1]) print(a[-6])</pre>	y f



5.11.2 Adding Element to a List

We can add an element to any list using the following methods:

- Using append() method
- Using insert() method
- Using extend() method

Using append() method

Table 5.13: Adding Elements to a List

Task	Sample Code	Output
Using append() method	<pre>List = [] print("Initial blank List: ") print(List) # Addition of Elements in the List List.append(1) List.append(2) List.append(4) print("\nList after Addition : ") print(List) # Addition of List to a List List2 = ['Good', 'Morning'] List.append(List2) print("\nList after Addition of a List: ") print(List)</pre>	Initial blank List: [] List after Addition [1, 2, 4] List after Addition of a List: [1, 2, 4, ['Good', 'Morning']]
Using insert() method	<pre># Creating a List List = [1,2,3,4] print("Initial List: ") print(List) # Addition of Element at a specific Position # (using Insert Method) List.insert(3, 12) List.insert(0, 'Kabir') print("\nList after Insert Operation: ") print(List)</pre>	Initial List: [1, 2, 3, 4] List after Insert Operation: ['Kabir', 1, 2, 3, 12, 4]
Using extend() method	<pre># Creating a List List = [1,2,3,4] print("Initial List: ") print(List) # Addition of multiple elements # to the List at the end(using Extend method) List.extend([8, 'Artificial', 'Intelligence']) print("\nList after Extend Operation: ") print(List)</pre>	Initial List: [1, 2, 3, 4] List after Extend Operation: [1, 2, 3, 4, 8, 'Artificial', 'Intelligence']



Elements can be added to the List by using built-in `append()` function. Only one element at a time can be added to the list by using `append()` method, for addition of multiple elements with the `append()` method, loops are used.

Using `insert()` Method

`append()` method only works for addition of elements at the end of the List, for addition of element at the desired position, `insert()` method is used. Unlike `append()` which takes only one argument, `insert()` method requires two arguments (position, value).

Using `extend()` method

Other than `append()` and `insert()` methods, there's one more method for Addition of elements, `extend()`, this method is used to add multiple elements at the same time at the end of the list.

5.11.3 Removing Elements from a List

Elements from a list can be removed using two methods :

- Using `remove()` method
 - Elements can be removed from the List by using built-in `remove()` function but an Error arises if element doesn't exist in the set. `Remove()` method only removes one element at a time, to remove range of elements, iterator is used. The `remove()` method removes the specified item.
 - **Note:** Remove method in List will only remove the first occurrence of the searched element.
- Using `pop()` method
 - `Pop()` function can also be used to remove and return an element from the list, but by default it removes only the last item of the list; to remove an element from a specific position of the List, index of the element is passed as an argument to the `pop()` method.

Table 5.14: Removing Elements Form a List

Task	Sample Code	Output
Using <code>remove()</code> method	<pre># Creating a List List = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,12] print("Initial List: ") print(List) # Removing elements from List using Remove() method</pre>	Initial List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] List after Removal: [1, 2, 3, 4, 7, 8, 9, 10, 11, 12]



Using pop() method	<pre> List.remove(5) List.remove(6) print("\nList after Removal: ") print(List) # Removing element from the Set using the pop() method List.pop() print("\nList after popping an element: ") print(List) # Removing element at a specific location from the Set # using the pop() method List.pop(2) print("\nContent after pop ") print(List) </pre>	List after popping an element: [1, 2, 3, 4] List after popping a specific element: [1, 2, 4]
--------------------	--	--

5.11.4 Slicing of a List

In Python List, there are multiple ways to print the whole List with all the elements, but to print a specific range of elements from the list, we use Slice operation. Slice operation is performed on Lists with the use of colon(:). To print elements from beginning to a range use [:Index], to print elements from end use[:-Index], to print elements from specific Index till the end use [Index:], to print elements within a range, use [Start Index: End Index] and to print whole List with the use of slicing operation, use [:]. Further, to print whole List in reverse order, use[::-1].

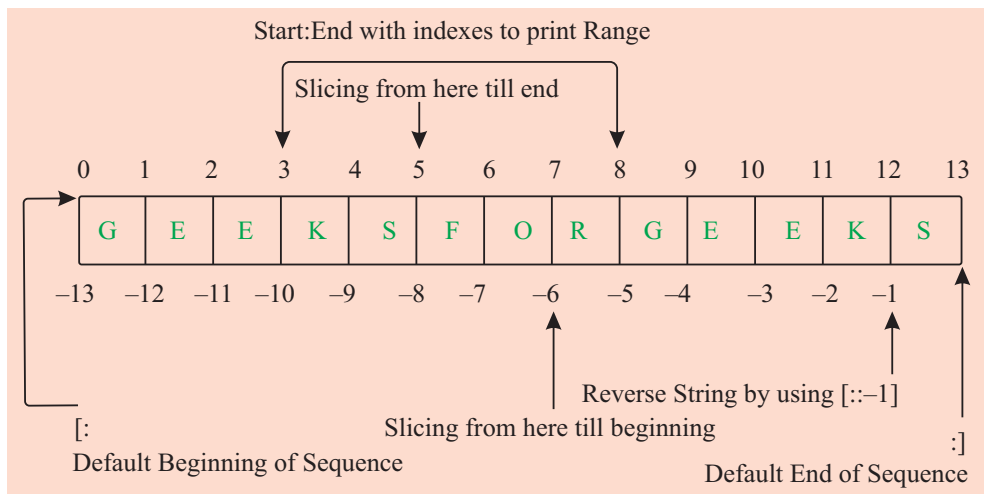


Fig. 5.7: Slicing in Python

Note: To print elements of List from rear end, use Negative Indexes.



Notes

Table 5.15: Slicing of a List

Task	Sample Code	Output
Slicing	<pre># Creating a List List= ['G','O','O','D','M','O','R','N', 'I','N','G'] print("Initial List: ") print(List) # using Slice operation Sliced_List = List[3:8] print("\nSlicing elements in a range 3-8: ") print(Sliced_List) # Print elements from a pre-defined point to end Sliced_List = List[5:] print("Elements sliced from 5th element till the end: ") print(Sliced_List) # Printing elements from beginning till end Sliced_List = List[:] print("\nPrinting all elements using slice operation:") print(Sliced_List)</pre>	Initial List: ['G','O','O','D','M','O','R','N','I','N','G'] Slicing elements in a range 3-8: ['D','M','O','R','N'] Elements sliced from 5th element till the end: ['O','R','N','I','N','G'] Printing all elements using slice operation: ['G','O','O','D','M','O','R','N','I','N','G']
Slicing using negative index of list	<pre># Creating a List List= ['G','O','O','D','M','O','R','N','I','N','G'] print("Initial List: ") print(List) # Print elements from beginning to a pre-defined point using Slice Sliced_List = List[:-6] print("\nElements sliced till 6th element from last: ") print(Sliced_List) # Print elements of a range using negative index List slicing Sliced_List = List[-6:-1] print("\nElements sliced from index -6 to -1") print(Sliced_List) # Printing elements in reverse using Slice operation Sliced_List = List[::-1] print("\nPrinting List in reverse: ") print(Sliced_List)</pre>	Initial List: ['G','O','O','D','M','O','R','N','I','N','G'] Elements sliced till 6th element from last: ['G','O','O','D','M'] Elements sliced from index -6 to -1 ['O','R','N','I','N'] Printing List in reverse: ['G','N','I','N','R','O','M','D','O','O','G']

5.11.5 Introduction to Tuples

Tuple is a collection of Python objects which is ordered and unchangeable. The sequence of values stored in a tuple can be of any type, and they are indexed by



integers. Values of a tuple are syntactically separated by commas. Although it is not necessary, it is more common to define a tuple by closing the sequence of values in parentheses. This helps in understanding the Python tuples more easily.

For example:

```
fruits = ("apple", "banana", "cherry")
```

Immutability of a tuple ?

Tuples are immutable by design: they are created to store fixed, unchangeable data. Once a tuple is created, its elements and structure cannot be altered, making them safe for protecting data from accidental changes.

Python enforces immutability: when you try to modify a tuple (like changing an element, adding, or removing items), Python raises a `TypeError` and prevents the operation, ensuring the tuple stays constant throughout your program.

How to Create a tuple?

In Python, tuples are created by placing sequence of values separated by ‘comma’ with or without the use of parentheses for grouping of data sequence. It can contain any number of elements and of any datatype (like strings, integers, list, etc.).

```
#Creating an empty Tuple
Tuple1 = ()

#Creating a Tuple with the use of string
Tuple1 = ('Artificial', 'Intelligence')

#Creating a Tuple with Mixed Datatype
Tuple1 = (5, 'Artificial', 7, 'Intelligence')
```

For example:

Accessing of Tuples

We can use the index operator `[]` to access an item in a tuple where the index starts from 0. So, a tuple having 6 elements will have indices from 0 to 5. Trying to access an element outside of tuple (for example, 6, 7...) will raise an `IndexError`. The index must be an integer; so, we cannot use float or other types. This will result in `TypeError`.

For example:

```
fruits = ("apple", "banana", "cherry")
print(fruits[1])
```

The above code will give the output as **“banana”**.



Notes

Deleting a Tuple

Tuples are immutable and hence they do not allow deletion of a part of it. Entire tuple gets deleted using `del()` method.

For example:

```
num = (0, 1, 2, 3, 4)
del num
```

In the programs we have seen till now, there has always been a series of statements executed by Python in exact top-down order. What if you wanted to change the flow of how it works? For example, you want the program to take some decisions and do different things depending on different situations, such as printing 'Good Morning' or 'Good Evening' depending on the time of the day? As you might have guessed, this is achieved using control flow statements. There are three control flow statements in Python - if, for and while.

5.12.1 If Statement

On World Health Day, one of the schools in the city decided to take an initiative to help students maintain their health and be fit. Let's observe an interesting conversation happening between the students when they come to know about the initiative.

There comes situations in real life when we need to make some decisions and based on these decisions, we need to decide what should we do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code.

Decision making statements in programming languages decide the direction of flow of program execution. Decision making statements available in Python are:

- if statement
- if...else statements
- if-elif ladder

If Statement

The if statement is used to check a condition: *if* the condition is true, we run a block of statements (called the *if-block*). (See Fig. 5.7)



Syntax:

```
if test expression:
    statement(s)
```

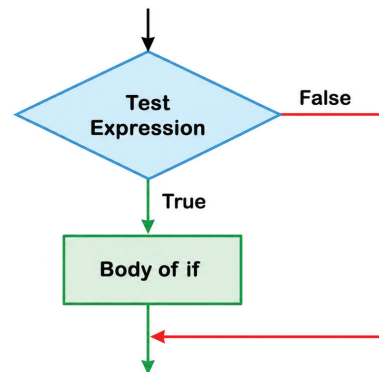


Fig. 5.8: Flowchart: If statement

Here, the program evaluates the test expression and will execute statement(s) only if the test expression is True. If the test expression is False, the statement(s) is not executed.

In Python, the body of the if statement is indicated by the indentation. Body starts with an indentation and the first unindented line marks the end. Python interprets non-zero values as True. None and 0 are interpreted as False.

```
#if the number is positive, we print an appropriate message

num = 3
if num > 0:
    print(num, "is a positive number.")
print("this is always printed")
num = -1
if num > 0:
    print(num, "is a positive number.")
print("this is also always printed")
```

For example:

When you run the program, the output will be:

Here:

- num > 0 is the test expression. The body of if is executed only if this evaluates to True.
 - When variable num is equal to 3, test expression is true and body inside body of if is executed.
 - If variable num is equal to -1, test expression is false and the statement(s) inside the if block doesn't execute.



Notes

- The print() statement falls outside of the if block (unindented). Hence, it is executed regardless of the test expression.

if...else Statement

```
3 is a positive number
this is always printed
this is also always printed
```

The if...else statement evaluates test expression and will execute body of if only when test condition is True. If the condition is False, body of else is executed. Indentation is used to separate the blocks. (See Fig. 5.9)

```
if test expression:
    Body of if
else:
    Body of else
```

Syntax:

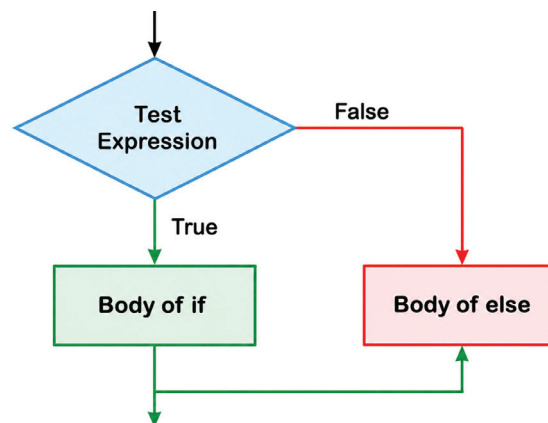


Fig. 5.9: Flowchart: If: else statement

```
#A program to check if a person can vote
age = input("Enter Your Age")
if age >= 18:
    print("You are eligible to vote")
else:
    print("You are not eligible to vote")
```

For example:

In the above example, when if the age entered by the person is greater than or equal to 18, he/she can vote. Otherwise, the person is not eligible to vote



if...elif...else Statement

The elif is short for else if. It allows us to check for multiple expressions. If the condition for if is False, it checks the condition of the next elif block and so on. If all the conditions are False, body of else is executed. Only one block among the several if...elif...else blocks is executed according to the condition. The if block can have only one else block. But it can have multiple elif blocks. (See Fig. 5.10)

Syntax:

```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```

For example:

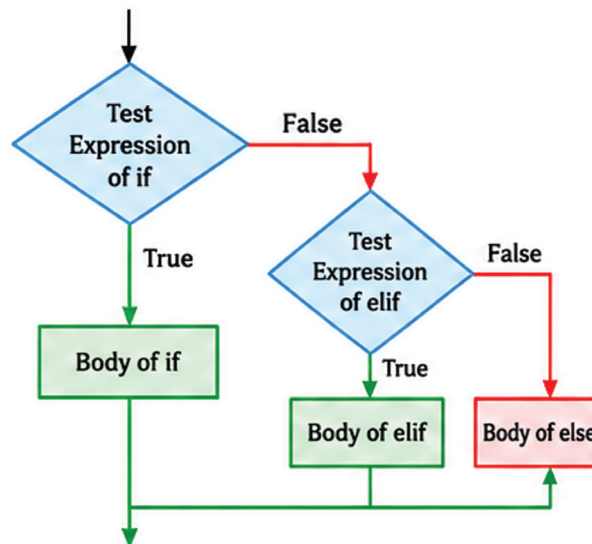


Fig. 5.10: Flowchart: If...elif...else statement

Here, the output comes as “You get a C grade,” as the Marks variable is 60 and both if and elif conditions here are False.

```
#To check the grade of a student
Marks = 60

if Marks > 75:
    print("You get an A grade")
elif Marks > 60:
    print("You get a B grade")
else:
    print("You get a C grade")
```



Notes

Nested if statements

We can have an if...elif...else statement inside another if...elif...else statement. This is called nesting in computer programming. Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting. This can get confusing, so must be avoided if it can be.

```
# Input a number & check if the number is positive or negative or zero and display a message
# This time we use nested if

num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

For example:

Output 1:

```
Enter a number: -1
Negative number
```

Output 2:

```
Enter a number: 0
Zero
```

Output 3

```
Enter a number: 5
Positive number
```

5.12.2 The For Loop

The forin statement is another looping statement which *iterates* over a sequence of objects i.e. go through each item in a sequence. (See Fig. 5.11). What you need to know right now is that a sequence is just an ordered collection of items.

Syntax:

```
for val in sequence:
    Body of for
```

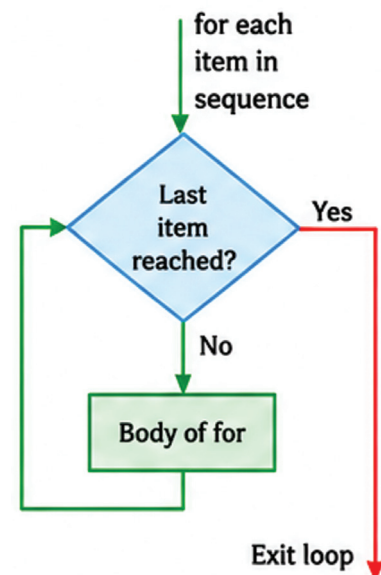


Fig. 5.11: Syntax of for loop



Here, `val` is the variable that takes the value of the item inside the sequence on each iteration. Loop continues until we reach the last item in the sequence. The body of for loop is separated from the rest of the code using indentation.

```
# Program to find the sum of all numbers stored in a list
# List of numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
# variable to store the sum
sum = 0
# iterate over the list
for val in numbers:
    sum = sum+val
print("The sum is", sum)
# Output: The sum is 48
```

For example:

When you run the program, the output will be: The sum is 48

5.12.3 The while Statement

The while statement allows you to repeatedly execute a block of statements if a condition is true. A while statement is an example of what is called a *looping* statement. A while statement can have an optional else clause.(See Fig. 5.12)

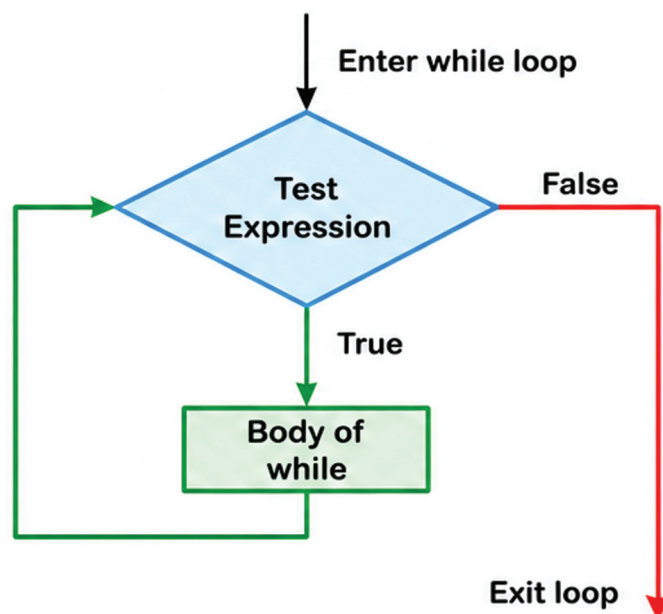


Fig. 5.12: Flowchart While Statement

**Syntax:**

```
while test_expression:  
    Body of while
```

In while loop, test expression is checked first. The body of the loop is entered only if the test_expression evaluates to True. After one iteration, the test expression is checked again. This process continues until the test_expression evaluates to False. In Python, the body of the while loop is determined through indentation. Body starts with indentation and the first unindented line marks the end. Python interprets any non-zero value as True. None and 0 are interpreted as False.

For example:

```
# Program to print numbers from 1 to 5 using while loop

number = 1                # starting value

while number <= 5:        # loop condition
    print(number)          # print the current number
```

Here:

- The loop starts with number = 1.
- The condition number <= 5 is checked.
- Till the condition is **True**, the loop keeps running.
- Each time, it prints the number and increases it by 1.
- When number becomes 6, the condition becomes False and the loop stops.

We need to increase the value of the counter variable in the body of the loop. This is very important (and mostly forgotten). Failing to do so will result in an infinite loop (never ending loop).

**WHAT HAVE YOU LEARNT**

- A step-by-step procedure to solve a problem is an **Algorithm**, which can be described informally using **Pseudocode** or visually using a **Flowchart**.

Logic Building and Basic Programming Concepts

- Standard symbols (Terminals, Processes, Decisions, I/O) are used to map out the flow of execution.
- Python uses straightforward syntax. Programs are built using basic **Data Types** (int, str, bool) and **Variables**.
- Program execution order is managed by **Conditional Statements** (if/else) for decision-making and **Loops** (for/while) for repetition.



TERMINAL EXERCISE

Short-answer Questions

1. Define an algorithm.
2. What is a flowchart?
3. What is a Python statement?
4. What is the purpose of Comments and Keywords in Python?
5. What is the difference between an identifier and a variable?
6. Name any two Python data types.
7. What are comparison operators used for?
8. What is a conditional statement?
9. When do we use elif?
10. When should we use a while loop instead of a for loop?
11. Why is indentation important in if/for/while statements?

Long-answer Questions

1. Explain the importance of logic building in programming.
2. Explain the difference between lists and tuples with examples.
3. Write a Python program to calculate the area of a rectangle and explain the code.
4. Write a Python program to calculate the average of three subject marks.



Notes



Notes

Skill Based Questions

1. A student wrote a while loop, but it never stops running. What could be the possible reasons?
2. Imagine a program that calculates marks for 100 students. Which loop (for or while) would you choose and why?
3. How can writing an algorithm before coding help avoid errors in Python programs?
4. Write a python code to calculate the Selling Price of an Item based on its cost price and discount percentage



ANSWERS TO INTEXT QUESTIONS

5.1

1. Logic
2. Reasoning
3. Logical
4. Patterns
5. Foundation

5.2

1. (b)
2. (c)
3. (b)

5.3

1. (c)
2. (c)
3. (b)
4. (c)
5. (b)

5.4

1. www.python.org,
2. IDLE
3. Integrated Development Environment
4. Script,
5. Shell

5.5

1. Statement
2. Comments
3. Keywords
4. Digit,
5. '='
- 6 String
7. None Datatype



Notes

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- explain the purpose of algorithm
- draw flowchart for flow of execution of a computer program
- write a program using python programming language
- use if statement and if...else statement in Python Programming language
- use for loop and while statement in python

References:

Introduction to Python:

- https://www.w3schools.com/python/python_intro.asp
- <https://www.programiz.com/python-programming/first-program>
- <https://www.geeksforgeeks.org/python-programming-language/>
- Introduction to Flowchart and Algorithm: <https://www.edrawsoft.com/explain-algorithm-flowchart.php>
- Python Lists: https://www.w3schools.com/python/python_lists.asp



APPLIED DATA HANDLING AND VISUALIZATION

Building on the foundational concepts of Python programming, including syntax, data structures, and control flow, this lesson introduces the essential tools used for working with real-world data. Effective data handling is a key component of Artificial Intelligence, Machine Learning, and Data Science, as these fields rely on the ability to process, organize, and analyze large volumes of information.

In this lesson, learners will explore powerful Python libraries such as **NumPy** and **Pandas**, which are widely used for data processing, manipulation, transformation, and analysis. These libraries enable users to efficiently work with structured and numerical data, perform computations, and prepare datasets for further exploration and decision-making.

The lesson also introduces **Matplotlib**, a visualization library that helps represent data through charts and graphs. Visualizing data makes it easier to identify patterns, trends, and relationships, thereby supporting meaningful interpretation and communication of insights. Tools such as **Tableau** may also be used to create interactive visualizations and dashboards.

By the end of this lesson, learners will have developed the skills required to organize, analyze, and visualize data effectively, providing a strong foundation for advanced applications in Artificial Intelligence, Machine Learning, and Data Science.



OBJECTIVES

After learning this lesson, you will be able to:

- explain the need for specialized data structures (like NumPy Arrays and Pandas DataFrames) over standard Python lists;



- utilize the NumPy library to perform efficient numerical and mathematical operations on multi-dimensional array structures;
- create and manipulate Pandas DataFrames and Series to load, inspect, clean, and organize structured data;
- apply basic data cleaning techniques (e.g., handling missing values, filtering data);
- design and generate various types of data visualizations (line plots, scatter plots, bar charts) using Matplotlib or an equivalent visualization tool like Tableau;
- interpret visual representations of data to draw meaningful conclusions;
- apply the fundamental concepts of the NumPy and Pandas libraries to perform data manipulation and analysis tasks;
- import and export data between CSV files and Pandas Data Frames, ensuring data integrity and consistency.

6.1 NUMERICAL COMPUTING WITH NUMPY

6.1.1 Introduction to NumPy

NumPy, which stands for Numerical Python, is the fundamental package for mathematical and logical operations on arrays in Python. It is a commonly used package when it comes to working around numbers. NumPy gives a wide range of arithmetic operations around numbers giving us an easier approach in working with them. NumPy also works with arrays, which is nothing but a homogenous collection of data.

An array is nothing but a set of multiple values which are of the same datatype. They can be numbers, characters, Booleans, etc. but only one datatype can be accessed through an array. In NumPy, the arrays used are known as ND-arrays (N-Dimensional Arrays) as NumPy comes with a feature of creating n-dimensional arrays in python.

An array can easily be compared to a list. Let us take a look how different they are:

NumPy Arrays	Lists
1. Homogenous collection of Data.	1. Heterogenous collection of Data.
2. Can contain only one type of data, hence not flexible with datatypes.	2. Can contain multiple types of data, hence flexible with datatypes.



Notes

Applied Data Handling and Visualization

3. Cannot be directly initialized. Can be operated with Numpy package only.	3. Can be directly initialized as it is the part of python syntax.
4. Direct numerical operations can be done. For example, dividing the whole array by 3 divides every element by 3.	4. Direct numerical operations are not possible. For example, dividing the whole list by 3 cannot divide every element by 3.
5. Widely used for arithmetic operations.	5. Widely used for data management.
6. Arrays take less memory space.	6. Lists acquire more memory space.
7. Example: To create a Numpy array 'A': <code>import numpyA=numpy.array([1,2,3,4,5,6,7,8,9,0])</code>	7. Example: To create a list:A = [1,2,3,4,5,6,7,8,9,0]

Since the NumPy package is not included in the basic Python installation, we need to install it separately. Once it is installed, it can be readily used in any Python code whenever imported.

Need for NumPy: While standard Python lists can store data, they are slow and memory-inefficient when performing complex mathematical operations on large scales. NumPy is implemented in C, making array manipulation significantly faster (often 50 times faster) and more memory-efficient. This performance is crucial for handling the massive datasets processed in Data Science and AI.

The Array Object: The core object in NumPy is the **ndarray (N-dimensional array)**. Unlike Python lists, where elements can have different data types, all elements in a NumPy array must be of the same data type, such as integers or floating-point numbers. This uniformity allows NumPy to perform computations more efficiently and makes it suitable for handling large datasets.

NumPy arrays can have one or more dimensions:

- **1-Dimensional (1D) Array:** Represents a single row of elements, similar to a list or vector.

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50])
print(arr1)
```

Output: [10 20 30 40 50]

- **2-Dimensional (2D) Array:** Represents data arranged in rows and columns, similar to a table or matrix.

```
arr2 = np.array([[1, 2, 3],
                 [4, 5, 6]])
print(arr2)
```

Output:[[1 2 3][4 5 6]]



- **N-Dimensional (3D and Higher) Array:** Represents data with multiple layers or dimensions. Such arrays are commonly used in image processing, scientific computing, and machine learning applications.

```
arr3 = np.array([[[1, 2], [3, 4]],
                 [[5, 6], [7, 8]]])
```

```
print(arr3)
```

Output: [[[1 2][3 4]][[5 6][7 8]]]

Exploring NumPy!

The NumPy package provides us with various features and functions which helps us in arithmetic and logical operations. Let us look at some of them:

1. **NumPy Arrays:** As discussed before, arrays are homogenous collections of datatypes. With NumPy, we can create n-dimensional arrays (where n can be any integer) and operate on them using other mathematical functions.

Here are some ways by which you can create arrays using the NumPy package assuming the NumPy packages are imported already.

Function	Code	Output
Creating a Numpy Array	<code>numpy.array([1,2,3,4,5])</code>	<code>[1, 2, 3, 4, 5]</code>
Creating a 2-Dimensional zero array (4X3 – 4 rows and 3 columns)	<code>numpy.zeros((4,3))</code>	<code>[[0., 0., 0.]</code> <code>[0., 0., 0.]</code> <code>[0., 0., 0.]</code> <code>[0., 0., 0.]</code>
Creating an array with 5 random values	<code>random.rand(5)</code>	<code>[0.4109496 0.4465852</code> <code>0.5969157 0.9251727</code> <code>0.5892841]</code>
Creating a 2-Dimensional constant value array (3X4 – 3 rows and 4 columns) having all 6s	<code>numpy.full((3,4),6)</code>	<code>[[6, 6, 6, 6],</code> <code>[6, 6, 6, 6],</code> <code>[6, 6, 6, 6]]</code>
Creating a sequential array from 0 to 30 with gaps of 5	<code>numpy.arange(0,30,5)</code>	<code>[0, 5, 10, 15, 20, 25]</code>

Arithmetic Operations on NumPy Arrays

One of the key advantages of NumPy arrays is that arithmetic operations can be performed directly on all elements of an array without the need for explicit loops.



Notes

Applied Data Handling and Visualization

When an operation is applied to an array, NumPy automatically performs that operation on each element of the array.

Let us assume that the array **ARR** has been initialized as follows:

```
ARR = numpy.array([1, 2, 3, 4, 5])
```

Various arithmetic operations such as addition, subtraction, multiplication, and division can then be performed directly on the array to obtain the desired results efficiently.

Now, let us look at various operations that could be implemented on this array:

Function	Code	Output
Adding 5 to each element	<code>ARR + 5</code>	<code>array([6, 7, 8, 9, 10])</code>
Divide each element by 5	<code>ARR / 5</code>	<code>array([0.2, 0.4, 0.6, 0.8, 1.])</code>
Squaring each element	<code>ARR ** 2</code>	<code>array([1, 4, 9, 16, 25])</code>
Accessing 2 nd element of the array (elementcount starts from 0)	<code>ARR[1]</code>	2
Multiplying 2 arrays {consider BRR = <code>numpy.array([6,7,8,9,0])</code> }	<code>ARR * BRR</code>	<code>array([5, 8, 9, 8, 5])</code>

As you can see, direct arithmetical operations can be implemented on individual array elements just by manipulating the whole array variable.

Let us look at the functions which talk about the properties of an array:

```
import numpy as np
```

```
ARR = np.array([[1, 2, 3], [4, 5, 6]])
```

Function	Code	Output
Type of an array	<code>type(ARR)</code>	<class 'numpy.ndarray'>
Check the dimensions of an array	<code>ARR.ndim</code>	2
Shape of an array	<code>ARR.shape</code>	(2, 3)
Size of an array	<code>ARR.size</code>	6
Datatype of elements stored in the array	<code>ARR.dtype</code>	int64

Some other mathematical functions available with NumPy are:



Function	Code	Output
Finding out maximum element of an array	ARR.max()	6
Finding out row-wise maximum elements	ARR.max(axis = 1)	[3, 6]
Finding out column-wise minimum elements	ARR.min(axis = 0)	[1, 2, 3]
Sum of all array elements	ARR.sum()	21

6.1.2 Creating and Manipulating NumPy Arrays

Creating Arrays: You can create arrays from existing Python lists using `numpy.array()`, or create pre-filled arrays using powerful functions like `numpy.zeros((rows, cols))` or `numpy.ones((rows, cols))` for quick initialization, or `numpy.arange()` to create a range of numbers.

Basic array properties: Every ndarray has important attributes that describe its structure:

- **shape:** A tuple indicating the size of each dimension (e.g., (3, 4) for 3 rows and 4 columns).
- **ndim:** The number of dimensions (e.g., 1 for a vector, 2 for a matrix).
- **dtype:** The data type of the elements (such as `int64` or `float32`).

Indexing and Slicing Arrays: Elements are accessed using zero-based indices. Slicing allows you to extract sub-arrays using the familiar Python notation `[start:end:step]`. For 2D arrays (matrices), indexing typically follows the format `[row_index, column_index]`.

Vectorized Operations: This is NumPy's greatest advantage. Instead of writing slow, explicit loops to add two lists element by element, NumPy allows you to simply use arithmetic operators (+, -, *, /) on two arrays of the same shape. This performs the operation on all corresponding elements simultaneously and extremely fast. NumPy also includes common mathematical functions (e.g., `np.sqrt()`, `np.log()`) that can be applied to an entire array at once.



INTEXT QUESTIONS 6.1

Fill in the blanks:

1. NumPy stands for
2. In NumPy, arrays are called because they can have multiple dimensions.



Notes

3. The function to create a NumPy array from a list is
4. The function `ARR.max()` returns the element of the array.
5. The core data structure in NumPy is called the

6.2 DATA STRUCTURING AND ANALYSIS WITH PANDAS

6.2.1 Introduction to Pandas

Definition: Pandas is a library built on top of NumPy and is the most important tool for data cleaning, transformation, and analysis. It provides flexible data structures designed to make working with labeled and relational data both intuitive and fast.

Core Data Structures: Pandas introduces two key structures that represent data logically:

- **Series:** A one-dimensional labeled array capable of holding any data type. Think of it as a single column in a spreadsheet, where the labels (known as the index) are used to access the data.
- **DataFrame:** A two-dimensional, size-mutable (rows/columns can be added/dropped), tabular data structure with labeled axes (rows and columns). It is the primary way we handle structured data, resembling a database table or an Excel sheet. Each column in a DataFrame is essentially a Series object.

In simple terms, a **Series** is like a single column of data, while a **DataFrame** is a collection of multiple Series arranged together in rows and columns. These structures form the foundation of data analysis and processing in Pandas.

6.2.2 Data Loading and Inspection

Loading data: Data in real-world projects often comes from external files. Pandas simplifies this process using functions like `pd.read_csv('file.csv')` to load data directly from a comma-separated values file into a DataFrame.

Inspecting data: Once data is loaded, it is crucial to understand its structure and composition before analysis.

- `df.head(5)` and `df.tail(5)`: Show the first and last five rows, respectively, for a quick view of the data.
- `df.info()`: Provides a summary of the DataFrame, including the data type (dtype) of each column and the total count of non-null (non-missing) values, which is essential for identifying data quality issues.
- `df.describe()`: Generates descriptive statistics (count, mean, standard deviation, minimum, quartiles, maximum) for numerical columns.



- **Accessing Data:** Data is accessed by selecting specific columns (`df['ColumnName']`) and rows (`df.loc[...]` for label-based or `df.iloc[...]` for position-based indexing).

6.2.3 Data Cleaning and Preparation

Handling Missing Data: Real-world data is rarely perfect and often contains null or missing values (represented as NaN). These must be handled before modeling:

- **Identification:** Checking for null values using `df.isnull().sum()`.
- **Removal:** `df.dropna()` removes rows or columns containing missing values entirely.
- **Imputation (Filling):** `df.fillna(value)` replaces missing values with a specified value (e.g., the mean or median of the column, or a fixed value like zero).

Filtering Data: This involves selecting specific subsets of data based on criteria. You can filter rows using Boolean indexing (e.g., `df[df['Age'] > 30]`).

Sorting Data: Data often needs to be ordered for better viewing or analysis. The function `df.sort_values(by='ColumnName', ascending=True)` sorts the DataFrame rows based on the values in one or more specified columns.

6.3 DATA VISUALIZATION

6.3.1 Principles of Data Visualization

Data Visualisation is the process of displaying information in the form of pictures, such as charts, graphs, and maps. Instead of looking at long tables full of numbers, visualisation helps us see patterns, trends, and comparisons easily.

When data is shown visually, our brain understands it much faster. For example, a bar graph showing a student's marks across subjects is easier to read than a list of numbers. In real life, data visualisation is used everywhere—weather apps, fitness apps, stock market charts, school reports, and even election results.

Data visualisation becomes especially important in the world of AI and data science because computers deal with huge amounts of information. Turning this information into graphs helps us:

- Spot trends (like rising temperatures)
- Find patterns (such as which products sell the most)
- Make decisions (like planning a budget or improving performance)
- Explain data clearly to others



Notes

Common visualisation tools include bar graphs, line charts, pie charts, histograms, and scatter plots, each used for different kinds of data. In simple words, data visualisation makes data easy to understand, helping us learn, analyse, and make better decisions.

6.3.2 Types of Data Visualisation

Different types of charts and graphs are used to show different kinds of information. Here are some common and easy-to-understand types:

Bar Graph

A bar graph uses rectangular bars to compare different categories or groups. The length or height of each bar shows the value of that category. It can be used to compare quantities of different items and to show differences clearly.

For example, comparing marks in English, Maths, and Science using bars helps you instantly see which subject has the highest or lowest marks.

Sample Data:

Subject	Marks
English	75
Mathematics	90
Science	85
Social Science	80

The corresponding bar graph would display a separate bar for each subject, with the height of each bar representing the marks obtained.

Example Bar Chart:

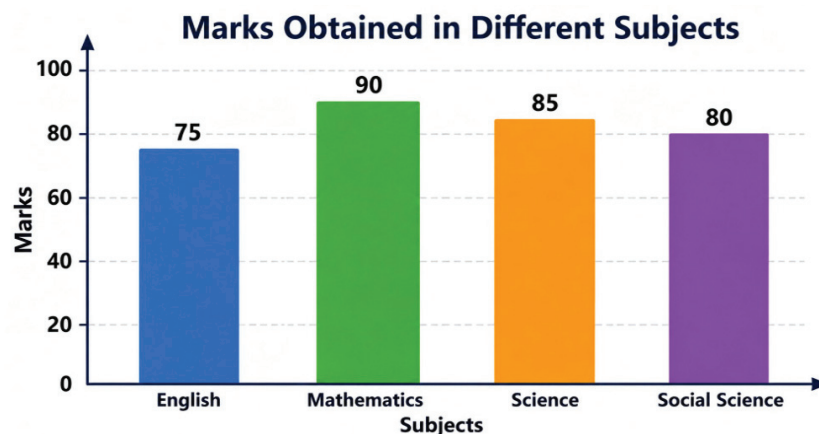


Fig. 6.1: Bar Chart



Line Chart

A line chart connects data points with a continuous line. It is mainly used to show how a value changes over time or across an ordered sequence. Line charts help identify trends, patterns, growth, and decline in data.

For example, plotting daily temperature readings over a month using a line chart can clearly show periods of increasing or decreasing temperature. Similarly, line charts are commonly used to represent stock prices, population growth, rainfall patterns, and sales trends.

Sample Data:

Month	Sales (Units)
January	120
February	135
March	150
April	145
May	170
June	190

The line chart below illustrates how sales changed over six months.



Fig. 6.2: Line Chart

Pie Chart

A pie chart is a circular graph divided into slices, where each slice represents a proportion or percentage of the whole. It is useful for showing how a total quantity



Notes

is distributed among different categories and for comparing the relative contribution of each category.

For example, a pie chart can be used to show how students spend their time during a day. By observing the size of each slice, it becomes easy to identify which activity occupies the largest or smallest portion of time.

Sample Data:

Activity	Hours
Sleeping	8
Studying	6
School	5
Playing	3
Other Activities	2

The pie chart below illustrates how a student’s 24-hour day is distributed among different activities.

Sample Pie Chart:

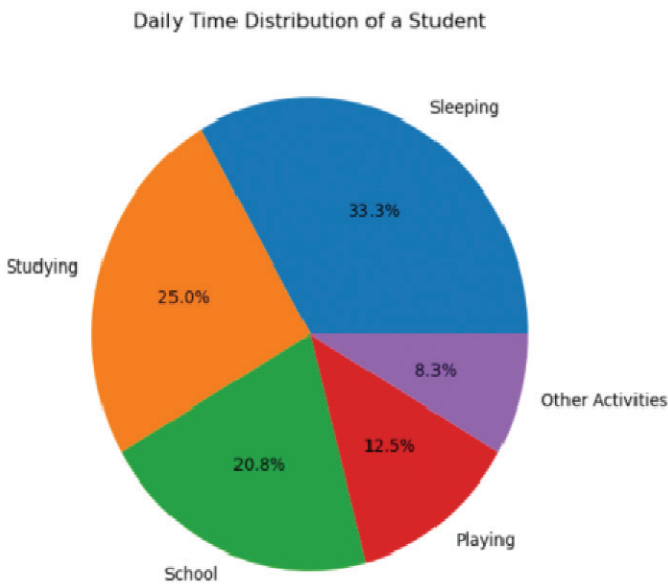


Fig. 6.3: Pie Chart

Histogram

A histogram is a graphical representation of the distribution of numerical data. It resembles a bar chart, but instead of comparing categories, it groups continuous



data into intervals called **bins** and displays the frequency of values within each interval. Histograms are useful for understanding the spread of data, identifying common value ranges, and observing patterns such as concentration, variation, or skewness.

For example, a histogram can be used to display the distribution of marks obtained by students in a science test. The marks are grouped into ranges, and the height of each bar represents the number of students whose scores fall within that range.

Sample Data

marks = [45, 52, 56, 60, 62, 65, 68, 70, 72, 75,
78, 80, 82, 85, 88, 90, 92, 95, 97, 99]

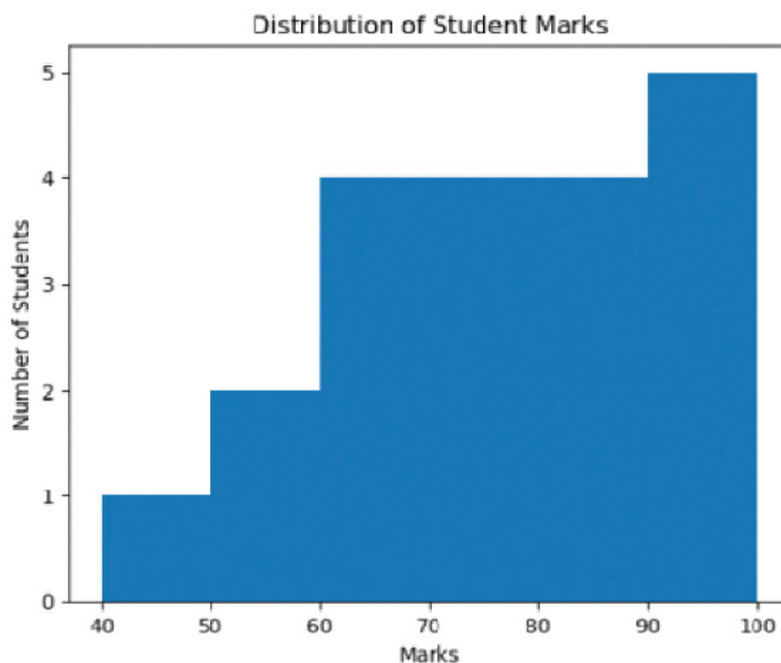


Fig. 6.4: Histogram

Scatter Plot

A scatter plot uses dots to show the relationship between two variables. Each dot represents one observation. Scatter plots help identify correlations or relationships of data points (negative, positive, or none) and helps study patterns in paired data.

For example, plotting hours studied vs. marks scored can show whether studying more increases marks.



Notes

Maps and Heat Maps

A map visualisation shows data across geographical areas. Heat maps use colours to show intensity – darker colours mean higher values. They can be used while working with regional or location-based data, and to compare values across places.

They help us identify high and low activity areas, regional differences, and problem zones. They are popularly used in weather forecasting, population studies, and environmental monitoring.

For example, a heat map showing air pollution levels in different Indian cities – Delhi might be shown in a darker colour compared to Shimla.

Choosing the Right Plot: The choice of graph depends entirely on the type of data and the question being asked:

- **Line Plot:** Ideal for showing trends over a continuous range, typically time (Time Series Data).
- **Bar Chart:** Best for comparing the size or frequency of different categories (Categorical Data).
- **Scatter Plot:** Used to examine the relationship or correlation between two numerical variables (Relationship Data).
- **Histogram:** Used to visualize the distribution of a single numerical variable (Distribution Data).

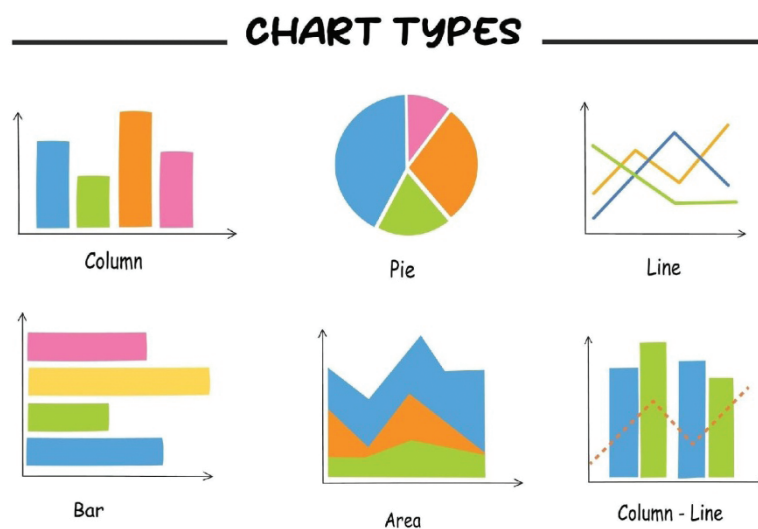


Fig. 6.5: Types of Charts

**INTEXT QUESTIONS 6.2**

Notes

Multiple-Choice Questions:

1. Pandas is mainly used for:
 - (a) Creating 3D animations
 - (b) Data processing, manipulation, and analysis
 - (c) Designing websites
 - (d) Playing audio files
2. To replace missing values with a fixed value, we use:
 - (a) `df.fillna()`
 - (b) `df.replace()`
 - (c) `df.nullfix()`
 - (d) `df.addvalue()`
3. Which Pandas structure is similar to an Excel worksheet?
 - (a) Tuple
 - (b) Series
 - (c) DataFrame
 - (d) Dictionary
4. Data visualization helps us to:
 - (a) Convert data into text
 - (b) Delete data
 - (c) Understand patterns, trends, and relationships in data
 - (d) Hide data
5. A histogram is primarily used to represent:
 - (a) Relationships between two variables
 - (b) Categories
 - (c) Frequency distribution of data
 - (d) Regions
6. The choice of a graph depends on:
 - (a) The size of the computer screen
 - (b) What the teacher prefers
 - (c) The type of data and the question being answered
 - (d) The number of colours available

6.4 DATA VISUALIZATION WITH MATPLOTLIB**6.4.1 Introduction**

Matplotlib is one of the most popular libraries in Python used for data visualization. It allows us to create different types of charts and graphs such as line charts, bar graphs, scatter plots, pie charts, and more. Matplotlib is extremely helpful because:



Notes

- It makes data easy to understand through pictures.
- It can create simple as well as advanced visualizations.
- It works well with other data-handling libraries like NumPy and Pandas.

The most used part of Matplotlib is `matplotlib.pyplot`, which works like a set of drawing tools for making graphs.

To use it in a Python program, we need to import it first:

