### Data Pre-processing

\*\*Task 3: Normalization of Selected Attribute\*\*

\*\*Attribute Selected: `PT08.S1(CO)`\*\*

#### Attribute Details:

The selected attribute for normalization is `PT08.S1(CO)`, which represents the concentration of Carbon Monoxide (CO) measured in the air. This attribute is part of the Air Quality UCI dataset, which is used to monitor air pollution levels. The attribute values are numerical, and it is essential to preprocess these values to ensure they are in a consistent scale for further analysis and machine learning tasks.

#### Normalization Technique:

Normalization is a technique used to scale the data to a fixed range, typically [0, 1]. This is particularly useful when dealing with numerical data that varies significantly in magnitude. In this case, we use Min-Max Scaling (or Min-Max Normalization) which transforms the data to fit within a specified range.

\*\*Why Min-Max Scaling?\*\*

Min-Max Scaling is appropriate for this attribute because:

- \*\*Uniform Scaling\*\*: It scales the data between 0 and 1, which is helpful for many machine learning algorithms that are sensitive to the scale of input features.

- \*\*Preservation of Relationships\*\*: This technique preserves the relationships between data points, making it suitable for attributes where the original scale is not relevant for analysis or modeling.

#### Python Code for Normalization:

```python

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

# Load the dataset

df = pd.read\_excel('AirQualityUCI.xlsx')

# Select an attribute for normalization

attribute = 'PT08.S1(CO)'

# Handle missing values (-200 values treated as NaN)

df[attribute] = df[attribute].replace(-200, pd.NA)

# Fill missing values in the selected attribute with its median

median\_value = df[attribute].median()

df[attribute] = df[attribute].fillna(median\_value)

# Normalize the attribute

scaler = MinMaxScaler()

df[attribute + '\_normalized'] = scaler.fit\_transform(df[[attribute]])

# Print results

print(f"Normalized values of {attribute}:")

print(df[[attribute, attribute + '\_normalized']].head())

```

#### Explanation of the Code:

1. \*\*Loading the Dataset\*\*: The dataset is loaded from an Excel file.

2. \*\*Handling Missing Values\*\*: We replace erroneous `-200` values with `NaN` to identify and handle missing values correctly.

3. \*\*Filling Missing Values\*\*: Missing values are filled using the median of the attribute to ensure that the data is complete and consistent.

4. \*\*Normalization\*\*: Min-Max Scaling is applied to the attribute to normalize its values between 0 and 1. This ensures that all values are on a uniform scale.

By applying these steps, the `PT08.S1(CO)` attribute is pre-processed to enhance its suitability for subsequent analysis and machine learning tasks.