

Course: Artificial Intelligence (Lab)

Week: 1

Lecture: 1

Mamoona Mushtaq

What is AI?

- AI stands for Artificial Intelligence. It refers to computer systems or machines that are capable of performing tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation.





Key Characteristics

- **Learning** - AI systems can learn from data and improve their capabilities over time without being explicitly programmed. Machine learning and deep learning are subsets of AI.

Reasoning

- Reasoning - AI can use logic and inference rules to solve problems. Systems can complete tasks by analyzing available data and then taking the most appropriate action.



Self-correction

AI systems can monitor their own performance, learn from experience, and adjust behaviors to improve over time.





Adaptability

AI can adapt to new inputs and scenarios that were not part of its initial training. This allows systems to handle real-world variability and complexity.

Natural language processing



Understanding and generating human language, including speech recognition, translation and dialogue systems.



Robotics

AI is being used to develop autonomous robots and vehicles that can perform tasks without human oversight.



Computer vision

Processing and analyzing visual inputs like images and videos using pattern recognition. Enables applications like facial recognition.



AI

While narrow AI can outperform humans in specialized tasks, general AI with capabilities rivaling human intelligence across all domains has not yet been achieved. But AI research and applications continue to evolve rapidly.



Types of AI



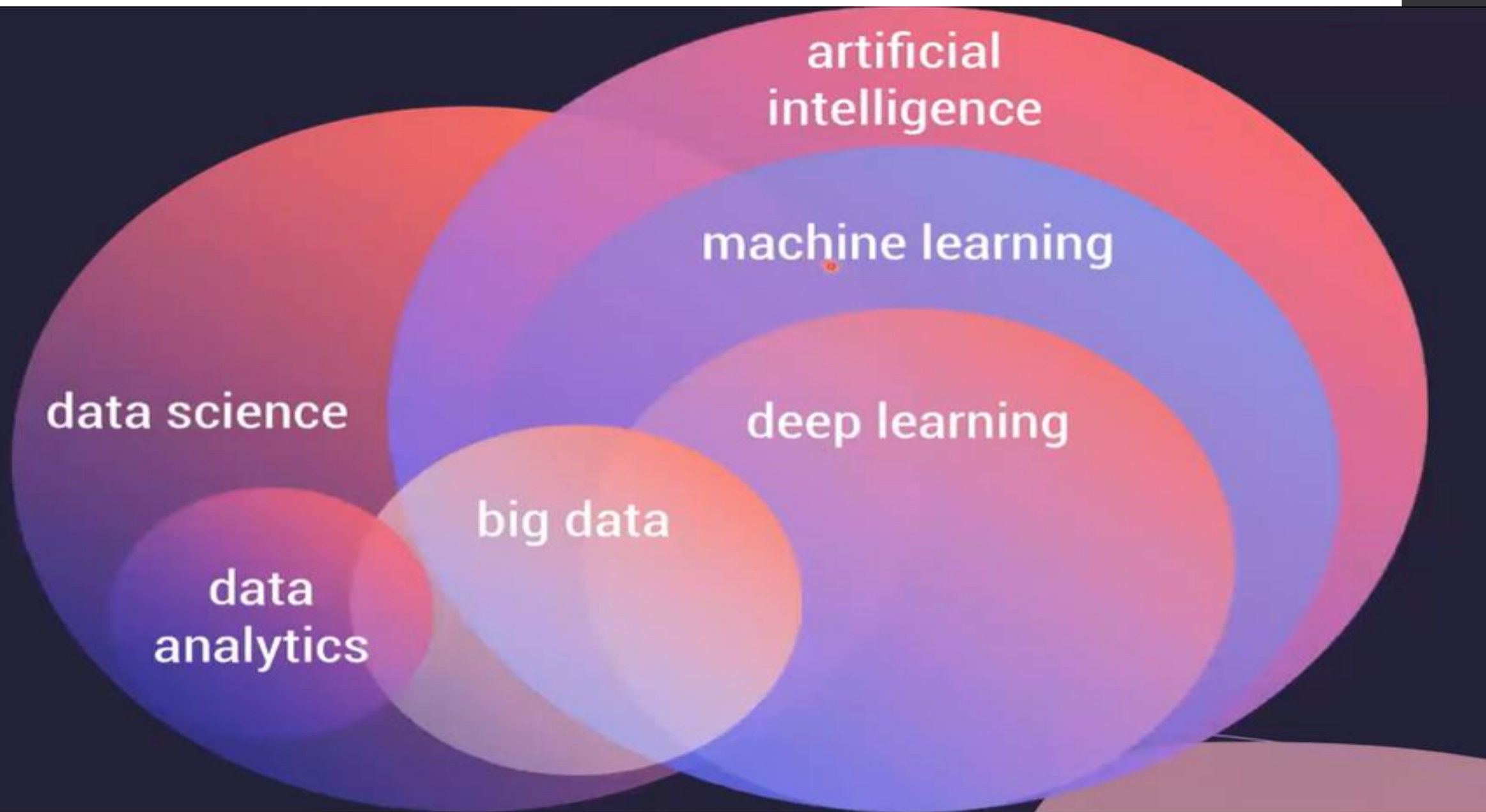
Narrow AI

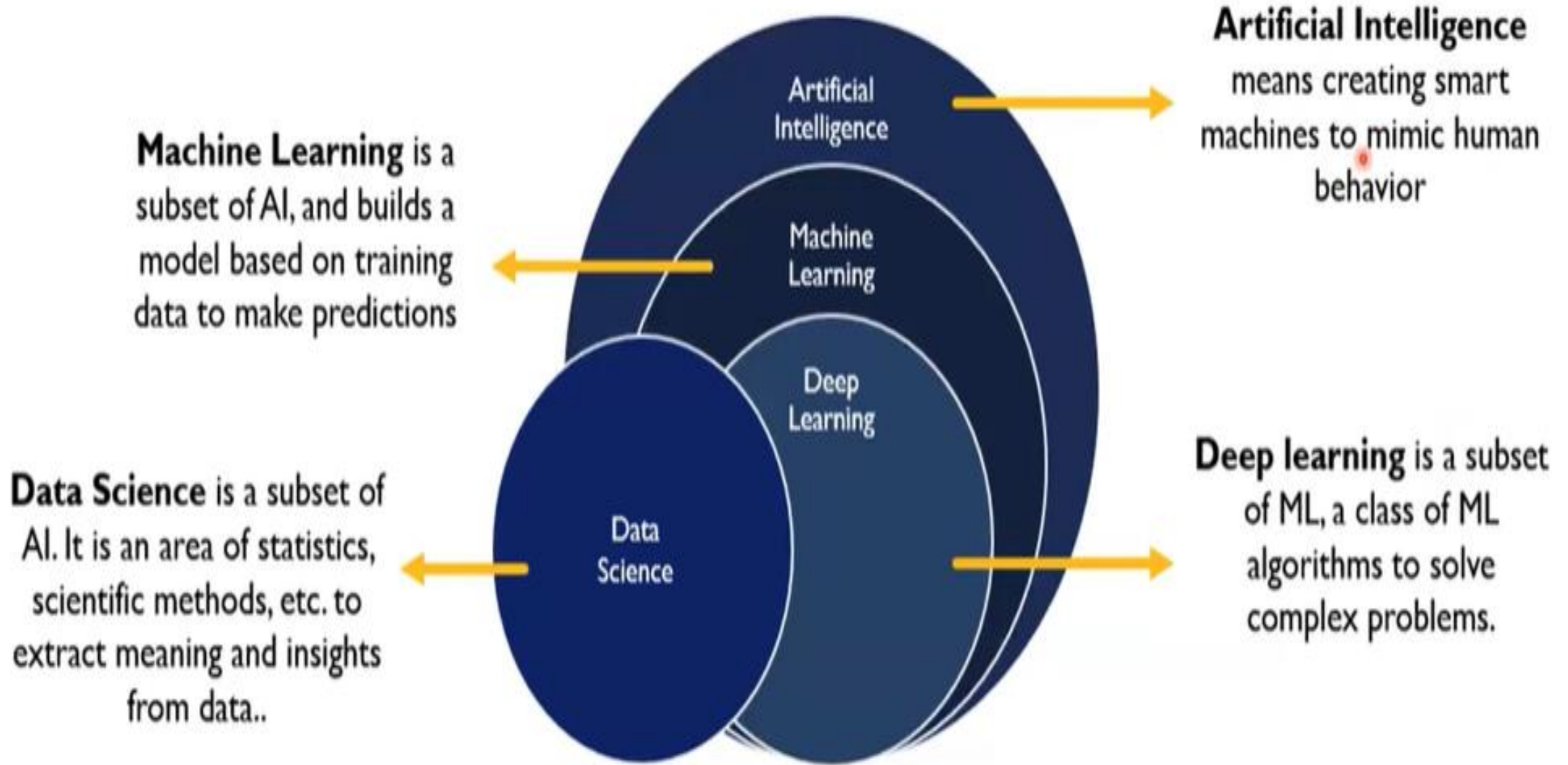
- Focused on single defined tasks
- Outperforms humans in specialized domains
- Based on current AI capabilities
- Examples are chess machines, self-driving cars