```
import matplotlib.pyplot as plt
```

Here, `plt` is just a short name we give to `pyplot` so that it becomes easier to write.

Importance of Matplotlib

Matplotlib is used in science projects, finance charts, weather visualisation, school projects, and many real-life applications. It is very useful to:

- To display data in a simple and visual form
- To identify patterns, trends, and comparisons
- To make reports and presentations clearer

6.4.2 Visualisation with Matplotlib

Matplotlib allows you to create different types of graphs with very little code. Here are the basic steps for visualising data using Matplotlib:

Creating a Line chart

A line chart is used to show how something changes over time. Following is a python code that will draw a simple line graph with labels and a title:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]      # days
y = [10, 20, 30, 25, 35] # temperature

plt.plot(x, y)

plt.xlabel("Days")
plt.ylabel("Temperature")
plt.title("Temperature Over 5 Days")

plt.show()
```




Output:

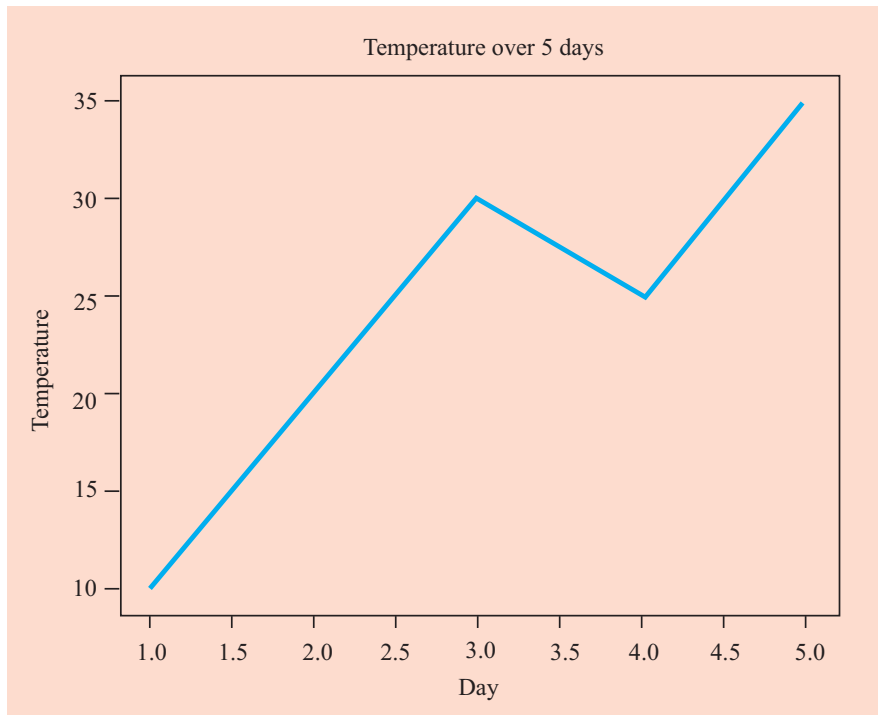


Fig. 6.6: Creating a Line chart using Matplotlib

Creating a Bar graph

Bar graphs help compare different categories. The following code will help plot a bar graph:

```
import matplotlib.pyplot as plt

subjects = ["Maths", "Science", "English"]
marks = [85, 92, 78]

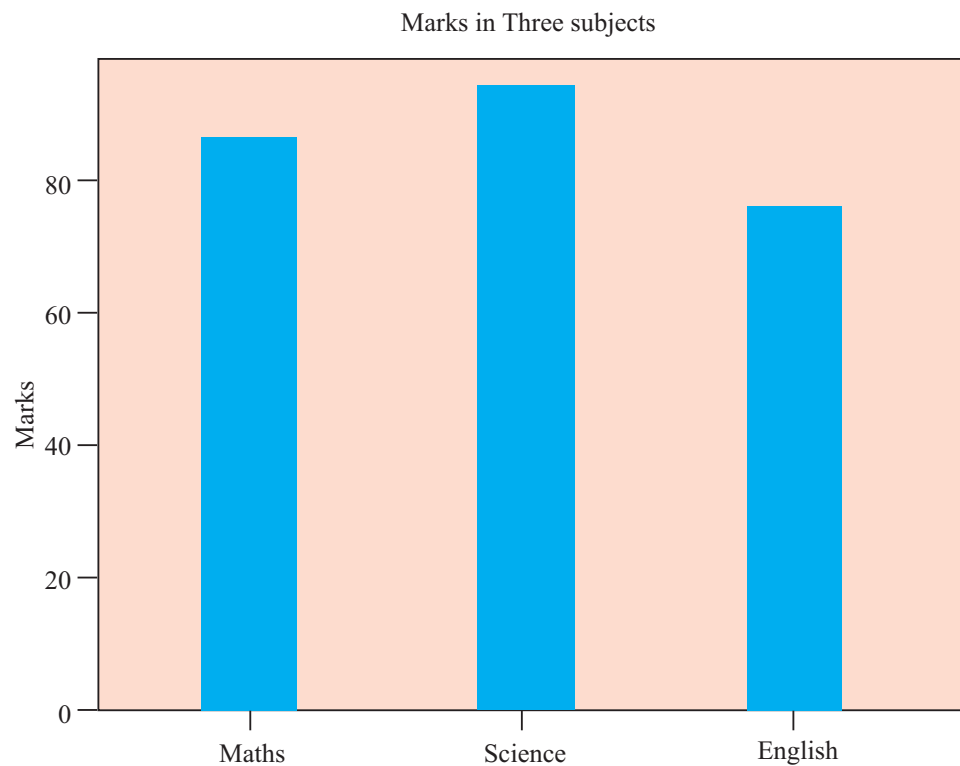
plt.bar(subjects, marks)

plt.ylabel("Marks")
plt.title("Marks in Three Subjects")

plt.show()
```



Notes

Output:**Fig. 6.7: Creating a Bar graph using Matplotlib****Creating a Scatter plot**

Scatter plots help show relationships between two variables. Here is a code snippet that will plot a basic scatter plot:

```
import matplotlib.pyplot as plt

plt.scatter([1, 2, 3, 4], [10, 20, 15, 25])

plt.title("Simple Scatter Plot")

plt.show()
```



Output:

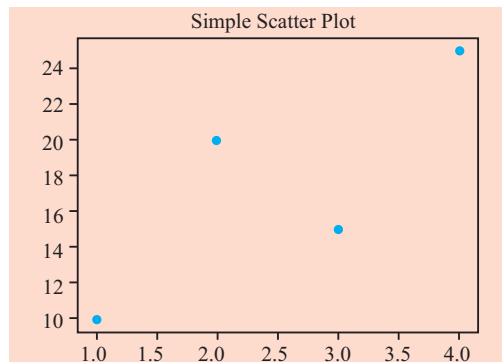


Fig. 6.8: Creating a Scatter Plot using Matplotlib

Creating a Pie chart

Pie charts show parts of a whole. Here is a code snippet plotting a pie chart on 'How I spend my day':

```
import matplotlib.pyplot as plt

activities = ["Study", "Play", "Sleep"]
time = [4, 2, 8]

plt.pie(time, labels=activities)

plt.title("How I Spend My Day")

plt.show()
```

Output:



Fig. 6.9: Creating a Pie Chart using Matplotlib

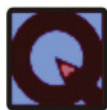


Notes

Basic steps for all Matplotlib graphs

Every graph in Matplotlib usually follows these steps:

1. Import the library – import matplotlib.pyplot as plt
2. Prepare the data – lists, arrays, files, and more
3. Choose the type of graph – plot, bar, pie, scatter, and more
4. Add labels and title – plt.xlabel(“X-axis”) plt.ylabel(“Y-axis”) plt.title(“Graph Title”)
5. Display the graph – plt.show()



INTEXT QUESTIONS 6.3

Short-answer questions

1. Why is Matplotlib important for data analysis?
2. What is the first step before creating a Matplotlib graph?
3. What does a pie chart represent?
4. What does the “Show Me” panel in Tableau provide?
5. What is Tableau?

6.5 DATA VISUALIZATION WITH TABLEAU

6.5.1 Introduction to Tableau

Tableau is a software that allows users to connect to different types of data, analyse it, and then create visualizations such as bar graphs, line charts, maps, and dashboards. It is especially helpful for students and beginners because it allows drag-and-drop visual creation without the need for coding.

Tableau is a powerful data visualization tool used to create interactive dashboards, charts, and reports. It helps people understand data through visuals rather than complex numbers or tables. Tableau is widely used in industries such as business, education, healthcare, and government because it makes data easy to explore and interpret.



How does it work?

Tableau works in three main steps:

1. **Connect to Data:** Users can connect Tableau to Excel files, CSV files, Google Sheets, databases, and more.
2. **Analyze Data:** Tableau allows filtering, sorting, grouping, and performing calculations on the data.
3. **Visualize Data:** Users can drag and drop fields to create graphs, charts, and dashboards.

6.5.2 Tableau Installation

1. Download Tableau from link <https://public.tableau.com/app/discover>
2. Install the package via the install wizard
3. Once installed, double click the program to open the Tableau Public Desktop application
4. This is what you should see once the app is open.
5. Now we are ready to pull in our data!

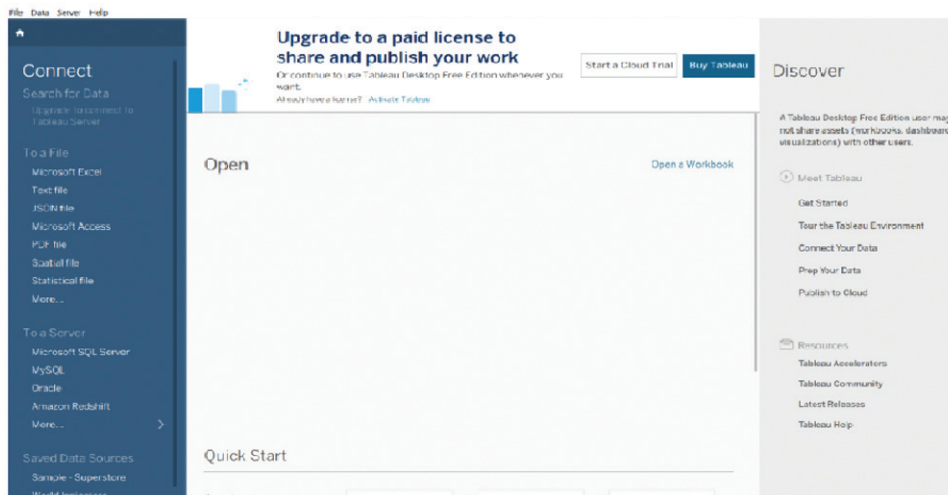


Fig. 6.10: Tableau

Dimensions and Measures: Tableau identifies each field as a dimension or measure in the Data pane, depending on the type of data the field contains.

Dimensions typically contain qualitative values (such as names, dates, or geographical data). You can use dimensions to categorize your data.



Notes

Applied Data Handling and Visualization

Measures typically contain numeric, quantitative values that you can measure. For example, SUM (Sales)

Tableau Example: Analysing Air Quality Data Across Indian States

Analysing Air Quality Data Using Tableau

Air pollution is one of the major environmental challenges faced by many regions across India. The Air Quality Index (AQI) is used to measure and communicate the quality of air in a particular location. It helps citizens and policymakers understand the level of pollution and take appropriate actions when necessary.

In this activity, we will use Tableau to analyse AQI data collected from different states and areas of India. The dataset contains information such as the state name, area, AQI value, prominent pollutant, number of monitoring stations, and air quality status. Through data visualisation, we can identify pollution patterns, compare regions, and draw meaningful conclusions.

6.5.3 Preparing the Dataset

You should already have a copy of Tableau Desktop installed on your computer. Download the AQI Dataset CSV file to your computer, then simply drag and drop the CSV file onto the Tableau Desktop icon or into an open Tableau window.

AirQuality

Connection

Live

Extract

Filters0 | Add

aqi.csv

9 fields 235785 rows

100rows

Name

aqi.csv

Description

No description available.

Fields

Type	Field Name	Phys...	Rem...
	Date	aqi.csv	date
	State	aqi.csv	state
Abc	Area	aqi.csv	area
#	Number O	aqi.csv	numb...
Abc	Prominent	aqi.csv	promi...
#	Aqi Value	aqi.csv	aqi_va...
Abc	Air Quality	aqi.csv	air_qu...
Abc	Unit	aqi.csv	unit
Abc	Note	aqi.csv	note

	aqi.csv	Abc aqi.csv	# aqi.csv	Abc aqi.csv
Date	State	Area	Number Of Monitoring St	Prominent Pollutants
28-07-2024	Uttar Pradesh	NOIDA	4	PM10
28-07-2024	Maharashtra	Dhule	1	PM10
28-07-2024	Gujarat	Surat	1	CO
28-07-2024	Rajasthan	Sikar	1	O3
28-07-2024	Rajasthan	Bikaner	1	PM10
28-07-2024	Rajasthan	Bhiwadi	2	PM10
28-07-2024	Haryana	Panchkula	1	O3
28-07-2024	Bihar	Muzaffarpur	2	O3
28-07-2024	Uttar Pradesh	Varanasi	1	PM10
28-07-2024	Haryana	Charkhi Dadri	1	PM2.5
28-07-2024	Madhya Pra...	Gwalior	4	PM10,O3,CO
28-07-2024	Madhya Pra...	Pithampur	1	PM10
28-07-2024	Haryana	Kurukshetra	1	PM10
28-07-2024	Haryana	Palwal	1	PM10
28-07-2024	Haryana	Gurgaon	2	PM2.5,O3

Fig. 6.11: Sample AQI Dataset Loaded in Tableau

Tableau will automatically recognize the file as a Text File Connection and configure the import settings. Review the data preview to ensure that fields such



as Area, AQI Value, Pollution Level, and other relevant attributes have been imported correctly, then click OK. You may see a dialog box asking whether you would like to create an extract or save the data connection; for this exercise, select No and continue working with the live data connection.

This will load the AQI dataset into Tableau, allowing you to create visualizations and analyze air quality patterns across different locations.

Editing Aliases and Data Labels

After importing the AQI dataset into Tableau, review the field names and values to ensure they are easy to understand and interpret.

For example, if the Air Quality Status field contains abbreviated values such as *G*, *S*, *M*, *P*, *VP*, and *Severe*, you can create more meaningful labels by right-clicking the field, selecting Edit Aliases, and replacing the abbreviations with their full forms:

- G → Good
- S → Satisfactory
- M → Moderate
- P → Poor
- VP → Very Poor
- Severe → Severe

Similarly, if any field names are unclear, they can be renamed for better readability. For instance, a field named AQI can be renamed to AQI Value, and Pollutants can be renamed to Prominent Pollutants.

It is also important to verify that Tableau has correctly classified the fields. Fields such as Date, State, Area, Air Quality Status, and Pollution Level should be treated as Dimensions, since they are used for categorization and grouping rather than numerical calculations.

If any of these fields appear under Measures, simply drag them to the

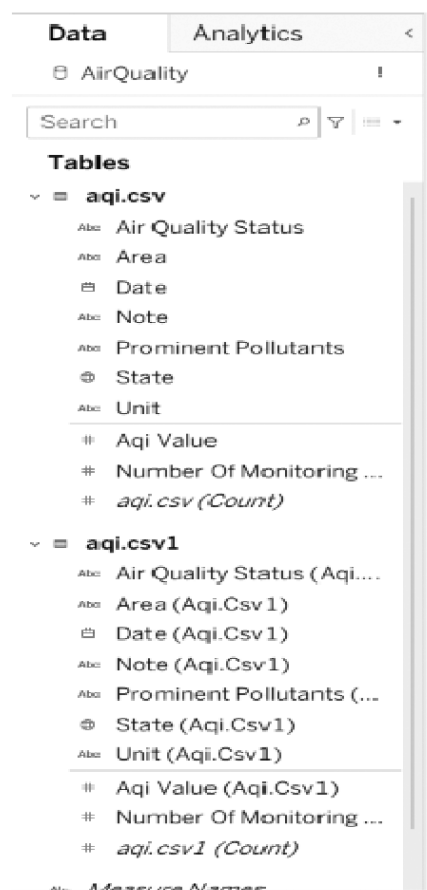


Fig. 6.12: Dimensions and Measures in Tableau



Notes

Dimensions section in the Data Pane. This ensures that the data can be analyzed and visualized correctly.

Creating Calculated Fields

In Tableau, it's easy to create new fields based on existing data. Our AQI dataset contains a numerical field called **AQI Value**, but it would be more useful if we could group these values into broader pollution categories.

To do this, right-click in the Data Pane and choose **Create Calculated Field**. Name the new field **Pollution Level** and enter the following formula:

1. IF [AQI Value] <= 100 THEN "Low Pollution"
2. ELSEIF [AQI Value] <= 200 THEN "Moderate Pollution"
3. ELSE "High Pollution"
4. END

Click **OK** to create the field.

The formula examines the AQI Value for each record and assigns it to one of three categories. Values up to 100 are classified as **Low Pollution**, values between 101 and 200 are classified as **Moderate Pollution**, and values above 200 are classified as **High Pollution**.

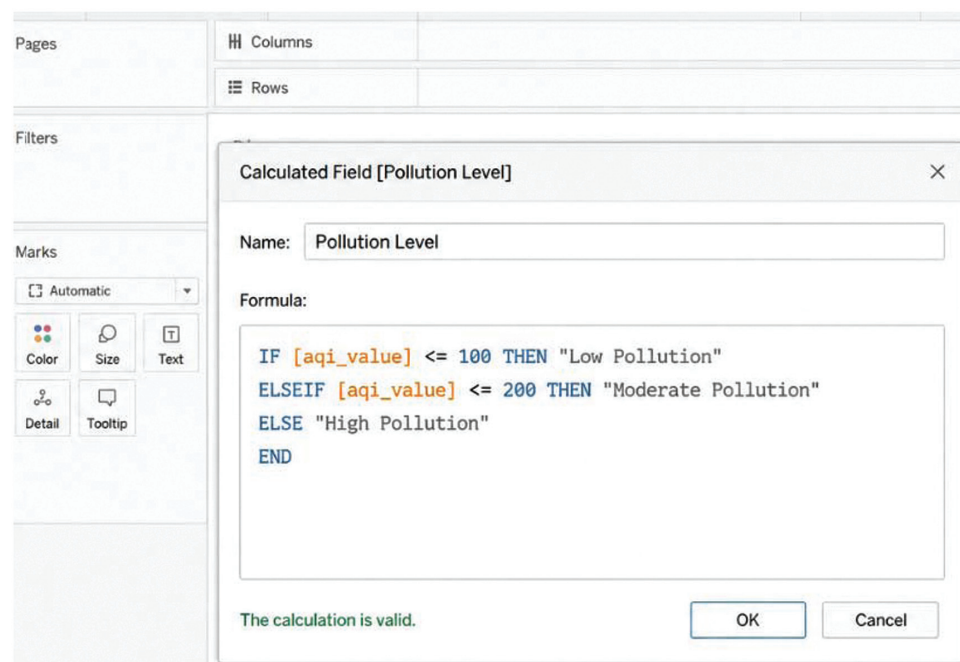


Fig. 6.13: Creating a Calculated Field in Tableau



The newly created **Pollution Level** field appears under **Dimensions** and can be used in charts, filters, and dashboards. This makes it easier to compare air quality across different areas without having to interpret individual AQI values.

Observation

Based on the given dataset:

Area	AQI Value	Pollution Level
Amravati	78	Low Pollution
Purnia	56	Low Pollution
Katni	98	Low Pollution
Tumidih	103	Moderate Pollution
Imphal	106	Moderate Pollution

6.6 CREATING CHARTS

6.6.1 Bar Chart

Bar charts are useful for comparing values across different categories. In our AQI dataset, a bar chart can be used to compare the **AQI Value** of different areas and identify locations with higher or lower pollution levels.



Fig. 6.14: Bar Chart Comparing AQI Values



Notes

Applied Data Handling and Visualization

To create a bar chart, drag **Area** to the **Columns** shelf and **AQI Value** to the **Rows** shelf. Tableau automatically creates a visualization showing the AQI values for each area. You can also select **Bar Chart** from the **Show Me** panel to ensure the data is displayed as bars.

To make the chart easier to interpret, sort the bars in descending order by AQI Value. This places the most polluted areas at the beginning of the chart and the least polluted areas at the end.

The resulting chart displays the AQI values of all areas included in the dataset, making it easy to compare air quality levels across locations and identify areas with relatively high or low pollution.

Observation

The chart shows that:

- Imphal recorded the highest AQI value.
- Tumidih recorded the second-highest AQI value.
- Nalbari recorded the lowest AQI value among the selected locations.

Such comparisons help identify regions with relatively poorer air quality.

6.6.2 Analysing Air Quality Status

Creating a Pie Chart

The dataset contains the field **Pollution Level**, which classifies locations into categories such as **Low Pollution**, **Moderate Pollution**, and **High Pollution**. A pie chart can be used to visualize the proportion of locations belonging to each pollution category.

To create a pie chart, drag **Pollution Level** to the **Marks** card and select **Pie Chart** as the chart type. Next, drag **Number of Records** to the **Angle** shelf. Tableau automatically divides the pie into slices based on the number of records in each pollution category.

To make the chart easier to interpret, display labels on the chart so that each slice shows its corresponding pollution category and value.

The resulting chart provides a clear view of the distribution of locations across different pollution levels, making it easy to identify which category contains the largest share of locations in the dataset.

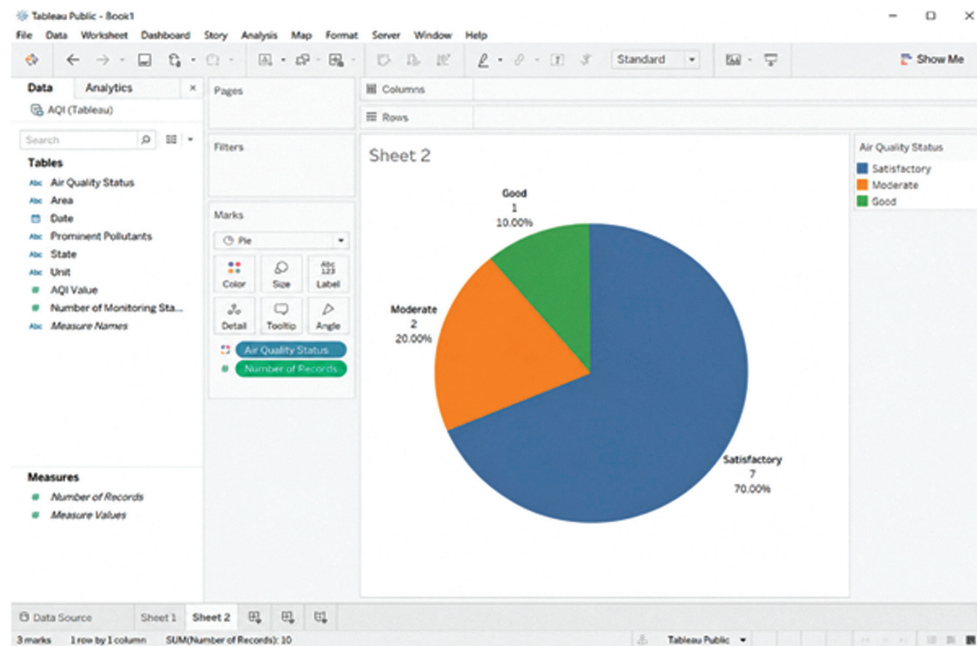


Fig. 6.15: Distribution of Air Quality Status

Observation

The pie chart indicates that most locations fall under the **Satisfactory** category, while only a few locations are classified as **Good** or **Moderate**.

6.6.3 Identifying Prominent Pollutants

Different pollutants contribute to air pollution in different regions. The **Prominent Pollutants** field in the dataset identifies the pollutant that has the greatest impact on air quality in a particular area. By visualizing this field, we can determine which pollutants occur most frequently across the dataset.

To create this visualization, drag **Prominent Pollutants** to the **Columns** shelf and **Number of Records** to the **Rows** shelf. Tableau automatically displays the count of records associated with each pollutant. You can also select **Bar Chart** from the **Show Me** panel to present the data more clearly.

The resulting chart displays the frequency of each prominent pollutant in the dataset. This makes it easy to compare pollutants and identify those that contribute most frequently to air pollution across different locations. Such analysis can help environmental agencies and policymakers focus their efforts on controlling the pollutants that have the greatest impact on air quality.



Notes

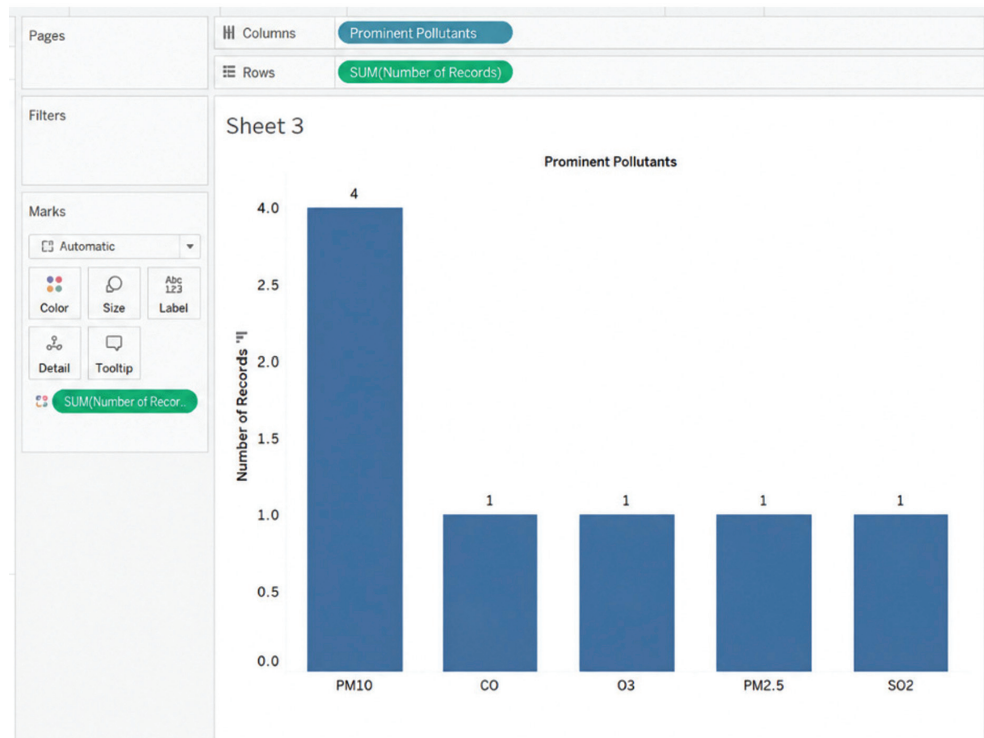


Fig. 6.16: Frequency of Prominent Pollutants

Observation

The chart shows that **PM10** appears most frequently in the dataset. Other pollutants such as CO, O₃, PM2.5, and SO₂ are observed in fewer locations.

6.6.4 Applying Filters

Filters help users focus on specific parts of the dataset by displaying only the records that meet selected criteria. They make it easier to explore the data and perform detailed analysis without being distracted by irrelevant information.

In the AQI dataset, filters can be applied to fields such as **State**, **Air Quality Status**, and **Prominent Pollutants**. To add a filter, drag the desired field to the **Filters** shelf and select the values that you want to include in the analysis. The filter can also be displayed on the worksheet, allowing users to interactively choose different categories.

Applying filters enables users to examine specific regions, air quality categories, or pollutants of interest. For example, a user may choose to view data for a particular state, analyze locations with a specific air quality status, or identify areas affected by a particular pollutant.

The use of filters makes visualizations more interactive and helps users gain deeper insights from the dataset by focusing on the information that is most relevant to their analysis.



Notes

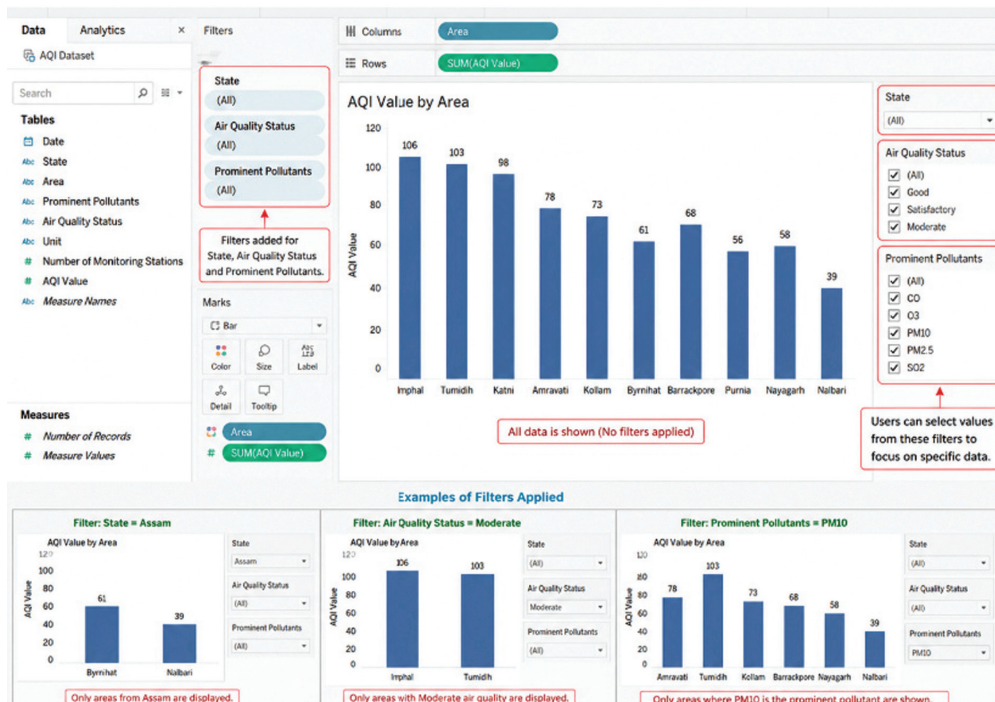


Fig. 6.17: Applying Filters in Tableau

6.6.5 Drawing Conclusions from the Visualisations

After creating the visualisations, meaningful insights can be obtained from the AQI dataset. Charts and graphs make it easier to identify trends, compare values, and understand the distribution of air quality indicators across different locations.

By analysing the visualisations, users can identify areas with high AQI values, determine the most frequently occurring pollutants, and observe the distribution of air quality categories. The visualisations also help highlight patterns that may not be immediately apparent when viewing the raw data.

Some key observations from the analysis include the identification of locations with relatively high pollution levels, the pollutants that contribute most frequently to poor air quality, and the proportion of areas belonging to different air quality categories. Such findings can help environmental agencies, researchers, and policymakers better understand air pollution patterns and make informed decisions.



Notes

Applied Data Handling and Visualization

Overall, data visualisation transforms raw AQI data into meaningful information, making it easier to communicate findings and support effective environmental analysis.

