◆ **Week 2: Data Cleaning and Preprocessing Using Python (Jupyter Notebook)**

◆ Objective

The objective of Week 2 was to perform **data cleaning and preprocessing using Python**, understand real-world data issues, and implement **step-by-step cleaning operations programmatically** using Pandas in a Jupyter Notebook environment.

◆ Tools & Technologies Used

- Python
- VS Code
- Jupyter Notebook
- Libraries:
  - Pandas
  - NumPy

◆ Dataset Description

The dataset used in this week was a **Customer Call List dataset**, which contained customer details such as:

- First Name
- Last Name
- Phone Number
- Address
- Do Not Contact flag

The dataset had multiple **real-world data quality issues** including:

- Duplicate records
- Inconsistent formatting
- Missing values
- Unnecessary columns
- Invalid contact entries

◆ Step-by-Step Data Cleaning Process

1 Importing Required Libraries and Dataset

The dataset was imported using Pandas from an Excel file into the Jupyter Notebook environment.

```
import pandas as pd


df = pd.read_excel("Customer Call List.xlsx")
df
```

2️⃣ Understanding Dataset Structure

Initial exploration was done to understand column names and structure.

```
df.columns
```

This step helped identify unnecessary columns and inconsistencies.

3️⃣ Removing Duplicate Records

Duplicate entries were identified and removed to ensure data integrity.

```
df.drop_duplicates()
```

This step prevents repeated customer records from affecting analysis.

4️⃣ Dropping Unnecessary Columns

Columns that were not useful for analysis were removed.

```
df = df.drop(columns=["Not_Useful_Column"])
```

This improves clarity and reduces noise in the dataset.

5️⃣ Cleaning Text Columns (Last Name)

Special characters and unwanted symbols were removed from text fields.

```
df["Last_Name"] = df["Last_Name"].str.strip("/..._")
```

This ensures consistency in textual data.

6️⃣ Standardizing Phone Numbers

Phone numbers contained inconsistent separators such as -, /, and |.
These were cleaned to maintain uniform formatting.

```
df["Phone_Number"] = df["Phone_Number"].str.replace("-", "")

df["Phone_Number"] = df["Phone_Number"].str.replace("/", "")
```

df["Phone_Number"] = df["Phone_Number"].str.replace("|", "")

7️⃣ Handling Missing Values

Missing values were replaced with blank spaces for consistency.

df = df.fillna(" ")

8️⃣ Replacing Inconsistent Categorical Values

Different representations of similar values were standardized.

df = df.replace("Na", " ")

df = df.replace("N/a", " ")

df = df.replace("Yes", "Y")

df = df.replace("No", "N")

This step ensures uniform categorical values.

9️⃣ Removing "Do Not Contact" Customers

Customers who opted out of contact were removed from the dataset.

df = df[df["Do_Not_Contact"] != "Y"]

This is critical for ethical and compliant data usage.

🔟 Filling Missing Contact Flags

Empty values in the **Do Not Contact** column were replaced with default values.

df["Do_Not_Contact"] = df["Do_Not_Contact"].str.replace(" ", "N")

1️⃣1️⃣ Removing Records Without Phone Numbers

Rows with missing phone numbers were removed, as they are unusable for call analysis.

df = df[df["Phone_Number"] != " "]

1️⃣2️⃣ Splitting Address Column

The address column was split into **Street** and **City** for better structure.

df[["Street", "City"]] = df["Address"].str.split(",", expand=True)

1️⃣3️⃣ Resetting Index

After row removal operations, the index was reset.

```
df.reset_index(drop=True)
```

1️⃣4️⃣ Renaming Columns

Column names were updated for clarity.

```
df = df.rename(columns={"Last_Name": "Test_Name"})
```

1️⃣5️⃣ Reverting Categorical Values for Readability

Final replacements were made for better readability.

```
df = df.replace("Y", "Yes")

df = df.replace("NO", "No")
```

1️⃣6️⃣ Final Dataset Validation

The cleaned dataset was reviewed to ensure:

- No duplicates
- No invalid contacts
- Clean text and numeric fields
- Structured address data

```
df
```

🔶 Outcome of Week 2

By the end of Week 2:

- Successfully cleaned a real-world customer dataset using Python
- Gained hands-on experience with Pandas operations
- Learned how to handle missing values, duplicates, and inconsistent data
- Prepared a clean dataset ready for analysis and visualization

---

🔶 Key Learning

“Python provides flexibility, automation, and repeatability in data cleaning, making it highly effective for handling real-world datasets.”