# Task 4 - Pairwise Similarity Analysis

## 1. Introduction

This document describes the approach taken to analyze pairwise similarity among data objects selected from the dataset. The task involves sampling a subset of data, computing similarity scores using appropriate measures, identifying the pair with the maximum similarity, and evaluating if the selected pair is truly similar.

## 2. Sampling Strategy

To address the task, the following sampling strategy was employed:

1. Dataset: The dataset used for this analysis is the 'Air Quality UCI' dataset, which contains various air quality metrics.

2. Sampling Technique: A random sampling approach was used to select 20 data objects from the dataset. This random sampling ensures that the selected subset is representative of the overall data distribution.

## 3. Similarity Measure

The similarity between data objects was computed using the Cosine Similarity measure. We used cosine similarity as the dataset is high-dimensional with 15 features. This measure evaluates the cosine of the angle between two non-zero vectors, providing a value between -1 and 1. A value closer to 1 indicates higher similarity.

### 3.1. Preprocessing

The following Python code was used for preprocessing:

import pandas as pd  
import numpy as np  
from sklearn.preprocessing import StandardScaler  
  
# Load and preprocess your dataset  
df = pd.read\_excel('AirQualityUCI.xlsx')  
  
# Replace -200 with NaN  
df.replace(-200, np.nan, inplace=True)  
  
# Separate numeric and non-numeric columns  
numeric\_df = df.select\_dtypes(include=[np.number])  
non\_numeric\_df = df.select\_dtypes(exclude=[np.number])  
  
# Compute medians for numeric columns and fill missing values  
df[numeric\_df.columns] = numeric\_df.fillna(numeric\_df.median())  
  
# Reset index and sample 20 data objects  
df.reset\_index(drop=True, inplace=True)  
sampled\_df = df.sample(n=20, random\_state=1)  
  
# Extract numeric features for similarity calculation  
features = sampled\_df.select\_dtypes(include=[np.number])  
  
# Standardize features  
scaler = StandardScaler()  
scaled\_features = scaler.fit\_transform(features)

### 3.2. Cosine Similarity Computation

The following Python code was used to compute cosine similarity:

from sklearn.metrics.pairwise import cosine\_similarity  
  
# Compute Cosine Similarity  
cosine\_sim = cosine\_similarity(scaled\_features)

### 3.3. Identification of Maximum Similarity Pair

The following Python code was used to identify the pair with maximum similarity and to print the results:

def get\_max\_similarity\_pair(similarity\_matrix):  
 np.fill\_diagonal(similarity\_matrix, 0) # Set diagonal to 0 to avoid self-similarity  
 max\_similarity\_idx = np.unravel\_index(np.argmax(similarity\_matrix, axis=None), similarity\_matrix.shape)  
 max\_similarity\_score = similarity\_matrix[max\_similarity\_idx]  
 return max\_similarity\_idx, max\_similarity\_score  
  
def print\_pair\_info(df, idx, measure\_name, score):  
 # Extract numeric columns for displaying pairs  
 numeric\_columns = df.select\_dtypes(include=[np.number]).columns  
 pair\_1 = df.iloc[idx[0]][numeric\_columns]  
 pair\_2 = df.iloc[idx[1]][numeric\_columns]  
   
 print(f"\n{measure\_name} Similarity:")  
 print(f"Pair with maximum similarity (Index {idx[0]} and {idx[1]}):")  
 print(f"Pair 1:\n{pair\_1}\n")  
 print(f"Pair 2:\n{pair\_2}\n")  
 print(f"Similarity Score: {score:.4f}")  
  
 # Check if they are really similar  
 if score > 0.8: # You can adjust this threshold based on your context  
 print("The pairs are really similar.")  
 else:  
 print("The pairs are not very similar.")  
  
# Get the pair with maximum similarity and the similarity score  
cosine\_max\_idx, cosine\_max\_score = get\_max\_similarity\_pair(cosine\_sim)  
print\_pair\_info(sampled\_df, cosine\_max\_idx, 'Cosine Similarity', cosine\_max\_score)

## 4. Results

The pair with the maximum similarity was identified as follows:

Cosine Similarity Similarity:

Pair with Maximum Similarity (Index 0 and 2):

Pair 1:  
CO(GT) 1.1  
PT08.S1(CO) 1047.333333  
NMHC(GT) 74.0  
C6H6(GT) 4.932008  
PT08.S2(NMHC) 760.0  
NOx(GT) 64.0  
PT08.S3(NOx) 1032.0  
NO2(GT) 74.0  
PT08.S4(NO2) 1378.666667  
PT08.S5(O3) 1003.0  
T 11.466667  
RH 61.433333  
AH 0.830289  
Name: 822, dtype: object

Pair 2:  
CO(GT) 1.8  
PT08.S1(CO) 1129.5  
NMHC(GT) 56.0  
C6H6(GT) 5.191654  
PT08.S2(NMHC) 773.0  
NOx(GT) 70.0  
PT08.S3(NOx) 1130.25  
NO2(GT) 82.0  
PT08.S4(NO2) 1451.75  
PT08.S5(O3) 1050.5  
T 12.1  
RH 61.100001  
AH 0.860316  
Name: 82, dtype: object

Similarity Score: 0.9681

The pairs are really similar.

**Answer :** The pairs really are indeed very similar. By examining the individual attributes, it is evident that all the points are relatively close to each other.

## 5. Conclusion

This document summarizes the methodology and results of the pairwise similarity analysis. The sampling strategy, similarity measure used, and the computation process were outlined, along with the evaluation of the most similar pair from the dataset.