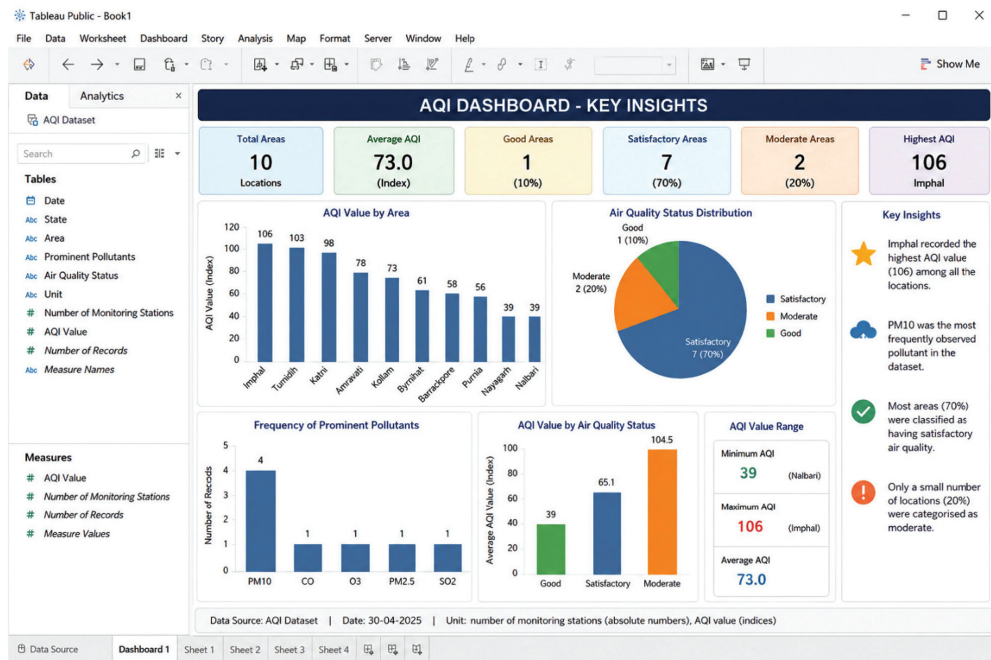


Fig. 6.18: AQI Dashboard Showing Key Insights

Activity

Using Tableau, analyse the AQI dataset and answer the following questions:

1. Which area recorded the highest AQI value?
2. Which pollutant appeared most frequently in the dataset?
3. How many locations fall under each air quality category?
4. Which state has the highest AQI value in the dataset?
5. What conclusions can be drawn from the visualisations?

**INTEXT QUESTIONS 6.4****Short-answer questions**

1. Which Pandas function is best suited to check the total count of non-null values and the data type of each column in a DataFrame?
(a) head() (b) describe() (c) info()
2. **True or False:** NumPy arrays allow for faster mathematical operations than standard Python lists.
3. Write the NumPy command to create a 2D array (matrix) of size 3×3 containing all zeros.
4. If a column contains a missing value (NaN), what is one way to handle it using a Pandas DataFrame method?
5. What type of chart would you use to show the trend of global temperature over the last 50 years?

**WHAT HAVE YOU LEARNT**

- NumPy: You can now create efficient ndarray structures to handle large numerical datasets quickly.
- Pandas: You can read, organize, and prepare data using the DataFrame structure, performing essential data cleaning and inspection tasks.
- Matplotlib: You can generate various graphical representations of data (plots and charts) using Matplotlib to identify trends and communicate findings effectively.
- Data visualization helps us understand patterns, trends, and comparisons quickly through pictures.
- Different graphs serve different purposes – bar graphs compare categories, line charts show trends, pie charts show parts of a whole, and scatter plots show relationships.
- Tableau is a software tool that creates interactive charts and dashboards using drag-and-drop without coding.



Notes



Notes

Applied Data Handling and Visualization

- Data can be prepared in Excel and imported into Tableau to create visuals like bar charts, bubble charts, and maps.
- Visualization tools like Matplotlib and Tableau make data easy to explore, understand, and present.



TERMINAL QUESTIONS

Short-answer questions

1. What is the main purpose of using NumPy instead of Python lists?
2. What are Pandas Series and DataFrames?
3. Why is matplotlib.pyplot imported as plt?
4. Why is data visualization important? Give one reason.
5. What type of data is best represented using a line chart?
6. Define a scatter plot.
7. What is the purpose of plt.show()?
8. What does df.sort_values() help achieve?

Long-answer questions

1. Explain the differences between NumPy arrays and Python lists. Why are NumPy arrays preferred for numerical operations?
2. Describe the steps involved in loading and inspecting data using Pandas. Include functions like head(), info(), and describe().
3. Explain the different ways to handle missing data in Pandas. Give examples of functions used.
4. Describe any four types of data visualisation and explain when each one should be used.
5. Explain the steps involved in creating visualisations using Matplotlib. Mention the functions used in each step.
6. What is Tableau? Explain how it works, including connecting data, analysing, and visualising it.



7. Explain the process of creating a bar chart in Tableau using the given example of “songs per genre.”
8. Compare Matplotlib and Tableau as data visualisation tools. How does each support data analysis?

Skill Based Questions

1. You are given a dataset with missing values, incorrect formatting, and mixed data types. Describe a step-by-step plan using Pandas to clean and prepare this dataset for analysis.
2. If you were analysing pollution levels across India, which visualisation type would best represent the data? Justify your choice.
3. You created a bar graph in Matplotlib but feel it is unclear. Suggest modifications (titles, labels, styles, etc.) that would make the visualisation more meaningful.
4. You need to explain a large dataset to your classmates. Would you choose Matplotlib or Tableau? Provide reasons considering ease of use, interactivity, and visual quality.



ANSWER TO INTEXT QUESTIONS

6.1

1. Numerical Python
2. N-dimensional arrays
3. `numpy.array()`
4. maximum
5. `ndarray`

6.2

1. (b)
2. (a)
3. (c)
4. (c)
5. (c)
6. (c)

6.3

1. It helps convert raw data into easy-to-understand visual graphs.
2. The first step is importing the library using `import matplotlib.pyplot as plt`.



Notes

Applied Data Handling and Visualization

3. A pie chart represents data distribution in parts of a whole as slices of a circle.
4. It provides different types of visualization options like bar charts, maps, and bubble charts.
5. Tableau is a data visualization software used to create charts, dashboards, and reports.

6.4

1. **Multiple Choice:** (c) info()
2. **True or False:** True.
3. `numpy.zeros((3, 3))`
4. One way is using `df.dropna()` (to remove rows with missing values) or `df['ColumnName'].fillna(value)` (to fill missing values with a specified value).
5. A **Line Plot** is typically used to show trends over time.

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- use NumPy arrays and Pandas for mathematical and logical operations.
- create data visualizations using Matplotlib.
- use Tableau for data visualization

References

1. CBSE Academic, *Python Content Manual*.



Notes

7

INTRODUCTION TO MACHINE LEARNING

In the previous lesson, you learned about Data Handling and Data Visualisation. You discovered how data is collected from various sources and visualized to understand trends. However, having data is only the first step. Data alone is just a collection of numbers and facts. To make it truly powerful, we need to extract intelligence from it.

In this lesson, we will explore how computers can go beyond simply storing data to actually learning from it. We will move from traditional programming, where humans give every command, to Machine Learning, where computers discover patterns and make decisions on their own. This technology is the engine behind many modern conveniences, from unlocking your phone with your face to recommending your next favorite song.



OBJECTIVES

After learning this lesson, the learner will be able to:

- define the concepts of Artificial Intelligence and Machine Learning;
- differentiate between Rule-Based approaches and Machine Learning approaches;
- explain the three main families of Machine Learning models: Supervised, Unsupervised, and Reinforcement Learning;
- evaluate the performance of models using metrics like the Confusion Matrix and R-squared score.



Notes

7.1 MACHINE LEARNING

Have you ever wondered how your smartphone unlocks by scanning your face or how streaming services seem to know exactly what movie you want to watch next? It isn't magic – it's Machine Learning. To understand Machine Learning, we first need to look at the bigger picture of Artificial Intelligence (AI).

7.1.1 What is AI?

Artificial Intelligence (AI) is any technique that enables computers to mimic human intelligence. It is the broad science of making machines “smart.”

7.1.2 What is Machine Learning (ML)?

Machine Learning is a specific subset of AI. Instead of just following strict instructions, ML enables machines to improve at tasks with experience. Think of it like learning to ride a bicycle. You didn't learn by memorizing a physics textbook; you learned by trying, falling and adjusting your balance. Machine Learning allows computers to learn in a similar way – by looking at data and finding patterns.

7.1.3 Hierarchy of Intelligence

It is helpful to visualize these concepts as concentric circles:

- AI is the largest circle (the broad concept).
- Machine Learning (ML) sits inside AI.
- Deep Learning (DL) is a specialized subset of ML where software trains itself to perform tasks using vast amounts of data.

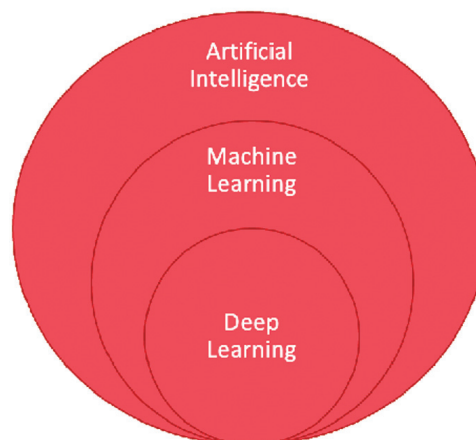


Fig. 7.01



7.2 HOW IS ML DIFFERENT FROM TRADITIONAL PROGRAMMING?

Until this lesson, you have been made aware about Traditional programming; In traditional programming (often called a Rule-Based Approach), you must explicitly define every rule for the computer to follow.

- **Rule-Based Approach:** We give the computer Rules and Data, and it gives us Answers. For example, If we want a computer to convert Celsius to Fahrenheit, we must explicitly program the formula ($F = C * 1.8 + 32$), where F is temperature in degrees Fahrenheit and C is Temperature in degrees Celsius.

In a Machine Learning Approach, we flip this logic. We use machines to figure out the relationship for us.

- **Machine Learning Approach:** We give the computer Data and the Answers, and the machine figures out the Rules. *Example:* We show the computer thousands of pictures labeled “Cat” and “Not Cat.” The computer figures out the visual “rules” (like pointy ears or whiskers) that define a cat on its own.

Under ML, we have different ways to train a model; just like humans learn in different ways – sometimes by a teacher correcting us, sometimes by exploring on our own, and sometimes by trial and error – computers also have different ways of learning. The specific method a computer uses depends on the type of data available and the problem we want to solve. Generally, there are three main families of machine learning models:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

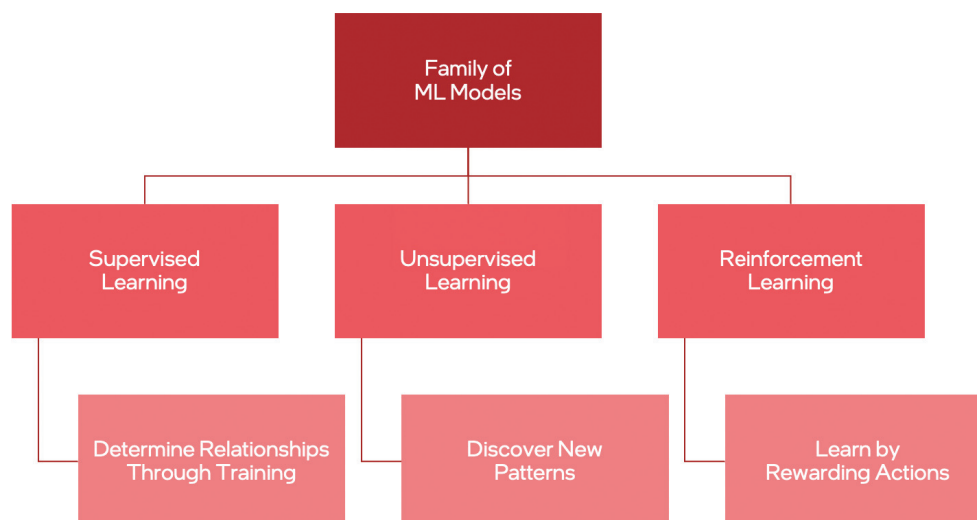


Fig. 7.02



Notes

Let us start looking at each of these ‘ways of training’ the Models, with one algorithm each.

7.2.1 Supervised Learning

This is the most common type of Machine Learning. Imagine a classroom where a teacher is supervising a student. The teacher knows the answers and guides the student. In Supervised Learning, the computer is trained using data that is “Labeled.” This means every piece of data comes with a “tag” or an answer key. The machine looks at the data (Input) and the label (Output) and tries to figure out the relationship between them.

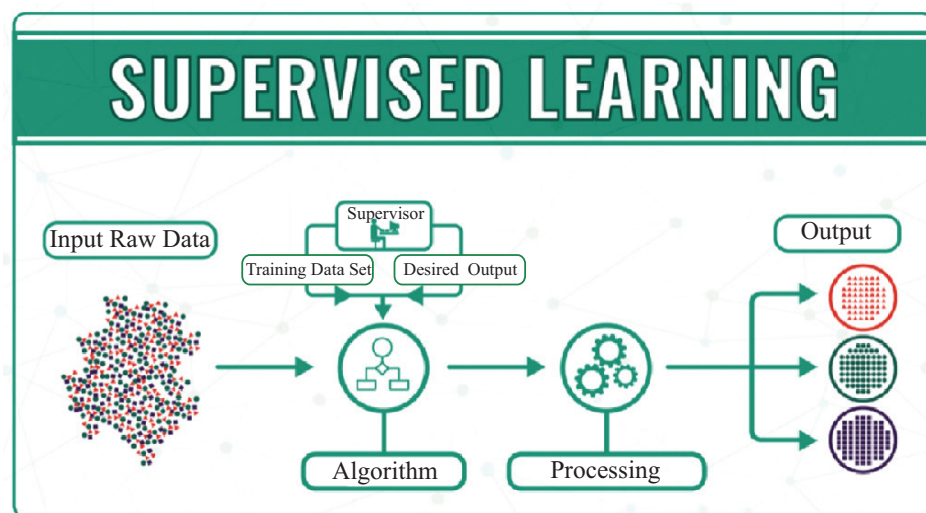


Fig. 7.03

Analogy: Imagine you are learning a new language using flashcards. On one side is a picture of an apple, and on the back, it says “Apple.” You look at the picture, guess the name, and check the back to see if you are right. If you are wrong, you correct yourself. After doing this hundreds of times, you can identify an apple instantly.

7.2.2 Types of Supervised Learning

Supervised learning is further divided into two main categories:

1. **Classification (Predicting a Category):** Here, the goal is to sort items into specific groups or “classes.” The answer is usually a “Yes/No” or a specific label.

Example: Spam Filters. Your email provider looks at millions of emails. Some are labeled “Spam” and others “Inbox.” The computer learns that words like “Lottery,” “Free,” and “Winner” often mean “Spam.”

Real-World Use: Identifying if a tumor in an X-ray is “Benign” or “Malignant.”



2. **Regression (Predicting a Quantity):** Here, the goal is to predict a continuous number, not a category.

Example: Predicting House Prices. A computer looks at data like “Size of house,” “Number of bedrooms,” and “Location” to predict the exact price (e.g., Rs. 250,0000). It isn’t a “Yes/No” question; it’s a mathematical calculation based on trends.

Real-World Use: Predicting the temperature for tomorrow or estimating how long your Uber ride will take.

7.2.3 Unsupervised Learning

Now, imagine you are an alien who has just landed on Earth. You enter a library. You don’t know what “books” or “magazines” are. You don’t know how to read. However, you start noticing patterns. You see that some items are hardbound and have more pages (Books), while others are glossy and have few pages (Magazines). You start separating them into two piles based on their look and feel. This is how Unsupervised Learning works, the data is “**Unlabeled.**” The computer is given a massive amount of data with no answer key. It has to explore the data and find structures, patterns, or groupings on its own.

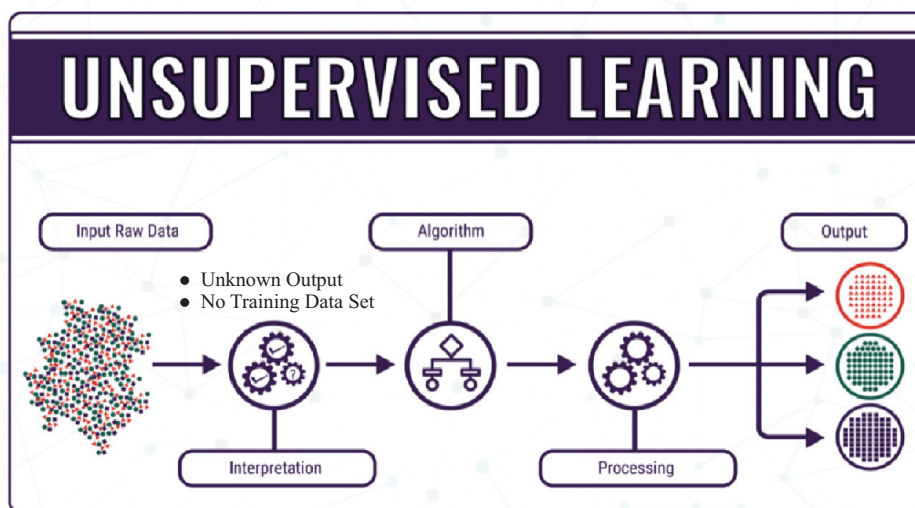


Fig. 7.04

7.2.4 Types of Unsupervised Learning:

Clustering (Grouping Patterns): The algorithm groups data points that are similar to each other.

Example: **Customer Clustering.** A shopping website like Amazon looks at the buying habits of millions of people. It notices that “People who buy bread often buy



Notes

butter.” It groups these users into a “Bread-Butter Buyer” cluster, even without explicitly asking them anything.

Real-World Use: Netflix grouping movies into genres like “Suspense thrillers” based on plot keywords, even if no human labeled them that way.

7.2.5 Reinforcement Learning

Think about how you learn to play a new video game, like Super Mario. You don’t read a manual. You just pick up the controller and try.

- **The puppy sits when asked:** You give it a treat → (**Reward**).
- **The puppy chews on your shoe:** You say “No!” and take the shoe away → (**Penalty**).
- **The puppy fetches the ball:** You rub its head and say “Good boy!” → (**Positive Reward**).

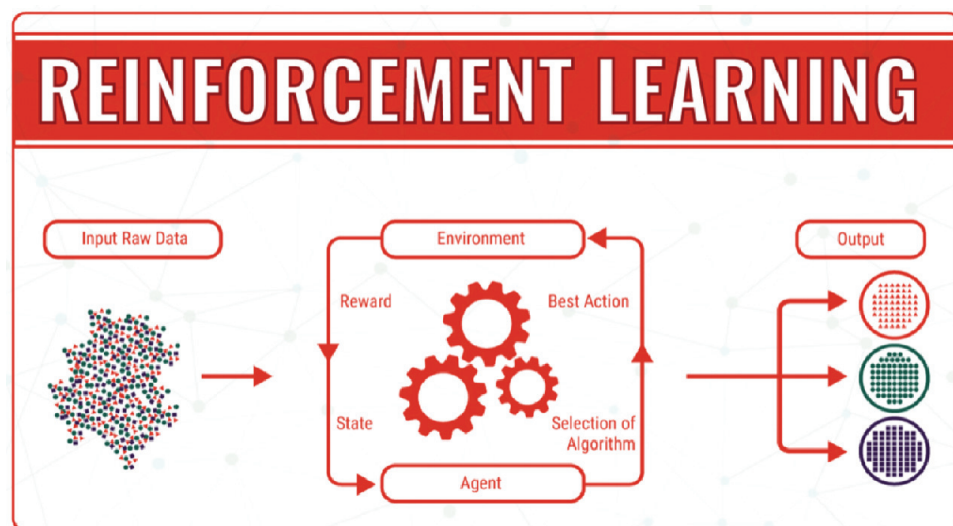


Fig. 7.05

Over time, you learn to avoid the bad things and seek out the good things to get the highest score. Reinforcement Learning (RL) is all about learning through trial and error. It involves an Agent (the AI) interacting with an Environment.

- **Action:** The move the agent makes.
- **Reward/Penalty:** The feedback the environment gives based on the action.

The goal of the agent is to maximize its total reward. Unlike Supervised Learning, where the answer is given immediately, in RL, the reward might come much later (like winning a chess game after 50 moves).



Real-World Application:

- **Self-Driving Cars:** The car gets a “reward” for staying in the lane and a “penalty” for going off-road. Over millions of simulations, it learns to drive safely.
- **Robotics:** A robot dog learning to walk. If it balances, it gets a numerical reward. If it falls, it gets a penalty.

7.3 EVALUATING MODELS

Imagine you studied hard for a math test and felt confident. You walked out thinking you got 100%. But when you got your paper back, you only got 60%. What happened? Maybe you memorized the wrong formulas, or maybe you made silly mistakes.

In Machine Learning, training the model is only half the battle. We can’t just assume our model is smart; we need to test it. Evaluation is the process of giving our model a “Report Card” to see how well it actually works on new data.

7.3.1 Evaluating Classification Models: The Confusion Matrix

When we use a Classification model (where the answer is a Category like “Yes/No” or “Cat/Dog”), a simple “Accuracy” score (like 90%) can be misleading.

To get the full picture, we use a Confusion Matrix. Despite the name, it isn’t meant to confuse you! It is a specific table that helps you see where your model is getting confused.

The Concept: Let’s take an example of a “Forest Fire Prediction” AI.

- Positive Case (1): There is a Forest Fire.
- Negative Case (0): There is No Fire.

When the AI makes a prediction, there are four possible outcomes:

1. True Positive (TP):
 - *Reality:* Fire.
 - *AI Prediction:* Fire.
 - *Result:* Correct! The alarm rings, and firefighters save the forest.
2. True Negative (TN):
 - *Reality:* No Fire.



Notes

- *AI Prediction:* No Fire.
 - *Result:* Correct! The alarm stays silent, and everyone is peaceful.
3. False Positive (FP) - “Type 1 Error”:
- *Reality:* No Fire.
 - *AI Prediction:* Fire!
 - *Result:* False Alarm. The alarm rings, firefighters rush out, but there is no fire. It’s annoying and a waste of resources, but not a disaster. This is also known as the “Boy Who Cried Wolf” error.
4. False Negative (FN) - “Type 2 Error”:
- *Reality:* Fire.
 - *AI Prediction:* No Fire.
 - *Result:* Dangerous! The alarm stays silent while the forest burns. This is the worst kind of error in this scenario.

7.3.2 Evaluating Regression Models: R-Squared (R²) Score

When we are doing Regression (e.g., Predicting house prices), we don’t have “Yes/No” answers. If a house costs Rs.50,00000 and our AI predicts Rs.49,99999, it is technically “wrong,” but it is a great prediction!

For this, we use the R-squared (R²) Score.

- What it is: It is a statistical measure that tells us how close the data is to the fitted regression line.
- The Analogy: Think of it like playing darts.
- If all your darts (predictions) land tight around the bulls eye (actual value), you have a High R² (close to 1).
- If your darts are scattered all over the wall, you have a Low R² (close to 0).

The Scale:

- 1 (or 100%): Perfect fit. The model explains all the variation in the data.
- 0 (or 0%): The model explains none of the variation (it’s basically guessing).



INTEXT QUESTIONS 7.1

Multiple-choice questions:

1. What is Artificial Intelligence (AI)?
 - (a) A programming language
 - (b) The simulation of human intelligence by machines
 - (c) A type of computer hardware
 - (d) A database system
2. Which type of Machine Learning uses labeled data?
 - (a) Unsupervised Learning
 - (b) Reinforcement Learning
 - (c) Supervised Learning
 - (d) Deep Learning
3. Which learning method uses rewards and penalties?
 - (a) Supervised Learning
 - (b) Unsupervised Learning
 - (c) Reinforcement Learning
 - (d) Transfer Learning
4. What is a dataset?
 - (a) A computer program
 - (b) A collection of data used for training and testing models
 - (c) A hardware device
 - (d) A network protocol
5. The R-squared (R^2) score is mainly used to evaluate
 - (a) Classification models
 - (b) Clustering models
 - (c) Regression models
 - (d) Reinforcement learning models



WHAT YOU HAVE LEARNT

- The concepts of Artificial Intelligence and Machine Learning;
- Difference between Rule-Based approaches and Machine Learning approaches;



Notes



Notes

Introduction to Machine Learning

- The three main families of Machine Learning models: Supervised, Unsupervised, and Reinforcement Learning;
- evaluate the performance of models using metrics like the Confusion Matrix and R-squared score



TERMINAL QUESTIONS

1. Define Artificial Intelligence and Machine Learning.
2. Write the difference between Rule-Based approaches and Machine Learning approaches.
3. Explain the three main families of Machine Learning models.
4. Explain the Confusion Matrix and R-squared score.



ANSWERS TO INTEXT QUESTIONS

1. (b)
2. (c)
3. (c)
4. (b)
5. (c)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- explain the three core types of Machine Learning i.e. Supervised learning, Unsupervised learning and Reinforcement learning.
- evaluate the performance of models using metrics like the Confusion Matrix and R-squared score.



8

NO-CODE AND LOW-CODE AI TOOLS

In the previous lessons, you learned the theory behind Machine Learning – how computers learn from data using complex algorithms like Neural Networks and Decision Trees. Historically, building these models required advanced mathematics and thousands of lines of complex programming code. It was a superpower reserved for computer scientists.

But today, that has changed. We have entered the era of **No-Code and Low-Code AI**. Just like you don't need to know how to wire an engine to drive a car, you no longer need to be a master coder to build powerful AI models. In this lesson, we will explore tools that let you build, train, and test AI models using simple visual interfaces and drag-and-drop actions. You will go from “learning about AI” to “building AI” in minutes!



OBJECTIVES

After reading this lesson, you will be able to:

- define the concepts of No-Code and Low-Code AI development;
- navigate and utilize Google's Teachable Machine to create image or audio classifiers;
- explain the workflow of Lobe.ai for desktop-based model training;
- utilize Dialogflow to understand service chatbot and agents;
- build a functional prototype of an AI model without writing code.



Notes

8.1 NO-CODE AND LOW-CODE

Imagine wanting to build a LEGO castle. In the past (Traditional Coding), you would have to mold the plastic and manufacture the bricks yourself. Today (No-Code), you just open the box and start snapping pre-made bricks together.

- **No-Code Tools:** These are platforms where you can build software or AI models using a graphical user interface (GUI). You drag, drop, click, and upload. You don't write a single line of code.
 - *Target Audience:* Students, Business Analysts, Designers.
- **Low-Code Tools:** These platforms handle most of the heavy lifting but allow you to add small snippets of code to customize special features.
 - *Target Audience:* Developers who want to work faster.

Why is this important? It “democratizes” AI. It means a doctor can build an AI to spot diseases or a musician can build an AI to sort songs, without needing a degree in Computer Science.

8.2 GOOGLE TEACHABLE MACHINE

Teachable Machine is a web-based tool by Google that makes creating machine learning models fast, easy, and accessible to everyone. It is primarily a Supervised Learning tool, meaning you act as the “teacher” by providing examples (labeled data), and the machine learns from them.

How it Works:

Teachable Machine simplifies the complex ML cycle into three intuitive steps:

1. **Gather (Input):** This is where you collect your data. You create “Classes” (categories) and fill them with examples.
 - *Input Types:* You can use images (via webcam or upload), sounds (via microphone), or poses.
 - *Example:* To train a “Pet Sorter,” you would create a “Cat” class and upload 50 photos of cats, then a “Dog” class with 50 photos of dogs.
2. **Train (Processing):** The computer looks at your examples and learns to distinguish between them.
 - *Under the Hood:* It builds a Neural Network in your browser. It uses a process called Transfer Learning – it starts with a model that already



knows general shapes (like “edges” or “circles”) and fine-tunes it to recognize your specific classes (like “ears” or “whiskers”).

- **Epochs:** During training, you might see a timer for “Epochs.” An Epoch is one complete cycle through your entire training dataset. The more epochs, the more the machine studies the data (though too many can lead to memorization/ overfitting!).
3. **Export (Output):** You test the model instantly. The output is usually a Confidence Score (e.g., “I am 95% sure this is a Cat”). If it works, you can export the model to use in an App, Website, or even a robot.

Practical Activity: Rock, Paper, Scissors Classification

Goal: Create an AI model that can referee a game of Rock-Paper-Scissors by recognizing hand gestures. Unlike the previous activity where you used a webcam to capture live data, here you will simulate a professional workflow by using a **pre-existing dataset**.

Prerequisite: Download the dataset → **Dataset Link:** Rock-Paper-Scissors Images (Kaggle). Download the ZIP file and extract it. You will see folders named “rock,” “paper,” and “scissors.”

Step-by-Step Instructions:

1. **Open Teachable Machine:** Go to teachablemachine.withgoogle.com and click **Get Started** → **Image Project** → **Standard Image Model**.
2. **Create Class 1 (Rock):**
 - Rename “Class 1” to “Rock”.
 - Click **Upload** → **Choose images from your files**.
 - Open the “rock” folder from your downloaded dataset, select about 20-50 images, and upload them.
3. **Create Class 2 (Paper):**
 - Rename “Class 2” to “Paper”.
 - Repeat the upload process using images from the “paper” folder.
4. **Create Class 3 (Scissors):**
 - Click **Add a Class** at the bottom.
 - Rename it to “Scissors” and upload images from the “scissors” folder.



Notes

5. **Train:** Click the **Train Model** button. Wait for the “Preparing training data...” bar to finish. (Do not switch tabs!)
6. **Test:**
 - Once trained, the “Preview” panel on the right will turn on your webcam.
 - Hold your own hand up to the camera in a “Rock” shape.
 - Does the bar for “Rock” go up to 100%? Try “Paper” and “Scissors.”
 - *Challenge:* If you tilt your hand or move it far away, does the model get confused? This helps you understand the limitations of your training data!

Export the model and Run in Colab (low code)

One of the most powerful features of **Teachable Machine** is that your model isn’t stuck inside the website. You can **Export** it to run on other computers, mobile apps, or even specialized hardware like a Raspberry Pi.

In this section, we will take our **Rock-Paper-Scissors** model and run it inside **Google Colab** using Python. This moves us from “No-Code” to “Low-Code” – we will use code, but it is pre-written for us!

Step 1: Export the Model

1. In your trained Teachable Machine project, click the **Export Model** button.
2. A pop-up window will appear. Select the **TensorFlow** tab (not .js or Lite).
3. Select **Keras** as the format.
4. Click **Download my model**.
5. After a few moments, a file named `converted_keras.zip` will download.
 - *Unzip this file.* You will see two files: `keras_model.h5` (the brain) and `labels.txt` (the answer key).

Step 2: Set Up Google Colab

1. Open Google Colab.
2. Click **New Notebook**.
3. On the left sidebar, click the **Folder icon** (Files).
4. Drag and drop your `keras_model.h5` and `labels.txt` files into this sidebar.

5. Also, find a test image of a hand gesture (e.g., test_rock.jpg) and drag it into the sidebar.

Step 3: The “Low-Code” Script

Teachable Machine actually writes the Python code for you! You can copy it directly from the Export window, or use the clean version below.

8.3 DIALOG FLOW

We just looked at a no code tool for Vision (Teachable Machine). Now, let’s look at a professional tool used by major companies to understand Natural Language: Google’s Dialogflow. Unlike simple “If/Then” bots, Dialogflow uses Natural Language Understanding (NLU). This means it doesn’t just match keywords; it tries to understand the *meaning* behind what you say.

To use Dialogflow, you need to understand three words:

1. Agent: This is your bot. Think of it as a call center employee.
2. Intent: What the user *wants*. (e.g., User says “I’m hungry” → Intent: order_food).
3. Training Phrases: Examples of how users might express that intent. (e.g., “I want food,” “Can I eat?”, “Menu please”).
4. Response: What the bot says back.

Practical Activity Pizza Agent

Goal: We will import a pre-made “Coffee Shop” bot, test it, and then “re-train” it to sell Pizza instead. This shows how easily AI models can be adapted.

Tool: dialogflow.cloud.google.com (Dialogflow ES Console)

Step 1: Create an Agent

1. Open the link above and log in with your Google account.
2. Click Create Agent in the left sidebar.
3. Name: Pizza-Bot-Demo (No spaces allowed).
4. Default Time Zone: Select your time zone.
5. Click Create.



Notes



Notes

Step 2: Import a Prebuilt Agent (The “Coffee” Brain)

Instead of building from scratch, we will borrow a brain.

1. In the left menu, click Prebuilt Agents.
2. Scroll down to find Coffee Shop (or “Fast Food” / “Restaurant”).
3. Hover over it and click Import.
4. Click Ok to confirm. The system will now create all the Intents and Responses for a full coffee shop automatically!

Step 3: Test the Coffee Bot

1. Look at the “Try it now” box on the right side of the screen.
2. Type: *“I would like a latte.”*
3. Observe: The bot should reply with something like *“What size would you like?”*
4. Observe the Intent: Notice that under the user input, it says Intent: order_drink. The AI correctly understood you wanted a drink.

Step 4: The Transformation (Coffee → Pizza)

Now, let’s hack the bot. We want to sell Pizza, not Coffee.

1. Click Intents in the left menu.
2. Find the intent named order_drink (or similar) and click to open it.
3. Modify Training Phrases:
 - You will see phrases like “I want a cappuccino” or “Pour me a latte”.
 - Delete a few of these.
 - Add new ones: *“I want a pepperoni pizza”, “Give me a slice”, “I want a cheese pizza”*.
4. Modify Responses:
 - Scroll down to the Responses section.
 - You will see: *“What size drink would you like?”*
 - Change it to: *“Great choice! Large or Medium Pizza?”*
5. Click Save (Blue button at the top). Wait for the *“Agent Training started/completed”* notification.

**Step 5: Test the Pizza Bot**

1. Go back to the “Try it now” box.
2. Type: “*I want a cheese pizza.*”
3. Result: The bot should now reply, “*Great choice! Large or Medium Pizza?*”
4. Experiment: Type “*I want a latte*” again.
 - *What happens?* It might still work (if you didn’t delete all coffee examples), or it might trigger the Fallback Intent (“I didn’t get that”), because you retrained its brain to focus on pizza!

**INTEXT QUESTIONS 8.1**

1. In Google Teachable Machine, what do we call the “buckets” where we upload our example images (e.g., “Rock”, “Paper”, “Scissors”)?
 - (a) Neurons (b) Epochs (c) Classes (d) Intents
2. When you test your model in Teachable Machine, it gives you a percentage (e.g., “Rock: 95%”). What does this percentage represent?
 - (a) The battery life of the computer.
 - (b) The Confidence Score (how sure the model is of its guess).
 - (c) The amount of memory used.
 - (d) The number of images in the bucket.
3. In Dialogflow, what is an “Intent”?
 - (a) The password to log in.
 - (b) The user’s intention or goal (e.g., wanting to order food).
 - (c) The speed of the internet connection.
 - (d) The customized response text.
4. Why do we provide multiple “Training Phrases” for one Intent in Dialogflow?
 - (a) To confuse the bot.
 - (b) To teach the bot the many different ways people say the same thing.
 - (c) To fill up the hard drive.
 - (d) It is mandatory to have exactly 100 phrases.



Notes

No-Code and Low-Code AI Tools

5. If you type something the Dialogflow bot absolutely does not understand, which Intent is usually triggered?
 - (a) Default Welcome Intent
 - (b) Default Fallback Intent
 - (c) Coffee Intent
 - (d) System Error
6. What action turns a “No-Code” project into a “Low-Code” project?
 - (a) Using the visual interface to drag-and-drop blocks.
 - (b) Exporting the model as code (e.g., HTML/Python) and integrating it into a custom website.
 - (c) Turning off the computer.
 - (d) Creating a new account.



WHAT YOU HAVE LEARNT

- No-Code (GUI-based, drag-and-drop for non-programmers) and Low-Code (minimal coding required for customization) platforms and their impact on democratizing AI.
- The three-step cycle – Gather (Input/Classes), Train (Processing/EPOCHS), and Export (Output/Confidence Score) – using Google Teachable Machine.
- The architecture of conversational AI using Dialogflow, specifically the relationship between Agents (the bot), Intents (user goals), Training Phrases (user inputs), and Responses.
- Transition from a No-Code prototype to a Low-Code application by exporting models (e.g., Keras/TensorFlow) for use in environments like Google Colab.



TERMINAL QUESTIONS

1. Explain NO-CODE & LOW-CODE.
2. Explain the three-step cycle – Gather (Input/Classes), Train (Processing/EPOCHS), and Export (Output/Confidence Score) – using Google Teachable Machine.



ANSWERS TO INTEXT QUESTIONS



Notes

8.1

1. (c) Classes
2. (b) The Confidence Score
3. (b) The user's intention or goal.
4. (b) To teach the bot the many different ways people say the same thing
5. (b) Default Fallback Intent.
6. (b) Exporting the model as code

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- distinguish between No-Code and Low-Code platforms and their impact on democratizing AI.
- master the Machine Learning Workflow i.e. Apply the three-step cycle - Gather (Input/Classes), Train (Processing), and Export (Output/ Confidence Score) using Google Teachable Machine.
- explain the architecture of conversational AI using Dialogflow
- recognize how to transition from a No- Code prototype to a Low-Code application by exporting models (e.g., Keras/TensorFlow) for use in environments like Google Colab.



GENERATIVE AI AND OPEN AI TOOLS

In the previous lesson, you explored the fascinating world of Machine Learning. You learnt how to teach computers to recognize patterns using **Supervised Learning** and how to build simple models using tools like **Teachable Machine**. You taught the computer to answer questions like, “Is this a photo of an apple or a banana?” or “Is this email spam or safe?” In those cases, the AI was analyzing existing data to make a decision.

But what if the computer didn’t just analyze the apple – what if it could paint a picture of a new fruit that doesn’t exist yet? Or write a poem about it?

This is the world of **Generative AI**. Unlike the regression or classification models you built previously, Generative AI doesn’t just sort information; it creates new content. It can write stories, generate computer code, draw complex images, and even compose music. It does this by learning from billions of examples and predicting what should come next.

In this lesson, we will move from being “AI Trainers” to “AI Creators.” We will explore powerful tools like **ChatGPT**, **DALL·E**, and **other Open Source Models**. However, with great power comes great responsibility. We will also learn the art of talking to these machines (Prompt Crafting) and the ethical rules we must follow to use them safely.



OBJECTIVES

After reading this lesson, you will be able to:

- distinguish between **Discriminative AI** (Analysis) and **Generative AI** (Creation);

- explain how **Large Language Models (LLMs)** predict text;
- utilize **ChatGPT** and **Microsoft Copilot** for creative and professional tasks;
- navigate **Hugging Face** to find and test open-source models;
- apply **Prompt Engineering** techniques to control AI outputs;
- create original visuals using **Text-to-Image** models responsibly.



9.1 DISCRIMINATIVE VS. GENERATIVE AI

To understand Generative AI, we first need to see how it differs from the AI you have used before.

Discriminative AI: The Judge Imagine an art competition. **Discriminative AI** acts like the **judge**. It looks at a painting and decides, “This is a landscape,” or “This is a portrait.” It doesn’t create the art; it only classifies or labels what it sees. Its main goal is to separate data into categories, like deciding if an email is “Spam” or “Not Spam”.

