Session 4

ENG : Shereen saad

The main topics we will cover are:

- **Introduction to Data Acquisition:**
- **Working with APIs:**
- **Data Integration Basics:**
- **Practical Project – Fetch and Prepare Data**

1_Introduction to Data Acquisition:

- What data acquisition is and why it’s crucial in the field of data analysis.
- An overview of various data sources, including databases, CSV/JSON files, and sensor data.

What is Data Acquisition?

- Definition: The process of collecting, measuring, and analyzing real-world data from different sources.

- Used in various fields such as engineering, research, and industrial automation.

- Data can come from sensors, signals, or manual input.

Why is Data Acquisition Important?

- Helps in monitoring and controlling processes.
- Enables decision-making based on real-world data.
- Improves efficiency and accuracy in scientific research and industrial applications.
- Supports automation and artificial intelligence.

Data Acquisition Process

- Signal Detection – Sensors collect raw data from the environment.
- Signal Conditioning – Data is cleaned and adjusted for accuracy.
- Data Conversion – Analog signals are converted to digital.
- Data Storage & Processing – Data is stored in databases or cloud systems.
- Data Analysis & Visualization – Software tools analyze and display the results.

Future Trends in Data Acquisition

- Internet of Things (IoT): Remote & real-time data collection.
- Artificial Intelligence & Machine Learning: Automated data analysis.
- Cloud Computing: Scalable data storage & processing.
- 5G & Edge Computing: Faster and more efficient data transmission.

Coding Example in Python for Data Acquisition

- Python can be used to collect and analyze data from sensors or files.

➢Reading Data from a Local CSV File:

```
import pandas as pd

# Load data from a CSV file
data = pd.read_csv("sensor_data.csv")
print(data.head())
```

➢Reading Data from Google Drive in Google Colab:

```
from google.colab import drive
import pandas as pd
drive.mount('/content/drive')
data = pd.read_csv('/content/drive/My Drive/sensor_data.csv')
print(data.head())
```

➢ Reading Data from Kaggle in Google Colab Without Downloading:

DAataSet link :https://www.kaggle.com/c/home-credit-default-risk/overview

- **Step 1: Get the API Token from Kaggle**

- Log in to your Kaggle account.
- Go to My Account in the account settings.
- Scroll down to the API section and click on Create New API Token.
- A kaggle.json file will be downloaded automatically.

Step : 2

```
from google.colab import files
uploaded = files.upload()


!pip install kaggle
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle competitions download -c home-credit-default-risk
!unzip home-credit-default-risk.zip
import pandas as pd
# البيانات تحميل
credit_card_balance = pd.read_csv('credit_card_balance.csv')
credit_card_balance.head(10)
```

2_Working with APIs:

- What is an API?

- An API (Application Programming Interface) is a way for programs to communicate and exchange data in a structured and secure manner

Steps to Fetch Data from an API

- Identify the API Type
- The most common type is REST API, which works over HTTP and uses methods like:
- GET to fetch data
- POST to send new data
- PUT to update data
- DELETE to remove data

- Send a Request to the API
- You can use tools like Postman or write code in Python or JavaScript.
- Example using Python’s requests library

```
• import requests

• url = "https://api.example.com/data"
• response = requests.get(url)

• if response.status_code == 200:  # الطلب نجاح من التأكد
•     data = response.json()  # إلى البيانات تحويل JSON
•     print(data)  # لمشاهدتها البيانات طباعة
• else:
•     print("Failed to fetch data")
```

3_Data Integration Basics:

- **Defining data integration and its importance in ensuring data is clean, consistent, and ready for analysis.**

- **An overview of the ETL process: Extracting, Transforming, and Loading data.**

- **Tools and techniques for cleaning and transforming data, with a focus on using Python’s pandas library.**

What is Data Integration?

- **Data Integration is: the process of collecting data from different sources, transforming it, and merging it into a unified system to ensure consistency and facilitate analysis and usage.**

Why Do We Need Data Integration?

- **Data Exists in Multiple Systems: In companies or organizations, data may be stored in different databases, Excel files, cloud applications, or even external sources.**
- **Improving Data Accuracy: When data from different sources is integrated, duplicates or errors can be detected and corrected.**
- **Easier Analysis and Decision-Making: Instead of analyzing each data source separately, data integration provides a comprehensive view that helps in making better decisions.**
- **Continuous Data Updates: Integration ensures that new or modified data from the original source is automatically transferred to the central system.**

How is Data Integration Performed?

- **The data integration process is carried out through several key steps:**
- **Data Extraction: Collecting data from different sources.**
- **Data Transformation: Cleaning, formatting, and converting the data to make it compatible with the new system.**
- **Data Loading: Storing the integrated data in a central database or a data warehouse.**

4_Practical Project

– Fetch and Prepare Data from a Public API:

- **A hands-on project where we’ll fetch data from a public API (for example, a weather or news API).**

- **Step-by-step guidance on choosing an API, reading its documentation, writing the code, and cleaning the data.**

- **Visualizing the results using data visualization libraries such as matplotlib.**

Thank YOU