# ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is a branch of computer science that focuses on developing systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving, understanding natural language, recognizing patterns, decision-making, and adapting to new situations.

The term Artificial Intelligence was first coined by John McCarthy in 1956.

## Types of AI

1. Narrow AI – Designed for specific tasks (e.g., Google Maps, Siri, chatbots).
2. General AI – A hypothetical form of AI that can perform any intellectual task like a human.
3. Superintelligent AI – AI that surpasses human intelligence.

## Advantages

* Automates repetitive tasks.
* Processes large amounts of data efficiently.
* Enhances accuracy in predictions and decision-making.

## Challenges

* Job displacement.
* Bias in AI systems.
* Data privacy concerns.
* Ethical issues.

# MACHINE LEARNING

The term Machine Learning (ML) was first introduced by Arthur Samuel in 1959.

Machine Learning is a subset of AI that enables computers to learn patterns from data and improve their performance automatically, without explicit programming.

## Types of Machine Learning

1. Supervised Learning – Learns from labeled data (input + correct output). Examples: House price prediction, spam detection, stock price forecasting.
2. Unsupervised Learning – Learns from unlabeled data by identifying hidden patterns. Examples: Customer segmentation, product grouping, market basket analysis.
3. Reinforcement Learning – Learns through trial and error, using rewards and penalties. Examples: Self-driving cars, game-playing AI (e.g., AlphaGo).

## Key Steps in Machine Learning

1. Data Collection → Gather raw data.
2. Data Preprocessing → Clean and prepare the data.
3. Feature Selection/Engineering → Identify useful variables (features).
4. Model Training → Apply algorithms (e.g., Decision Trees, Neural Networks).
5. Testing & Evaluation → Assess performance using test data.
6. Deployment → Use the trained model in real-world applications.

## Common ML Algorithms

* Linear Regression → Predicting continuous values (e.g., salary prediction).
* Logistic Regression → Classification (e.g., spam detection).
* Decision Trees & Random Forests → Used for both classification and regression.
* Support Vector Machines (SVM) → Classification using margin separation.
* K-Means Clustering → Unsupervised grouping of data.
* Neural Networks & Deep Learning → Advanced tasks such as image and speech recognition.

## Advantages

* Automates decision-making.
* Learns and improves from experience.
* Handles large-scale data analysis.

## Challenges

* Requires large datasets.
* Risk of bias in training data leading to unfair outcomes.
* Lack of transparency in complex models (“black box problem”).

# NUMPY

* NumPy is a Python library used for numerical computing.
* It is widely used for working with arrays and performing mathematical operations.
* NumPy stands for Numerical Python.

# NumPy Examples

## Creating a NumPy Array

import numpy as np  
  
arr = np.array([1, 2, 3, 4, 5])  
  
print(arr)  
print(type(arr))

Output: [1 2 3 4 5]  
<class 'numpy.ndarray'>

## 0-D Arrays

arr = np.array(42)  
print(arr)

Output: 42

## 1-D Arrays

arr = np.array([1, 2, 3, 4, 5])  
print(arr)

Output: [1 2 3 4 5]

## 2-D Arrays

arr = np.array([[1, 2, 3], [4, 5, 6]])  
print(arr)

Output:  
[[1 2 3]  
 [4 5 6]]

## 3-D Arrays

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])  
print(arr)

Output:  
[[[1 2 3]  
 [4 5 6]]  
  
 [[1 2 3]  
 [4 5 6]]]

## Accessing Array Elements

arr = np.array([1, 2, 3, 4])  
print(arr[0])

Output: 1

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

Output: 7

## Array Slicing

Slicing allows selection of elements using [start:end:step].

arr = np.array([1, 2, 3, 4, 5, 6, 7])  
print(arr[1:5])

Output: [2 3 4 5]

arr = np.array([1, 2, 3, 4, 5, 6, 7])  
print(arr[:4])

Output: [1 2 3 4]

arr = np.array([1, 2, 3, 4, 5, 6, 7])  
print(arr[-3:-1])

Output: [5 6]

## NumPy Array Shape

arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])  
print(arr.shape)

Output: (2, 4)

## Joining NumPy Arrays

arr1 = np.array([1, 2, 3])  
arr2 = np.array([4, 5, 6])  
arr = np.concatenate((arr1, arr2))  
print(arr)

Output: [1 2 3 4 5 6]

## Sorting Arrays

arr = np.array([3, 2, 0, 1])  
print(np.sort(arr))

Output: [0 1 2 3]

## Generating Random Numbers

from numpy import random  
x = random.randint(100)  
print(x)  
  
x = random.rand()  
print(x)  
  
x = random.randint(100, size=(5))  
print(x)

## Additional NumPy Examples

### Generate Random Float

from numpy import random  
  
x = random.rand()  
print(x)  
  
Output:  
0.8578710965891362

### Generate Random Array

from numpy import random  
  
x = random.randint(100, size=(5))  
print(x)  
  
Output:  
[55 1 96 12 63]

### Initializing numpy array with zeros

import numpy as np  
  
N1 = np.zeros((3,3))  
print(N1)  
  
Output:  
[[0. 0. 0.]  
 [0. 0. 0.]  
 [0. 0. 0.]]