General AI

- Has general intelligence and capabilities
- Can reason, plan, communicate across domains
- Adapts to new environments and tasks
- Aspirationally aims to mimic human intelligence
- Still remains hypothetical





AI and Python

Python's link with Artificial Intelligence (AI) is so strong that it is often called the "**lingua franca**" (the common language) of AI. While AI is a set of mathematical concepts and algorithms, Python is the **bridge** that makes those concepts easy for humans to build and for computers to execute.

History and Discovery

Python was not created by a large corporation, but by a single individual with a vision for a more "human" programming language.

It is conceived by **Guido van Rossum**, a Dutch programmer.

Contrary to popular belief, it isn't named after the snake. Van Rossum was a fan of BBC comedy show "**Monty Python's Flying Circus**." He wanted a name that was short, unique, and slightly mysterious.

The intent was to create a language that was as easy to read as English, filling the gap between the complex C language and the shell scripts used for administration.

Foundation (The "Expert" Perspective)

- Python is built on a philosophy known as the "**Zen of Python.**" Its core strengths are:
- **Readability:** Python uses **indentation** (whitespace) to define code blocks. This forces programmers to write clean, organized code that others can read easily.
- **Interpreted Nature:** Unlike languages like C++ that need to be "compiled" into machine code all at once, Python is **Interpreted**. This means it executes code line-by-line, making it much faster to write and test (debug)
- **Dynamically Typed:** You don't need to tell Python that a variable is a number or text; it figures it out automatically.
- **Extensive Standard Library:** It is often called a "**Batteries Included**" language because it comes with built-in tools for everything from web browsing to file management.

Real-World Applications (Uses)

- Python is the "Swiss Army Knife" of the digital age. Its versatility allows it to dominate several industries:
- **Data Science & Analytics:** It's the best language for cleaning, analyzing, and visualizing massive amounts of data.
- **Artificial Intelligence (AI):** Frameworks like TensorFlow and PyTorch (used to build things like ChatGPT) are primarily written in Python.
- **Web Development:** Powering the "back-end" (server-side) of giants like **Instagram, Pinterest, and Spotify**.

- **Automation & Scripting:** Used by engineers to automate repetitive tasks, like moving files, scraping websites for data, or sending bulk emails.
- **Scientific Computing:** Used by NASA and CERN for complex physical simulations and astronomical calculations.

Basic Syntax

1. In most languages, we use curly braces { } to show that a group of code belongs together. In Python, we use **spaces** (usually 4 spaces or one Tab).

RULE

Every line of code within a function, loop, or condition must be indented at the same level.

Why?

This forces the programmer to keep the code visually organized. If you miss an indentation, Python will throw an “indentation error”

2. Variables and Data Types

In Python, you don't need to declare what "kind" of data a variable holds. You just assign it.

