МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
Національний технічний університет   
«Харківський політехнічний інститут»  
Кафедра ГМКГ

**Лабораторна работа №5**

З дисципліни «Інтелектуальний аналіз даних»

Виконав:

Студент групи ІКМ-220 г.

Ульянов Кирило Юрійович

Перевірив:

Доц. Дашкевич А.О.

Харків 2023

**Мета роботи:** : вивчення базових алгоритмiв зниження розмiрностi для задач кластеризацiї та вiзуалiзацiї даних.

**Завдання на роботу:** завантаження набору даних, зниження розмiрностi даних лiнiйними та нелiнiйними методам, знаходження способу зниження розмiрностi та оптимальної розмiрностi для розв’язання задачi кластеризацiї на наборi iз меншою розмiрнiстю, порiвняльний аналiз лiнiйного та нелiнiйного алгоритмiв.

**Варіант: 20**

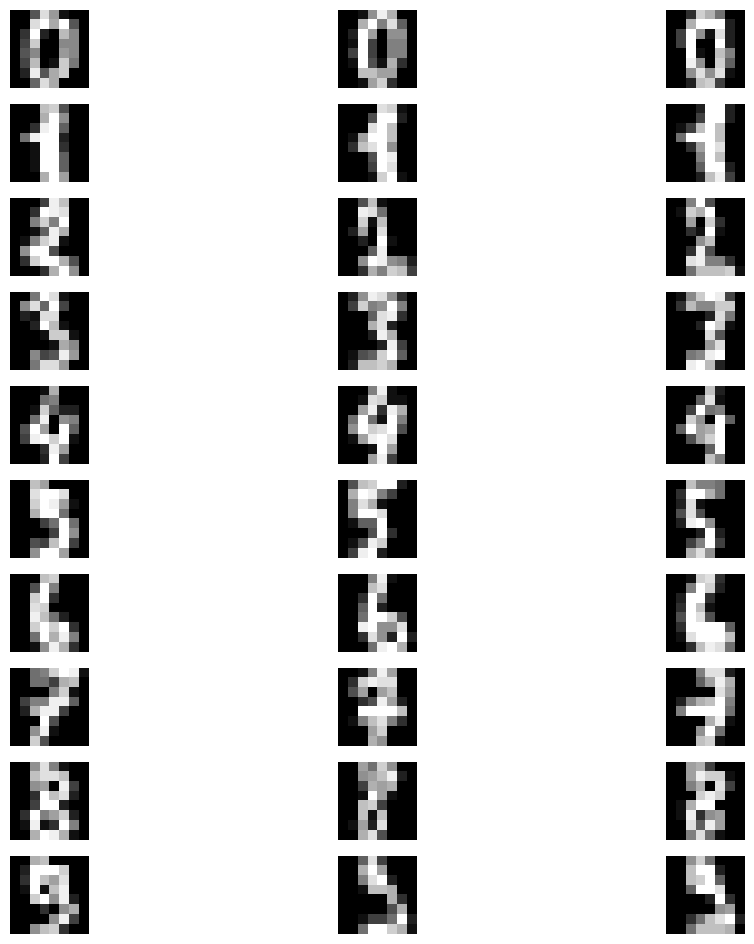
*N = 100*

*Метод: Isomap*

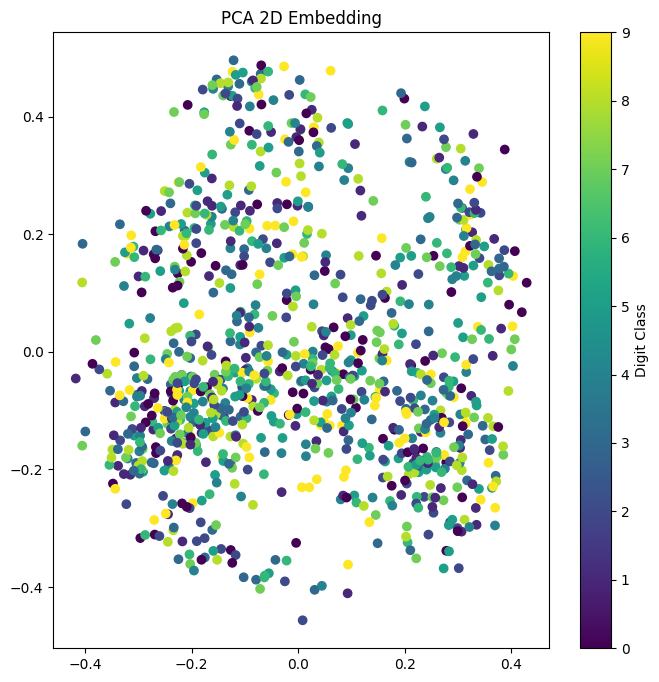
*D1 = 5*

*D2 = 15*

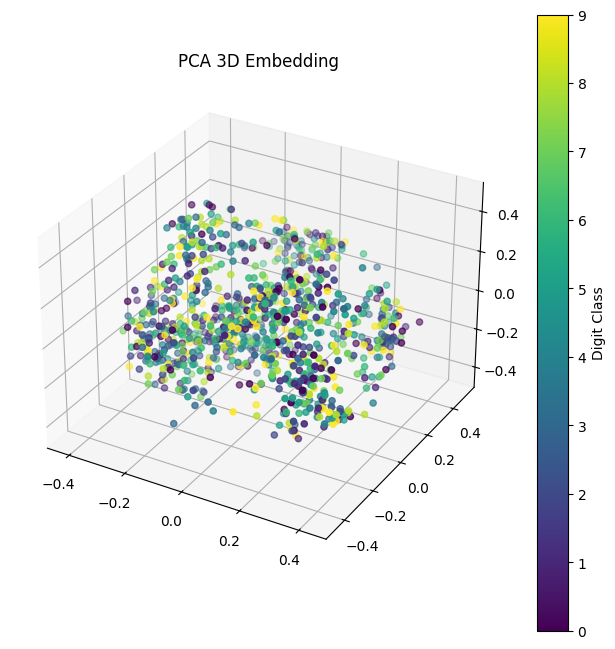
1. Візуалізація даних з якими будемо працювати.



1. Вкладення даних у простори розмірності 2 та 3 за допомогою методу головних компонентів (PCA)