### Initializing numpy array with same number

import numpy as np  
  
n1 = np.full((3,3), 1)  
print(n1)  
  
Output:  
[[1 1 1]  
 [1 1 1]  
 [1 1 1]]

### Vstack and Set Operations

import numpy as np  
  
n1 = np.array([1,2,3])  
n2 = np.array([1,2,3])  
  
print(np.intersect1d(n1, n2)) # Intersection  
Output:  
[1 2 3]  
  
print(np.setdiff1d(n1, n2)) # Set Difference  
Output:  
[]

# Pandas

• Pandas is a Python library.  
• Pandas is used to analyze data.

### Loading a CSV file into Pandas DataFrame

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
print(df.to\_string())  
  
Output:  
 Duration Pulse Maxpulse Calories  
0 60 110 130 409.1  
1 60 117 145 479.0  
2 60 103 135 340.0  
3 45 109 175 282.4  
4 45 117 148 406.0  
5 60 102 127 300.5  
...

### Series

import pandas as pd  
  
a = [1, 7, 2]  
myvar = pd.Series(a)  
print(myvar)  
  
Output:  
0 1  
1 7  
2 2  
dtype: int64

### DataFrames

import pandas as pd  
  
data = {  
 "calories": [420, 380, 390],  
 "duration": [50, 40, 45]  
}  
  
myvar = pd.DataFrame(data)  
print(myvar)  
  
Output:  
 calories duration  
0 420 50  
1 380 40  
2 390 45

### Labels

import pandas as pd  
  
a = [1, 7, 2]  
myvar = pd.Series(a, index = ["x", "y", "z"])  
print(myvar)  
  
Output:  
x 1  
y 7  
z 2  
dtype: int64

### Viewing the Data

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
print(df.head(5))  
print(df.tail())  
print(df.info())  
  
Output (head):  
 Duration Pulse Maxpulse Calories  
0 60 110 130 409.1  
1 60 117 145 479.0  
2 60 103 135 340.0  
3 45 109 175 282.4  
4 45 117 148 406.0  
  
Output (tail):  
 Duration Pulse Maxpulse Calories  
164 60 105 140 290.8  
165 60 110 145 300.4  
166 60 115 145 310.2  
167 75 120 150 320.4  
168 75 125 150 330.4  
  
Output (info):  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 169 entries, 0 to 168  
Data columns (total 4 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Duration 169 non-null int64   
 1 Pulse 169 non-null int64   
 2 Maxpulse 169 non-null int64   
 3 Calories 164 non-null float64  
dtypes: float64(1), int64(3)  
memory usage: 5.4 KB  
None

**MATPLOTLIB**

* Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
* Matplotlib was created by John D. Hunter.
* Matplotlib is open source and we can use it freely.
* Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.
* **Matplotlib Pyplot:**

1. Draw a line in a diagram from position (0,0) to position (6,250):

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([0, 6])  
ypoints = np.array([0, 250])  
  
plt.plot(xpoints, ypoints)  
plt.show()

A graph with a blue line

AI-generated content may be incorrect.

* **Matplotlib Plotting:**

1. Draw a line in a diagram from position (1, 3) to position (8, 10):

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([1, 8])  
ypoints = np.array([3, 10])  
  
plt.plot(xpoints, ypoints)  
plt.show()

A line graph with numbers

AI-generated content may be incorrect.

* **Matplotlib Markers:**

1. Mark each point with a circle:

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, marker = 'o')  
plt.show()

A graph with blue lines

AI-generated content may be incorrect.

* **Format Strings:**

1. Mark each point with a circle:

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, 'o:r')  
plt.show()

A graph with red dots

AI-generated content may be incorrect.

* **Line Reference:**

Line Syntax Description

'-' Solid line

:' Dotted line

'--' Dashed line

'-.' Dashed/dotted line

* **Color Reference:**

Color Syntax Description

'r' Red

'g' Green

'b' Blue

'c' Cyan

'm' Magenta

'y' Yellow

'k' Black

'w' White

* **Marker Size:**

Set the size of the markers to 20:

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, marker = 'o', ms = 20)  
plt.show()

A graph with blue dots and lines

AI-generated content may be incorrect.

* **Marker Color:**

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')  
plt.show()

A graph with blue dots and red dots

AI-generated content may be incorrect.

* **Linestyle:**

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, linestyle = 'dotted')  
plt.show()

A graph with a line

AI-generated content may be incorrect.

* **Line color:**

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, color = 'r')  
plt.show()

A red line graph on a white background

AI-generated content may be incorrect.

* **Multiple Lines:**

import matplotlib.pyplot as plt  
import numpy as np  
  
y1 = np.array([3, 8, 1, 10])  
y2 = np.array([6, 2, 7, 11])  
  
plt.plot(y1)  
plt.plot(y2)  
  
plt.show()

A line graph with blue and orange lines

AI-generated content may be incorrect.

* **Create Labels for a Plot:**

import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])  
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])  
  
plt.plot(x, y)  
  
plt.xlabel("Average Pulse")  
plt.ylabel("Calorie Burnage")  
  
plt.show()

A graph of a pulse

AI-generated content may be incorrect.

* **Create a Title for a Plot:**

import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])  
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])  
  
plt.plot(x, y)  
  
plt.title("Sports Watch Data")  
plt.xlabel("Average Pulse")  
plt.ylabel("Calorie Burnage")  
  
plt.show()

A graph of a line

AI-generated content may be incorrect.