Generative AI: The Artist **Generative AI**, on the other hand, acts like the **artist**. It has studied millions of paintings to understand what art looks like. When you ask it to “paint a sunset,” it creates a brand-new image that has never existed before. It doesn’t just choose from a list of options; it generates new data.

Feature	Discriminative AI (The Judge)	Generative AI (The Artist)
Goal	To classify or predict labels.	To create new data.
Example	Unlocking your phone with Face ID.	Writing a new Harry Potter story.
Output	A label (e.g., “Cat”, “Dog”).	A new creation (e.g., Image, Text).

9.2 HOW LARGE LANGUAGE MODELS (LLMS) PREDICT TEXT

Tools like ChatGPT are built on technology called **Large Language Models (LLMs)**. But how does a computer actually “write” an essay or a poem? It isn’t thinking or being creative in the way humans are. Instead, it is playing a very complex game of “**Fill in the Blank.**”

Imagine you have a friend who has read every book, website, and article in the world. If you started a sentence with “*Once upon a...*”, that friend would immediately know the next word is likely “*time.*” This is exactly what an LLM does, but with billions of words.



Notes

The Prediction Engine The core function of an LLM is **Next-Token Prediction**. It breaks text down into small chunks called “tokens” (which can be words or parts of words). When you type a prompt, the model looks at your words and calculates the mathematical **probability** of what should come next based on everything it has ever read.

How it works step-by-step:

1. **Input:** You type “*The students opened their...*”
2. **Calculation:** The AI looks at its training data. It knows that “*books*” is 85% likely, “*laptops*” is 10% likely, and “*umbrellas*” is 5% likely.
3. **Selection:** It picks the most probable word: “*books*”.
4. **The Loop:** Now the sentence is “*The students opened their books*”. The AI reads this *new* sentence and predicts the *next* word (e.g., “*to*”).

It repeats this process thousands of times per second. It doesn’t just look at the last word; it looks at the entire context of your conversation. If you were talking about rain earlier, it might have chosen “*umbrellas*” instead of “*books*”. This ability to remember context and predict patterns is what makes it seem like it is “understanding” you, even though it is just doing advanced math!

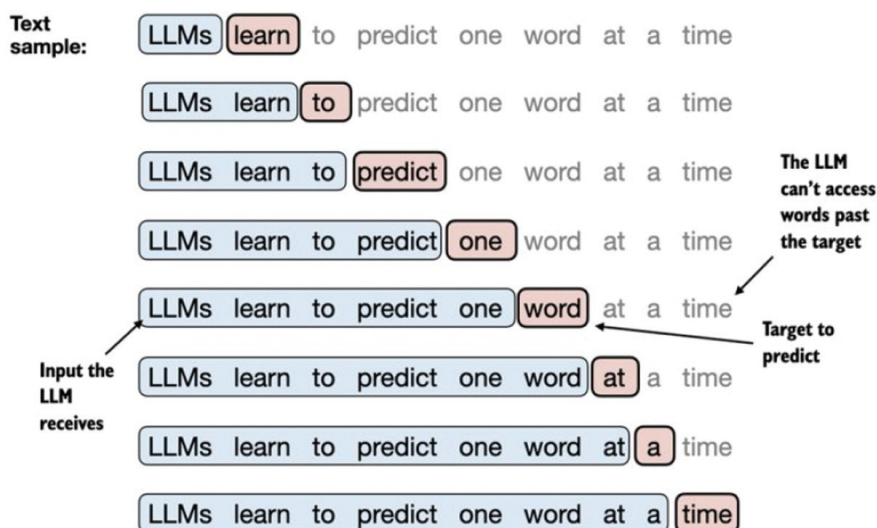


Fig 9.1: Next Token Prediction in LLM

Generative AI tools are like having a smart, creative assistant available 24/7. Whether you are working on a school assignment or pursuing a personal hobby,



these tools can help you work faster and think bigger. Instead of doing the work *for* you, think of AI as a partner that helps you get started.

Here are four powerful ways you can use these tools in your daily life:

- **Smart Summarization (The Study Buddy)** Sometimes school chapters or articles can be long and difficult to understand. You can paste a complex paragraph into an AI tool and ask: “*Explain this to me like I am 10 years old*” or “*Summarize this history chapter into 5 simple bullet points.*” This helps you understand the core concepts quickly before you dive into the details.

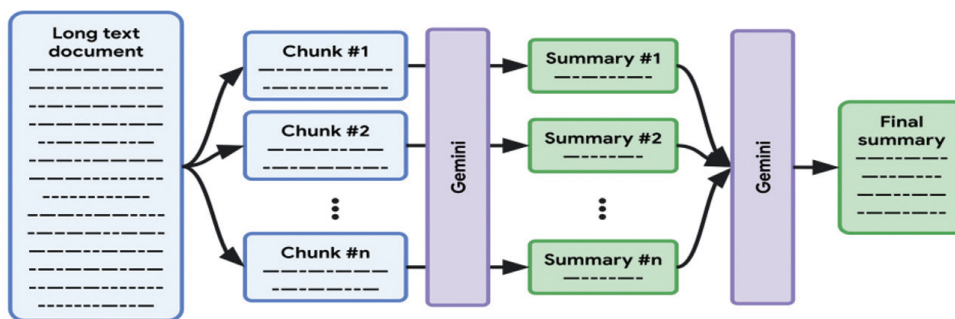


Fig 9.2: Smart Summarization

- **Drafting and Writing (The Editor)** Have you ever stared at a blank screen, not knowing how to start an email or an essay? This is called “Writer’s Block.” You can ask AI to unblock you. For example: “*Draft a polite email to my teacher asking for extra time on my assignment.*” You can also paste your own rough draft and ask the AI to check your grammar or suggest better vocabulary. (See Fig. 9.3)
- **Brainstorming Ideas (The Spark)** When you are stuck on a creative project, AI acts like a brainstorming partner. If you have a science fair coming up, you can ask: “*Give me 10 unique ideas for a science project using magnets and water.*” If you are writing a story, you can ask: “*What are three plot twists for a mystery story set on a train?*” It gives you options, and you choose the best one.
- **Language Learning (The Tutor)** Learning a new language (like French, Spanish, or even Python code) is easier with practice. AI can act as a conversation partner that never gets tired. You can ask it to quiz you on vocabulary, translate difficult phrases, or role-play a conversation in a new language to improve your fluency.



Notes

Generative AI and Open AI Tools

The screenshot shows a web-based AI writing tool interface. On the left, there are three dropdown menus: 'Language' set to 'English', 'Tone' set to 'Convincing', and 'Use Case' set to 'Poetry'. Below these is a text box labeled 'Idea' containing the text 'Poetry about stars and planets'. At the bottom left is a blue button labeled 'Generate article'. On the right, a large text area displays the generated poem:

Stars and planets circling
In the universe
Piercing the darkness
Losing the light of day.

You are my morning star
Blessing my soul in the dark
Radiating a love that gives me strength.

Will you keep on shining down on me?
Fill me up with your light?
So I can shine for you.

You are my darkest hour
Giving me something to hold onto
The shadow that disappears when you're around

You're the brightest star I see gleaming in the sky. You were the first thing to
come into focus when my eyes opened You are the brightest star and I'll love
you forever.

Fig 9.3: Drafting and Writing

9.3 PROMPT ENGINEERING: THE ART OF COMMUNICATING WITH AI

The text or instruction you type into an AI tool is called a **Prompt**. If you ask a vague question, the AI will give you a vague answer. However, if you give specific, clear instructions, the AI can produce amazing results. The skill of crafting these effective instructions is known as **Prompt Engineering**.

Think of the AI as a brilliant actor. If you just tell the actor to “act,” they won’t know what to do. You need to give them a script, a character, and a scene. Similarly, to get the best out of AI, you need to provide structure.

The Components of a Perfect Prompt To communicate effectively with an AI, your prompt should include three key elements, often remembered as the **RTC Framework**:

- 1. Role (Who):** Tell the AI who it should pretend to be. This helps the AI set the right tone.
 - *Example:* “Act as a strict English Teacher” or “Act as a helpful Tour Guide.”
- 2. Task (What):** Clearly state what you want the AI to do. Use active verbs like *Write*, *List*, *Explain*, or *Summarize*.
 - *Example:* “Write a short story about space travel.”



3. **Context (For Whom):** Explain who the audience is or give specific details.

- *Example:* “The story should be funny and written for children aged 10.”

Analyzing Prompts: Weak vs. Strong

Let us look at an example of how adding these elements changes the result.

- **Weak Prompt:** “Write about pollution.”
 - *Why it fails:* This is too general. The AI doesn’t know if you want a scientific definition, a poem, or a solution to pollution. It will likely give a boring, generic dictionary definition.
- **Strong Prompt:** “Act as an Environment Activist (**Role**). Write a persuasive speech about the dangers of plastic pollution (**Task**) to be delivered at a school assembly (**Context**).”
 - *Why it works:* Now the AI knows exactly what to do. It will use emotional language suitable for a speech and focus specifically on plastic pollution for students.

Topic	Weak Prompt (What to avoid)	Strong Prompt (What to use)
Science	“Explain gravity.”	“Act as a Physics Teacher (Role) . Explain the concept of gravity (Task) to a class of 5th-grade students using the analogy of a trampoline (Context).”
History	“Who was Ashoka?”	“Act as a Historian (Role) . Write a short biography of Emperor Ashoka focusing on his conversion to Buddhism (Task). Keep it under 100 words for a school magazine (Context).”
Coding	“Write python code for a calculator.”	“Act as a Coding Tutor (Role) . Write a simple Python script for a calculator that adds two numbers (Task). Add comments to every line explaining what the code does for a beginner (Context).”
English	“Write a letter.”	“Act as a School Captain (Role) . Write a formal application to the Principal requesting permission to organize a tree plantation drive (Task). The tone should be polite and persuasive (Context).”

Try providing both the prompts to any Large Language Model and you’ll notice a clear difference in the answers.



Notes

9.4 ETHICS AND RESPONSIBILITY IN GENERATIVE AI

With Large Language Model tools like ChatGPT and DALL·E, you have the power to create almost anything. But just because you *can* create something, does it mean you *should*? As an AI user, you must understand the risks involved.

1. **Bias:** Generative AI models are trained on vast amounts of data from the internet. If that data contains unfair opinions or stereotypes, the AI will learn them. This is known as **Bias**.
 - **Real-Life Example:** A famous company once built an AI tool to review job resumes. Because it was trained on resumes from the previous 10 years (which came mostly from men), the AI learnt to downgrade resumes from women.
 - **The Lesson:** AI is not neutral. It reflects the data it was fed. Always check AI-generated content for unfair stereotypes before using it.
2. **Hallucinations and Errors** AI models can sometimes be confidently wrong. They might invent historical facts or citations.
3. **Deepfakes and Misinformation** Generative AI can create realistic images and voices of people saying things they never said (Deepfakes). Using these tools to spread rumors or fake news is unethical and dangerous.

Watermarks and Signatures: To fight misinformation, responsible AI companies now add digital watermarks or cryptographic signatures to the content they generate. These are often invisible codes embedded in an image or text file that allow detection tools to verify that the content was created by an AI, not a human. The next time you look at an image, do have all of this in mind!

9.5 EVALUATING IMPACT: THE FUTURES WHEEL

Before you create or deploy an AI solution, you need to predict its consequences. A useful tool for this is the **Futures Wheel**.

1. The futures wheel is used to identify first order (in red) and second order (in green) effects as shown in Fig. 9.4.
2. We fill up the red circles with direct effects of our AI solution.
3. We fill up the green circles with indirect effects that are a result of the direct effects.

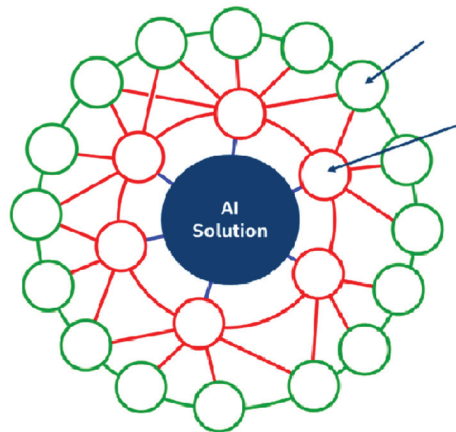


Fig. 9.4: Futures Wheel

It helps you see beyond the immediate result.

- **Center:** Write down the AI Action (e.g., “Students use ChatGPT to do all their homework”).
- **First Layer (Direct Effects):** What happens immediately? (e.g., “Students get homework done faster,” “Grades might go up initially”).
- **Second Layer (Indirect Effects):** What happens because of the first layer? (e.g., “Students stop learning basic writing skills,” “Exams become much harder to catch cheating”).

Using the Futures Wheel helps you decide if an AI use case is actually “good” for everyone in the long run.



INTEXT QUESTIONS 9.1

1. While Discriminative AI acts like a judge to classify data, Generative AI acts like an to create new data.
2. Large Language Models (LLMs) like ChatGPT generate text by:
 - (a) Thinking like a human brain
 - (b) Copy-pasting answers from a database
 - (c) Predicting the most probable next word based on patterns
 - (d) Searching for the best website link



Notes

Generative AI and Open AI Tools

3. In the **RTC Framework** for prompt engineering, what do the letters **R**, **T**, and **C** stand for?
4. **True or False:** If an AI model confidently states a historical fact, it is guaranteed to be true because computers cannot lie.
5. Which tool is used to map out the *direct* and *indirect* consequences of deploying an AI solution?
 - (a) The RTC Framework
 - (b) The Futures Wheel
 - (c) Teachable Machine
 - (d) Digital Watermarking



WHAT YOU HAVE LEARNT

- **Differentiate AI Types:** Distinguish between **Discriminative AI** (which classifies existing data like a judge) and **Generative AI** (which creates new content like an artist).
- **Explain LLM Mechanics:** Describe how **Large Language Models (LLMs)** function as prediction engines using probability to generate text, rather than “thinking” like humans.
- **Master Prompt Engineering:** Apply the **Role-Task-Context (RTC)** framework to craft clear, effective prompts that control AI outputs for creative and professional tasks.
- **Evaluate Ethical Risks:** Identify critical ethical pitfalls in AI, including **data bias**, **hallucinations**, and **deepfakes**.
- **Assess Impact:** Utilize the **Futures Wheel** method to map out the direct and indirect consequences of deploying AI solutions in society.
- **Practice Responsibility:** Adopt a “Human-in-the-Loop” mindset to verify AI-generated content and recognize the importance of **digital watermarks** for transparency.



ANSWERS TO INTEXT QUESTIONS

9.1

1. Artist (Generative AI creates, while Discriminative AI judges/classifies).

2. (c) Predicting the most probable next word based on patterns (LLMs function as prediction engines, not thinking brains).
3. Role, Task, Context (These are the three pillars of effective Prompt Engineering).
4. False (AI models can “hallucinate” and confidently present incorrect information as facts).
5. (b) The Futures Wheel (This tool helps visualize the ripple effects of technology).



Notes

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- Distinguish between Discriminative AI and Generative AI
- Explain Large Language Models (LLMs)
- Utilize the Futures Wheel method to map out the direct and indirect consequences of deploying AI solutions in society.
- Identify critical ethical pitfalls in AI, including data bias, hallucinations, and deepfakes.



AI SYSTEMS AND FRAMEWORK

Have you ever wondered how a single developer can build a complex application like a voice assistant or a recommendation system in just a few days? Or how researchers can quickly test new ideas without writing thousands of lines of complex mathematical code from scratch? The secret lies in powerful tools known as AI Frameworks and Libraries.

Imagine you want to build a house. You could make every single brick, mix the cement yourself, and cut every plank of wood from a tree. This would take years! Alternatively, you could buy pre-made bricks, ready-mix cement, and pre-cut wood, allowing you to focus on the design and structure of the house. AI Frameworks are like those pre-made building materials. They provide pre-written code and tools that handle the difficult, repetitive mathematical calculations required for Artificial Intelligence, allowing developers to focus on solving problems and building smart applications.

In this lesson, we will explore the AI Project Cycle to understand how an idea becomes a reality. We will look at the tools – like TensorFlow, Keras, and Scikit-learn – that make this process easier and faster. Finally, we will see these tools in action by running a “pre-trained” model, which is like using a robot that has already been taught how to perform a task!



OBJECTIVES

After reading this lesson, you will be able to:

- explain the stages of the AI Project Cycle, from Problem Scoping to Deployment;

- define AI Frameworks and Libraries and explain why they are essential for efficient development;
- differentiate between key libraries like Scikit-learn (for classical machine learning), TensorFlow, and Keras (for deep learning);
- understand the concept of a Pre-trained Model and its advantages in saving time and computational resources;
- demonstrate how to load and run a simple pre-trained model using a cloud-based environment like Google Colab.



Notes

10.1 THE AI PROJECT CYCLE

Just like building a house or cooking a complex meal, creating an AI model follows a specific life cycle. We cannot simply “code AI” without first understanding what we are trying to solve. The **AI Project Cycle** provides a roadmap for developers to ensure they build useful and accurate models. It typically consists of five key stages as shown in fig 10.1

1. **Problem Scoping:** This is the planning phase. Before touching a computer, we define the goal. We ask questions like: *What is the problem? Who is it for? What value will this AI solution bring?* For example, are we trying to detect burglars or just count the number of people entering a shop?
2. **Data Acquisition:** AI needs data to learn. In this stage, we collect accurate and reliable data from various sources. This could be images from a camera, text from documents, or sensor readings. This is like gathering high-quality ingredients for a recipe.
3. **Data Exploration:** Once we have the data, we must inspect it to understand its nature. We look for patterns, trends, or missing pieces (like “cleaning” the data). We often use charts and graphs here to visualize the data before the machine sees it.
4. **Modeling:** This is where the magic happens. We select an AI algorithm (mathematical rules) and “train” it using our data. The computer analyzes the data to find patterns and builds a “Model” that can make decisions.
5. **Evaluation & Deployment:** Finally, we test the model on new data it has never seen before to check its accuracy (Evaluation). If it works well, we deploy it into the real world, like installing it onto a security camera or a mobile app.



Notes

AI Systems and Framework

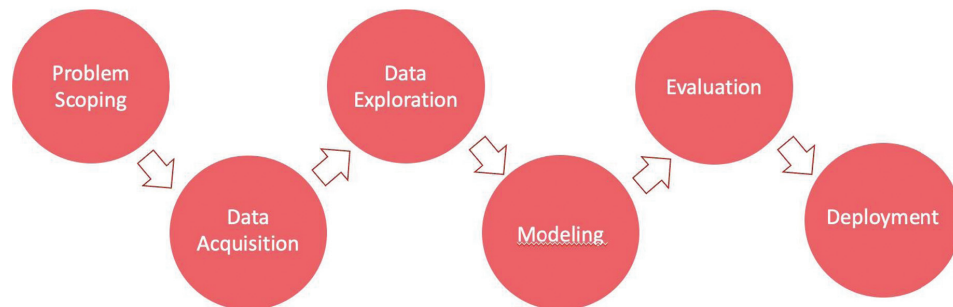


Fig. 10.1: AI Project Cycle

10.2 UNDERSTANDING AI FRAMEWORKS AND LIBRARIES

To follow the AI Project Cycle efficiently, developers rely on Frameworks and Libraries. (See Fig. 10.2)

- **Libraries:** A library is a collection of pre-written code routines that you can use to perform specific tasks. Think of it as a “toolbelt” containing specific tools like a hammer or a screwdriver. You pick the specific tool (function) you need for a specific job.
- **Frameworks:** A framework is a larger, more comprehensive platform that provides a foundation for developing applications. It is like the “skeleton” of a house. It dictates the structure, and you build your specific rooms (features) on top of it.

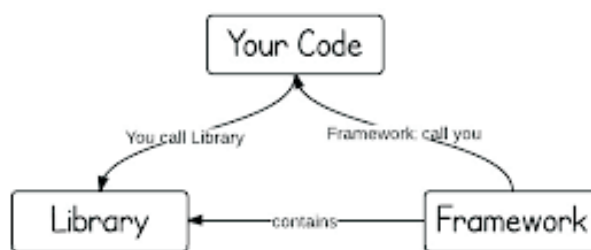


Fig. 10.2: AI Frameworks and Libraries

10.3 THE “EASY WAY” VS. THE “HARD WAY”

Why are these tools so important? Let’s look at an example of building a Neural Network (a complex AI model) to recognize handwritten numbers.

- **The Hard Way (Coding from Scratch):** You would have to write thousands of lines of mathematical code. You would need to manually program complex



calculus, matrix multiplications, and backpropagation algorithms. It would be like trying to build a car by mining the metal and forging the engine parts yourself.

- **The Easy Way (Using a Framework):** You use a framework like TensorFlow or Keras. Instead of writing the math yourself, you simply use a pre-made “layer”. The framework handles all the complex math in the background. You can build the same powerful AI model in just 10-15 lines of code instead of 1,000.

10.4 KEY AI TOOLS AND FRAMEWORKS

There are three main tools you will often hear about in the world of Python and AI. Each serves a different purpose:

10.4.1 Scikit-learn

- **What it is:** A simple, efficient library for classical Machine Learning.
- **Best for:** Beginners and projects involving tables of data (like Excel sheets). It is excellent for tasks like **Classification** (e.g., Is this email Spam or Not Spam?) or **Regression** (e.g., Predicting house prices based on size).
- **Key Feature:** It includes many built-in datasets and simple algorithms like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN).

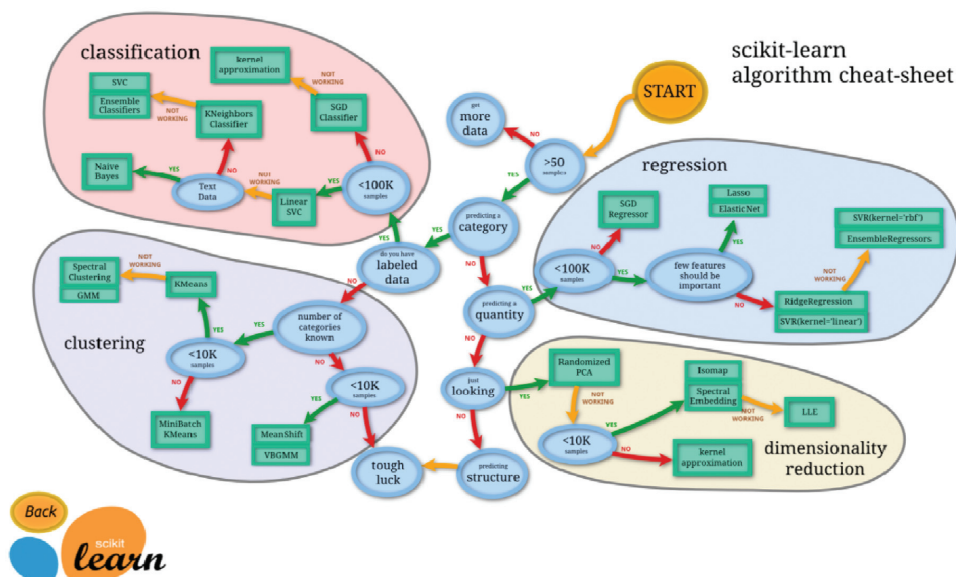


Fig. 10.3: Scikit-Learn Algorithm



Notes

This diagram is a handy decision-making flowchart designed by scikit-learn to help you choose the right machine learning algorithm for your specific dataset. By following the path based on your data's size, whether it has labels, and your predictive goals, it guides you to the most appropriate classification, regression, clustering, or dimensionality reduction model.

10.4.2 TensorFlow

- **What it is:** A powerful, open-source framework created by Google. It is designed for “Deep Learning” and Neural Networks.
- **Best for:** Complex projects that require analyzing massive amounts of unstructured data, such as millions of images or hours of audio. It is widely used in industry for things like self-driving cars and large-scale image recognition.
- **Key Feature:** It allows for “Tensor” processing (multidimensional arrays of data) and runs efficiently on specialized hardware like GPUs.

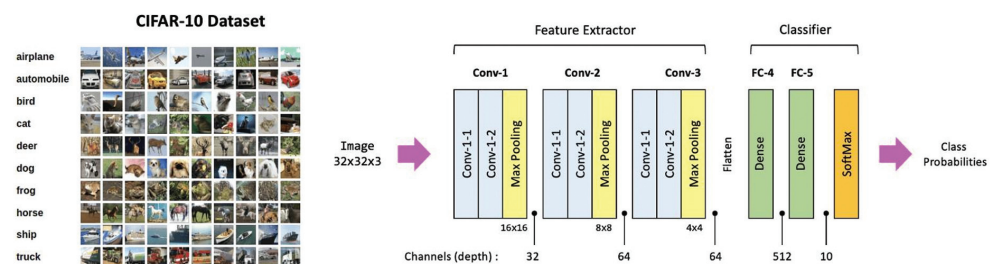


Fig. 10.4: CIFAR – 10 Dataset

This diagram illustrates the architecture of a Convolutional Neural Network (CNN) designed to classify images from the CIFAR-10 dataset. It shows how a raw input image is processed through a “Feature Extractor” (using convolutional and pooling layers to find visual patterns) and then fed into a “Classifier” (dense layers) to predict which of the 10 categories the image belongs to.

10.4.3 Keras

- **What it is:** A high-level API that runs *on top* of TensorFlow. It was designed specifically to be easy for humans to understand and write.
- **Best for:** Fast prototyping and education. It allows you to build deep learning models quickly using simple, readable code blocks like Dense, Conv2D, or Flatten.

- **Key Feature:** It stacks layers of neurons like LEGO blocks. You can snap powerful pieces together to build a Convolutional Neural Network (CNN) without needing to know the complex mathematics behind every connection.



Notes

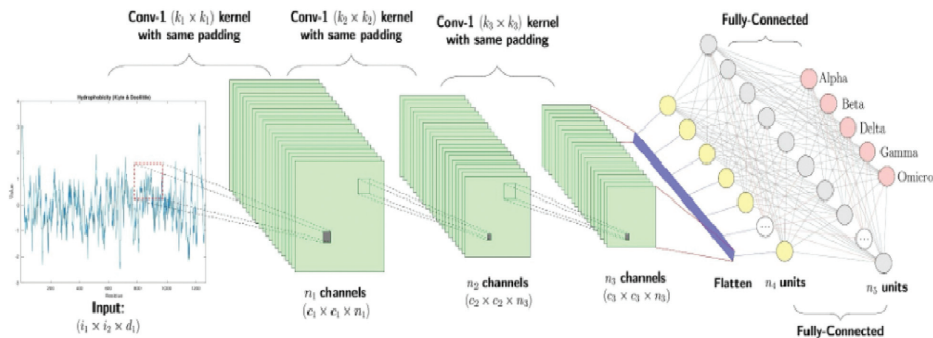


Fig. 10.5: Convolutional Neural Network

This diagram illustrates a Convolutional Neural Network (CNN) that processes biological sequence data – specifically a protein’s hydrophobicity plot – through a series of convolutional layers to extract important patterns. These extracted features are then flattened and fed into a fully-connected classifier to predict which specific viral variant (like Alpha, Delta, or Omicron) the sequence belongs to.

10.5 THE POWER OF PRE-TRAINED MODELS

One of the greatest advantages of modern AI frameworks is the ability to use Pre-trained Models.

Imagine if you wanted to learn how to play the piano. It would be very difficult if you first had to learn how to control your fingers, how to hear sound, and how to read. Fortunately, you already possess those basic skills (features), so you only need to learn the specific skill of playing keys.

Transfer Learning works the same way. Researchers at big companies (like Google or Microsoft) train huge models on millions of images for weeks using supercomputers. These models learn to recognize basic “features” like edges, curves, eyes, and shapes.

- **Pre-trained Model:** A model that has already been trained on a large dataset and saved.
- **Advantage:** Instead of training a model from scratch (which takes a lot of time and data), we can download a pre-trained model and “fine-tune” it for our specific task. For example, a model trained to recognize “animals” can be easily tweaked to specifically recognize “types of bears” with very little new data.



Notes

10.5.1 Popular Pre-trained Models and Their Uses

Instead of building a brain from scratch, developers often “download” a brain that is already good at a specific task. Here are three famous models you might use:

1. MobileNet (The “Lightweight” Vision Model)

- **What it is:** A small, fast computer vision model designed to run on mobile phones and web browsers.
- **Best For:** Real-time apps like a “Dog Breed Identifier” on your phone or a “Face Filter” on Instagram. It trades a tiny bit of accuracy for incredible speed.

2. VGG-16 (The “Standard” Vision Model)

- **What it is:** A powerful, highly accurate model for image classification. It is much larger than MobileNet.
- **Best For:** Projects where accuracy is more important than speed, such as analyzing medical X-rays or sorting high-quality photos.

3. BERT (The “Language” Master)

- **What it is:** A model developed by Google that understands the *context* of words in a sentence, not just the words themselves.
- **Best For:** Natural Language Processing (NLP) tasks like summarizing a long article, translating languages, or determining if a movie review is positive or negative (Sentiment Analysis).

10.6 TOOL SPOTLIGHT: HUGGING FACE TRANSFORMERS

If frameworks like TensorFlow are the “factory,” then **Hugging Face** is the “App Store” for AI models.

What is it? Hugging Face Transformers is an open-source library that gives you instant access to thousands of pre-trained models (like BERT and GPT).

Why is it great for students?

- **Simplicity:** You don’t need to understand the complex math behind a “Transformer” neural network.
- **Ease of Use:** With just 3 lines of code, you can download a state-of-the-art AI model that understands English, French, or even Python code.
- **Community:** It is a hub where researchers share their models for free, allowing you to use the same technology as top scientists.



10.6.1 Industry Relevance of These Tools

- **Rapid Prototyping:** Companies can bring ideas to market in days rather than months by building on top of established frameworks instead of coding complex math from scratch.
- **Cost Efficiency:** Using pre-trained models saves millions of dollars in computing power and server costs that would otherwise be spent training massive networks from the ground up.
- **Scalability:** Frameworks like TensorFlow are designed to run anywhere, from massive cloud server farms to small edge devices like smartphones and Raspberry Pis.
- **Standardization:** Using common libraries ensures that teams of developers across the globe can collaborate efficiently using the same universal coding standards.

10.6.2 Case Study: Building a Plant Disease App

Scenario: Riya wants to help local farmers by building a smartphone app that detects crop diseases simply by taking a photo of a leaf.

Here is how she uses the AI Project Cycle and tools to build it quickly:

- **TensorFlow !** She uses it as the powerful underlying engine to process the massive matrices of image data.
- **Pre-trained Model !** Instead of teaching the AI what a generic leaf looks like from scratch, she downloads a pre-trained **MobileNet** model. This saves weeks of training time and ensures the model is lightweight enough to run smoothly on a farmer's phone.
- **Keras !** She uses this simple, Lego-like interface to add a few custom layers to the top of MobileNet, fine-tuning it to specifically recognize the difference between “healthy” and “diseased” plant leaves.



INTEXT QUESTIONS 10.1

Multiple-Choice Questions

1. A data scientist is working on a complex Computer Vision problem involving millions of high-resolution images. They need a framework that offers full



Notes

control over the mathematical operations of the neural network and runs efficiently on GPUs. Which tool is the most appropriate choice?

- (a) Scikit-learn
- (b) TensorFlow
- (c) Pandas
- (d) Matplotlib

2. Keras is often described as a “high-level API.” What does this mean in the context of building AI models?

- (a) It runs only on supercomputers located at high altitudes.
- (b) It focuses on visual data processing only.
- (c) It abstracts away complex mathematical details, allowing developers to build models using simple, human-readable commands.
- (d) It provides higher accuracy than any other framework automatically.

3. You are designing a smart-glasses application that helps visually impaired users identify objects in real-time. The device has very limited battery life and memory. Which pre-trained model architecture would be the optimal choice?

- (a) VGG-16 (High accuracy, large size)
- (b) BERT (Language understanding)
- (c) MobileNet (Optimized for speed and efficiency)
- (d) Linear Regression (Statistical prediction)

4. When performing “Transfer Learning” with a pre-trained model on a new, smaller dataset, what is the standard procedure for the initial layers of the network?

- (a) They are usually “frozen” (kept unchanged) because they have already learned to detect universal features like edges and curves.
- (b) They are deleted to save space.
- (c) They are randomized to ensure the model learns from scratch.
- (d) They are replaced with Scikit-learn algorithms.

5. Which of the following scenarios describes the best use-case for the Scikit-learn library?

- (a) Training a Deep Convolutional Neural Network to generate deepfake videos.



- (b) Predicting customer churn (Yes/No) based on a structured spreadsheet of customer transaction history.
 - (c) Translating live audio from French to English using a Transformer model.
 - (d) Detecting specific bounding boxes around pedestrians in a self-driving car video feed.
6. The “Hugging Face” library has revolutionized Natural Language Processing (NLP) by providing easy access to which type of architecture?
- (a) Decision Trees
 - (b) Transformers (like BERT and GPT)
 - (c) Support Vector Machines (SVM)
 - (d) Random Forests
7. In a Convolutional Neural Network (CNN) built with Keras/TensorFlow, what is the primary function of a “Pooling Layer” (often seen in pre-trained models like VGG-16)?
- (a) To colorize the image.
 - (b) To increase the size of the image to see more detail.
 - (c) To reduce the spatial dimensions (size) of the image while retaining the most important features, effectively summarizing the data.
 - (d) To classify the final output as a “Cat” or “Dog”.



WHAT YOU HAVE LEARNT

- **The AI Project Cycle is Essential:** Building an AI model requires a structured five-stage roadmap - Problem Scoping, Data Acquisition, Data Exploration, Modeling, Evaluation & Deployment to ensure the final solution is accurate and useful.
- **Tools Simplify Development:** AI Libraries and Frameworks save developers immense time by handling the complex, repetitive mathematical calculations behind the scenes.
- **Tool Selection Depends on the Data:** Scikit-learn is ideal for classical machine learning and structured tabular data, whereas TensorFlow and Keras are powerful frameworks designed for Deep Learning and unstructured data like images or audio.



Notes

AI Systems and Framework

- **Pre-trained Models Provide a Head Start:** Models like MobileNet, VGG-16, and BERT are existing AI “brains” that have already learned basic features from massive datasets, saving significant time and computing power.
- **Transfer Learning is Efficient:** Instead of training a model from scratch, developers can download a pre-trained model and fine-tune it for a specific, related task using a much smaller dataset.
- **Centralized Hubs Democratize AI:** Platforms like Hugging Face act as an “App Store” for AI, providing developers and students instant access to thousands of state-of-the-art models with just a few lines of code.



TERMINAL QUESTIONS

Short Answer Questions

1. List the five stages of the AI Project Cycle in order.
2. Which Python library is best suited for tabular data (like Excel sheets) and classical machine learning algorithms?
3. What is the primary advantage of using a high-level API like Keras over writing deep learning code from scratch?
4. What type of tasks is the pre-trained BERT model primarily used for?

Long Answer Questions

1. Explain the primary difference between a “Library” and a “Framework” in the context of AI development. Provide an analogy to support your answer.
2. Describe the concept of “Transfer Learning.” Why is this technique particularly beneficial for individual developers or small startups with limited computational resources?
3. When performing Transfer Learning on an image classification model, why are the initial layers of the pre-trained network typically “frozen”?



ANSWERS TO INTEXT QUESTIONS

1. (b) TensorFlow

2. (c) It abstracts away complex mathematical details, allowing developers to build models using simple, human-readable commands.
3. (c) MobileNet (Optimized for speed and efficiency)
4. (a) They are usually “frozen” (kept unchanged) because they have already learned to detect universal features like edges and curves.
5. (b) Predicting customer churn (Yes/No) based on a structured spreadsheet of customer transaction history.
6. (b) Transformers (like BERT and GPT)
7. (c) To reduce the spatial dimensions (size) of the image while retaining the most important features, effectively summarizing the data.



Notes

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- define the AI Project Cycle and the five key stages: Problem Scoping, Data Acquisition, Data Exploration, Modeling, and Evaluation & Deployment.
- differentiate between Libraries and Frameworks
- distinguish between Scikit-learn (best for classical Machine Learning and tabular data) and TensorFlow/Keras (best for Deep Learning and unstructured data like images).



11

NATURAL LANGUAGE PROCESSING

Have you ever talked to Siri or Alexa, used Google Translate, searched something on the internet or chatted with a customer-support bot? If yes, you've already interacted with Natural Language Processing (NLP).

Why does NLP matter?

Language is one of the most natural ways humans communicate. If computers can understand language, they can:

- help us find information faster
- translate across cultures
- assist people with disabilities
- make digital experiences more human

From WhatsApp spam filters to ChatGPT-style chatbots, NLP powers countless tools we use every day.

In this Lesson you will be introduced to Natural Language Processing (NLP) and its importance. You will receive an overview of the various stages of NLP and test processing techniques used in NLP. You will undertake activities to appreciate the distinction between the Code vs No-code NLP. Henceforth also understand the Bag of Words algorithm and the concept of TF-IDF (Term Frequency-Inverse Document Frequency).



OBJECTIVES

After learning this lesson, the learner will be able to:

- describe NLP, its importance and various applications of NLP;



- explain different stages of NLP and various text processing techniques used in NLP;
- use No-Code NLP tools and the Bag of Words model;
- explain concept of TFIDF and No code Orange Data Mining Tool.

11.1 WHAT IS NATURAL LANGUAGE PROCESSING (NLP)?

NLP is a branch of Artificial Intelligence that teaches computers how to **understand, interpret, and generate human language**. It is the reason machines can read emails, answer questions, summarize long articles, and even write stories or code.

Features of Natural Languages

- They are governed by set rules that include syntax, lexicon, and semantics.
- All natural languages are redundant, i.e., the information can be conveyed in multiple ways.
- All natural languages change over time.

Test Yourself:

Choose the right word:

1. I am so tired; I want to take a?

break

brake

2. Let's her a letter.

right

write

Do you see how same-sounding words can have totally different meanings?

- **Different meanings in different contexts.**

Let's consider these sentences:

His face turned red after he found out that he took the wrong bag.

What does this mean? Is he feeling ashamed because he took another person's bag instead of his? Is he feeling angry because he did not manage to steal the bag that he has been targeting?

The red car zoomed past his nose.

Probably talking about the colour of the car

His face turns red after consuming the medicine.



Notes

Natural Language Processing

Is he having an allergic reaction? Or was he ashamed because he lost a bet (“I will not fall sick because of this”)? Or was he taking a medicine that dilates the artery?

Here we can see that context is important. We understand a sentence almost intuitively, depending on our history of using the language and the memories that have been built within. In all three sentences, the word red has been used in three different ways which according to the context of the statement changes its meaning completely. Thus, in natural language, it is important to understand that a word can have multiple meanings and the meanings fit into the statement according to the context of it.

Think of some other words which can have multiple meanings and use them in sentences.

Computer Language

Computer languages are languages used to interact with a computer, such as Python, C++, Java, HTML, etc.

