**Лабораторна робота №9**

Зменшення розмірів за допомогою SVD

1. Використання PCA для візуалізації даних у двовимірному та тривимірному (2D та 3D) просторах. Використовуйте клас PCA для бібліотеки sklearn <https://scikitlearn.org/stable/modules/generated/sklearn.decomposition.PCA.html>.

Рішення:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import TruncatedSVD

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, save\_as):

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.savefig(save\_as, bbox\_inches='tight') # Save the plot as JPG

plt.close() # Close the plot to release memory

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

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.savefig(save\_as, bbox\_inches='tight') # Save the plot as JPG

plt.close() # Close the plot to release memory

# PCA зменшення до 2D

pca\_2d = PCA(n\_components=2)

X\_pca\_2d = pca\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_pca\_2d, y, 'PCA 2D', 'pca\_2d.jpg')

# PCA зменшення до 3D

pca\_3d = PCA(n\_components=3)

X\_pca\_3d = pca\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_pca\_3d, y, 'PCA 3D', 'pca\_3d.jpg')

# SVD (TruncatedSVD) зменшення до 2D

svd\_2d = TruncatedSVD(n\_components=2)

X\_svd\_2d = svd\_2d.fit\_transform(X\_scaled)

plot\_2d(X\_svd\_2d, y, 'SVD 2D', 'svd\_2d.jpg')

# SVD (TruncatedSVD) зменшення до 3D

svd\_3d = TruncatedSVD(n\_components=3)

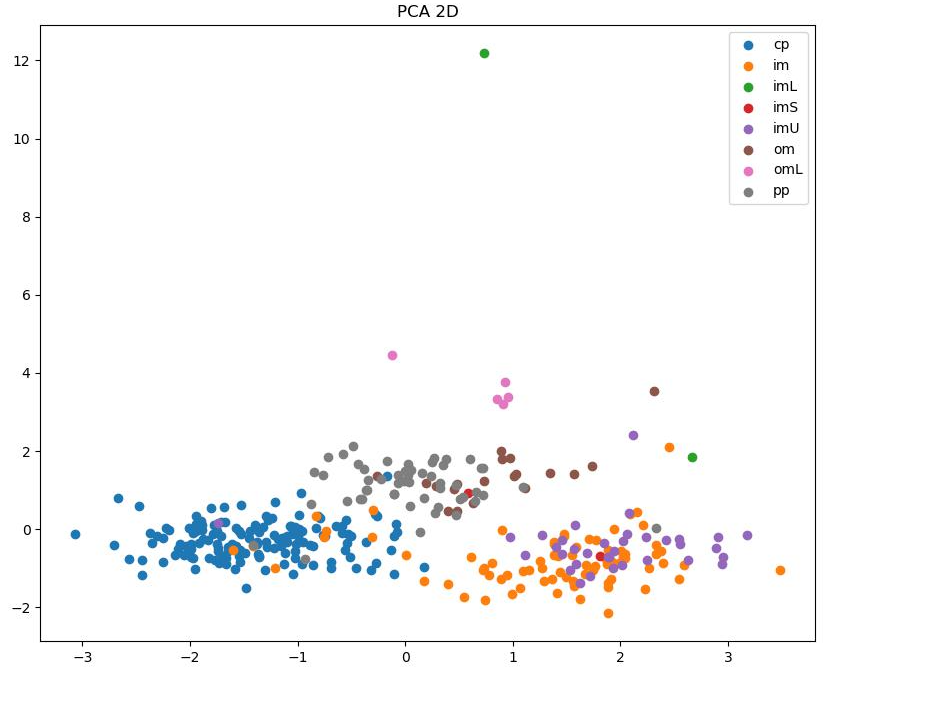
X\_svd\_3d = svd\_3d.fit\_transform(X\_scaled)

plot\_3d(X\_svd\_3d, y, 'SVD 3D', 'svd\_3d.jpg')

# Перевірка збереження дисперсії для PCA

print("PCA explained variance ratio (2D):", np.sum(pca\_2d.explained\_variance\_ratio\_))

print("PCA explained variance ratio (3D):", np.sum(pca\_3d.explained\_variance\_ratio\_))



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

Автоматично згенерований опис

Зображення, що містить знімок екрана, Барвистість

Автоматично згенерований опис

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

Автоматично згенерований опис

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

Автоматично згенерований опис

Виконала: Павленко Д.А.

Група: ІКМ-М223в