**Лабораторна робота №8**

Нелінійне зменшення розмірів

1. Зменшення розмірності за допомогою ізомапи, локально-лінійного вбудовування, багатовимірного масштабування (MDS), t-розподіленого стохастичного сусіда Вбудовування (t-SNE) у 2D та 3D простір.

Рішення:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.manifold import Isomap, LocallyLinearEmbedding, MDS, TSNE

from sklearn.preprocessing import StandardScaler

from mpl\_toolkits.mplot3d import Axes3D

# Завантаження даних

column\_names = ['Sequence Name', 'mcg', 'gvh', 'lip', 'chg', 'aac', 'alm1', 'alm2', 'class']

data = pd.read\_csv('ecoli.data', sep='\s+', header=None, names=column\_names)

# Підготовка даних

X = data.iloc[:, 1:8].values

y = data['class'].values

# Стандартизація даних

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Функція для візуалізації результатів

def plot\_2d(X, y, title):

plt.figure(figsize=(10, 8))

unique\_classes = np.unique(y)

for cls in unique\_classes:

plt.scatter(X[y == cls, 0], X[y == cls, 1], label=cls)

plt.title(title)

plt.legend()

plt.show()

def plot\_3d(X, y, title):

fig = plt.figure(figsize=(10, 8))

ax = fig.add\_subplot(111, projection='3d')

unique\_classes = np.unique(y)

for cls in unique\_classes:

ax.scatter(X[y == cls, 0], X[y == cls, 1], X[y == cls, 2], label=cls)

ax.set\_title(title)

ax.legend()

plt.show()

# Isomap

isomap\_2d = Isomap(n\_components=2)

X\_isomap\_2d = isomap\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_isomap\_2d, y, 'Isomap 2D')

isomap\_3d = Isomap(n\_components=3)

X\_isomap\_3d = isomap\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_isomap\_3d, y, 'Isomap 3D')

# Locally Linear Embedding

lle\_2d = LocallyLinearEmbedding(n\_components=2)

X\_lle\_2d = lle\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_lle\_2d, y, 'LLE 2D')

lle\_3d = LocallyLinearEmbedding(n\_components=3)

X\_lle\_3d = lle\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_lle\_3d, y, 'LLE 3D')

# Multi-dimensional Scaling

mds\_2d = MDS(n\_components=2)

X\_mds\_2d = mds\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_mds\_2d, y, 'MDS 2D')

mds\_3d = MDS(n\_components=3)

X\_mds\_3d = mds\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_mds\_3d, y, 'MDS 3D')

# t-SNE

tsne\_2d = TSNE(n\_components=2)

X\_tsne\_2d = tsne\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_tsne\_2d, y, 't-SNE 2D')

tsne\_3d = TSNE(n\_components=3)

X\_tsne\_3d = tsne\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_tsne\_3d, y, 't-SNE 3D')

Зображення, що містить знімок екрана, текст, схема, карта

Автоматично згенерований опис

Зображення, що містить текст, схема, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

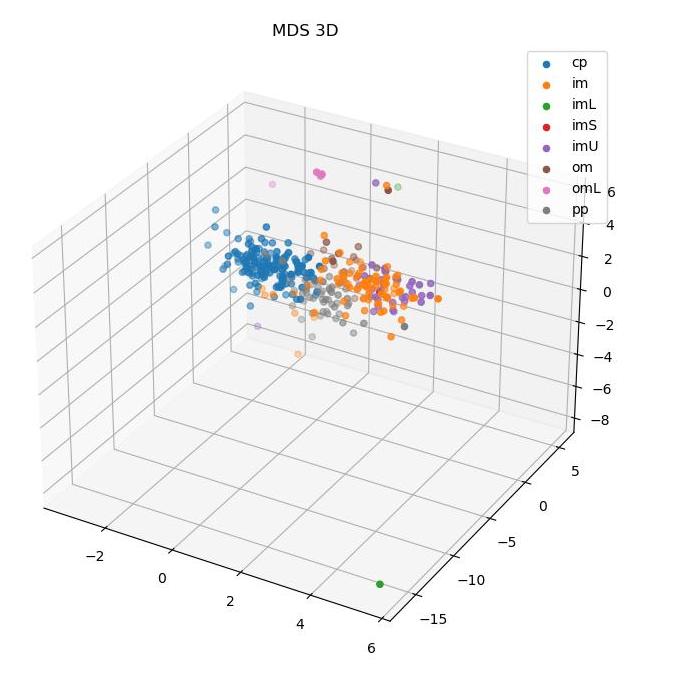
Автоматично згенерований опис

Зображення, що містить схема, текст, знімок екрана, ряд

Автоматично згенерований опис

Зображення, що містить знімок екрана, Барвистість

Автоматично згенерований опис



Зображення, що містить знімок екрана, схема

Автоматично згенерований опис

Зображення, що містить схема, знімок екрана, текст, дизайн

Автоматично згенерований опис

Виконала: Павленко Д.А.

Група: ІКМ-М223в