Can computers understand our language?

Computers require a specific set of instructions to understand human input called programs. To talk to a computer, we convert natural language into a language that a computer understands. We need Natural Language Processing to help computers understand natural language.



Fig. 11.1

Why is NLP important?

Computers can only process electronic signals in the form of binary language. Natural Language Processing facilitates this conversion to digital form from the natural form. Thus, the whole purpose of NLP is to make communication between computer systems and humans possible. This includes creating different tools and techniques that facilitate better communication of intent and context.



Fig. 11.2



Demystify Natural Language Processing (NLP)

Natural Language Processing (NLP), is the sub-field of AI that is focused on enabling computers to analyse, understand and process human languages to derive meaningful information from human language.

11.2 APPLICATIONS OF NATURAL LANGUAGE PROCESSING

Since Artificial Intelligence now a days is becoming an integral part of our lives, its applications are very commonly used by the majority of people in their daily lives. Some of the applications of Natural Language Processing which are used in the real-life scenario are:

1. **Voice assistants:** Voice assistants take our natural speech, process it, and give us an output. These assistants leverage NLP to understand natural language and execute tasks efficiently.

For example:

Hey Google, set an alarm at 3.30 pm Hey Alexa, play some music Hey Siri, what's the weather today.



Fig. 11.3

2. **Autogenerated captions:** Captions are generated by turning natural speech into text in real-time. It is a valuable feature for enhancing the accessibility of video content.

For example: Auto-generated captions on YouTube and Google Meet.

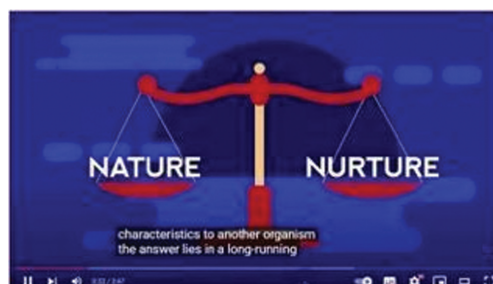


Fig. 11.4



Notes

3. **Language Translation:** It incorporates the generation of translation from another language. This involves the conversion of text or speech from one language to another, facilitating cross-linguistic communication and fostering global connectivity.

For example: Google Translate

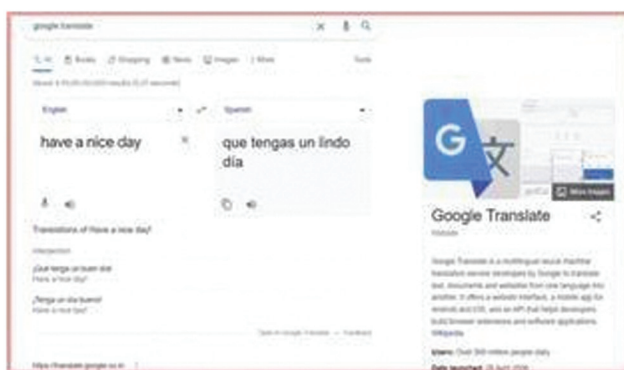


Fig. 11.5

4. **Sentiment Analysis:** Sentiment Analysis is a tool to express an opinion, whether the underlying sentiment is positive, negative, or neutral. Customer sentiment analysis helps in the automatic detection of emotions when customers interact with the products, services, or brand.



Fig. 11.6

5. **Text Classification:** Text classification is a tool which classifies a sentence or document category-wise. In the example, we can observe news articles containing information on various sectors, including Food, Sports, and Politics, being categorized through the text classification process. This process classifies the raw texts into predefined groups or categories.

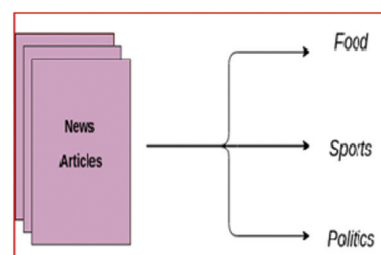


Fig. 11.7



- Keyword Extraction:** Keyword extraction is a tool that automatically extracts the most used, important words and expressions from a text. It can give valuable insights into people's opinions about any business on social media. Customer Service can be improved by using a Keyword extraction tool.

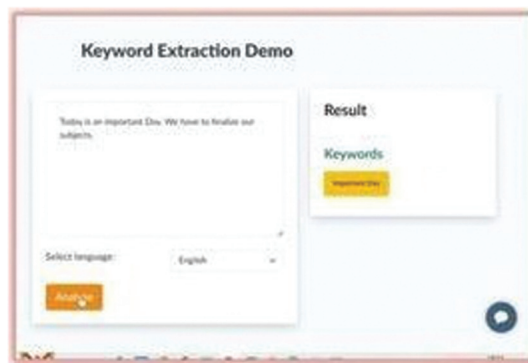


Fig. 11.8

7. Chat GPT

HANDS ON ACTIVITY 1: Keyword Extraction

STEP – 1: Go to the given website: <https://cloud.google.com/natural-language>

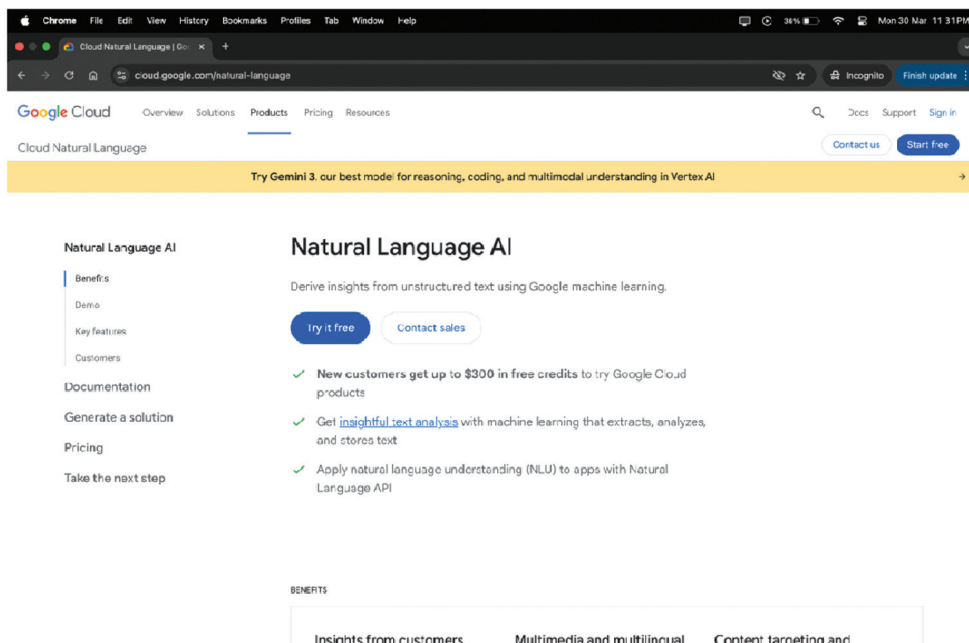


Fig. 11.9



Notes

Natural Language Processing

STEP – 2: Click on ‘Analyze’ and check the results.

Google Cloud Overview Solutions Products Pricing Resources

Cloud Natural Language

Natural Language AI

- Benefits
- Demo
- Key features
- Customers
- Documentation
- Generate a solution
- Pricing
- Take the next step

Natural Language API demo

Try the API

Try the API

Google, headquartered in Mountain View (1600 Amphitheatre Pkwy, Mountain View, CA 940430), unveiled the new Android phone for \$799 at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.

See supported languages

KEY FEATURES

Natural language solutions that work with your text

Natural Language API

The powerful pre-trained models of the [Natural Language API](#) empowers developers to easily apply natural language understanding (NLU) to their applications with features including sentiment analysis, entity analysis, entity sentiment analysis, content

Fig. 11.10

Google Cloud Overview Solutions Products Pricing Resources

Cloud Natural Language

Natural Language AI

- Benefits
- Demo
- Key features
- Customers
- Documentation
- Generate a solution
- Pricing
- Take the next step

Google, headquartered in Mountain View (1600 Amphitheatre Pkwy, Mountain View, CA 940430), unveiled the new Android phone for \$799 at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.

1. \$799	PRICE	2. 1600	NUMBER
3. 940430	NUMBER	4. 799	NUMBER
5. Mountain View (1600 Amphitheatre Pkwy)	ADDRESS	6. 1600 Amphitheatre Pkwy	LOCATION
7. Android	CONSUMER GOOD	8. CA	LOCATION
9. Consumer Electronic Show	EVENT	10. Google	ORGANIZATION
11. Mountain View	LOCATION	12. Sundar Pichai	PERSON
13. keynote	OTHER	14. phone	CONSUMER GOOD
15. phones	CONSUMER GOOD	16. users	PERSON

Fig. 11.11

Natural Language Processing

- The keywords from the paragraph in the textbox have been highlighted in different colours e.g., Google, Mountain View, etc.
- Click on other options to check the output.
- Use your own text in the text box and observe the results.



Notes

11.3 STAGES OF NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) breaks human language into structured data using five different stages. The different stages of Natural Language Processing (NLP) serve various purposes in the overall task of understanding and processing human language. The stages of Natural Language Processing (NLP) typically involve the following:

1. Lexical Analysis
2. Syntactic Analysis
3. Semantic Analysis
4. Discourse Integration
5. Pragmatic Analysis

11.3.1 Lexical Analysis

NLP starts with identifying the structure of input words. It is the process of dividing a large chunk of words into structural paragraphs, sentences, and words. Lexicon stands for a collection of the various words and phrases used in a language.

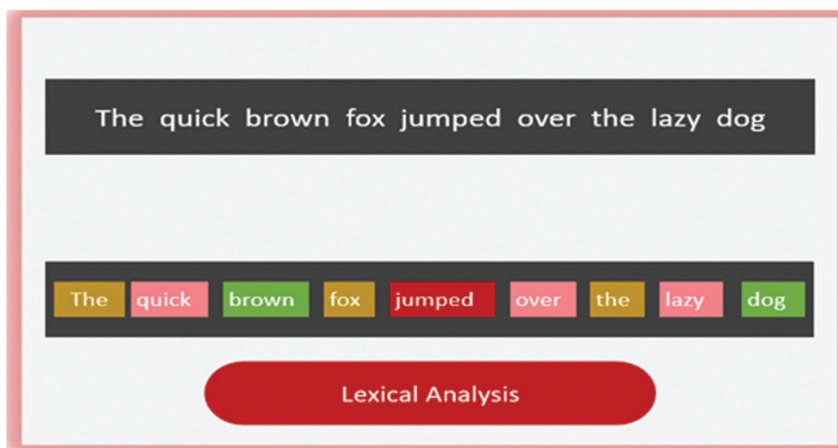


Fig. 11.12

Lengthy text is broken down into chunks.



Notes

11.3.2 Syntactic Analysis / Parsing

It is the process of checking the grammar of sentences and phrases. It forms a relationship among words and eliminates logically incorrect sentences.

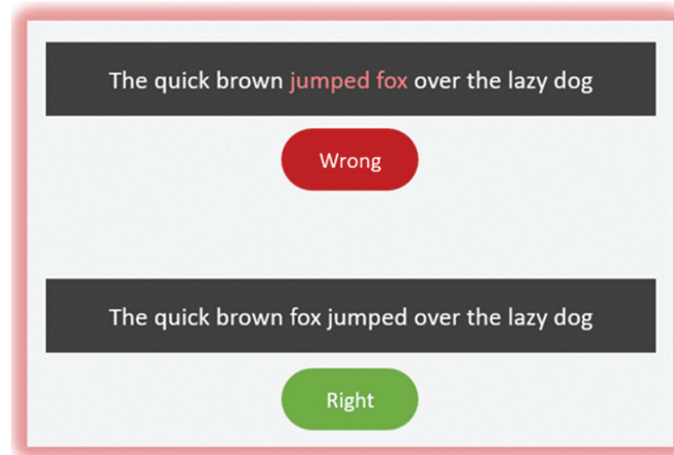


Fig. 11.13

The grammar is correct!

11.3.3 Semantic Analysis

In this stage, the input text is now checked for meaning.

For example:

It will reject a sentence that contains 'hot ice cream' in it. The fox jumped into the dog.

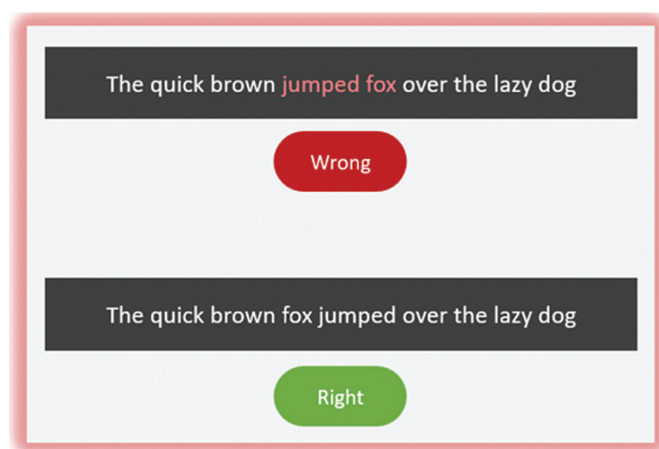


Fig. 11.14



11.3.4 Discourse Integration

It is the process of forming the story of the sentence. Every sentence should have a relationship with its preceding and succeeding sentences.

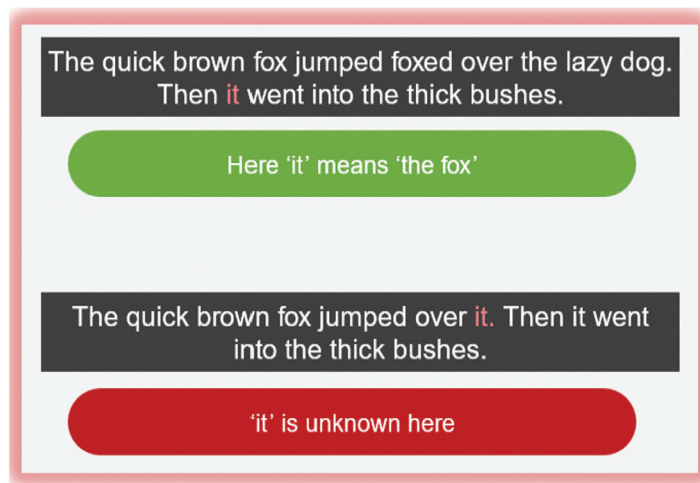


Fig.11.15

The flow of words makes sense!

11.3.5 Pragmatic Analysis

In this stage, sentences are checked for their relevance in the real world. Pragmatic means practical or logical, i.e., this step requires knowledge of the intent in a sentence. It also means to discard the actual word meaning taken after semantic analysis and take the intended meaning.

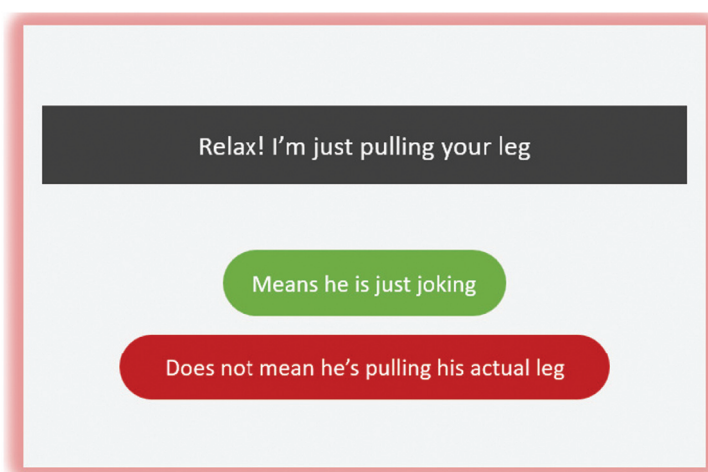


Fig. 11.16



Notes

Natural Language Processing

In summary,

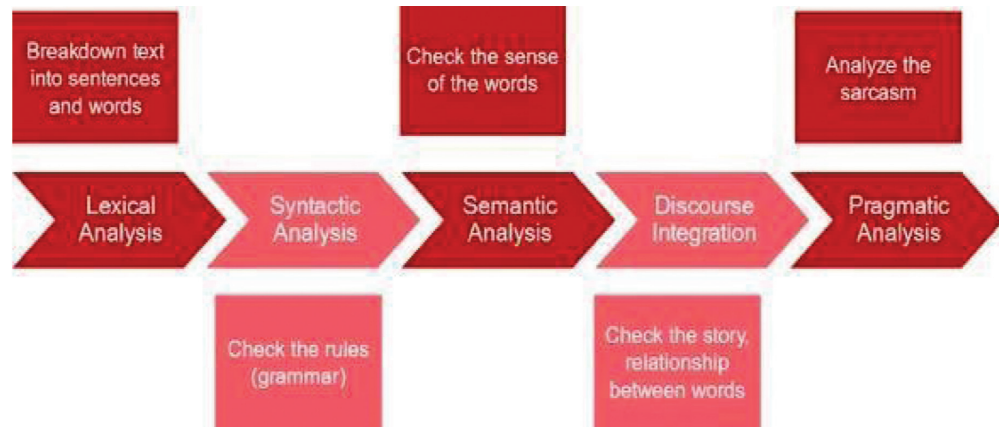


Fig. 11.17



INTEXT QUESTIONS 11.1

Choose the right word:

1. Syntax refers to the grammatical structure of a sentence.

YES

NO

2. Which technique is used to assess the meaningfulness of the input text?

Pragmatic Analysis

Lexical Analysis

Semantic Analysis

Discourse Integration

11.4 CHATBOTS

One of the most common applications of Natural Language Processing is a chatbot. A chatbot is a computer program that's designed to simulate human conversation through voice commands or text chats or both. It can learn over time how to best interact with

humans. It can answer questions and troubleshoot customer problems, evaluate and qualify prospects, generate sales leads and increase sales on an ecommerce site. There are a lot of chatbots available. Let us try some of the chatbots and see how they work. A common application of chatbots is chat GPT.



ACTIVITY 2: Play with chatbots

How a Chatbot Works (Basic Flow)

To understand how these programs talk to us, here is a simple explanation of what happens behind the scenes:

User input → **NLP processing** → **Intent detection** → **Response**

- **User input:** You type or say a message.
- **NLP processing:** The computer breaks down your words to understand the language.
- **Intent detection:** The system figures out what your goal or underlying question is.
- **Response:** The chatbot generates the most appropriate reply.

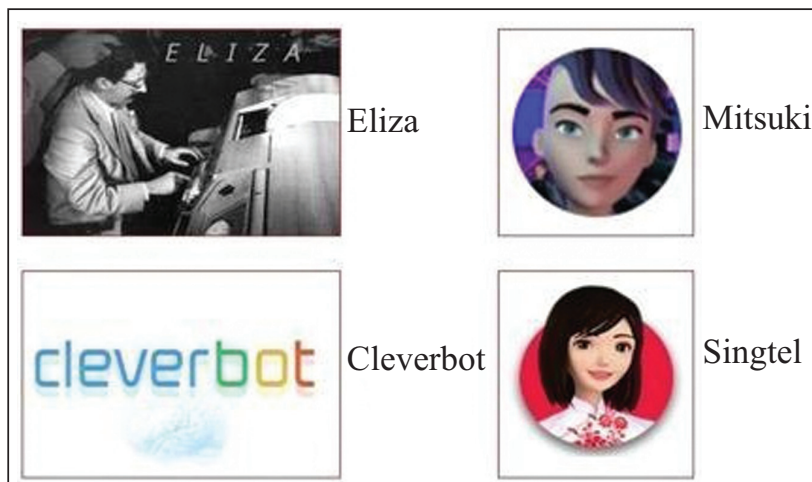


Fig. 11.18

Elizabot: <https://www.masswerk.at/elizabot/>

Mitsuki: <https://www.kuki.ai/>

Cleverbot: <https://www.cleverbot.com/>

Singtel: <https://www.singtel.com/personal/support>

Let us discuss it!

- Which chatbot did you try? Name anyone.
- What is the purpose of this chatbot?



Notes

Natural Language Processing

- How was the interaction with the chatbot?
- Did the chat feel like talking to a human or a robot? Why do you think so?
- Do you feel that the chatbot has a certain personality?

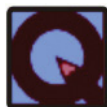
As you interact with more and more chatbots, you would realise that some of them are scripted or in other words are traditional chatbots while others are AI-powered and have more knowledge. With the help of this experience, we can understand that there are 2 types of chatbots around us: Script-bot and Smart-bot. Let us understand what each of them means in detail.

Script bot

- Script bots are easy to make.
- Script bots work around a script which is programmed in them.
- Mostly they are free and are easy to integrate to a messaging platform.
- No or little language processing skills.
- Limited functionality.

Smart bot

- Smart-bots are flexible and powerful.
- Smart bots work on bigger databases and other resources directly.
- Smart bots learn with more data.
- Coding is required to take this up on board.
- Wide functionality.



INTEXT QUESTIONS 11.2

1. Natural Language Processing majorly deals with processing.
(a) Numeric data (b) Textual data
(c) Image data (d) Visual data
2. is an NLP tool to express an opinion, whether the underlying sentiment is positive, negative, or neutral.
(a) Text Classification (b) Machine Translation
(c) Sentiment Analysis (d) Automatic Text Summarization
3. What is the first stage of Natural Language Processing (NLP)?
(a) Semantic Analysis (b) Pragmatic Analysis
(c) Lexical Analysis (d) Syntactic Analysis



4. Words that we want to filter out before doing any analysis of the text are called
 - (a) Rare words
 - (b) Stop words
 - (c) Frequent words
 - (d) Filter words
5. What does discourse integration involve in the context of sentence formation?
 - (a) Identifying individual words in a sentence
 - (b) Forming a coherent story within a sentence
 - (c) Establishing relationships between preceding and succeeding sentences
 - (d) Applying punctuation and grammar rules to a sentence

11.5 TEXT PROCESSING

Humans interact with each other very easily. For us, the natural languages that we use are so convenient that we speak them easily and understand them well too. But for computers, our languages are very complex. As you have already gone through some of the complications in human languages above, now it is time to see how Natural Language Processing makes it possible for machines to understand and speak in Natural Languages just like humans.

Since we all know that the language of computers is Numerical, the very first step that comes to our mind is to convert our language to numbers. This conversion takes a few steps to happen. The first step to it is Text Normalisation. Since human languages are complex, we need to first of all simplify them in order to make sure that understanding becomes possible. Text Normalisation helps in cleaning up the textual data in such a way that it comes down to a level where its complexity is lower than the actual data. Let us go through Text Normalisation in detail.

11.5.1 Text Normalisation

In Text Normalisation, we undergo several steps to normalise the text to a lower level. Before we begin, we need to understand that in this section, we will be working on a collection of written text. That is, we will be working on text from multiple documents and the term used for the whole textual data from all the documents altogether is known as **corpus**. Not only would we go through all the steps of Text Normalisation, we would also work them out on a corpus. Let us take a look at the steps:



Notes

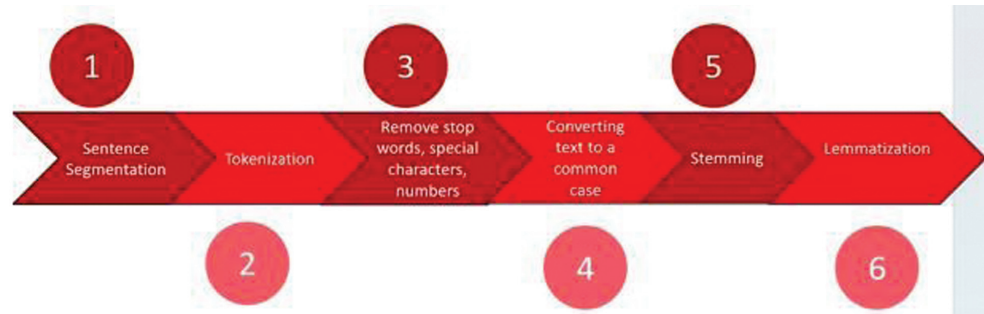


Fig. 11.19

11.5.2 Sentence Segmentation

Under sentence segmentation, the whole corpus is divided into sentences. Each sentence is taken as a different data so now the whole corpus gets reduced to sentences.

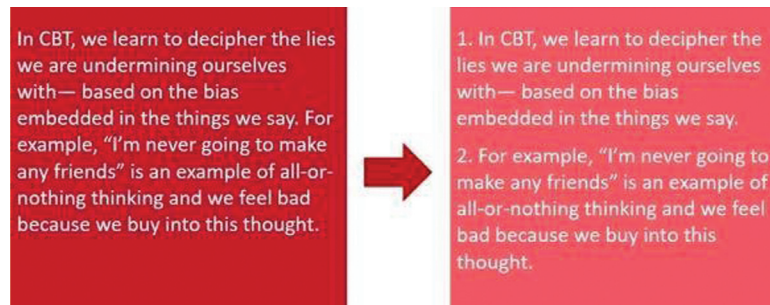


Fig. 11.20

11.5.3 Tokenization

After segmenting the sentences, each sentence is then further divided into tokens. Tokens is a term used for any word or number or special character occurring in a sentence. Under tokenisation, every word, number and special character is considered separately and each of them is now a separate token.

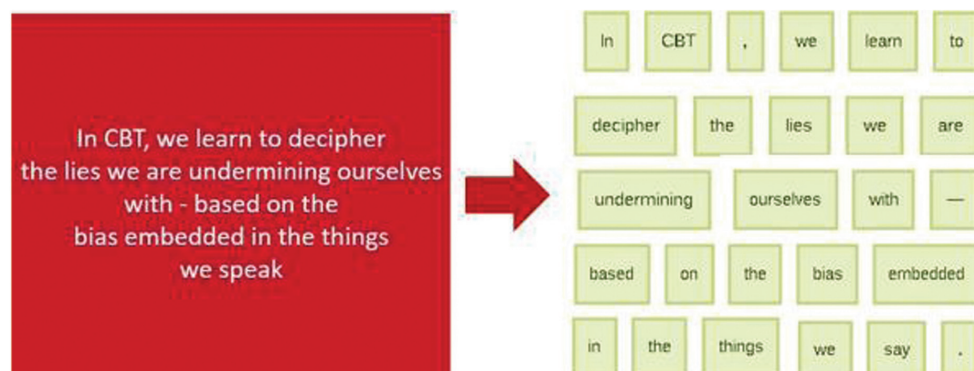


Fig. 11.21



Below is a representation of 4 steps, about how the operation works.

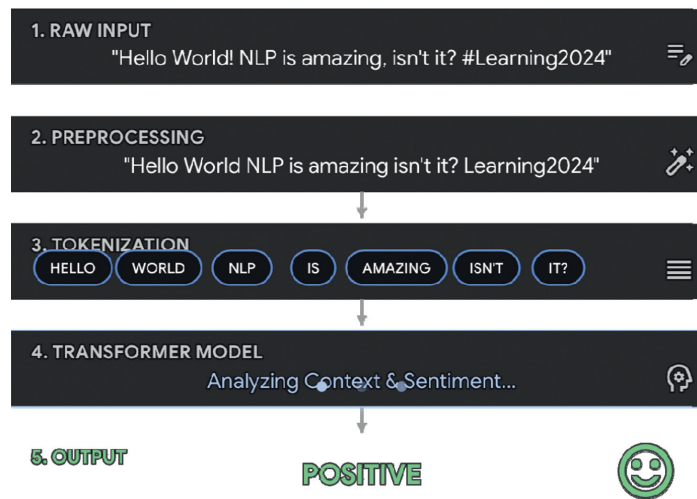


Fig. 11.22

11.5.4 Removing Stop words, Special Characters and Numbers

In this step, the tokens which are not necessary are removed from the token list. What are the possible words which we might not require?

Stop words are the words which occur very frequently in the corpus but do not add any value to it. Humans use grammar to make their sentences meaningful for the other person to understand. But grammatical words do not add any essence to the information which is to be transmitted through the statement hence they come under stop words. Some examples of stop words are:

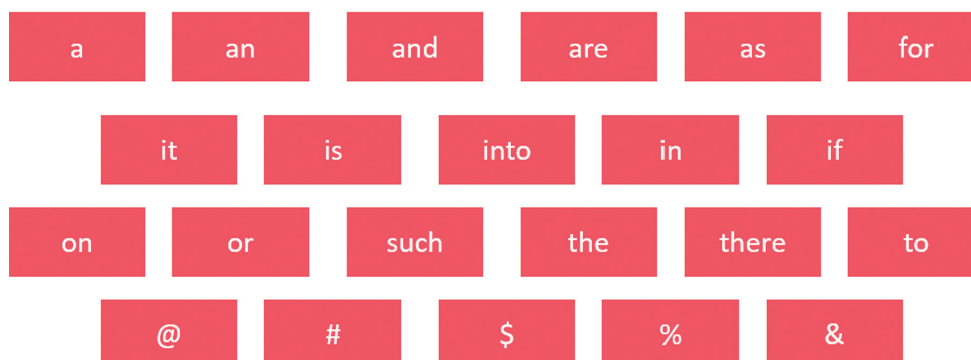


Fig. 11.23

These words occur the most in any given corpus but talk very little or nothing about the context or the meaning of it. Hence, to make it easier for the computer to focus on meaningful terms, these words are removed.



Notes

Along with these words, a lot of times our corpus might have special characters and/or numbers. Now it depends on the type of corpus that we are working on whether we should keep them in it or not. For example, if you are working on a document containing email IDs, social media post then you might not want to remove the special characters and numbers whereas in some other textual data if these characters do not make sense, then you can remove them along with the stop words.

11.5.5 Converting Text to a Common Case

After the stop words removal, we convert the whole text into a similar case, preferably lowercase. This ensures that the case sensitivity of the machine does not consider the same words as different just because of different cases.

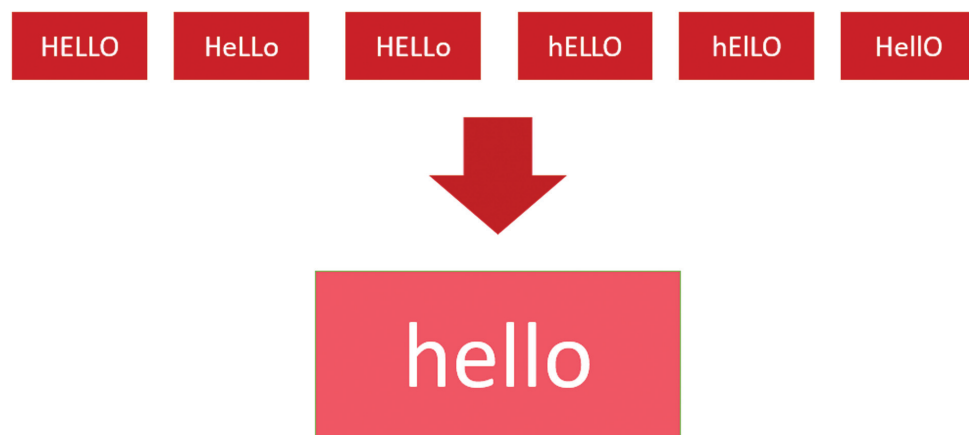


Fig. 11.24

Here in this example, all the 6 forms of hello would be converted to lowercase and hence would be treated as the same word by the machine.

11.5.6 Stemming

In this step, the remaining words are reduced to their root words. In other words, stemming is the process in which the affixes of words are removed and the words are converted to their base form.

Word	Affixes	Stem
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	studi
studying	-ing	study



Note that in stemming, the stemmed words (words that we get after removing the affixes) might not be meaningful. Here in this example as you can see: healed, healing and healer all were reduced to heal but studies were reduced to study after the affix removal which is not a meaningful word. Stemming does not take into account whether the stemmed word is meaningful or not. It just removes the affixes hence it is faster.

11.5.7 Lemmatization

Stemming and lemmatization both are alternative processes to each other as the role of both the processes is same – removal of affixes. But the difference between both of them is that in lemmatization, the word we get after affix removal (also known as lemma) is a meaningful one. Lemmatization makes sure that a lemma is a word with meaning and hence it takes a longer time to execute than stemming.

Word	Affixes	Stem
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	study
studying	-ing	study

As you can see in the same example, the output for studies after affix removal has become study instead of studi.

The difference between stemming and lemmatization can be summarized by this example:

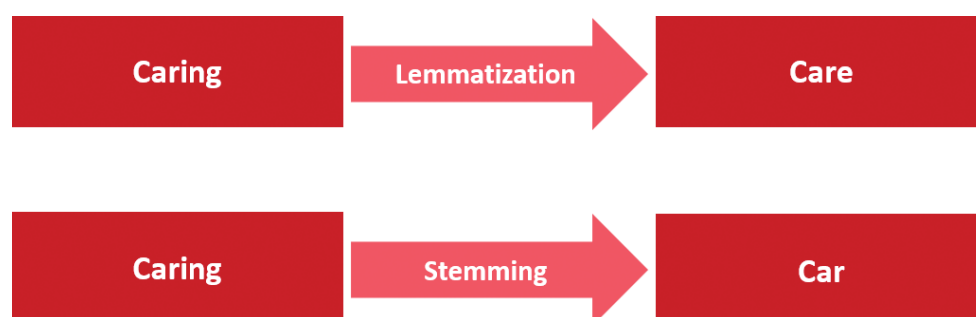


Fig. 11.25

With this, we have normalised our text to tokens which are the simplest form of words present in the corpus. Now it is time to convert the tokens into numbers. For this, we would use the Bag of Words algorithm.



Notes

11.5.8 Bag of Words

Bag of Words is a Natural Language Processing model which helps in extracting features out of the text which can be helpful in machine learning algorithms. In the bag of words, we get the occurrences of each word and construct the vocabulary for the corpus.

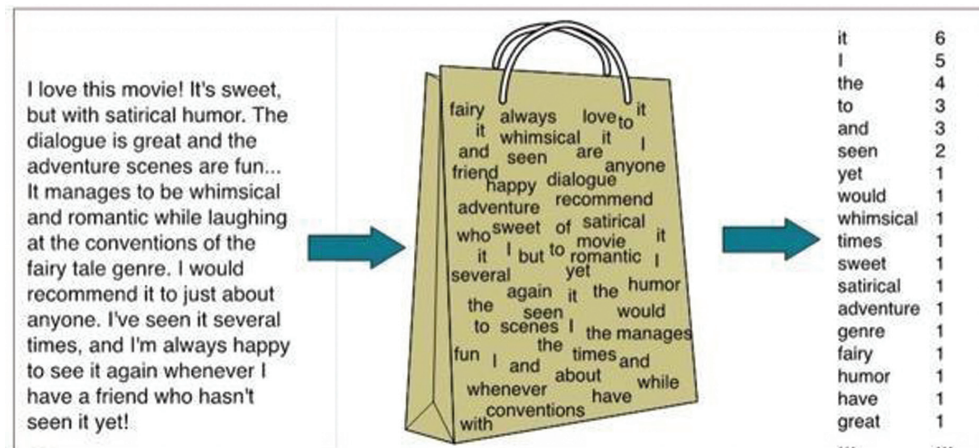


Fig. 11.26

This image gives us a brief overview of how the bag of words works. Let us assume that the text on the left in this image is the normalised corpus which we have got after going through all the steps of text processing. Now, as we put this text into the bag of words algorithm, the algorithm returns to us the unique words out of the corpus and their occurrences in it. As you can see on the right, it shows us a list of words appearing in the corpus and the numbers corresponding to it show how many times the word has occurred in the text body. Thus, we can say that the bag of words gives us two things:

1. A vocabulary of words for the corpus
2. The frequency of these words (number of times it has occurred in the whole corpus).

Here calling this algorithm a “bag” of words symbolises that the sequence of sentences or tokens does not matter. In this case, all we need are the unique words and their frequency.

Here is the step-by-step approach to implementing the bag of words algorithm:

1. **Text Processing:** Collect data and pre-process it
2. **Create a Dictionary:** Make a list of all the unique words occurring in the corpus. (Vocabulary)



3. **Create document vectors:** For each document in the corpus, find out how many times the word from the unique list of words has occurred.
4. Create document vectors for all the documents.

Let us go through all the steps with an example:

ACTIVITY

Step 1: Collecting data and pre-processing it.

Document 1: Aman and Avni are stressed

Document 2: Aman went to a therapist

Document 3: Avni went to download a health chatbot

Here are three documents having one sentence each. After text normalisation, the text becomes:

Document 1: [aman, and, avni, are, stressed]

Document 2: [aman, went, to, a, therapist]

Document 3: [avni, went, to, download, a, health, chatbot]

Step 2: Create a Dictionary

Go through all the steps and create a dictionary i.e., list down all the words which occur in all three documents:

Dictionary:

aman	and	avni	are	stressed	went
download	health	chatbot	therapist	a	to

Note that even though some words are repeated in different documents, they are all written just once as while creating the dictionary, we create the list of unique words.

Step 3: Create a document vector

In this step, the vocabulary is written in the top row. Now, for each word in the document, if it matches the vocabulary, put a 1 under it. If the same word appears again, increment the previous value by 1. And if the word does not occur in that document, put a 0 under it.



Notes

Natural Language Processing

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0

Since, in the first document, we have words: aman, and, avni, are, stressed. So, all these words get a value of 1 and the rest of the words get a 0 value.

Step 4: Create document vectors for all the documents.

The same exercise has to be done for all the documents. Hence, the table becomes:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

In this table, the header row contains the vocabulary of the corpus and three rows correspond to three different documents. Take a look at this table and analyse the positioning of 0s and 1s in it.

Finally, this gives us the **document vector table** for our corpus. However, the tokens have still not been converted to numbers. This leads us to the final step of our algorithm: TFIDF.

11.6 TFIDF: TERM FREQUENCY & INVERSE DOCUMENT FREQUENCY

Suppose you have a book. Which characters or words do you think would occur the most in it?

The bag of words algorithm gives us the frequency of words in each document we have in our corpus. It gives us an idea that if the word is occurring more in a document, its value is more for that document. For example, if I have a document on air pollution, air and pollution would be the words which occur many times in it. And these words are valuable too as they give us some context around the document. But let us suppose we have 10 documents and all of them talk about different issues. One is on women's empowerment; the other is on unemployment and so on. Do you think air and pollution would still be one of the most occurring words in the whole corpus? If not, then which words do you think would have the highest frequency in all of them? **And, this, is, the, etc.** are the words which occur the most in almost all the documents. But these words do not talk about the corpus at all. Though they are important for humans as they make the statements



understandable to us, for the machine they are a complete waste as they do not provide us with any information regarding the corpus. Hence, these are termed as stop words and are mostly removed at the pre-processing stage only.