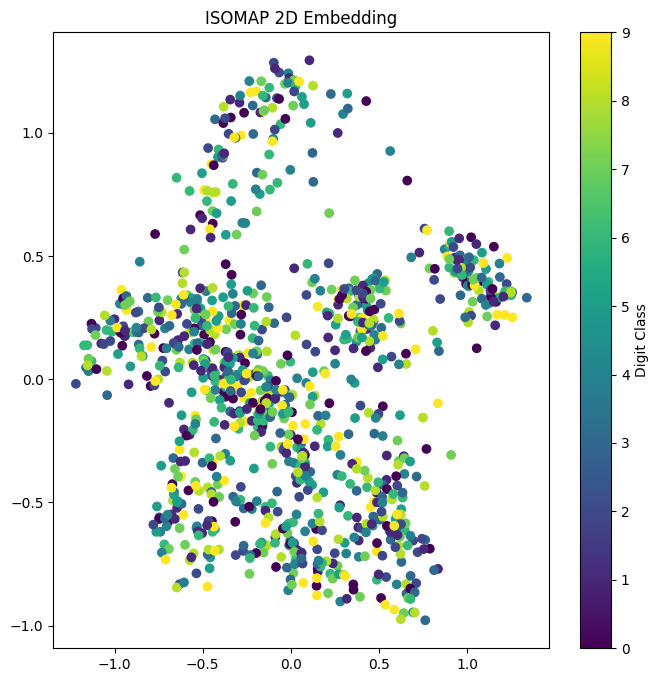


*Візуалізація у 2-вимірному просторі*

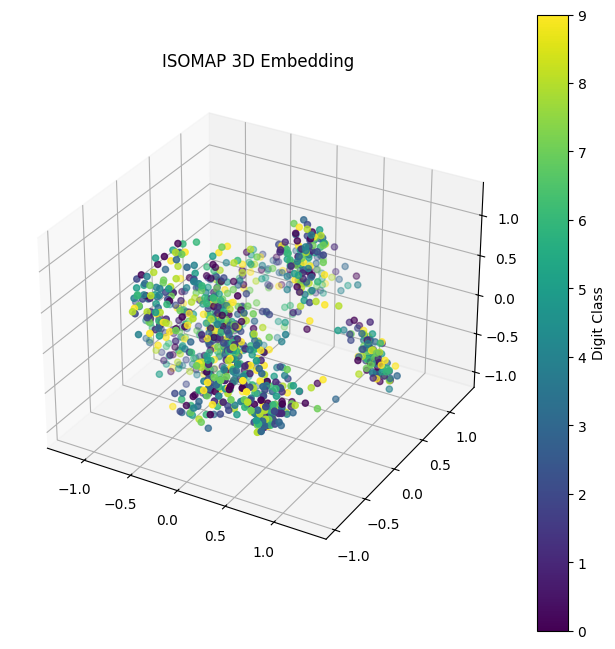


*Візуалізація у 3-вимірному просторі*

1. Вкладення даних у простори розмірності 2 та 3 за допомогою методу (ISOMAP)



*Візуалізація у 2-вимірному просторі*



*Візуалізація у 3-вимірному просторі*

1. Порівняльна таблиця результатів кластерізації вкладених даних за допомогою різних методів зниження розмірності даних.



|  |  |  |  |
| --- | --- | --- | --- |
| Кількість компонентів | точність з PCA | точність з ISOMAP | точність з T-SNE |
| 5 | 0.7611 | 0.7582 | 0.7704 |
| 6 | 0.7591 | 0.7609 | 0.7611 |
| 7 | 0.7611 | 0.7612 | 0.7590 |
| 8 | 0.7611 | 0.7613 | 0.7611 |
| 9 | 0.7611 | 0.7613 | 0.7591 |
| 10 | 0.7590 | 0.7613 | 0.7611 |
| 11 | 0.7612 | 0.7591 | 0.7590 |
| 12 | 0.7587 | 0.7611 | 0.7612 |
| 13 | 0.7611 | 0.7584 | 0.7584 |
| 14 | 0.7703 | 0.7611 | 0.7611 |
| 15 | 0.7611 | 0.7612 | 0.7704 |

Можна побачити що в цілому не сильно змінювалася точність алгоритмів на визначеному діапазоні. Але все ж таки на деяких проміжках найкращу точність отримали алгоритми TSNE та PCA. Помітно, що при зміни кількості компонент не сильно змінюється точність кластерізатору KMeans.

**Код програми:**

*# %%*

**from** sklearn.cluster **import** KMeans

**from** sklearn.datasets **import** load\_digits

**from** sklearn.preprocessing **import** Normalizer

**from** sklearn.metrics **import** rand\_score

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** sklearn.decomposition **import** PCA

**from** sklearn.manifold **import** Isomap

**from** sklearn.manifold **import** TSNE

*# %%*

digits = load\_digits()

*# кількіть елементів класу*

N = 100

selected\_data = np.empty((0, digits.data.shape[1]))

**for** i **in** range(10):

    digit\_data = digits.data[digits.target == i][:N]

    selected\_data = np.concatenate((selected\_data, digit\_data), axis=0)

num\_samples\_per\_class = 3

*# Нормалізація*

normalized\_data = Normalizer().fit\_transform(selected\_data)

*# Вивід на єкран прикладів*

fig, axs = plt.subplots(10, num\_samples\_per\_class, figsize=(12, 12))

**for** i **in** range(10):

**for** j **in** range(num\_samples\_per\_class):

        index = i \* N + j

        axs[i, j].imshow(normalized\_data[index].reshape(8, 8), cmap='gray')

        axs[i, j].axis('off')

plt.show()

**def** **plot\_methods**(data, method\_name=None, visualization\_type="2d"):

    """

    Візуалізація даних за допомогою методів ISOMAP та PCA в дво- або тривимірному просторі.

    Параметри:

    - `data (numpy.ndarray)`: Дані для візуалізації. Має мати форму (n\_samples, n\_features).

