Sample Dataset:

You can import using this:

"from sklearn.datasets import fetch_california_housing
import pandas as pd

 # Load the California housing dataset
california = fetch_california_housing()

 # Create a DataFrame
df = pd.DataFrame(california.data, columns=california.feature_names)

 # Display the first few rows
df.head()"

Tasks:

1. **Outlier Detection and Removal**
   Description:

   Steps:

   a. **Data Loading:**
   - **Provide a dataset (e.g., a housing dataset, sales data, or any other dataset with numerical attributes).**
   - **Load the dataset and display basic statistics using pandas.describe().**

   b. **Task 1: Outlier Detection Using Z-score:**
   - **Instruct students to calculate the Z-scores for a specific numerical feature (e.g., price or size).**
   - **Set a threshold for detecting outliers (e.g., Z-score > 3 or Z-score < -3).**
   - **Identify and count the number of outliers based on the Z-score threshold.**
   - **Visualize the outliers using scatter plots or box plots.**

   c. **Task 2: Outlier Detection Using IQR:**
   - **Calculate the 25th percentile (Q1) and 75th percentile (Q3) of a specific feature.**
   - **Compute the IQR (Q3 - Q1) and identify outliers as values below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.**
   - **Remove the outliers and compare the dataset before and after outlier removal (e.g., using visualizations or statistical summaries).**

2. **Feature Scaling Using Standardization and Min-Max Scaling**

Steps:

1. **Data Loading:**
    - **Provide a dataset (e.g., California Housing, Titanic dataset, etc.).**
    - **Load the dataset and describe the features using summary statistics (pandas.describe()).**
2. **Task 1: Standardization (Z-score Scaling):**
    - **Select one or more numerical features from the dataset (e.g., price, age, or salary).**
    - **Apply Z-score standardization to these features using sklearn.preprocessing.StandardScaler.**
    - **Display the transformed features and compare the results to the original data.**
    - **Visualize the original and standardized data using histograms and box plots.**
3. **Task 2: Min-Max Scaling:**
    - **Apply Min-Max scaling to the same features using sklearn.preprocessing.MinMaxScaler.**
    - **Display and compare the transformed features with the original data.**
    - **Visualize the original and Min-Max scaled data.**