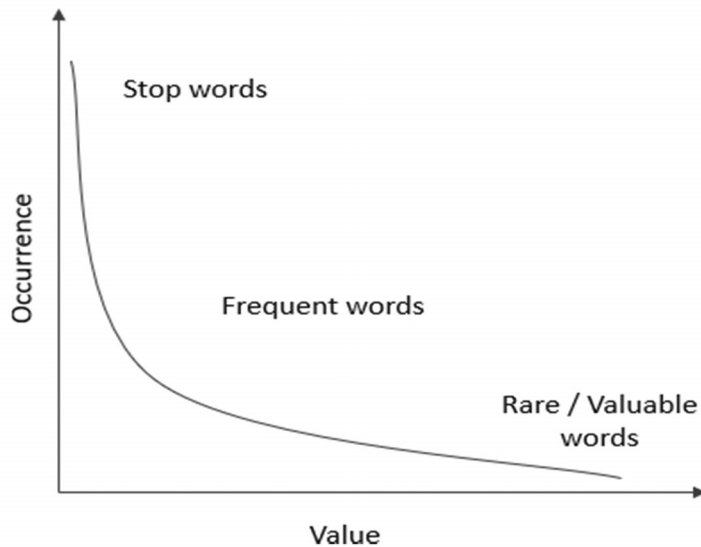


Fig. 11.27

Take a look at this graph. It is a plot of the occurrence of words versus their value. As you can see, if the words have the highest occurrence in all the documents of the corpus, they are said to have negligible value hence they are termed as stop words. These words are mostly removed at the pre-processing stage only. Now as we move ahead from the stop words, the occurrence level drops drastically and the words which have adequate occurrence in the corpus are said to have some amount of value and are termed as frequent words. These words mostly talk about the document's subject and their occurrence is adequate in the corpus. Then as the occurrence of words drops further, the value of such words rises. These words are termed as rare or valuable words. These words occur the least but add the most value to the corpus. Hence, when we look at the text, we consider frequent and rare words.

Let us now demystify TFIDF. TFIDF stands for Term Frequency and Inverse Document Frequency. TFIDF helps us identify the value of each word. Let us understand each term one by one.

11.6.1 Term Frequency

Term frequency is the frequency of a word in one document. Term frequency can easily be found in the document vector table as in that table we mention the frequency of each word of the vocabulary in each document.



Notes

Natural Language Processing

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

Here, you can see that the frequency of each word for each document has been recorded in the table. These numbers are nothing but the Term Frequencies!

11.6.2 Inverse Document Frequency

Now, let us look at the other half of TFIDF which is Inverse Document Frequency. For this, let us first understand what document frequency means. Document Frequency is the number of documents in which the word occurs irrespective of how many times it has occurred in those documents. The document frequency for the exemplar vocabulary would be:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
2	1	2	1	1	2	2	2	1	1	1	1

Here, you can see that the document frequency of 'aman', 'avni', 'went', 'to' and 'a' is 2 as they have occurred in two documents. The rest of them occurred in just one document hence the document frequency for them is one.

Talking about inverse document frequency, we need to put the document frequency in the denominator while the total number of documents is the numerator. Here, the total number of documents is 3, hence inverse document frequency becomes:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
3/2	3/1	3/2	3/1	3/1	3/2	3/2	3/2	3/1	3/1	3/1	3/1

Finally, the formula of TFIDF for any word W becomes:

$$\text{TFIDF}(W) = \text{TF}(W) * \log(\text{IDF}(W))$$

Here, log is to the base of 10. Don't worry! You don't need to calculate the log values by yourself. Simply use the log function in the calculator and find out!

Now, let's multiply the IDF values by the TF values. Note that the TF values are for each document while the IDF values are for the whole corpus. Hence, we need to multiply the IDF values to each row of the document vector table.



aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
$1 \cdot \log(3/2)$	$1 \cdot \log(3)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3)$	$1 \cdot \log(3)$	$0 \cdot \log(3/2)$	$0 \cdot \log(3/2)$	$0 \cdot \log(3/2)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$
$1 \cdot \log(3/2)$	$0 \cdot \log(3)$	$0 \cdot \log(3/2)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$
$0 \cdot \log(3/2)$	$0 \cdot \log(3)$	$1 \cdot \log(3/2)$	$0 \cdot \log(3)$	$0 \cdot \log(3)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3/2)$	$1 \cdot \log(3/2)$	$0 \cdot \log(3)$	$1 \cdot \log(3)$	$1 \cdot \log(3)$	$1 \cdot \log(3)$

Here, you can see that the IDF values for Aman in each row are the same and a similar pattern is followed for all the words of the vocabulary. After calculating all the values, we get:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
0.176	0.477	0.176	0.477	0.477	0	0	0	0	0	0	0
0.176	0	0	0	0	0.176	0.176	0.176	0.477	0	0	0
0	0	0.176	0	0	0.176	0.176	0.176	0	0.477	0.477	0.477

Finally, the words have been converted to numbers. These numbers are the values of each for each document. Here, you can see that since we have less amount of data, words like 'are' and 'and' also have a high value. But as the IDF value increases, the value of that word decreases. That is, for example:

Total Number of documents: 10

Number of documents in which 'and' occurs: 10 Therefore, $IDF(\text{and}) = 10/10 = 1$

Which means: $\log(1) = 0$. Hence, the value of 'and' becomes 0.

On the other hand, the number of documents in which 'pollution' occurs: 3
 $IDF(\text{pollution}) = 10/3 = 3.3333...$

This means $\log(3.3333) = 0.522$; which shows that the word 'pollution' has considerable value in the corpus.

Summarising the concept, we can say that:

1. Words that occur in all the documents with high term frequencies have the lowest values and are considered to be the stop words.
2. For a word to have a high TFIDF value, the word needs to have a high term frequency but less document frequency which shows that the word is important for one document but is not a common word for all documents.
3. These values help the computer understand which words are to be considered while processing the natural language. The higher the value, the more important the word is for a given corpus.



Notes

11.6.3 Applications of TFIDF

TFIDF is commonly used in the Natural Language Processing domain. Some of its point wise applications are:

Document Classification	Topic Modelling	Information Retrieval System	Stop word filtering
Helps in classifying the type and genre of a document.	It helps in predicting the topic for a corpus.	To extract the important information out of a corpus.	Helps in removing unnecessary words from a text body.

11.7 NATURAL LANGUAGE PROCESSING: USE CASE WALK THROUGH EXAMPLES OF CODE AND NO-CODE NLP TOOLS

Code NLP	No-Code NLP
NLTK package: Natural Language Tool Kit or NLTK is a package readily available for text processing in Python. The package contains functions and modules which can be used for Natural Language Processing.	Orange Data Mining: It is a machine learning tool for data analysis through Python and visual programming. We can perform operations on data through simple drag-and-drop steps.
SpaCy: SpaCy is an open-source natural language processing (NLP) library designed to build NLP applications. It offers various features such as tokenization, part-of- speech tagging, named entity recognition, dependency parsing, and more.	MonkeyLearn: MonkeyLearn is a text analysis platform that offers NLP tools and machine learning models for text analysis, supporting tasks such as classification, sentiment analysis, and entity recognition. Users can create custom models or use pre-trained ones for tasks like social media monitoring and customer feedback analysis.

Applications of NLP



Sentiment (opinion)

Fig.11.28



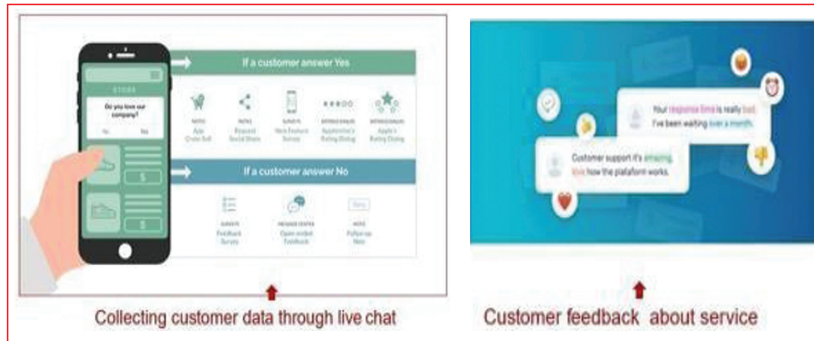
Analysis

Fig. 11.29



11.8 INTRODUCTION TO SENTIMENT ANALYSIS

Sentiment analysis is a natural language processing (NLP) technique used to analyze whether a given textual data is positive, negative, or neutral.



11.8.1 Applications of Sentiment Analysis-Customer Service

Customer sentiment analysis helps in the automatic detection of emotions when customers interact with products, services, or brands.



11.8.2 Applications of Sentiment Analysis – Voice of the Customer

- Voice of the customer analysis helps to analyze customer feedback and gain actionable insights from it.
- It measures the gap between what customers expect and what they actually experience when they use the products or services.

Understanding and utilizing these NLP tools has become highly valuable across various business sectors. Here is why industries rely on them:

- **Rapid Prototyping and Cost Efficiency:** No-code tools allow businesses to build and deploy text analysis models quickly without needing a team of highly specialized software engineers.



Notes

Natural Language Processing

- **Real-time Brand Monitoring:** Companies use sentiment analysis to constantly monitor social media and product reviews, allowing them to instantly catch and address negative PR or customer dissatisfaction.
- **Automating Workflow and Routing:** Text classification is heavily used to automatically read incoming customer support tickets and instantly route them to the correct department (e.g., billing, tech support), saving hundreds of human hours.
- **Data-Driven Decision Making:** Management teams use keyword extraction and topic modeling to digest thousands of survey responses or market reports, turning overwhelming text into clear market trends.

In a corporate setting, an AI Assistant acts as a digital worker, leveraging these exact NLP techniques to perform complex daily tasks:

- **Intelligent Triage and Support:** The AI reads incoming emails, uses *Intent Detection* and *Sentiment Analysis* to gauge urgency and either drafts an immediate, empathetic reply or escalates an angry email to a human manager.
- **Document Summarization:** Instead of an executive reading a 50-page legal contract or financial report, the AI Assistant uses NLP to extract the most critical clauses, keywords, and summaries in seconds.
- **Market Intelligence Gathering:** The assistant can scrape daily news articles or competitor reviews, apply *Bag of Words* or *TFIDF*, and send the marketing team a daily brief on what keywords competitors are suddenly focusing on.
- **Internal Knowledge Retrieval:** Acting as an HR or IT bot, the AI processes an employee's natural language question (e.g., "How do I apply for sick leave?"), understands the semantics, and retrieves the exact company policy from a massive internal database.



INTEXT QUESTIONS 11.3

1. Why is context and ambiguity a major challenge for Artificial Intelligence systems attempting to understand human language?
 - (a) Computers can only process text written in the Roman alphabet.
 - (b) Human languages constantly evolve, making rule-based parsing obsolete.
 - (c) The computational cost of translating words into binary code is too high.
 - (d) A single word or phrase can have multiple meanings depending on the surrounding text or situation.



2. What is the primary challenge faced by computers in understanding human languages?
 - (a) Complexity of human languages
 - (b) Lack of computational power
 - (c) Incompatibility with numerical data
 - (d) Limited vocabulary
3. How do voice assistants utilize NLP?
 - (a) To analyze visual data
 - (b) To process numerical data
 - (c) To understand natural language
 - (d) To execute tasks based on computer code
4. Which of the following is NOT a step in Text Normalisation?
 - (a) Tokenization
 - (b) Lemmatization
 - (c) Punctuation removal
 - (d) Document summarization
5. In the context of text processing, what is the purpose of tokenisation?
 - (a) To convert text into numerical data
 - (b) To segment sentences into smaller units
 - (c) To translate text into multiple languages
 - (d) To summarize documents for analysis
6. What distinguishes lemmatization from stemming?
 - (a) Lemmatization produces meaningful words after affix removal, while stemming does not.
 - (b) Lemmatization is faster than stemming.
 - (c) Stemming ensures the accuracy of the final word.
 - (d) Stemming generates shorter words compared to lemmatization.
7. What is the primary purpose of the Bag of Words model in Natural Language Processing?
 - (a) To translate text into multiple languages
 - (b) To extract features from text for machine learning algorithms
 - (c) To summarize documents for analysis
 - (d) To remove punctuation marks from text



Notes

8. In the context of text processing, what are stop words?
 - (a) Words with the frequent occurrence in the corpus
 - (b) Words with negligible value that are often removed during preprocessing
 - (c) Words with the lowest occurrence in the corpus
 - (d) Words with the most value added to the corpus
9. What is the characteristic of rare or valuable words in the described plot?
 - (a) They have the highest occurrence in the corpus
 - (b) They are often considered stop words
 - (c) They occur the least but add the most value to the corpus
 - (d) They are typically removed during preprocessing
10. What information does the document vector table provide?
 - (a) The frequency of each word across all documents
 - (b) The frequency of each word in a single document
 - (c) The total number of words in the entire corpus
 - (d) The average word length in the entire corpus
11. What is the primary purpose of TFIDF in text processing?
 - (a) To identify the presence of stop words in documents
 - (b) To remove punctuation marks from text
 - (c) To identify the value of each word in a document
 - (d) To translate text into multiple languages
12. **Assertion:** Pragmatic analysis in natural language processing (NLP) involves assessing sentences for their practical applicability in real-world scenarios.
Reasoning: Pragmatic analysis requires understanding the intended meaning behind sentences and considering their practical or logical implications, rather than solely relying on literal word meanings obtained from semantic analysis.
 - (a) Both Assertion and Reasoning are true, and Reasoning is the correct explanation of the Assertion.
 - (b) Assertion is true, but Reasoning is false.
 - (c) Both Assertion and Reasoning are true, but Reasoning is not the correct explanation of the Assertion.
 - (d) The assertion is false, but the reasoning is true.



13. **Assertion:** Converting the entire text into lowercase following stop word removal is a crucial preprocessing step in natural language processing.

Reasoning: This process ensures uniformity in word representation, preventing the machine from treating words with different cases as distinct entities, thereby enhancing the accuracy of subsequent text analysis.

- (a) Both Assertion and Reasoning are true, and Reasoning is the correct explanation of the Assertion.
- (b) Assertion is true, but Reasoning is false.
- (c) Both Assertion and Reasoning are true, but Reasoning is not the correct explanation of the Assertion.
- (d) Assertion is false, but Reasoning is true



WHAT YOU HAVE LEARNT

- The concept of NLP and its applications
- To clean text using tokenization, stop-word removal, stemming, lemmatization, and lowercasing.
- Understanding Bag-of-Words, TF-IDF - how text is converted into numerical vectors.
- Building models for sentiment analysis
- Various text processing techniques used in NLP.



TERMINAL QUESTIONS

1. Mention a few features of natural languages.
2. What is the significance of NLP?
3. What do you mean by lexical analysis in NLP?
4. What do you mean by a chatbot?
5. What does the term “Bag of Words” refer to in Natural Language Processing (NLP)?



Notes

Natural Language Processing

6. Describe two practical uses of Natural Language Processing in real-world scenarios.
7. Explain the process of stemming and lemmatization in text processing, supported by an example.
8. Describe any four applications of TFIDF.
9. Samiksha, a student of class X was exploring the Natural Language Processing domain. She got stuck while performing the text normalisation. Help her to normalise the text on the segmented sentences given below:

Document 1: Akash and Ajay are best friends.

Document 2: Akash likes to play football but Ajay prefers to play online games.

10. Through a step-by-step process, calculate TFIDF for the given corpus

Document 1: Johny Johny Yes Papa,

Document 2: Eating sugar? No Papa

Document 3: Telling lies? No Papa

Document 4: Open your mouth, Ha! Ha



ANSWER TO INTEXT QUESTIONS

11.1

1. (a)
2. (c)

11.2

1. (b)
2. (c)
3. (c)
4. (b)
5. (c)

11.3

1. (b)
2. (a)
3. (c)
4. (d)
5. (b)
6. (a)
7. (b)
8. (b)
9. (c)
10. (b)
11. (c)
12. (a)
13. (a)



KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- describe the importance of pre-processing NLP data.
- list several applications of NLP which can be implemented without code.
- enlist different No-Code NLP tools
- Outline the concept of the Bag of Words algorithm

References:

1. https://cbseacademic.nic.in/web_material/Curriculum26/publication/secondary/AI_Facilitators_Handbook_X.pdf



12

COMPUTER VISION

Have you ever wondered how your smartphone unlocks just by looking at your face? Or how a self-driving car knows exactly when to stop for a pedestrian? Or perhaps you have used a funny filter on social media that puts puppy ears perfectly on your head, even when you move around? This is not magic; it is a field of Artificial Intelligence called **Computer Vision (CV)**.

While humans see the world effortlessly using eyes and brains, computers “see” the world very differently. To a computer, a beautiful photograph of a sunset or a picture of your pet dog is just a massive grid of numbers called **pixels**. The challenge of Computer Vision is to teach machines not just to record these images, but to *understand* them.

In previous lessons, you learnt about data and basic machine learning. In this lesson, we will focus specifically on **Visual Data**. We will explore how computers process images, how they distinguish a cat from a dog (Image Classification), and how they can pinpoint exactly where an object is located (Object Detection). Finally, we will move from theory to practice by building our own “smart eye” using tools like **Lobe**.



OBJECTIVES

After reading this lesson, the learner will be able to:

- define Computer Vision and explain how computers interpret images using pixels and RGB values;
- distinguish between key CV tasks: Image Classification, Object Detection, and Image Segmentation;
- explain the concept of “Features” (such as edges, colors, and shapes) and how machines use them to recognize objects;

- identify real-world applications of Computer Vision, including facial recognition, autonomous vehicles, and augmented reality filters;
- create and test a custom Image Classifier model using a no-code tool like Lobe.



Notes

12.1 HOW COMPUTERS “SEE”: IMAGES AND PIXELS

To understand Computer Vision, we must first understand how a computer “sees” a digital image. When you look at a photograph of a park, you see trees, children playing, and a blue sky. You perceive shapes and meaning instantly. However, a computer is blind to “meaning.” It does not see a “tree” or a “child.” Instead, it sees a massive spreadsheet full of numbers.

The Building Block: The Pixel

Every digital image is composed of tiny dots called pixels (short for “Picture Elements”). If you zoom into a photo on your computer, you will eventually see these small squares. To a computer, an image is just a grid (or a matrix) of these pixels.

Grayscale Images (Black and White):

In a simple black-and-white image, each pixel represents the brightness of that specific point. Computers store this brightness as a number:

- **0:** Represents pure **Black** (total darkness).
- **255:** Represents pure **White** (total brightness).
- **Numbers between 0-255:** Represent various shades of **Grey**.

For example, a dark grey shadow might have a value of **50**, while a bright grey cloud might have a value of **200**. Why 255? Because computers use 8 bits of memory (1 byte) to store this data, and 2^8 equals 256 distinct values (0 to 255). (See Fig. 12.1)

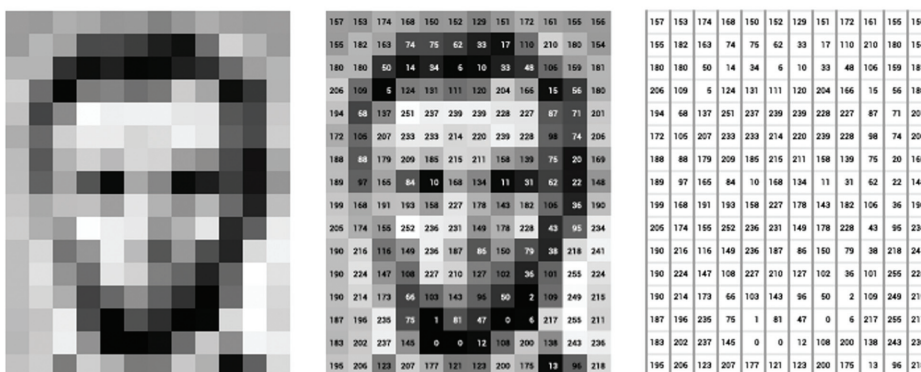


Fig. 12.1: Staring images in grayscale model



Notes

Color Images (The RGB Model):

Most images we see are in color. To create color, computers stack three of these grids on top of each other – one for Red, one for Green and one for Blue. This is known as the RGB Model.

- Every single pixel in a color photo is actually a combination of three numbers (e.g., Red: 200, Green: 50, Blue: 50 would produce a reddish color).
- By mixing these three primary colors at different intensities, computers can create millions of distinct colors, just like a painter mixing paints on a palette.

So, when you upload a selfie, the computer is actually processing three massive grids of numbers, analyzing the mathematical patterns to determine where your face ends and the background begins. (See Fig. 12.2)

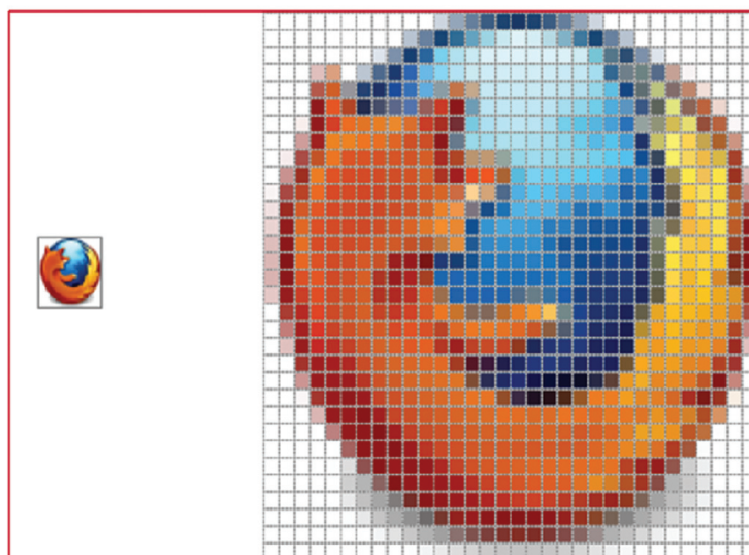


Fig. 12.2: RGB Color Model

12.2 CORE TASKS IN COMPUTER VISION

Computer Vision is not a single task; it is a collection of different jobs. Just as a human might “glance” at a scene to name it or “stare” to pick out details, computers use different techniques depending on the goal. As shown in the above, these tasks progress from simple naming to precise outlining.

1. **Image Classification (The “What”)** This is the starting point of Computer Vision. The computer looks at an image containing a single object and assigns it a label. It answers the question: “*What is the main object in this picture?*”



- **How it works:** The computer analyzes the features (colors, edges) and predicts the most probable label.
 - **Example:** In the first panel of the diagram, the computer analyzes the photo and simply outputs the label “CAT”. It knows a cat is present, but not where it is.
2. **Object Detection (The “Where”)** This is a step more advanced. The computer identifies *multiple* objects in one scene and pinpoints their locations. It answers the question: “*Where are the objects located in this picture?*”
- **How it works:** The computer draws a rectangular box, called a **Bounding Box**, around each distinct object it recognizes.
- Example:** Look at the third panel. The image contains a cat, a dog and a duck. The computer detects all three distinct objects and draws boxes around them: a **Red box** for the Cat, a **Blue box** for the Dog, and a **Green box** for the Duck.
3. **Instance Segmentation (The “Exact Shape”)** This is the most precise technique (shown in the fourth panel). Instead of drawing a generic box, the computer traces the exact outline of each object. It answers the question: “*Which specific pixels belong to each object?*”
- **How it works:** The computer classifies every single pixel in the image. It doesn’t just separate the animals from the background; it separates each animal from the others (Instance).
 - **Example:** In the final panel, the computer highlights the exact shape of each animal. The pixels belonging to the **Cat are colored Red**, the **Dog in Blue**, and the **Duck in Green**, perfectly separating them from the grassy background. (See Fig. 12.3)

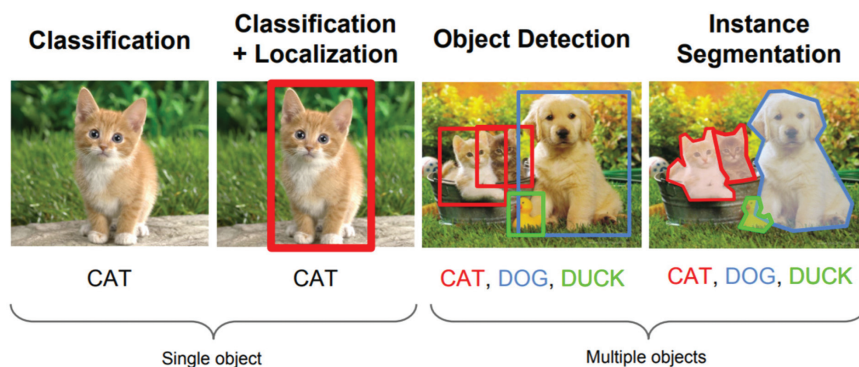


Fig. 12.3: Task in Computer Vision



Notes

How to capture Image Data

Before a computer can recognize a cat or a car, we must “teach” it by showing it examples. This collection of examples is called a **Dataset**. (See fig. 12.4) The most important rule in AI is: *Your model is only as good as your data*. If you feed the computer bad images, it will learn bad habits.



Fig. 12.4: Collection of images

Sources of Images You can collect images in two main ways:

- **Static Photos:** Uploading existing files (like JPEGs or PNGs) from your computer or the internet.
- **Live Webcam:** capturing a stream of images in real-time using a camera connected to your computer.

The Golden Rules for a Good Dataset To build a reliable AI model, you cannot just take 10 identical photos. You must follow these guidelines to ensure your data is robust:



- **Quality and Focus:** Ensure the images are sharp and clear. If the object is blurry, the computer cannot find the edges or features it needs.
- **Diversity:** This is the most critical factor. You need a **diverse dataset** so the AI recognizes the object in different situations.
 - **Different Angles:** Capture the object from the front, side, and top.
 - **Different Lighting:** Take photos in bright sunlight, dim indoor light, and shadow.
 - **Different Sizes & Distances:** Place the object close to the camera and far away so the AI learns scale.
- **Consistency:** Computers prefer structure. It helps if your training images have the same **size** (resolution) and **aspect ratio** (shape, like square or widescreen).

Quantity: How much is enough? A machine needs many examples to find a pattern. Just one or two photos are rarely enough. You need a sufficient **quantity** of data to cover all the variations mentioned.

12.3 CONVOLUTIONAL NEURAL NETWORKS (CNN)

To recognize complex objects like faces, computers use a special type of Artificial Intelligence model called a **Convolutional Neural Network (CNN)**. While traditional programming relies on manually defined rules (e.g., “if it has whiskers, it is a cat”), a CNN learns these rules by itself by looking at thousands of examples.

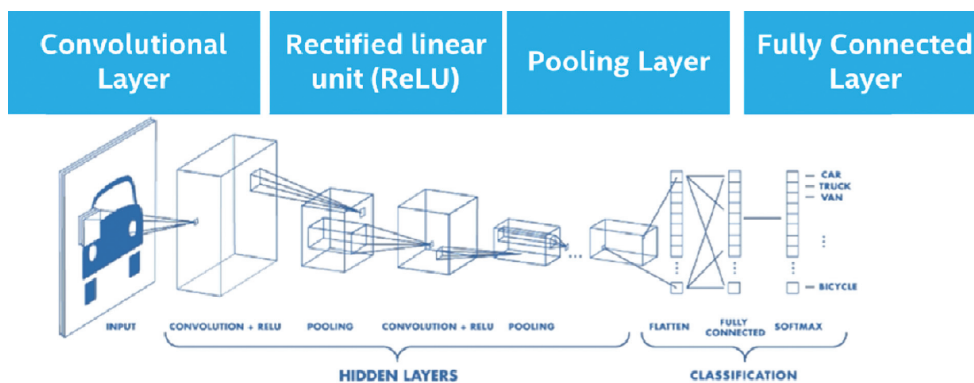


Fig. 12.5: Convolutional Neural Networks

Think of a CNN as a factory assembly line. An image enters at the start, goes through several stations (layers) where it is processed, and at the end, the machine labels what it is.



Notes

12.3.1 The 4 Stages of a CNN Pipeline:

1. **The Convolutional Layer (The Feature Hunter)** This is the first and most important step. Imagine viewing a large picture through a small square window (usually 3x3 pixels) that slides across it. This window is called a **Kernel** or **Filter**.
 - **How it works:** As the filter slides over the pixels, it performs math operations to detect specific **features**. One filter might look for vertical lines, another for horizontal edges, and another for curves.
 - **Result:** It creates a “Feature Map” that highlights where these specific shapes are in the image.

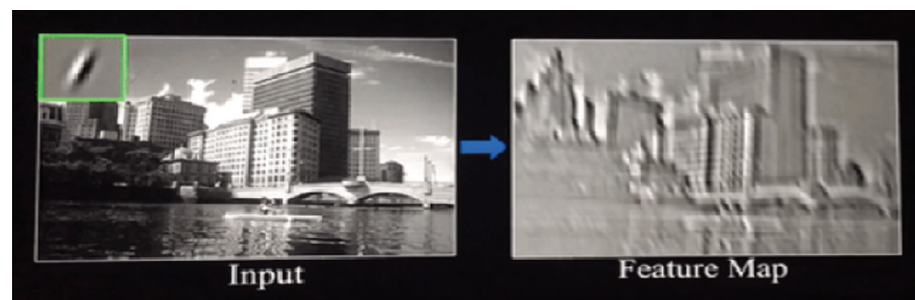


Fig. 12.6: Convolutional Neural (Feature Hunter)

2. **The ReLU Layer (The Cleaner)** ReLU stands for **Rectified Linear Unit**. It sounds complex, but its job is simple: it removes negative values. (See Fig. 12.7)
 - **How it works:** In image data, negative numbers (black/dark areas) might not carry useful information for feature detection. ReLU turns all negative values to 0.
 - **Result:** This acts like an image filter that cleans up the data, making the edges and features stand out more sharply (like increasing the contrast).

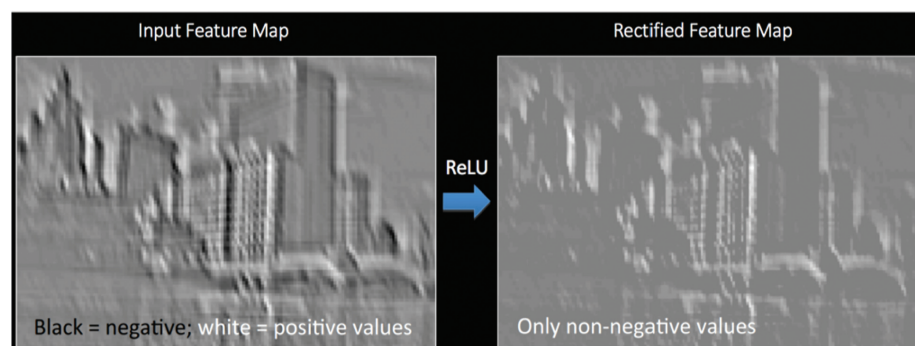


Fig. 12.7: Rectified Linear Unit



3. **The Pooling Layer (The Shrinker)** Images are big and have millions of pixels. Processing every single pixel takes a lot of time and computing power. The Pooling layer solves this by shrinking the image size. (See in fig 12.8)
 - **How it works:** It looks at a small grid (e.g., 2x2 pixels) and keeps only the **maximum** number (Max Pooling).
 - **Result:** The image becomes smaller, but the most important features (like the shape of an eye or a wheel) are preserved. This makes the AI faster and less sensitive to small distortions (e.g., if the cat is tilted slightly).

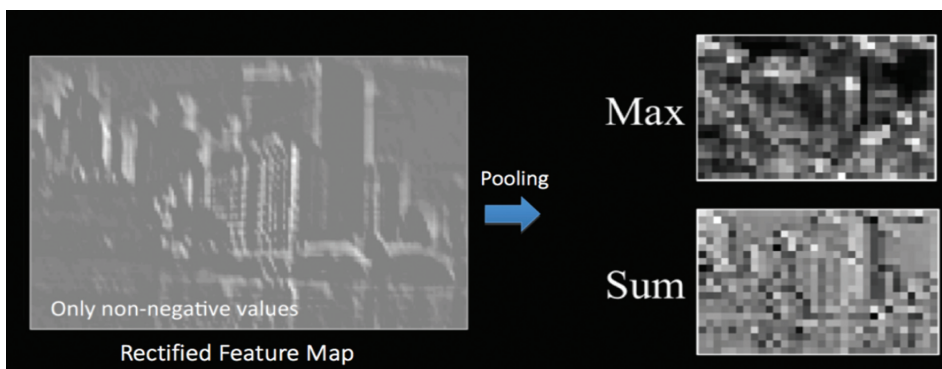


Fig. 12.8: Pooling Layer

4. **The Fully Connected Layer (The Decision Maker)** After passing through the above layers multiple times, the image is converted into a simple list of numbers. This final layer looks at these numbers and calculates the probability of what the object is. (See fig. 12.9)
 - **Result:** It gives a final answer, such as: “**Car: 70%, Truck: 20%, Bicycle: 10%**”.



Fig. 12.9: Fully Connected Layer



Notes



INTEXT QUESTIONS 12.1

1. In a grayscale image, the pixel value **0** usually represents:
 - (a) White
 - (b) Black
 - (c) Red
 - (d) Transparent
2. Which Computer Vision task involves detecting an object and drawing a **Bounding Box** around it?
 - (a) Classification
 - (b) Object Detection
 - (c) Image Segmentation
 - (d) Data Cleaning
3. What is the small window called that slides over an image in a CNN to detect features like edges?
 - (a) A Pixel
 - (b) A Kernel (or Filter)
 - (c) A Box
 - (d) A Pool
4. Which layer in a CNN is responsible for reducing the size of the image data to make processing faster?
 - (a) Convolutional Layer
 - (b) ReLU Layer
 - (c) Pooling Layer
 - (d) Output Layer
5. The **ReLU** layer in a neural network removes negative numbers to help define features more clearly.
6. **RGB** stands for:
 - (a) Red, Grey, Blue
 - (b) Real, Green, Black
 - (c) Red, Green, Blue
 - (d) Red, Gold, Blue
7. Which hardware device helps run powerful AI models on small computers (like a laptop or Raspberry Pi) without needing the internet?
 - (a) Intel® Neural Compute Stick 2 (NCS2)
 - (b) A standard USB Mouse
 - (c) A WiFi Router
 - (d) A Printer



WHAT YOU HAVE LEARNT

- **Computers See Numbers, Not Objects:** While humans see shapes and meaning, computers process digital images as massive grids of numbers (pixels) that represent brightness (Grayscale) or color combinations (Red, Green, Blue).
- **Computer Vision Has Distinct Tasks:** CV encompasses different levels of understanding, from simply naming an object (Image Classification), to locating it with a bounding box (Object Detection), to outlining its exact pixel boundaries (Instance Segmentation).
- **Data Diversity is Crucial:** An AI model is only as good as its dataset. Training requires capturing high-quality images with diverse angles, lighting and distances to ensure the machine learns robust patterns.
- **CNNs recognise Images in Stages:** Convolutional Neural Networks (CNNs) recognise images using specific layers like: Convolutional (finding patterns), ReLU (cleaning negative data), Pooling (shrinking image size), and Fully Connected (making the final prediction).
- **Using Tools Bridge the Gap:** Platforms like Lobe and Google Cloud Vision API allow users to build, train, and test custom image recognition models without writing complex code from scratch.



TERMINAL QUESTIONS

1. Explain the fundamental difference between how a human perceives a photograph of a landscape and how a computer “sees” that exact same digital file.
2. Describe how the RGB model works. If a computer wants to display a specific color for a single pixel, what exactly is it combining?
3. Compare and contrast “Object Detection” and “Instance Segmentation.” Which of the two is considered more precise, and why?
4. You are tasked with creating an AI model to recognize your specific water bottle. According to the “Golden Rules for a Good Dataset,” what are three different ways you should alter the environment or camera when capturing your 30 training photos?



Notes



Notes

5. In the pipeline of a Convolutional Neural Network (CNN), what is the specific role of a “Kernel” (or Filter)?
6. An engineer is building a self-driving car system, but the computer is struggling to process the high-resolution camera images fast enough. Which specific CNN layer is designed to solve this problem by shrinking the data while keeping the important features intact?
7. You successfully trained a Lobe model to distinguish between “Pens” and “Keys.” If you place a “Coffee Mug” in front of the webcam during the testing phase, how will the AI likely respond, and what limitation of machine learning does this demonstrate?
8. A wildlife reserve wants to use drone cameras to count the exact number of elephants in a herd from the sky. Should they use Image Classification, Object Detection, or Instance Segmentation? Justify your choice.



ANSWERS TO INTEXT QUESTIONS

12.1

1. (b) Black
2. (b) Object Detection
3. (b) A Kernel (or Filter)
4. (c) Pooling Layer
5. True
6. (c) Red, Green, Blue
7. (a) Intel® Neural Compute Stick 2 (NCS2)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- differentiate CV Tasks i.e. Distinguish between Classification (naming an object), Object Detection (locating it with a bounding box), and Segmentation (outlining its shape)
- explain the basic workflow of a Convolutional Neural Network (CNN).



13

RESPONSIBLE AND ETHICAL AI

Artificial Intelligence (AI) is no longer a futuristic concept; it determines who gets a loan, who gets hired, how long a prison sentence might be and what news you see on social media. As AI systems become more autonomous and integrated into society, the question shifts from “What *can* AI do?” to “What *should* AI do?”

This lesson explores the critical landscape of AI Ethics. It moves beyond science fiction fears (like robots taking over the world) to address real, immediate risks: privacy erosion, algorithmic bias, lack of transparency and accountability gaps. By the end of this lesson, you will not just understand these risks but learn the frameworks used to mitigate them.



OBJECTIVES

After going through this lesson, you will be able to:

- define the core concepts of AI ethics including privacy, bias, and fairness;
- analyze real-world case studies to identify where AI systems failed ethically;
- apply Responsible AI principles to hypothetical scenarios;
- evaluate the trade-offs between model accuracy, privacy, and explainability.

13.1 CORE CONCEPTS OF AI ETHICS

13.1.1 Part I: The Privacy Paradox

AI runs on data. Often, that data is about us.



Notes

1. Data Collection and Surveillance

Modern AI, particularly Deep Learning, requires massive datasets. This creates an incentive for “surveillance capitalism,” where user behavior is constantly tracked.

Informed Consent: Traditional “Terms of Service” are often too long and complex. True ethical AI requires that users genuinely understand what data is taken and how it is used.

Inference Risks: AI can infer sensitive data from non-sensitive data. For example, an AI might infer your political orientation based solely on your movie preferences, even if you never explicitly stated your politics.

2. Anonymization vs. De-identification

Many companies claim data is “anonymized” because they removed your name. However, AI is excellent at pattern matching.

Re-identification attacks: Researchers have proven that by combining “anonymized” Netflix viewing history with public IMDb ratings, they could identify specific individuals.

Differential Privacy: This is a responsible AI practice where statistical “noise” is added to data so that AI can learn patterns (e.g., “smoking causes cancer”) without learning about specific individuals (e.g., “John Doe smokes”).

13.1.2 Part II: Bias and Fairness

If you teach an AI using history books, it will learn the prejudices of history.

1. Sources of Bias

Bias isn’t usually evil programming; it is usually a data artifact.