Integers: `age = 25`

Floats (Decimals): `price = 99.99`

Strings (Text): `name = "Sana"` (Always use quotes)

Booleans: `is_active = True` (Note the capital T)

3. Basic Input and Output

This is how your program talks to you and how you talk to your program.

Printing: To show something on the screen, use `print()`.

- *Example:* `print("Hello, Class!")`

Input: To get information from a user, use `input()`.

- *Example:* `user_name = input("Enter your name: ")`

4. Comments

Comments are notes for yourself or other humans. Python ignores them when running the code.

Use a **hash symbol** (#) for a single-line comment.

Use **triple quotes** (""" """) for multi-line explanations.

5. Comparison and Logic

These are used to compare values and return a True or False.

Operator	Meaning	Example
==	Equal to	5 == 5 is True
!=	Not equal to	5 != 3 is True
and	Both must be true	(5 > 3) and (2 < 4)
or	At least one is true	(5 < 3) or (2 < 4)

A Simple Example (Putting it together)

```
# This program asks for a name and checks age
```

```
name = input("What is your name? ")
```

```
age = int(input("How old are you? ")) # We use int() to turn  
text into a number
```

```
if age >= 18:
```

```
    print(name + " is an adult.")
```

```
else:
```

```
    print(name + " is a minor.")
```

- **Indentation** defines the structure; there are no curly braces for blocks.
- **Case Sensitivity:** Variable and `variable` are two different things in Python.
- **Dynamic Typing:** You can change a variable from a number to text without errors.

Python Libraries for AI

NumPy

Numerical computing

Pandas

Data handling & analysis

Matplotlib

Data visualization

Seaborn

Statistical visualization

Scikit-learn

Machine Learning

TensorFlow / Keras

Deep Learning

PyTorch

Deep Learning (research-oriented)

NumPy (Numerical Python)

Purpose:

- Works with **arrays & matrices**
- Faster than Python lists
- Used in almost all AI algorithms

Code:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4])
```

```
print(arr)
```

```
print(arr * 2)
```

Output:

```
[1 2 3 4]
```

```
[2 4 6 8]
```


Pandas (Data Analysis Library)

Purpose:

- Used for **tables (rows & columns)**
- Data cleaning & preprocessing
- Handles CSV, Excel, databases

Code:

```
import pandas as pd

data = {
    "Name": ["Ali", "Sara", "Ahmed"],
    "Marks": [85, 90, 78]
}

df = pd.DataFrame(data)
print(df)
```

Output:

	Name	Marks
0	Ali	85
1	Sara	90
2	Ahmed	78

Matplotlib (Basic Visualization)

Purpose:

- Creates **graphs & plots**
- Used to visualize trends & results

Code:

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4]
```

```
y = [10, 20, 25, 30]
```

```
plt.plot(x, y)
```

```
plt.xlabel("X values")
```

```
plt.ylabel("Y values")
```

```
plt.show()
```

Output:



Line graph appears

Seaborn (Advanced Visualization)

Purpose:

- Built on Matplotlib
- Better & attractive statistical graphs

Code:

```
import seaborn as sns  
import matplotlib.pyplot as plt  
  
data = [10, 20, 20, 30, 40]  
sns.histplot(data)  
plt.show()
```

Output:



Histogram showing data distribution

Scikit-Learn (Machine Learning)

Purpose:

- Used for **ML algorithms**
- Classification, regression, clustering
- Very beginner-friendly

TensorFlow / Keras (Deep Learning)

Purpose:

- Used for **Neural Networks**
- Image, text & speech recognition

PyTorch (Deep Learning, Research-oriented)

Purpose:

- Used in research & academia
- More control over neural networks

AI Learning Flow

Python Basics



NumPy + Pandas



Matplotlib + Seaborn



Scikit-learn (ML)



TensorFlow / PyTorch (DL)

AI in Python works by processing data (NumPy, Pandas), visualizing it (Matplotlib, Seaborn), learning patterns (Scikit-learn), and building intelligent models (TensorFlow/PyTorch).

Thank
you