Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра комп’ютерного моделювання процесів і систем

ЗВІТ

з лабораторної роботи №11

“Зниження розмірності за допомогою PCA та SVD”

з курсу

«Алгоритми та моделі збору, аналізу та візуалізації даних»

Виконав: студент групи ІКМ-М222к  Черкас Ю.В.

Перевірила: аспірантка  Рикова В.О.

Харків 2023р

**Варіант №15**

В роботі для зниження розмірності використовується бібліотека scikit-learn <https://scikit-learn.org/stable/modules/manifold.html>

**Виконання**

Для відповідного датасету згідно з варіантом виконати пониження розмірності даних за допомогою PCA та SVD. Датасети розміщенні в теці datasets (<https://github.com/a-vodka/dv/tree/master/lab/dataset> ).

1. Використовуючи PCA візуалізувати данні у просторах з розмірностями два та три (2D та 3D).

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.decomposition import PCA

data = pd.read\_csv('Wholesale customers data.csv')

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

y = data.iloc[:,-1].values

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

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

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

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

x\_min, x\_max = np.min(X\_2d), np.max(X\_2d)

X\_2d = (X\_2d - x\_min) / (x\_max - x\_min)

x\_min, x\_max = np.min(X\_3d), np.max(X\_3d)

X\_3d = (X\_3d - x\_min) / (x\_max - x\_min)

plt.figure()

plt.scatter(X\_2d[:,0], X\_2d[:,1], c = y)

plt.title('PCA 2D')

plt.figure()

plt.subplot(111, projection = '3d')

plt.scatter(X\_3d[:,0], X\_3d[:,1], X\_3d[:,2], c = y)

plt.title('PCA 3D')

plt.show()

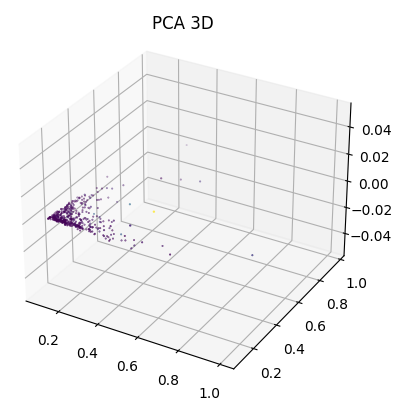
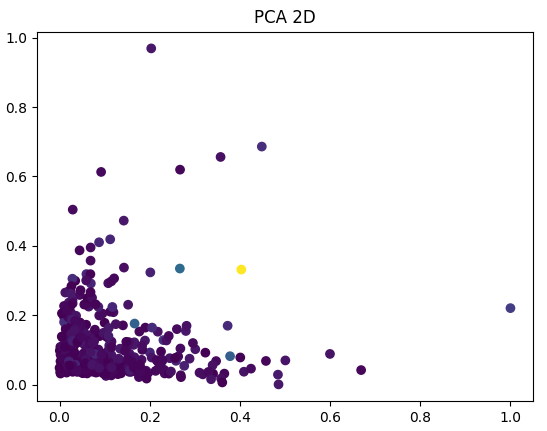


Рисунок 1 – 2D та 3D візуалізація даних за допомогою PCA

1. Використовуючи SVD, побудувати графік залежності власних значень матриці від їх номеру. Перед побудовою графіку впорядкувати власні значення у спадаючому порядку.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.decomposition import TruncatedSVD

data = pd.read\_csv('Wholesale customers data.csv')

svd = TruncatedSVD(n\_components=data.shape[1])

svd.fit(data)

own\_values = svd.singular\_values\_

idx\_sorted = np.argsort(own\_values)[::-1]

own\_values\_sorted = own\_values[idx\_sorted]

plt.plot(own\_values\_sorted, marker="\*")

plt.xlabel('Index')

plt.ylabel('Own Value')

plt.grid(True)

plt.show()

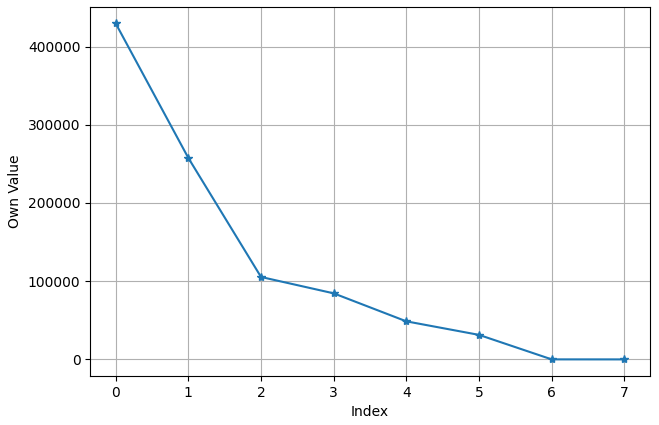


Рисунок 2 – Графік залежності власних значень матриці від їх номеру

1. Визначити таке найменше значення розміру простору *d*, для якого виконується співвідношення (1). Де λi – власні значення матриці, *n* – загальна кількість власних значень.