Historical Bias: The data reflects past inequalities. If a company historically hired mostly men, an AI trained on past resumes will learn to downgrade resumes containing the word “Women’s Chess Club.”

Representation Bias: The dataset is not diverse. If a facial recognition system is trained 80% on light-skinned faces, it will perform poorly on dark-skinned faces.

Proxy Variables: Even if you remove “Race” as a category, the AI might find a proxy, such as “Zip Code,” which highly correlates with race in segregated cities.

2. Defining Fairness

“Fairness” is mathematically difficult to define.



- Individual Fairness: Similar individuals should be treated similarly.
- Group Fairness: The system's error rates should be equal across different demographic groups.
- The Trade-off: Often, you cannot maximize accuracy and fairness simultaneously. You must choose what matters more for the specific context.

13.1.3 Part III: Responsible AI Principles & Governance

How do we fix this? We move from “Ethics” (philosophy) to “Responsible AI” (engineering and policy).

1. Core Principles

Most major tech companies and government institutions have converged on these principles:

- **Transparency & Explainability (XAI):** You should know *why* the AI made a decision. “The computer said so” is not an acceptable answer for denying a mortgage.
- **Accountability:** There must be a human in the loop. If the AI crashes a car or ruins a credit score, who is responsible? The developer? The user? The algorithm?
- **Safety & Security:** AI must be robust. It shouldn't be easily tricked (adversarial attacks) or behave unpredictably in new situations.
- **Human-Centric:** AI should augment human capabilities, not replace human dignity.

2. The Black Box Problem

Deep Neural Networks are “Black Boxes.” They digest data and spit out answers, but the internal math is so complex that even the engineers often don't know exactly which feature triggered the decision.

- **LIME and SHAP:** These are tools used in Responsible AI to highlight which parts of an input (e.g., a specific section of an image or text) contributed most to the AI's output.

13.2 ETHICAL FRAMEWORKS FOR AI

13.2.1 Frameworks

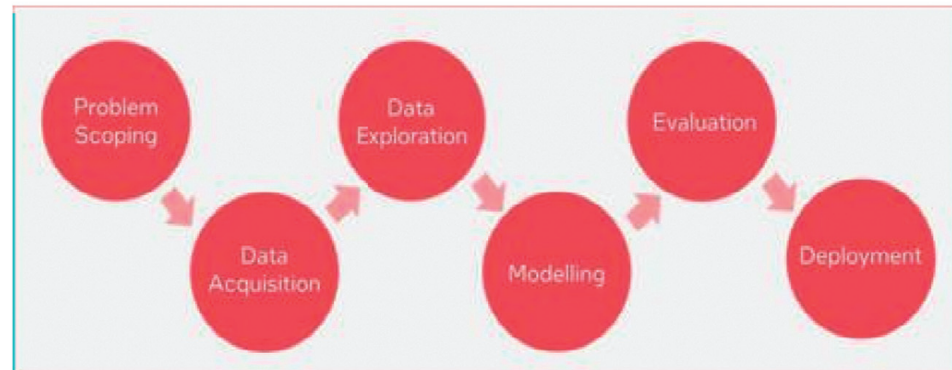
Frameworks are a set of steps that help us in solving problems. It provides a step-by-step guide for solving problems in an organized manner. Moreover, frameworks



Notes

offer a structured approach to problem-solving, ensuring that all relevant factors and considerations are taken into account. Additionally, they serve as a common language for communication and collaboration, facilitating the sharing of best practices and promoting consistency in problem-solving methodologies.

You may have used frameworks without knowing it! Can you think of one framework you have come across during your AI journey?



13.2.2 Ethical Frameworks

We know that ethics are a set of values or morals which help us separate right from wrong. Frameworks are step-by-step guidance on solving problems.

Hence, Ethical frameworks are frameworks which help us ensure that the choices we make do not cause unintended harm.

Furthermore, ethical frameworks provide a systematic approach to navigating complex moral dilemmas by considering various ethical principles and perspectives. By utilizing ethical frameworks, individuals and organizations can make well-informed decisions that align with their values and promote positive outcomes for all stakeholders involved.

13.2.3 Why do we need Ethical Frameworks for AI?

As we have seen how bias could result in unwanted outcomes in AI solutions. Think of the hiring algorithm which was biased against women applicants!

AI is essentially being used as a decision-making/influencing tool. As such we need to ensure that AI makes morally acceptable recommendations

Ethical frameworks ensure that AI makes morally acceptable choices. If we use ethical frameworks while building our AI solutions, we can avoid unintended outcomes, even before they take place!



So far, we have understood why ethical frameworks are needed. Let's try to guess what such a framework would be made of!

Activity: 1 My Goodness

Visit the website – <https://www.my-goodness.net/>

Take a quick look at the video to learn more about the game!

Players must make 10 decisions on how they would like to make a charitable donation. In most cases, players will receive details about the recipients. They will also be given information on the intended use of the money they are donating. In a few instances, this may be hidden from them, however they can choose to reveal it.

This activity aims to understand an individual's judgment. We are looking to discover potential biases within us!

Data is collected anonymously and with your consent.

If you wish to delve deeper and explore your own decision-making, click on "yes."

After clicking on "yes" you will be taken to a short survey.

After the survey has been filled, you will see interesting insights about your decisions

Did you discover any internal biases in your decisions?

Do you agree with the results shared by the game?

Factors which could influence your decisions without your realizing it include

- Identity of the charity recipient.
- Location of the recipient.
- Bias towards relatives.
- Uncovering information available.

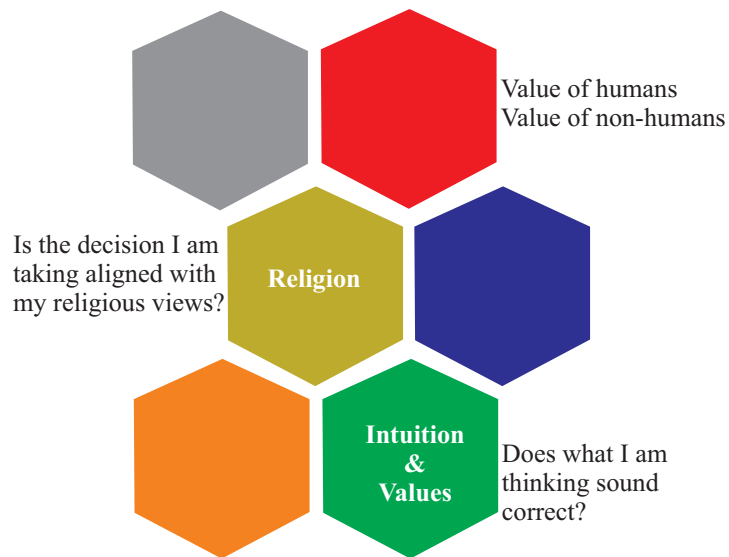
This is just an exercise to uncover our biases and thought processes behind making certain decisions. This will help us in producing a framework which can aid in making decisions which are ethically sounder.

Let's list at least **3 factors** which knowingly or unknowingly influence our decision-making.



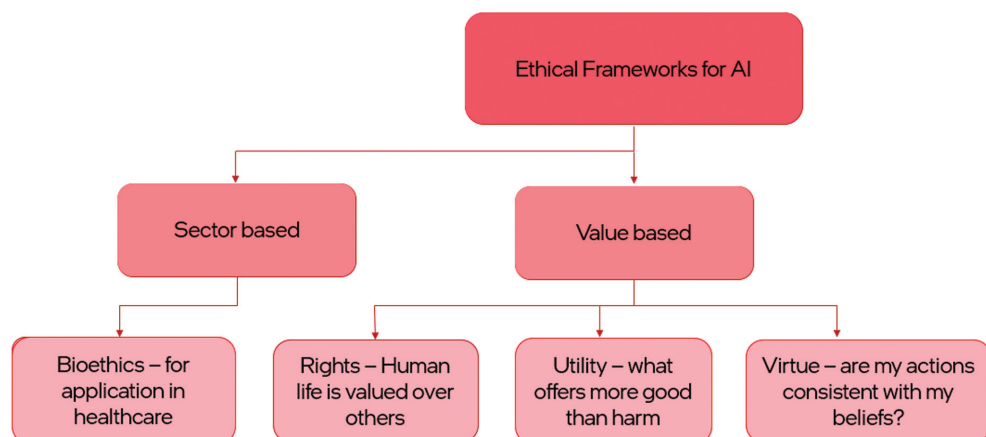
Notes

Responsible and Ethical AI



13.2.4 Types of Ethical Frameworks

The various types of ethical frameworks are classified as follows:



Let's delve into the classifications of ethical frameworks.

Ethical frameworks for AI can be categorized into **two main** types: sector-based and value-based frameworks.

1. Sector-based Frameworks:

These are frameworks tailored to specific sectors or industries. In the context of AI, one common sector-based framework is Bioethics, which focuses on ethical considerations in healthcare. It addresses issues such as patient privacy, data

security, and the ethical use of AI in medical decision-making. Sector-based ethical frameworks may also apply to domains such as finance, education, transportation, agriculture, governance, and law enforcement.

2. Value-based Frameworks:

Value-based frameworks focus on fundamental ethical principles and values guiding decision-making. It reflects the different moral philosophies that inform ethical reasoning. Value-based frameworks are concerned with assessing the moral worth of actions and guiding ethical behaviour. They can be further classified into three categories:

- (i) **Rights-based:** Prioritizes the protection of human rights and dignity, valuing human life over other considerations. It emphasizes the importance of respecting individual autonomy, dignity, and freedoms. In the context of AI, this could involve ensuring that AI systems do not violate human rights or discriminate against certain groups.
- (ii) **Utility-based:** Evaluates actions based on the principle of maximizing utility or overall good, aiming to achieve outcomes that offer the greatest benefit and minimize harm. It seeks to maximize overall utility or benefit for the greatest number of people. In AI, this might involve weighing the potential benefits of AI applications against the risks they pose to society, such as job displacement or privacy concerns.
- (iii) **Virtue-based:** This framework focuses on the character and intentions of the individuals involved in decision-making. It asks whether the actions of individuals or organizations align with virtuous principles such as honesty, compassion, and integrity. In the context of AI, virtue ethics could involve considering whether developers, users, and regulators uphold ethical values throughout the AI lifecycle.

These classifications provide a structured approach for addressing ethical concerns in AI development and deployment, ensuring that considerations relevant to specific sectors and fundamental ethical values are adequately addressed.

Let's explore a popular framework which is used in the healthcare industry.

13.3 BIOETHICS

Bioethics is an ethical framework used in healthcare and life sciences. It deals with ethical issues related to health, medicine, and biological sciences, ensuring that AI applications in healthcare adhere to ethical standards and considerations.



Notes



Notes

Principles of bioethics:

- Respect for Autonomy.
- Do not harm.
- Ensure maximum benefit for all.
- Give justice.

Let's look at a case study and see the impact of the application of an AI ethical framework on the end outcome.



CASE STUDY

A company aimed to support hospitals in optimizing patient care by creating an AI algorithm designed to identify individuals at high risk. The objective was to provide healthcare providers with valuable insights to allocate resources effectively and ensure those most in need receive appropriate attention. However, potential unintended consequences lead to some problems in the model, such as the algorithm inadvertently exacerbating existing biases or inaccuracies in the data, potentially leading to misclassification of patients or overlooking critical cases. Addressing concerns about the algorithm's accuracy and reliability becomes paramount, as any flaws in its design or training data could compromise patient care and outcomes



The problem it caused:

Patients from the Western region of a particular area, who were categorized at the same risk level by the algorithm, generally exhibited more severe health conditions compared to patients from other regions.

Why the problem happened:

- The algorithm utilized was trained on healthcare expense data as a measure for health metrics rather than actual physical illness.
- This algorithm was created in the United States where less money is spent on western region patient healthcare than other ethnic patient healthcare.

We can use principles of the Bioethics framework to ensure an ethical AI solution.

The **four principles** of bioethics can be used to ensure an ethical AI solution for the healthcare problem.

- (i) **Respect for autonomy:** Enabling users to be fully aware of decision-making. e.g., users of an AI algorithm should know how it functions.

- The data that models were trained on and used to make decisions should be reproducible and accessible to the patients.
- In the event of performance concerns, model predictions and data labels should be released.



- (ii) **Do not harm:** Harm to anyone (be it human or non-human) must be avoided at all costs. If no choice is available, the path of least harm must always be chosen.

- Promote well-being, minimize harm, and ensure that benefits and harms are distributed among stakeholders in a just manner.
- The AI algorithm must be trained on data sets that equitably reduce harm for all, not just harm for some groups.
- In this instance, patients from other regions excluding western part who were less ill would receive more intensive care from doctors than patients who actually require help.
- This algorithm, if implemented, would actively harm patients belonging to western region by inappropriately recommending healthcare resource allocation.



- (iii) **Maximum benefit:** Not only should we avoid harm, our actions must focus on providing the maximum benefit possible.

- The solution should be held to clinical practice standards, not merely technological ethics standards.



Notes



Notes

Responsible and Ethical AI

- It should go beyond nonmaleficence and strive for beneficence.
- Considering the example we discussed, the AI algorithm should not only avoid causing harm to patients from the western region but also provide benefits to these patients, as well as patients from other regions and of all races.
- Is there a better data set for training that reflects the healthcare needs and outcomes of patients of all races?
- The data we use for training must be unbiased.



(iv) **Justice:** All benefits and burdens of a particular choice must be distributed in a justified manner across people irrespective of their background.

- Solution development requires concerted and in-depth knowledge of social structures at play that result in issues like racism and sexism (a few types of societal biases).
- The solution needs to be aware of social determinants of healthcare and actively work against those structures.



We saw that abiding by bioethical principles could have helped us to avoid the unintended consequences of the AI solution.



INTEXT QUESTIONS 13.1

1. A bank uses an AI that inadvertently uses “zip code” to deny loans to minority communities, even though “race” was removed from the dataset. This is an example of:
 - (a) Algorithm Overfitting
 - (b) Proxy Variable Bias
 - (c) Explainable AI
 - (d) Data Anonymization

Responsible and Ethical AI

2. A hospital wants to use patient records to train an AI to detect cancer. They remove names and social security numbers. Is this data now perfectly safe? Why or why not?
3. An AI screens loan applicants. It has a 95% accuracy rate overall. However, it approves loans for qualified men 90% of the time, but for qualified women only 60% of the time. Is this model “fair”? Which type of bias might be present?
4. Why is “Explainability” more important for an AI diagnosing diseases than for an AI recommending songs on Spotify?



WHAT HAVE YOU LEARNT

- **AI is not neutral.** It reflects the data it is fed and the values of its creators.
- **Privacy** is more than just hiding names; it involves protecting against re-identification and inference.
- **Bias** can enter through historical data, lack of representation, or proxy variables.
- **Fairness** is a mathematical and social choice that must be deliberately designed into the system.
- **Responsible AI** requires transparency (XAI), accountability (Human-in-the-loop), and safety.



TERMINAL EXERCISE

1. Explain the concept of “Human-in-the-loop” and give one example of a high-stakes industry where this is mandatory.
2. You are building a facial recognition system for airport security. You discover the system is 99% accurate for men but only 85% accurate for women. The launch deadline is tomorrow. Using Responsible AI principles, what is the correct course of action?



Notes



Notes



ANSWERS TO INTEXT QUESTIONS

13.1

1. (b) (Proxy Variable Bias)
2. No, the data is not perfectly safe. This is “de-identification,” not true anonymization. Through **re-identification attacks**, hackers could combine this dataset with other public data (like voter registration or social media) to figure out who the patients are.
3. The model is likely **not fair** regarding Group Fairness (specifically, “Demographic Parity” or “Equal Opportunity”). Even if the overall accuracy is high, the discrepancy between men (90%) and women (60%) suggests **Historical Bias** (women might have been given fewer loans in the past data) or **Representation Bias** (the model learned mostly from male financial histories).
4. The **stakes** are different. If Spotify recommends a bad song, the cost is a few minutes of annoyance. If a medical AI misdiagnoses a disease because of a “black box” error, a patient could die or undergo unnecessary surgery. High-stakes decisions require high **Explainability** to build trust and allow doctors to verify the AI’s logic.

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- gain a robust understanding of AI ethics and the societal issues surrounding automated decision-making.
- use the ethical principles (Safety, Accountability, Fairness, Privacy) to design and advocate for safer AI systems.



Notes

14

EMPLOYABILITY AND FUTURE SKILLS

In today's competitive job market, only technical expertise is not sufficient for any employee.. Equally important are the employability skills that help individuals perform well in their professional journeys. These skills, including effective communication, IT proficiency, 21st-century competencies, work ethics, and traits like adaptability and dependability, are essential for navigating the modern workplace. Employability skills provide the means through which individuals effectively interact with colleagues, customers, and other stakeholders to foster good relationships that would help in an organization's success. They are as important as technical skills in determining job performance and career growth.

Additionally, self-management skills, including flexibility and responsibility, are critical to adapt to the changing work environment. These skills, besides strengthening the growth and reputation of companies, pave the way for entrepreneurial ventures.

This lesson will focus on the significance of employability skills and how they contribute to the success of the individual and organization as well.



OBJECTIVES

After learning this lesson, you will be able to:

- describe importance of employability skills;
- describe 21st-century skills and their importance;
- illustrate verbal and nonverbal communication skills.



Notes

14.1 EMPLOYABILITY SKILLS

An employee is a person who works for a salary. Employability skills are imperative skills which ensure one performs better in any workplace. These consist of both hard and soft skills, and these consist of key personal attributes much prized by employers. Employability skills are needed if one is to deliver effectively on the job and to continue resilient and effective in their careers.

Hard skills, or technical skills, are learned at school, in training, or through on-the-job experience. They are specific to a job and can be measured and taught.

Soft skills, being the most important skills, are generalized traits that can be applied in any job and any industry. The transferable or interpersonal skills such as communication, teamwork, attitude, and adaptability have been known as soft skills. Soft skills are essential to the success of professionals because they allow employees to thrive in any environment.

Employers not only seek technically skilled individuals but also value those who can make a broader contribution to the organization. Organizations look for people with the following qualities:

1. **Good oral and written communication skills:** Good oral and written communication are needed to communicate effectively with people and customers. Oral communication aims to effectively convey information through clear and concise vocal words. It fosters interaction, relationship building and facilitates collaboration among individuals.

Written communication relies on written words whereas oral communication involves direct interaction between individuals through spoken words. These skills enable employees to articulate ideas, collaborate with colleagues and present their thoughts convincingly to employers or clients. Written communication skills are crucial for crafting professional documents such as emails, reports, presentations and cover letters.

2. **Leadership Qualities:** It is said that Leaders are not born but they are made. Leaders provide direction, vision, motivate and inspire others and help in creating a conducive environment. Leadership skills are highly sought after by employers as they involve dealing with people, taking responsibility and initiative.

McKinsey is an American multinational strategy and management consulting firm which is the oldest and largest of the “Big Three” management consultancies has reported the analysis of academic literature as well as a



survey of nearly 200,000 people in 81 organizations all over the world, there are four types of behaviour that account for 89 percent of leadership effectiveness:

- being supportive z operating with a strong results orientation
- seeking different perspectives
- solving problems effectively

3. **Critical thinking and Problem solving:** Problems in any organization range from the simplest to the most complex and should be approached constructively and equitably. To this end, critical thinking and problem-solving individuals are indispensable. They analyze a situation, apply knowledge, and make appropriate decisions. An organization needs people who can take up problems based on facts, data, and knowledge.

Organizations would want those people who can deal with issues independently or make meaningful contributions as a team member. They don't need individuals who can take the initiative and provide solutions. The perfect employee would be able to critically think and come up with unique ideas, communicate knowledge, make proper judgments, and make wise decisions. Critical thinking would therefore entail making sense of available information, asking questions, and developing reasonable conclusions for solving problems better and making informed decisions in the workplace.

4. **Positive attitude:** "You can often change your circumstances by changing your attitude." –Eleanor Roosevelt

A positive attitude towards work can bring positive changes for you in an organization. People with a positive attitude are focussed towards work, support their colleagues, they work in collaboration and are compassionate. People with a positive attitude never complain but are always with a solution.

5. **Time Management and Organizational Skills:** Good time management and keeping organized are some of the most significant things employers seek in employees. It lets one be able to meet deadlines, balance priorities, and reduce stress. For example, a project manager with several tasks will have to organize their schedule, allocate time for each task, and avoid procrastination. This ability ensures that projects are completed efficiently and deadlines are met without compromising quality.

Organization, then, is not only time management but also a clean working environment, structured tasks and responsibilities, and the right systems for



Notes

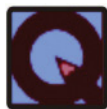
tracking progress. Be it a meeting, project timeline, or even emails, employees who are organized can make workflow smoother and help prevent bottlenecks in processes.

6. **Teamwork and Collaboration:** Employees who can work well with others contribute to a positive and productive work culture. Teamwork involves more than working with others. It requires understanding the dynamics of the group, respecting diverse perspectives, and working towards a shared goal. For instance, in a marketing campaign, a team that collaborates well can brainstorm ideas, divide tasks, and support one another in delivering the project successfully.

Collaboration is communication and compromise. When the ideas are shared and people listen to one another, there is a higher likelihood of solving any conflicts that might arise and even strengthens working relationships. Employers hire people who work well with other diverse teams, manage group dynamics effectively, and contribute to a collective goal.

7. **Adaptability and Flexibility:** Adaptability and flexibility are the most important employability skills in the fast-paced work environment. Organizations often change in terms of technology, market conditions, or business goals. Therefore, they expect employees to adjust accordingly. For example, in the pandemic era of COVID-19, employees had to rapidly shift from being in a work environment to being at home. Those employees flexible enough managed to adapt to this transition by setting up a home office and learning some communication tools with minimal disruption.

Being adaptable means openness to new ideas and changes in the workplace. It involves continuous learning of new skills and keeping updated in the modern trends of the industry and embracing change as an opportunity for growth. In dynamic work environments, the most valuable assets are those employees who can adapt to a given situation and remain productive.



INTEXT QUESTIONS 14.1

I. Fill in the blanks:

1. Rohan has completed his graduation in economics, he has developed his skills.
2. and are needed to communicate effectively with people and customers.



3. is the process of understanding information available, seeking answers and coming to a justified conclusion.
4. provide direction, vision, motivate and inspire others, and help in creating a conducive environment.
5. A is someone who works for a salary.

II. Multiple choice questions:

1. Employability skills are
 - (a) Skills for good communication
 - (b) Skill to be a good employee
 - (c) Interpersonal Skill
 - (d) None of the above
2. Which of the following is not an employability skill
 - (a) Excellent communication skill
 - (b) Positive attitude
 - (c) Sales and marketing
 - (d) Leadership qualities
3. Which are the skills needed for career growth
 - (a) Technical skills
 - (b) Soft skills
 - (c) Communication skills
 - (d) All of the above
4. An employee is someone
 - (a) Who is hired for work
 - (b) Who will paid for the job
 - (c) Both a and b
 - (d) None of the above
5. Meaning of employability skill is
 - (a) IT skills
 - (b) Skills for good communication
 - (c) To make successful at workplace
 - (d) None of the above



Notes

14.2 21ST-CENTURY SKILLS AND THEIR IMPORTANCE

The entire world has witnessed an era of intense transformation in digital technology, global trade, science, economy, education and innovations. The development of a society, a nation and the world go hand-in-hand with the development of an individual. When a society, nation or world develops, this has a direct impact upon the development of an individual and vice-versa.

21st Century skills: 21st Century Skills are the most essential skill in today's time. These skills are important for better professional, work habits, and character traits that are believed to be critically important to succeed in today's world. They are required to enable an individual to face the challenges of the 21st century world that is globally-active, digitally transforming, collaboratively moving forward and creatively progressing.

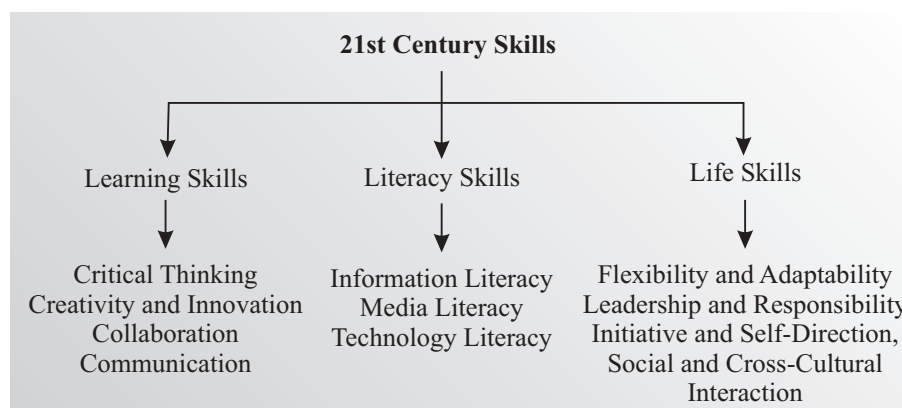


Fig. 14.1: 21st Century Skills

Learning Skills: The most popular 4 C's are part of 21st Century skills.

- **Critical thinking:** Critical thinking is when you find solutions to problems. In critical thinking you question, analyze, interpret, evaluate and come to a solution on your own.
- **Creativity:** Creativity is generating new solutions or ideas of a problem. It is thinking out of the box
- **Collaboration:** Collaboration is working in a team. Collaboration increases productivity and efficiency of work.
- **Communication:** Communication is a way of expressing our thoughts, beliefs. Effective communication is sharing our thoughts, ideas to others and in return get positive feedback or results.



Literacy Skills

- **Information Literacy:** With millions of websites today, there is a lot of information on the internet. Information literacy is the ability to locate, evaluate, use and communicate information efficiently and effectively. Information literate people use information for their learning and growth.
- **Media Literacy:** There are various forms of media through which we get information about the world. Different forms of media are print media, internet media and broadcast media. Media literacy is helpful in assessing and evaluating the information.
- **Technology skills:** This is the most relevant skill of today's time. A person should know the basics of computer, word processing, presentation and spreadsheets software. They should know how to search for appropriate information and should be able to work effectively with emails etc. There should be awareness on do's and don'ts of using social media.

Life Skills

- **Flexibility & adaptability:** Adaptability is the capacity to adjust, evolve and succeed in response to changes at the workplace. The faster you adapt yourself at the workplace, the faster your growth starts. You need to understand the working culture and environment. Flexibility is how you change in certain conditions or circumstances at any given point of time. It is how you handle problematic situations, and come out of it with proper solutions.
- **Leadership and responsibility:** Leader is someone who guides, influences, motivates others for the common purpose. Leader takes the responsibility for actions and directions of a team. If a certain assignment was successful, a good leader credits his team. But, he takes the ownership, if in any case there was failure in delivering the assignment.
- **Initiative and self direction:** Initiative is taking charge to solve a problem. It is a quality of an individual to achieve the goals and take the responsibility. Self direction is your inner desire to make your own decisions and organize your own work. Self-direction is achieving the goal with managing time and effort.
- **Social and cross cultural Interaction:** There will be people from different backgrounds and countries. It is the company's effort to ensure that people can work in a diverse environment. Cross cultural interactions open doors for new businesses and job opportunities worldwide.



Notes



INTEXT QUESTIONS 14.2

I. Fill in the blanks:

1. Information literacy, media literacy and technology literacy are skills.
2. Working together in a team is
3. Taking charge to solve a problem is taking
4. Generating ideas or new solutions to a problem is
5. Different forms of media are and

II. Multiple choice questions:

1. The 4Cs of 21st century refers to
 - (a) Communication
 - (b) Collaboration
 - (c) Communication, Critical thinking,
 - (d) Creativity
2. People from different backgrounds and countries work together in unison is
 - (a) Diverse environment
 - (b) Social and cross cultural interaction
 - (c) Social Interaction
 - (d) Cultural Interaction
3. Flexibility is how you
 - (a) Change in certain circumstances or conditions
 - (b) Handle problematic situation
 - (c) Both a and b
 - (d) None of the above
4. 21st century skills are classified into
 - (a) Learning skill
 - (b) Life skill
 - (c) Literacy skill
 - (d) All of the above
5. A person who guides or motivates its employees is called
 - (a) Leader
 - (b) Manager
 - (c) Boss
 - (d) Colleague



14.3 COMMUNICATION SKILLS

Communication is essential in life. It helps us to connect with others. Just as we need basic facilities such as food, water, shelter, and other basic resources to survive, we need the communication process to succeed in life. We use it to build relationships, share information, and resolve conflicts. Communication is more than just a way to express our thoughts and feelings. It helps us to learn things better. Communication is very important in the workplace or in business. It is very essential to communicate with colleagues, customers and other people we meet.

There are 5 elements of communication:

- **Sender:** Sender is one who initiates the communication process
- **Receiver:** Receiver is who receives the message from the sender.
- **Message:** Message is what is to be communicated.
- **Channel:** Channel is a medium of communication.
- **Feedback:** Response of receiver to the sender.

We communicate with each other by speaking, writing, facial expressions, body language, gestures, using symbols etc. There are two types of communication: verbal and nonverbal.

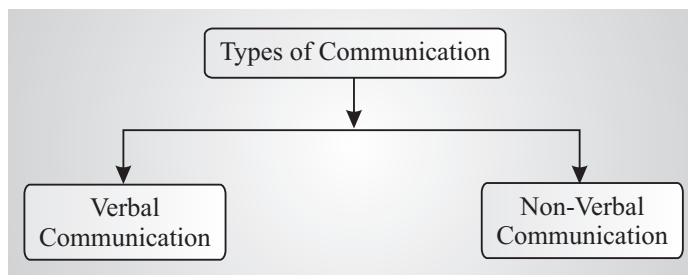


Fig. 14.2: Types of Communication

Verbal Communication: The term ‘verbal’ means by using words and language. Verbal communication is sharing our ideas, information, suggestions, thoughts and feelings either in oral or written.

- **Oral:** To communicate orally one should possess good command in language. There should be clarity in voice and a person should be able to express ideas clearly and effectively. Oral communication is of two types:
 - **Speaking:** Face-to-face conversations in meetings, group discussions, seminars, presentations, video conferencing, telephonic conversations etc.



Notes

- **Listening:** Active listening can help you to understand what others are saying in conversations and meetings. Look at the speaker and nod or smile that you are listening. Evaluate the conversation and if required ask questions.

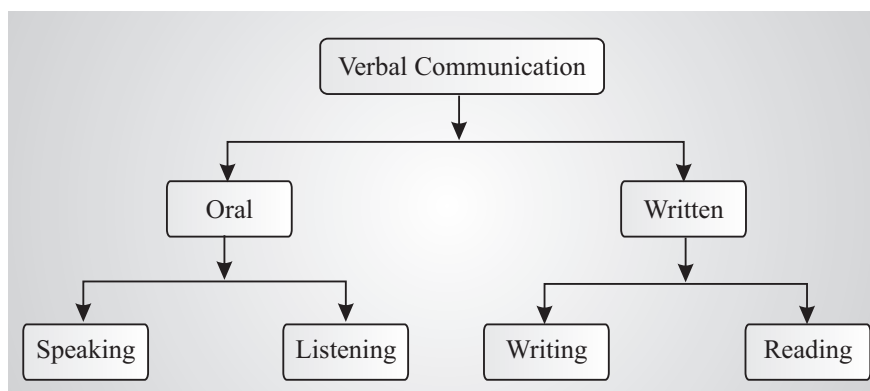


Fig. 14.3: Verbal Communication

- **Written:** Written communication is an important aspect of today's communication. Exchange of information can be either printed or handwritten. Well maintained written records are permanent evidence for the future and at any point of time for reference. It promotes uniformity in policies and procedures.
 - **Writing:** Written communication can be done through letters, emails, messages, contracts, forms, quotations etc.
 - **Reading:** Reading is a process of constructing meaning from the written text either printed or through digital media. It can be in the form of books, magazines, graphs, charts, websites, emails etc.

Non-Verbal Communication: Non verbal communication is communicating without using words or language but with body language, gestures, eye contact, facial expression etc. To communicate effectively using Non-Verbal communication, one should:

- | | |
|------------------------------|-----------------------|
| ● Maintain eye contact | ● Modulate voice tone |
| ● Maintain Proper distance | ● Smile when required |
| ● Maintain good body posture | ● Physical appearance |



INTEXT QUESTIONS 14.3



Notes

I. Fill in the blanks:

1. Body language is type of communication.
2. A person who initiates the communication process is
3. is just a way to express our thoughts and feelings.
4. It is preferable not to interrupt someone while
5. Email is a type of communication.

II. Multiple choice questions

1. The process of exchanging information between individual or a group is

(a) Transmission	(b) Communication
(c) Formal communication	(d) Informal communication
2. Which of the following is an example of non-verbal communication

(a) Speaking	(b) Listening
(c) Reading	(d) Body Posture
3. Active listening is

(a) Paying attention to the speaker	(b) Asking questions
(c) Using mobile phone	(d) Avoiding eye contact
4. Written communication is

(a) Communication over video call	(b) Sending an email
(c) Attending meeting	(d) Telephonic conversation
5. Which communication skill requires a voice tone and clarity

(a) Listening skill	(b) Reading skill
(c) Verbal communication	(d) Non-verbal communication

14.4 ESSENTIAL DIGITAL SKILLS

Nowadays, people in a wide range of fields and professions find it increasingly necessary to have specific digital abilities. The following are some fundamental digital abilities that are helpful in many situations:



Notes

Employability and Future Skills

- **Basic computer literacy:** Basic computer literacy refers to the ability to use computers, navigate an operating system, use a web browser, organise files and directories and perform simple troubleshooting.
- **Internet navigation and search:** Internet navigation and search are essential for discovering pertinent information and making the most of online resources. These skills include how to conduct efficient online searches, assess the reliability of sources, and move around websites and web pages.
- **Email communication:** Email communication is still the main way that people communicate both personally and professionally. Effective communication requires a variety of abilities, including the ability to write and send emails, manage email accounts, organise folders, and comprehend email etiquette.
- **Digital Security and Privacy:** securing digital identities and preserving privacy online requires an understanding of the fundamentals of online security, such as password management, spotting phishing scams, and securing personal information.
- **Online collaboration and communication tools:** Effective teamwork and distant collaboration need the use of digital collaboration tools, including as video conferencing platforms, project management software, shared document editing tools, and instant messaging programmes.
- **Digital research and information evaluation:** It is crucial for academic endeavours, professional research, and decision-making to be able to quickly obtain information from digital sources, critically assess its veracity and credibility and synthesise pertinent data.
- **Digital creativity and content creation:** This skill can improve one's capacity to articulate ideas, captivate audiences, and effectively communicate in the digital sphere. These skills include graphic design, photo and video editing, audio production, and basic coding.
- **Data literacy and analysis:** As data become more accessible, it is becoming increasingly important to be able to comprehend, examine, and evaluate data sets. Making educated decisions and spotting patterns and trends can be aided by knowledge of data visualisation, data analysis software and fundamental statistical ideas.
- **Social Media Management:** It's helpful for community development, marketing, and personal branding to be familiar with the various social media platforms, understand social media strategies, and effectively engage and interact with audiences.

- **Adaptability and Continuous Learning:** It's critical to have the capacity to adjust to new technologies, platforms, and tools in the quick-changing digital environment. Growing personally and professionally requires adopting an attitude of lifelong learning, remaining current on trends and being open to learning new digital skills.

14.5 ENTREPRENEURSHIP

The process of locating, developing, and pursuing possibilities to launch a new company or endeavour is referred to as entrepreneurship. It involves people—known as entrepreneurs—who take risks and invest money to create novel concepts, goods or services in an effort to create economic value.

Entrepreneurship encompasses several key elements:

- **Innovation and Creativity:** Entrepreneurs frequently identify and launch new concepts, goods or services to fill gaps in the market or address issues already in existence. They use their ingenuity to build original solutions and advertise them.
- **Risk-taking:** Taking measured risks is a key component of entrepreneurship. Even though they are aware that success is not guaranteed, entrepreneurs are eager to put their time, effort, and money into new endeavours. They welcome ambiguity and are equipped to deal with difficulties and setbacks.
- **Opportunity Recognition:** Entrepreneurs have a strong eye for spotting business possibilities. They notice patterns, gaps, and new demands, and they think of methods to take advantage of those possibilities by adding value to their offerings.
- **Business Planning and Execution:** Entrepreneurs frequently identify and launch new concepts, goods, or services to fill gaps in the market or address issues already in existence. They use their ingenuity to build original solutions and advertise them.
- **Resource Management:** Successful businesspeople effectively manage resources including money, people and time. They effectively allocate resources, seek funding or investment as necessary and take well-informed decisions to maximise their business operations.
- **Building Networks and Relationships:** Successful businesspeople recognise the value of developing networks and relationships. To obtain knowledge, access resources, and form cooperative relationships, they interact with mentors, business experts, possible partners and clients.



Notes



Notes

Employability and Future Skills

- **Adaptability and Resilience:** Entrepreneurship necessitates the capacity to adjust to shifting market conditions, technical developments, and consumer expectations. To remain competitive, entrepreneurs must be resilient in the face of difficulties, learn from mistakes, and adapt their plans as necessary.
- **Scaling and growth:** Over time, entrepreneurs want to see their businesses expand. They create plans for growing their business, gaining more clients, and breaking into new markets. To promote growth, this may entail bringing on new personnel, looking for additional funds or putting new technology into place.

14.6 CUSTOMER SERVICE

Customer service refers to the support and assistance provided to customers before, during, and after their interaction with a product, service or organization. It involves addressing customer inquiries, resolving issues or complaints, and ensuring customer satisfaction.

Effective customer service is crucial for building and maintaining strong relationships with customers. It encompasses various elements:

- | | |
|------------------------------|--------------------------------|
| (i) Communication | (ii) Knowledge and expertise |
| (iii) Problem solving | (iv) Empathy and understanding |
| (v) Timeliness | (vi) Personalization |
| (vii) Continuous improvement | (viii) Positive attitude |

14.7 PROCESS OF APPLICATION FOR A JOB AND INTERVIEW PREPARATION

Process of Application for a Job: There are various things you have to keep in mind before applying for a job through newspapers, job portals, links from family and friends etc.

- **Self assessment:** It is a tool to reflect your career progress, achievements, accomplishments, skill development. It helps in identifying your skills, strengths, interest and personal value. This will help in choosing the right job.
- **Preparation of resume:**
 - Choose the best format for your resume.
 - Brief summary about yourself.
 - Add complete personal details.

Employability and Future Skills

- Include all educational details.
- Add professional details
- List the relevant skills
- List the work experience in descending order.
- Highlight the achievements or awards.
- **Drafting a cover letter to a resume:** A cover letter is a document submitted with the job application showing interest in the open position.

Interview Preparation: To walk for an interview one should keep following things in mind.

- Review the job profile.
- Research about the company.
- Prepare the key points about yourself to communicate to the interviewer.
- Dress professionally.
- You should know your strengths.
- Be punctual.
- Enter with a positive attitude and smile and be confident.
- Listen carefully to the interviewer's questions and then answer them.



