# DAY – 4

## 1 August 2025

Artificial Intelligence refers to something which is made by humans or non-natural things and Intelligence means the ability to understand or think. AI is not a system but it is implemented in the system. There are many different types of AI, each with its own strengths and weaknesses.

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy.

Applications and devices equipped with AI can see and identify objects. They can understand and respond to human language. They can learn from new information and experience. They can make detailed recommendations to users and experts. They can act independently, replacing the need for human intelligence or intervention (a classic example being a self-driving car).

## Types of Artificial Intelligence

Artificial Intelligence (AI) has transformed industries, leading to significant advancements in technology, science, and everyday life. To understand AI better, we must first recognize that AI can be categorized into different types based on capabilities and functionalities.

**Type 1: Based on Capabilities of AI**

- Narrow AI
- General AI
- Super AI

**Type 2: Based on the Functionality of AI**

- Reactive Machines
- Limited Memory AI
- Theory of Mind
- Self-Aware AI

AI development is primarily powered by extensive, open-source libraries that provide pre-written functions for common tasks. The most prominent AI libraries are in Python, due to its ease of use and vast ecosystem.

**Top AI Libraries by Language/Domain**

Here are some of the leading AI libraries, categorized by their primary use and language:

**Python (Most Common):** Python is the dominant language for AI/ML development.

- **TensorFlow**: An end-to-end open-source platform developed by Google for machine learning, widely used for production-grade, large-scale deep learning applications such as image recognition and speech processing.
- **PyTorch:** Developed by Facebook AI Research (FAIR), it is popular in academic and research settings for its flexibility, dynamic computation graphs, and ease of use in prototyping complex neural networks.
- **Scikit-learn:** An efficient and simple library for traditional machine learning tasks like classification, regression, clustering, and data preprocessing. It is excellent for beginners and smaller datasets.
- **Keras:** A high-level, user-friendly API that runs on top of other frameworks (like **TensorFlow)** to simplify the building and training of deep learning models.
- **OpenCV:** A comprehensive library for computer vision tasks, including image processing, object detection, and real-time video analysis.
- **Hugging Face Transformers:** A library providing state-of-the-art pre-trained models (like BERT, GPT, T5) for Natural Language Processing (NLP) problems such as text analysis and translation.
- **NLTK & spaCy:** Libraries specifically for NLP tasks like sentiment analysis, tokenization, and named entity recognition.

**Java:** Used for enterprise-level AI applications due to its platform independence and performance.

- **Deeplearning4j (DL4J):** A deep learning library that integrates with Apache Spark for distributed computing.
- **Weka:** A suite of machine learning software that provides tools for data mining, classification, and visualization.

**C++:**

- **OpenCV:** The most popular AI library for C++, also available for Python and Java, for high-performance computer vision tasks.
- **Program Examples & Applications of AI**
- AI programs are used in numerous real-world applications, often without users realizing it.
- **Digital Assistants (Siri, Alexa, Google Assistant):** These use NLP libraries to understand voice commands and provide relevant information or actions.
- **Recommendation Systems (Netflix, Amazon):** Machine learning models analyze user data to suggest products or media based on past behavior.
- **Self-Driving Cars:** Utilize computer vision libraries like OpenCV to interpret road conditions, detect objects, and navigate safely.
- **Fraud Detection**: In finance, AI models analyze transaction patterns in real-time to flag unusual activity and prevent financial crimes.
- **Healthcare Diagnosis**: Deep learning models powered by TensorFlow or PyTorch can analyze medical images (e.g., MRI scans) to help detect diseases like tumors with high accuracy.
- **Text Editors/Autocorrect:** Software like Grammarly or smartphone autocorrect uses AI/ML and NLP to check grammar, suggest improvements, and predict words.

## What is Open-Source AI?

Open-source AI libraries are libraries that are released under a license but the source code is available to all. The copyright holders of such libraries allow users to use them and do some valuable modifications to their source codes to add new features, improve existing features, and fix bugs if any.

Projects on GitHub work across operating systems so teams can create custom solutions by leveraging existing tools. Open-source AI democratizes access to technology and enables applications for many use cases.

**Benefits of Open-Source AI Tools**

**Free:** Individuals and businesses of all sizes can use it.

**Customizable:** Users can modify the source code.

**Scalable:** Can be used for projects of all sizes from big to small.

**Community:** A large community of developers contribute to the software.

**Top 10 Open Source AI Libraries**

1. TensorFlow
2. PyTorch
3. Scikit-learn
4. Keras
5. OpenCV
6. Hugging Face Transformers
7. NLTK (Natural Language Toolkit)
8. SpaCy
9. Gensim
10. XGBoost

- **Examples-**

**Example 1 -**

```
from transformers import pipeline

# Load a pre-trained sentiment analysis model

classifier = pipeline("sentiment-analysis")
```

```
# Run the analysis on a sample text

text = "Python automation makes me feel unstoppable!"

result = classifier(text)

print(f"Text: {text}")

print(f"Sentiment: {result[0]['label']} with score {result[0]['score']}")
```

**Example 2 –**

```
from sklearn.cluster import KMeans

import numpy as np


# Sample data: three groups of two-dimensional points

X = np.array([[1, 2], [1, 4], [1, 0],

        [4, 2], [4, 4], [4, 0],

        [8, 2], [8, 4], [8, 0]])


# Define the model: we want to find 3 clusters

kmeans = KMeans(n_clusters=3, random_state=0, n_init=10) # Added n_init for modern scikit-learn versions


# Train the model on the data

kmeans.fit(X)


# Print the cluster labels for each data point
```

```
print("Cluster labels for the data points:")

print(kmeans.labels_)


# Print the coordinates of the cluster centers

print("Cluster centers:")

print(kmeans.cluster_centers_)
```

**Example 3 –**

```
import pandas as pd

import io


# Create a sample dataset in CSV format

data = """

Name,Age,City,Score

Alice,25,New York,88

Bob,30,Los Angeles,92

Charlie,,Chicago,75

David,28,New York,

"""

df = pd.read_csv(io.StringIO(data))

# Data cleaning: drop rows with missing values

df.dropna(inplace=True)
```

```
# Data analysis: calculate the average score

average_score = df['Score'].mean()


print("Original Data:")

print(df)

print(f"\nAverage Score (after cleaning): {average_score}")
```