Assignment 2: Advanced Data Preprocessing and Cleaning with Pandas

Objective:
In this assignment, you will focus on **advanced data preprocessing** techniques using the Pandas library. After completing Assignment 1, where you explored fundamental dataset analysis (head, tail, info, and descriptive statistics), this assignment will help you dive deeper into **handling missing values, encoding categorical variables, and transforming data** to prepare your dataset for analysis.

Instructions:

1. **Select a Dataset**
    - Choose a dataset in **CSV**, **Excel**, or from **Kaggle**. Ensure it contains both **numerical** and **categorical** variables, and includes **missing data** and **duplicates** for cleaning and preprocessing.
2. **Create a Jupyter Notebook**
    - **Filename:** Name your notebook as `Student_Name_Assignment2.ipynb`.
    - **Load the Data:**
    - `import pandas as pd`
    - `dataset = pd.read_csv(r"Path_to_your_dataset.csv")`
3. **Exploratory Data Analysis (EDA) (Referring to Assignment 1):**
   You have already explored the dataset in Assignment 1 using basic commands, such as head(), tail(), and info(). In this assignment, revisit the following commands for deeper insights into your dataset:
    - `dataset.head()`, `dataset.tail()`
    - `dataset.info()` (Check for missing data types)
    - `dataset.isnull().sum()` (Identify columns with missing values)
    - `dataset.describe()` (Understand distribution of numerical columns)
    - `dataset.corr()` (Check correlations between numerical features)
    - `dataset.dtypes` (Verify data types)
4. **Preprocess the Data:**
   This is the key part of the assignment. Apply the following preprocessing techniques to the dataset:
    - **Handle Missing Data:**
        - Use `fillna()` to fill missing values with mean, median, mode, or other methods.
        - Alternatively, you can use `dropna()` to remove rows or columns with missing values.
    - **Remove Duplicates:**
        - Use `drop_duplicates()` to remove duplicate rows from the dataset.
    - **Convert Data Types:**
        - For date columns, convert them to datetime format using `pd.to_datetime()`.
        - Ensure numeric columns are of the appropriate data type using `astype()`.
    - **Encode Categorical Variables:**
        - Use `pd.get_dummies()` to one-hot encode categorical variables.

        - Alternatively, if the dataset is large, consider **Label Encoding** or **Ordinal Encoding** for efficient representation.
    - **Feature Scaling:**
        - Normalize or standardise numeric columns using **MinMaxScaler** or **StandardScaler** from **sklearn**.
        - Scaling ensures that variables with different ranges are treated equally in machine learning models.
    - **Outlier Detection and Treatment:**
        - Detect outliers using statistical methods such as the Z-score or Interquartile Range (IQR).
        - Consider whether to remove or transform outliers (e.g., using logarithmic transformations or capping).
    - **Data Transformation:**
        - Create new features from existing ones, such as extracting day, month, or year from a date column, or creating categorical bins from continuous variables.

5. **Optional: Exploratory Data Analysis (EDA) for Insights:**
   While the primary focus is on preprocessing, you can optionally visualise the data using libraries such as **Matplotlib** or **Seaborn** to understand relationships, distributions, and potential transformations.
6. **Save and Submit:**
    - Save the notebook as **HTML**: File > Download as > HTML in Jupyter.
    - Submit the HTML file as part of the assignment.