(1)

1. Занулити λi, для яких *d* ≤ *i* ≤ *n.* Виконати зворотне перетворення та порівняти отримані данні з вихідними. За можливості побудувати візуалізацію отриманих даних після зворотного перетворення.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.decomposition import TruncatedSVD

data = pd.read\_csv('Wholesale customers data.csv').iloc[:,2:].values

svd = TruncatedSVD(n\_components=data.shape[1]-1)

X\_svd = svd.fit\_transform(data)

own\_values = svd.singular\_values\_

total = np.sum(own\_values)

target\_dimenstion\_size = 0

for i in range(len(own\_values)):

    current\_sum = np.sum(own\_values[:i+1])

    ratio = current\_sum / total

    if ratio >= 0.8:

        break

    target\_dimenstion\_size = i

print('Target dimenstion size: ', i)

own\_values[target\_dimenstion\_size-1:] = 0

svd.singular\_values\_ = own\_values

data\_inverse = svd.inverse\_transform(X\_svd)

np.set\_printoptions(precision=1)

print('Original dataset:')

print(data[:3, :])

print('Restored dataset:')

print(data\_inverse[:3, :])

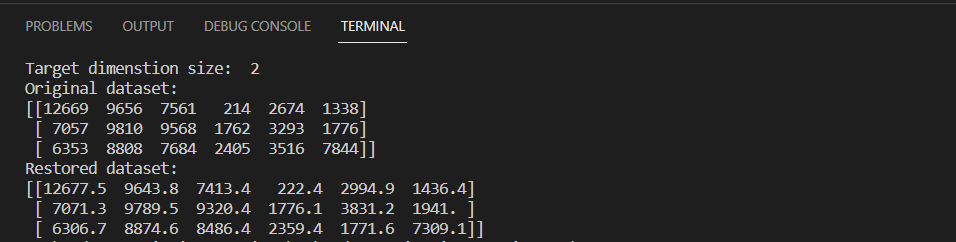
plt.figure()

plt.plot(data)

plt.figure()

plt.plot(data\_inverse)

plt.show()



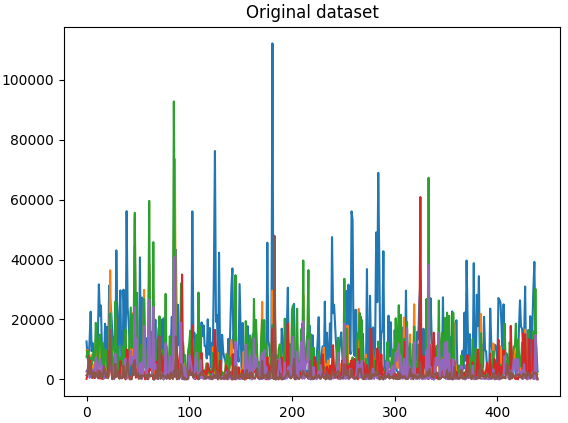


Рисунок 3 – Графік рядів значень початкового набору даних

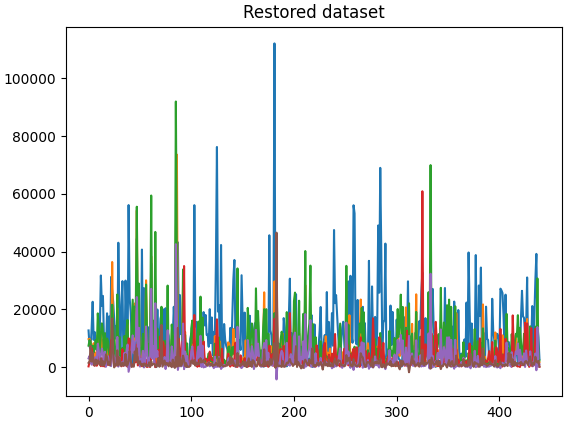


Рисунок 4 – Графік рядів значень відновленого набору даних

**Висновок:** на даній лабораторній роботі ми провели аналіз даних високої розмірності за допомогою PCA та SVD. Навчилися будувати графік залежності власних значень матриці від їх номеру. Дослідили вплив власних значень матриці при зворотному перетворені даних, а саме занулення тих власних значень, котрі мають мінімальний вплив.