INTEXT QUESTIONS 14.4

I. Fill in the blanks:

1. The financial that are made available to sectors in need.
2. A customer is likely to buy a product or avail a service.
3. It is not advised to do transactions using public wifi.
4. Right to pay equal remuneration to men and women is
5. One who sets his own business is called an

II. Multiple choice questions:

1. Resume should include
 - (a) Work experience
 - (b) Relevant skill
 - (c) Educational qualification
 - (d) All of the above.



Notes



Notes

Employability and Future Skills

2. A person who does not need any guidance for buying a product is
 - (a) Need based customer
 - (b) Knowledgeable customer
 - (c) Impulsive customer
 - (d) None of the above
3. On finding suspicious mail
 - (a) Open it at once
 - (b) Open the attached file
 - (c) Delete it immediately
 - (d) None of the above
4. Which one is a “Do’s” for interview
 - (a) Nervousness
 - (b) Be confident
 - (c) Dress casually
 - (d) Excessive gesture
5. A person who is associated of starting a business is
 - (a) Executive
 - (b) Manager
 - (c) Entrepreneur
 - (d) Employer



WHAT HAVE YOU LEARNT

- Employability skills are the essential skills that enable you to thrive in any workplace. It is a combination of soft and hard skills.
- Organizations look for people with Good oral and written communication skills, Leadership Qualities, Critical thinkers & Problem solvers, Positive attitude etc.
- 21st Century Skills refer to the skills that are required to enable an individual to face the challenges of the 21st century world.
- 21st century skills are broadly classified into learning Skills, literacy Skills and life Skills.
- Most popular 4 C’s are part of 21st Century skills are Critical thinking, Creativity Collaboration and Communication.
- Communication is of two types, verbal communication and non verbal communication. z Verbal communication is sharing our ideas, information, suggestions, thoughts and feelings either in oral or written. It is of two types: oral and written.

Employability and Future Skills

- Non verbal communication is communicating without using words or language but with body language, gestures, eye contact, facial expression etc.
- Job interview is an opportunity to present your skills, knowledge and experience to the interview panel. Be calm and confident and carry a positive attitude.



Notes



TERMINAL QUESTIONS

1. Why are employability skills important?
2. What suggestions are required to become a professional in 21st century?
3. How to improve communication skill to become a good professional?
4. Role of digital education in professionalism?
5. What is entrepreneurship and its elements?



ANSWERS TO INTEXT QUESTIONS

14.1

- I.**
- | | |
|----------------------|--|
| 1. Hard | 2. Good oral and written communication |
| 3. Critical Thinking | 4. Leader |
| 5. Employee | |
- II.**
- | | | | | |
|--------|--------|--------|--------|--------|
| 1. (b) | 2. (c) | 3. (d) | 4. (c) | 5. (c) |
|--------|--------|--------|--------|--------|

14.2

- I.**
- | | |
|--|------------------|
| 1. Literary skill | 2. Collaboration |
| 3. Initiative | 4. Creativity |
| 5. Print media/ internet media / broadcast media | |
- II.**
- | | | | | |
|--------|--------|--------|--------|--------|
| 1. (d) | 2. (b) | 3. (c) | 4. (d) | 5. (a) |
|--------|--------|--------|--------|--------|

14.3

- I.**
- | | | |
|---------------|------------|------------------|
| 1. non-verbal | 2. sender | 3. communication |
| 4. speaking | 5. Written | |



Notes

Employability and Future Skills

- II.** 1. (c) 2. (d) 3. (a) 4. (b) 5. (c)

14.4

- I.** 1. Services 2. potential 3. online
4. The Equal Remuneration Act 1976
5. Entrepreneur

- II.** 1. (a) 2. (b) 3. (c) 4. (b) 5. (c)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- describe 21st-century skills and their importance.
- illustrate effective communication.
- explain digital literacy.



15

PROJECT CYCLE

A Project is a series of tasks or a group of activities that need to be completed to reach a specific goal under given constraints. These tasks or activities are usually performed with limited resources and time. A Project cycle is a sequence of phases through which a project progresses. The number of phases and sequences of the cycle may vary depending on the type of project.

The students are taken through a use case where they create an AI project to predict future temperature. This use case can be helpful in creating awareness and estimating the level of catastrophe that can be avoided if climate change is combated at the right time.



OBJECTIVES

After reading this lesson, you will be able to:

- define project and Project Cycle;
- use the basic concept in development process of project;
- explain various data processing techniques in project development;
- explain the process of developing AI-enabled social impact solutions.

15.1 THE AI PROJECT CYCLE

The AI project cycle is a structured framework that guides the development of artificial intelligence solutions from start to finish. The AI Project Cycle includes five stages: Problem Scoping, Data Acquisition, Data Exploration, Modeling, Evaluation and Deployment.



Notes

Project Cycle

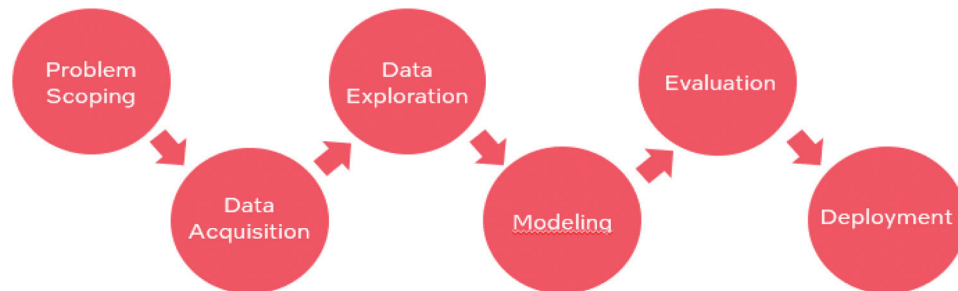


Fig. 15.1: Various Stage in AI Project Cycle

15.1.1 Understanding the Problem

Let's understand the problem solution through a visualization

- This color-coded map displays a progression of changing global surface temperature anomalies
- Normal temperatures are shown in white, higher than normal temperatures are shown in red and lower than normal temperatures are shown in blue
- Temperatures are calculated over a 30-year baseline period 1951-1980

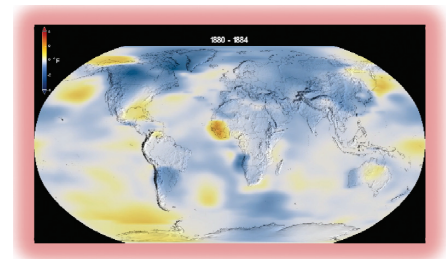


Fig. 15.2: Earth

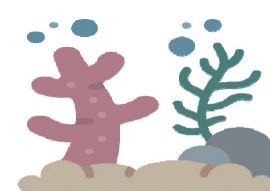
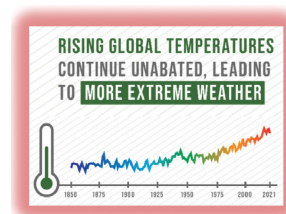
The temperature of the Earth is rising

This effect is called Global Warming

Some effects of global warming



Melting Polar Icecaps



Dying Corals

Fig. 15.3: Temperature on Earth

15.1.2 Let's start with our project!

Predict Future Temperature

Project Cycle

Step 1: Problem Scoping

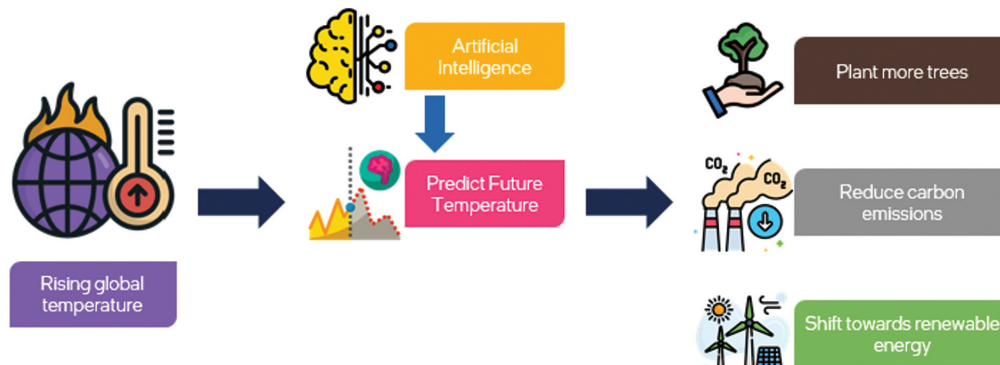


Fig. 15.4: Stage in Problem Scoping

Use Machine Learning (ML) to correctly predict the future temperature of a country and take appropriate measures to control global warming levels

Step 2: Data Acquisition

Step 2(a)

Import the necessary libraries and packages used for the problem solving



This sign will be displayed on the successful run of the lines of code

```
In [2]: # Import libraries
import time
import itertools
import panel as pn
import numpy as np
import pandas as pd

# Visualization libraries
import holoviews as hv
import folium
import seaborn as sns
import matplotlib.pyplot as plt

from functools import partial

from panel.widgets import Tqdm

from tornado import gen
from tornado.ioloop import PeriodicCallback

from folium.plugins import HeatMap, HeatMapWithTime, MarkerCluster

from scipy.linalg import LinAlgError
from scipy import stats
from scipy.stats import gaussian_kde, norm

from bokeh import models, plotting, io
from bokeh.layouts import column
from bokeh.plotting import figure, show
from bokeh.models import ColumnDataSource, RangeTool, BoxAnnotation

from holoviews import opts
from holoviews.streams import Pipe, Buffer
from holoviews.operation.timeseries import rolling, rolling_outlier_std
```

Fig. 15.5: Data Acquisition



Notes



Notes

Step 2(b)

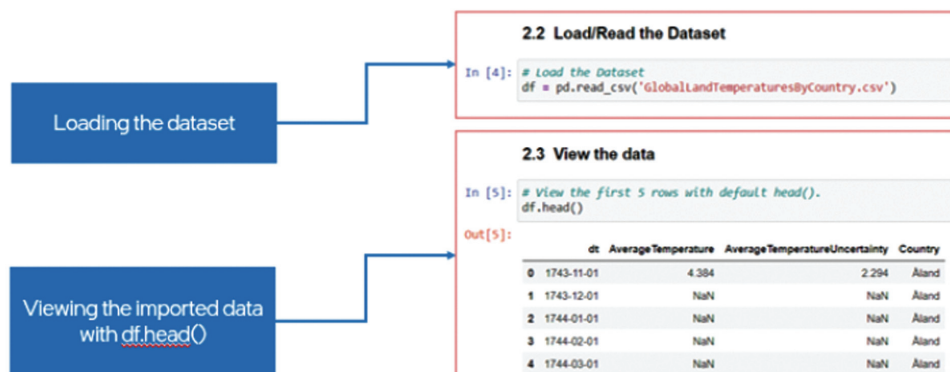


Fig. 15.6: Loading and Viewing the Data

Step 2(c)

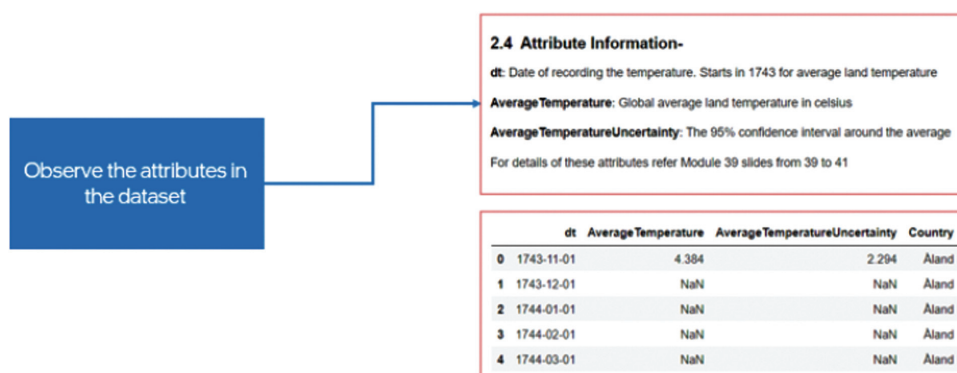


Fig. 15.7: Displaying Attributes in Dataset

15.1.3 Let us Dive into Exploring the Data now!

Step 3: Data Processing

Step 3(a): View the Information

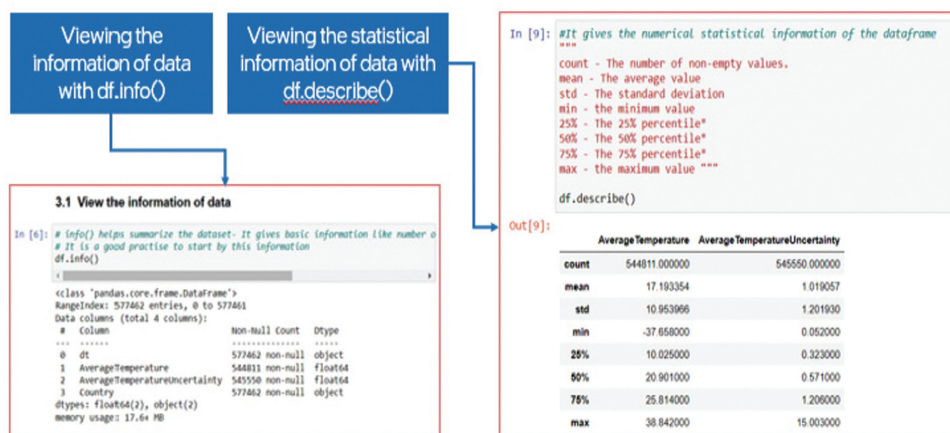


Fig. 15.8: Viewing Statistical Information of Data



Step 3(b): Handling the missing values

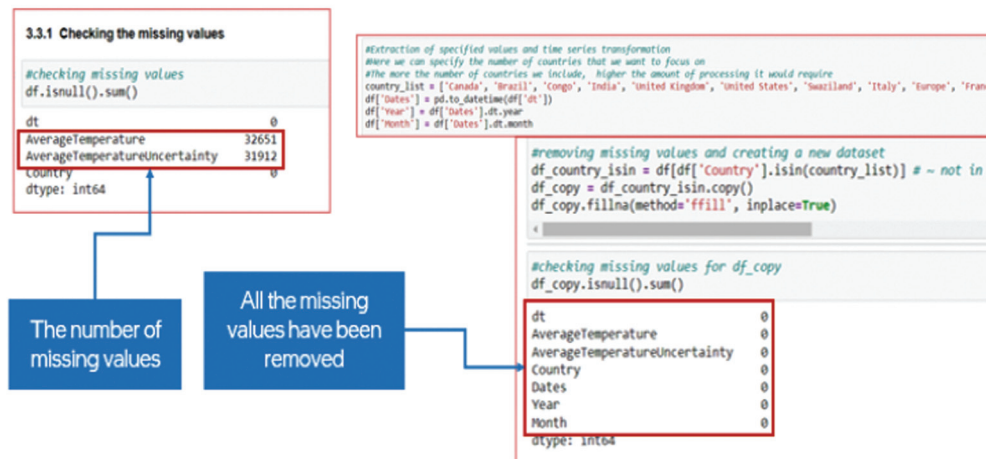
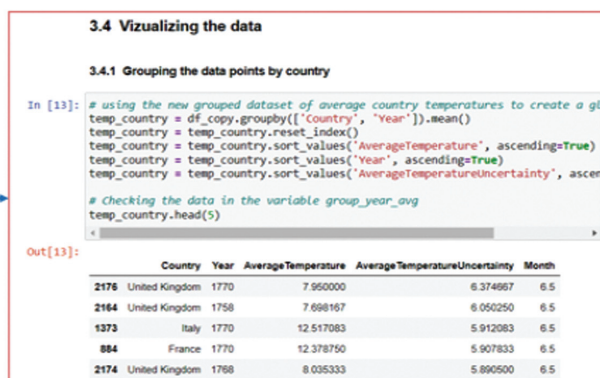


Fig. 15.9: Handling missing Values

Step 3(c): Plotly Globe Visualization

For our visualization we need to group the data by country and year



- With this interactive visualization we can observe the temperatures though the globe
- We can also take a snapshot of a specific part

```
# using plotly to create an interactive globe visualization
fig = px.choropleth(temp_country, locations='Country',
                    locationmode='country names', color='AverageTemperature',
                    color_continuous_scale='magma', hover_name='Country',
                    title='Global Temperature')
fig.update_projection(type='orthographic', showcoast=True, showlakes=True)
fig.update_layout(height=500, margin={'r':0,'t':50,'l':0,'b':0})
fig.show()

# shows lakes in the maps and change the color scale between red to green in t
fig = px.choropleth(temp_country, locations='Country',
                    locationmode='country names', color='AverageTemperature',
                    color_continuous_scale='magma', hover_name='Country',
                    title='Global Temperature with lakes')
fig.update_projection(type='orthographic', showcoast=True, showlakes=True)
fig.update_layout(height=500, margin={'r':0,'t':50,'l':0,'b':0})
fig.show()
```





Notes

Animating the globe visualization

- We can pan or zoom through the globe, and visualize the average temperature in a given year
- We can take a screenshot for any year

```
fig = px.choropleth(temp_country, locations='country',
                    locationmode='country names', color='AverageTemperature',
                    color_continuous_scale=temps, hover_name='country',
                    animation_frame='year', animation_group='country',
                    title = 'Global Temperature', range_color=temp_country['A']
fig.update_geos(projection_type='orthographic', showocean = True, showlakes=False)
fig.update_layout(height=700, margins={'r':0,'t':0,'l':0,'b':0})
fig.show()
```



Fig. 15.10: Plotly Globe Visualization

Step 3(d): Holoviews Visualization (Pythan Library) – Visualization 1

Holoviews is a high-level open-source Python library. It lets you build interactive, data-rich visualizations with just a few lines of code.

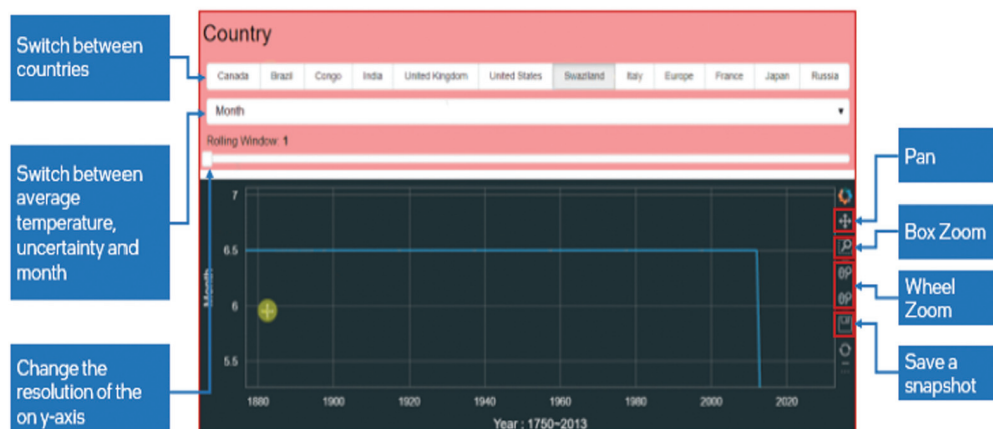


Fig. 15.11: Holoviews visualization

Step 3(e): Bar Visualization (Visualization 2)

- Bar visualization of temperature of different countries in different years
- Visualizations like can be used to compare different countries in terms of temperature

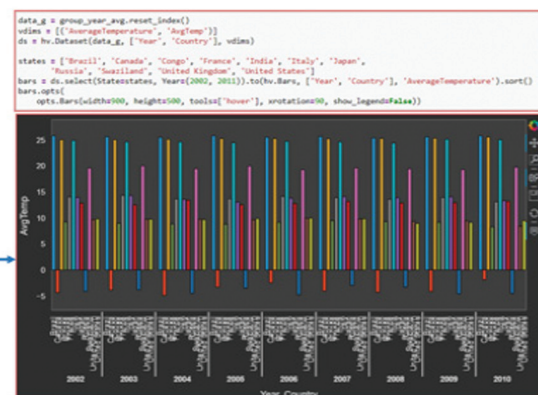


Fig. 15.12: Bar visualization



Step 3(f): Visualization 3

- A heat map that shows temperature in a country over the years
- Observe that the temperature is higher in the 5, 6, 7, and 8 months which is the time of summer

```
uk_data = df_copy[df_copy['country'] == 'United Kingdom']
uk_data_10 = uk_data.copy()
uk_data_10['year_mean'] = 10 * (uk_data['year'] // 10)

heatmap = hv.Heatmap(uk_data_10[['year_mean', 'month', 'AverageTemperature']],
                    labels='United Kingdom over 12 months').aggregate(function:np.mean)
aggregate = hv.Dataset(heatmap).aggregate('year_mean', np.mean, np.std)
aggre = hv.Errors(aggregate) * hv.Curve(aggregate)
vline = hv.VLine(1900)
vline_1 = hv.VLine(1900)
marker = hv.Points(1910, 10, 'Expanding the use of gas and oil. = ', halight='right')
marker_1 = hv.Points(1900, 10, 'The beginning of a period of rapid global growth. = ', halight='right')

overlay = (heatmap * aggre * vline * marker * vline_1 * marker_1).cols(1)
overlay.opts(
    hv.Heatmap(width=800, height=500, toolbar='hover', logz=True,
               invert_yaxis=True, label='data', toolbar='none',
               xaxis=None, colorbar=True, cmap='inferno', clim=(1, np.nan)),
    hv.Curve(width=800, height=500, show_title=False, rotation=0),
    hv.VLine(color='red', line_alpha=0.5))
```

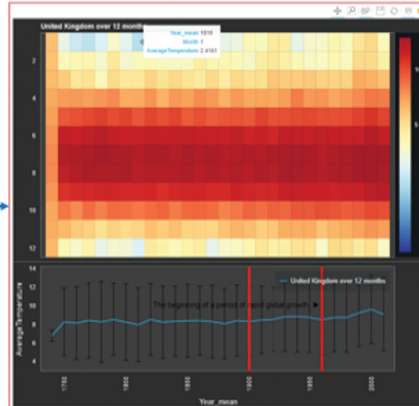


Fig. 15.13: A Heat Map showing Temperature in a Country

Step 3(g): Average temperature plot

- Plot of average temperature over many years against a rolling mean of 10 years
- The slider can be moved to change the range of the graph
- Observe that the waves are very complex and there is a lot of up variation in data

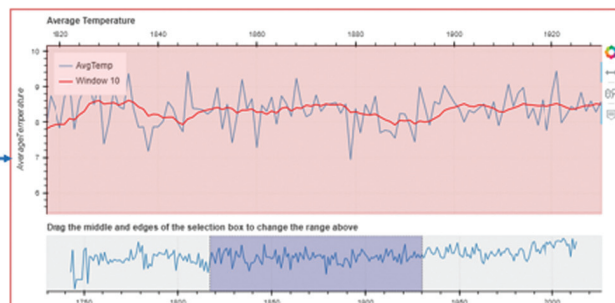


Fig. 15.14: Average Temperature Plot

15.1.4 Step 4: ARIMA/SARIMA Model is Modeling

ARIMA and SARIMA are foundational statistical models used for time series forecasting.

ARIMA (Autoregressive Integrated Moving Average) models predict future values based on past data.

SARIMA (Seasonal ARIMA) extends this to capture repeating, seasonal patterns (e.g., summer temperature spikes) by adding seasonal parameters.



Notes

Project Cycle

Step 4(a)

The challenges in setting up our SARIMA model:

1. We do not yet know the order that gives the best results
2. Our data contains both, seasonal parts and non seasonal parts

```
# As we do not know the which order to choose for the non-seasonal part(ARIMA) and seasonal part, we r
p = d = q = range(0, 3)
sp = sd = sq = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x, f, g, 12) for x, f, g in list(itertools.product(sp, sd, sq))]
```

Setting up variables for the p, q and d parameters of ARIMA model

Fig. 15.15: ARIMA Model

Step 4(b)

Finding the optimised order for our SARIMA model

Setting up a variable for AIC value to check what order works the best for our model

Finding results_df that shows the best order

```
def optimize_Sarima(train_sarima, pdq, seasonal_pdq):
    aic_results = []
    number = 0
    for param in tqdm_outer(pdq):
        for param_season in tqdm_inner(seasonal_pdq):
            try:
                model_sarimax = SARIMAX(train_sarima,
                                         order=param,
                                         seasonal_order=param_season,
                                         enforce_stationarity=False,
                                         enforce_invertibility=False)

            results = model_sarimax.fit()
            number = number + 1
            aic = results.aic
            aic_results.append([param, param_season, aic])
            except LinAlgError as lal:
                print(lal, 'Loop Number:', number, 's_order=', s_order)
                continue
            results_df = pd.DataFrame(aic_results, columns=['order', 's_order', 'aic'])
            results_df = results_df.sort_values(by='aic', ascending=True).reset_index(drop=True)
            return results_df

    pm.Column(tqdm_outer, tqdm_inner)
    results_df = optimize_Sarima(train_sarima, pdq, seasonal_pdq)
```

Fig. 15.16: SARIMA Model

15.1.5 Akaike Information Criterion (AIC)

- Akaike Information Criterion (AIC) is a method of estimating the error in prediction.
- It can be used to check the relative quality of statistical models.
- Lesser the AIC, better the prediction.
- The model with the maximum likelihood of correct prediction, is the one that 'fits' the data is the best

Project Cycle

Step 4(c)

Notice how our model is comparing different orders and ranking them based on best fit of the data

	no	s	F
	order	s_order	aic
0	(1, 1, 1)	(0, 1, 1, 12)	101.448656
1	(0, 1, 2)	(0, 1, 1, 12)	101.608648
2	(0, 1, 1)	(0, 1, 1, 12)	102.386678
3	(0, 0, 2)	(0, 1, 1, 12)	103.099140
4	(1, 1, 2)	(0, 1, 1, 12)	103.168544

Fig. 15.17: AIC for error in Prediction

After this progress bar finishes, we get the orders with the AIC values

Lower the AIC values, better the model works

```
pn.Column(temp_out, temp_inn)
100% 27/27 [00:22<00:00, 1.231it/s]
200% 0/0 [00:02<00:00, 4.982it/s]

# Running this cell starts the progress bars above, wait for the progress to finish.
results_of = optimize_sarima(train_sarima, pdq, seasonal_pdq)

# checking the best order for our model, which is in results_of
# This cell gives us the answer to which order and s_order is the best for our case,
# (which would lead to accurate predictions and also giving a low AIC)
# Lower the AIC, better the accuracy
results_of.head()

order    s_order    aic
0  (1, 1, 1)  (0, 1, 1, 12)  101.448656
1  (0, 1, 2)  (0, 1, 1, 12)  101.608648
2  (0, 1, 1)  (0, 1, 1, 12)  102.386678
3  (0, 0, 2)  (0, 1, 1, 12)  103.099140
4  (1, 1, 2)  (0, 1, 1, 12)  103.168544

Best order and Seasonal Order: SARIMA(pqd)(PQD)m
```

Fig. 15.18: Progress bar of AIC value

Step 4(d)

Here we finally train our model with the order that fits the data best and save it in 'results' variable

Looking at the summary of 'results' using the command 'results.summary()' we find about some other parameters that we do not need to discuss here

```
# Using the best order we train the SARIMA model according to the historical temperature data
final_model = SARIMAX(train_sarima,
                      order=order,
                      seasonal_order=s_order,
                      enforce_stationarity=False,
                      enforce_invertibility=False)
results = final_model.fit()
```

```
# After training the SARIMA model, we get the following results, from which we need to take into account
results.summary()

SARIMAX Results
Dep. Variable: AverageTemperature    No. Observations: 100
Model: SARIMAX(1, 1, 1, 12)    Log Likelihood: -46724
Date: Fri, 26 Aug 2022    AIC: 101.449
Time: 01:54:23    BIC: 110.610
Sample: 12-31-1999    HQIC: 105.190
Covariance Type: oas

coef    std err    z    P>|z|    [0.025    0.975]
intercept    1.0000    0.000    1.000    0.000    0.999    1.001
AR.L1    0.2272    0.127    1.793    0.075    -0.021    0.476
MA.L1    -0.8214    0.309    -2.658    0.009    -1.436    -0.207
MA.L12    -1.0002    0.1547    -6.480    0.000    -0.328    -1.672
sigma2    0.1660    0.000    0.000    0.000    0.000    0.1660
Ljung-Box (Ljung-Box): 0.00    Jarque-Bera (JB): 0.00
Prob(Q): 0.97    Prob(KS): 0.72
Heteroskedasticity (H): 1.00    Bias: 0.00
Prob(H0:Const): 0.22    Kurtosis: 2.02
```

Fig. 15.19



Notes



Notes

15.1.6 Step 5: Evaluation

Let's learn about some metrics that will help us identify if our model is working well

15.1.6.1 Residual in Time series

- Each value in a time series can be forecast using previous observations
- These values are called fitted values
- Different models can fit over a dataset in a different way like we observed in the SARIMA model
- The 'residuals' in a time series model are what is left over after fitting a model

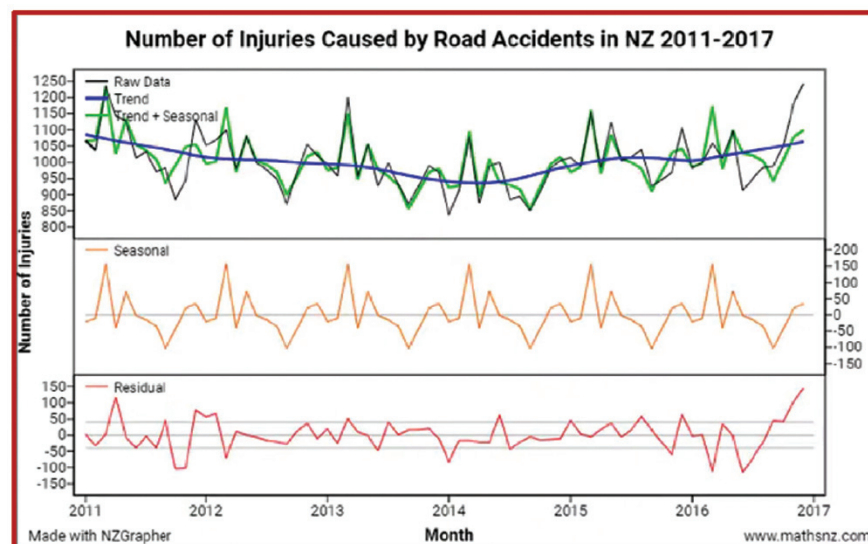


Fig. 15.20 Data SARIMA Model

Residual plots must match



Fig. 15.21: Residual Plots



Lesser the residual, better the model fits the data



Fig. 15.22

- Window 1, residual is a negative peak because raw data is lesser than the seasonal plus trend data
- Window 2, the difference is very less, so the residual line is quite close to zero line
- If there is a lot of up and down in the residual, our model does not fit the data

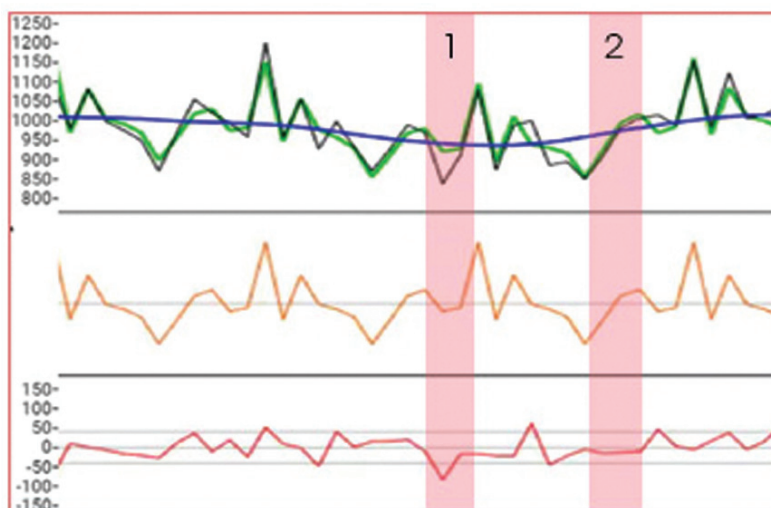


Fig. 15.23: Residual Line

15.1.6.2 Distribution of Residuals.

The residual needs to be normally distributed

- Distribution a function of data that defines how the values of a variable are distributed
- The normal distribution is a symmetric distribution where most of the observations cluster around the central peak

- Our model is working well if:
- Our fitted values are distributed closer to the mean of residual values
 - Residual values are small in magnitude

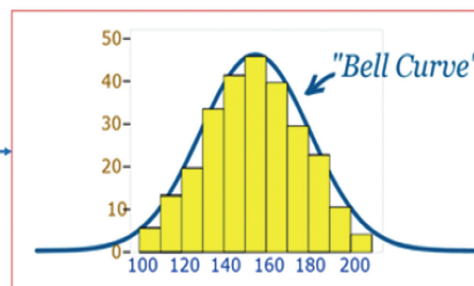


Fig. 15.24: Distribution of Residuals



Notes

15.1.6.3 Histogram

Examining the distribution: Histogram

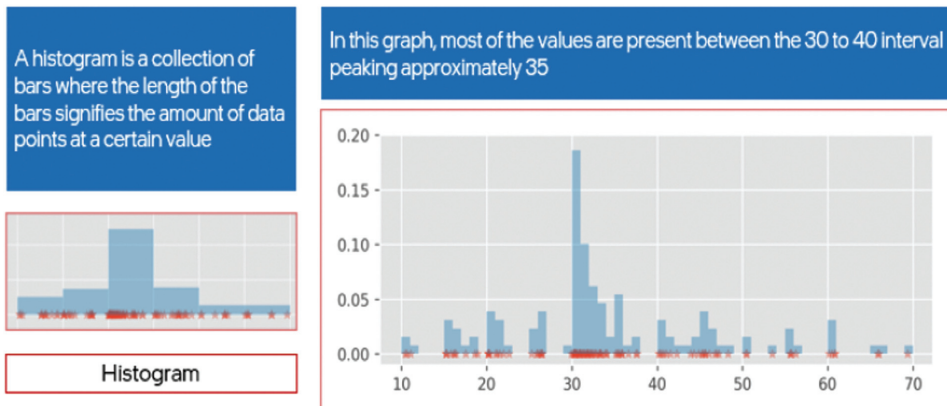


Fig. 15.25: Data Distribution as Histogram

15.1.6.4 Kernel Density Estimator (KDE)

Examining the distribution through Kernel Density Estimator (KDE)

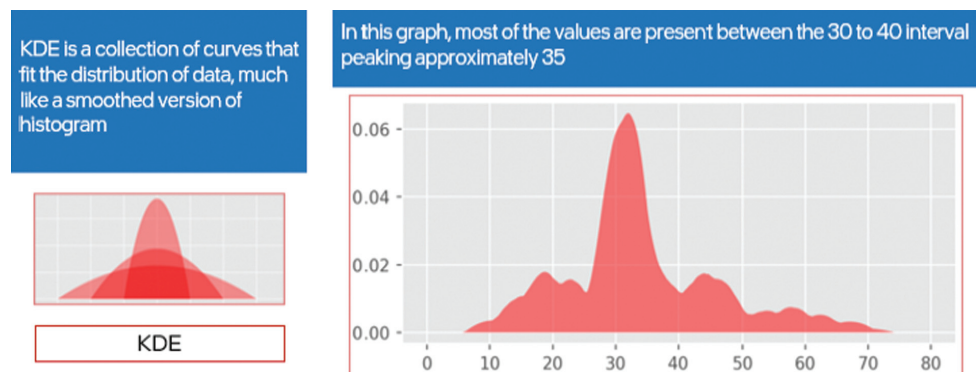


Fig. 15.26: Kernel Density Estimator (KDE)

15.1.6.5 Correlogram

Examining the distribution through Correlogram

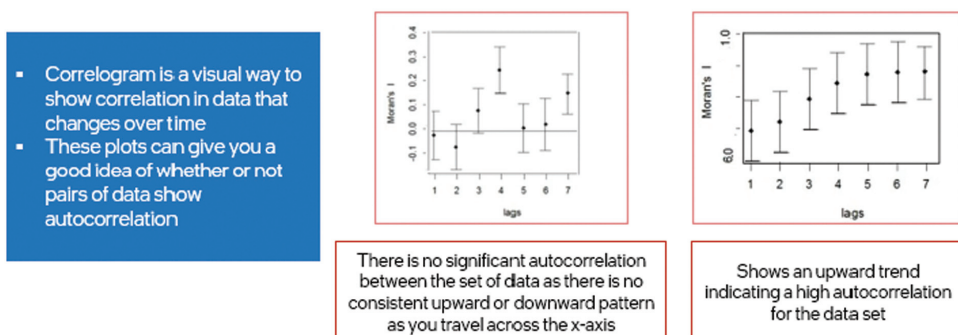


Fig. 15.27: Correlation Using Correlogram



Notes

Project Cycle

2. Which stage involves cleaning and organizing the collected data?
 - (a) Problem Scoping
 - (b) Data Exploration
 - (c) Deployment
 - (d) Evaluation
3. In which stage is an AI model trained?
 - (a) Modeling
 - (b) Problem Scoping
 - (c) Data Acquisition
 - (d) Evaluation
4. What does ARIMA stand for?
 - (a) Auto Regression Integrated Moving Average
 - (b) Automatic Regression Integrated Model Analysis
 - (c) Average Regression Integrated Moving Algorithm
 - (d) Auto Recursive Integrated Mean Analysis
5. Which model is specifically designed for seasonal time series data?
 - (a) Linear Regression
 - (b) ARIMA
 - (c) SARIMA
 - (d) Decision Tree



WHAT YOU HAVE LEARNT

- Project and Project Cycle
- The basic concept in development process of project
- various data processing techniques in project development
- ARIMA and SARIMA statistical models
- some metrics that will help us identify if our model is working well such as Residual in Time series, Kernel Density Estimator (KDE), Correlogram and Quantile-Quantile Plot (QQ plot)



TERMINAL QUESTIONS

1. How can we combat climate change on a local (immediate neighborhood) level?

Project Cycle

2. Which other domain of AI should we explore for possible AI use cases after Statistical Data?
3. What code will you write to get data info?
4. What code will you use to remove missing values?
5. Name the library you will use to plot the graphs.
6. Differentiate between ARIMA Model and SARIMA model.



Notes



ANSWERS TO INTEXT QUESTIONS

15.1

1. (b)
2. (b)
3. (a)
4. (a)
5. (c)

KEY LEARNING OUTCOMES

After learning this lesson, the learner will be able to:

- identify how AI can help to achieve goals of climate action through the guided use case
- recognize Time-Series data and extract meaningful insights with the help of exploratory data analysis.
- define and apply the ARIMA model to predict the data points in the Time-Series dataset.