    - `method\_name (str)`: Назва методу, яка буде використана в заголовку графіка.

    - `visualization\_type (str)`: Тип візуалізації, "2d" або "3d".

    Повертає:

    None

    """

*# Визуализация в двумерном пространстве ISOMAP*

**if** visualization\_type == "2d":

        plt.figure(figsize=(8, 8))

        plt.scatter(data[:, 0], data[:, 1], c=digits.target[: N \* 10])

        plt.colorbar(label="Digit Class")

        plt.title(f"{method\_name} 2D Embedding")

        plt.show()

*# Визуализация в трехмерном пространстве ISOMAP*

**elif** visualization\_type == "3d":

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

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

        scatter = ax.scatter(

            data[:, 0], data[:, 1], data[:, 2], c=digits.target[: N \* 10]

        )

        ax.set\_title(f"{method\_name} 3D Embedding")

        plt.colorbar(scatter, label="Digit Class")

        plt.show()

*# ініціалізація методу головних компонентів для 2д та 3д простору*

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

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

*# застосування методу головних компонентів для 2д та 3д простору*

data\_2d = pca\_2d.fit\_transform(normalized\_data)

data\_3d = pca\_3d.fit\_transform(normalized\_data)

plot\_methods(data\_2d, "PCA")

plot\_methods(data\_3d, "PCA", "3d")

*# Ініціалізація моделі ISOMAP для просторів*

isomap\_2d = Isomap(n\_components=2, n\_neighbors=30)

isomap\_3d = Isomap(n\_components=3, n\_neighbors=30)

*# Застосування ISOMAP*

data\_2d\_isomap = isomap\_2d.fit\_transform(normalized\_data)

data\_3d\_isomap = isomap\_3d.fit\_transform(normalized\_data)

plot\_methods(data\_2d\_isomap, "ISOMAP")

plot\_methods(data\_3d\_isomap, "ISOMAP", "3d")

*# задаємо діапазон d*

d\_range = range(5, 16)

isomap\_score = []

**for** d **in** d\_range:

*# метод зниження розмірності даних Isomap*

    isomap\_data = Isomap(n\_components=d).fit\_transform(normalized\_data)

*# метод кластеризації KMeans з кількістю кластерів 7*

    kmeans = KMeans(n\_clusters=7)

    labels = kmeans.fit\_predict(isomap\_data)

    score = rand\_score(digits.target[:N \* 10], labels)

    isomap\_score.append(score)

pca\_score = []

**for** d **in** d\_range:

*# метод зниження розмірності даних PCA*

    pca\_data = PCA(n\_components=d).fit\_transform(normalized\_data)

*# метод кластеризації KMeans з кількістю кластерів 7*

    kmeans = KMeans(n\_clusters=7)

    labels = kmeans.fit\_predict(isomap\_data)

    score = rand\_score(digits.target[:N \* 10], labels)

    pca\_score.append(score)

tsne\_score = []

**for** d **in** d\_range:

*# метод зниження розмірності даних TSNE*

    tsne\_data = TSNE(n\_components=d, method="exact").fit\_transform(normalized\_data)

*# метод кластеризації KMeans з кількістю кластерів 7*

    kmeans = KMeans(n\_clusters=7)

    labels = kmeans.fit\_predict(isomap\_data)

    score = rand\_score(digits.target[:N \* 10], labels)

    tsne\_score.append(score)

*# візуалізація результатів*

plt.figure(figsize=(10, 5))

plt.scatter(d\_range, isomap\_score, label='ISOMAP')

plt.scatter(d\_range, pca\_score, label='PCA')

plt.scatter(d\_range, tsne\_score, label='TSNE')

plt.title("Оцінка якості кластреризації з вкладеними даними")

plt.xlabel("Number of components")

plt.ylabel("RandScore")

plt.legend()

plt.show()

print(f'Кількість компонентів, точність з ISOMAP, точність з PCA, точність з T-SNE')

*# вивід оцінок*

**for** i, j **in** zip(d\_range, range(0, 11)):

    print(f'{i}, {isomap\_score[j]:.4f}, {pca\_score[j]:.4f}, {tsne\_score[j]:.4f}')