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 алгоритмів зниження розмірності для задач кластеризації та візуалізації даних.

Завдання на роботу: завантаження набору даних, зниження розмірності даних лінійними та нелінійними методам, знаходження способу зниження розмірності та оптимальної розмірності для розв’язання задачі кластеризації на наборі із меншою розмірністю, порівняльний аналіз лінійного та нелінійного алгоритмів.

Варіант: 20

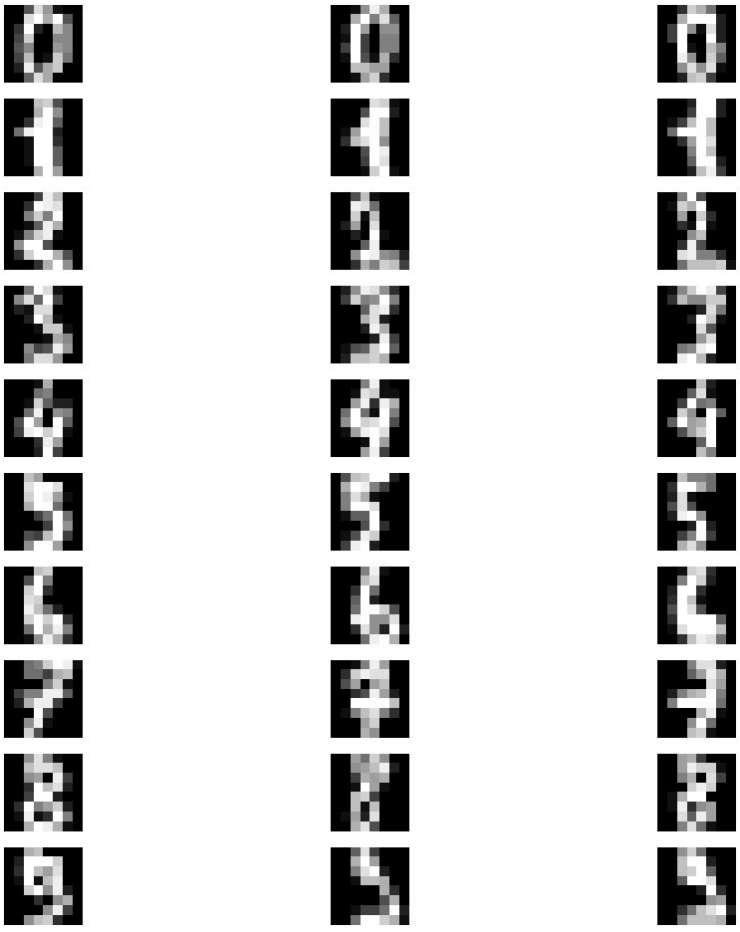
N = 100

Метод: Isomap

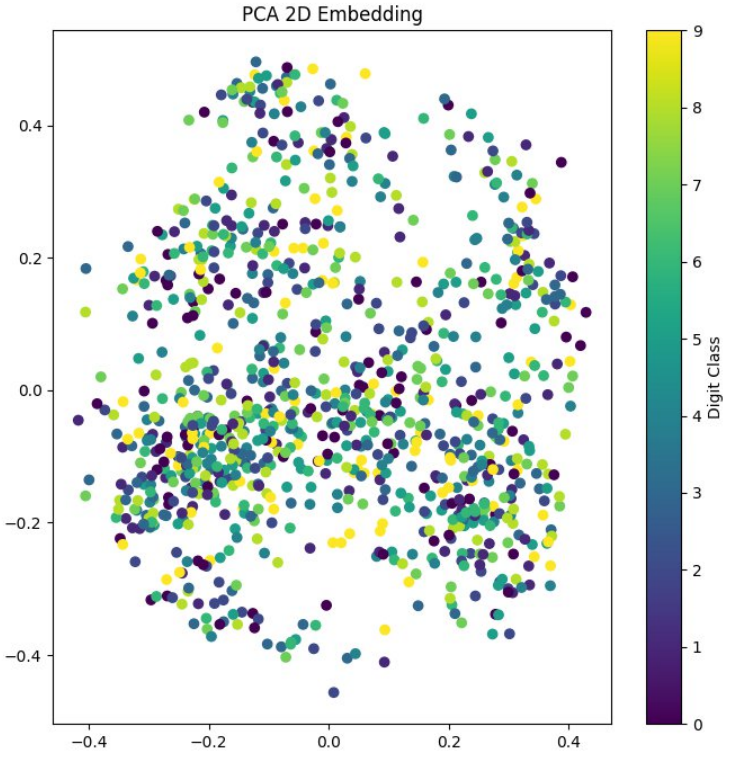
D1 = 5

D2 = 15

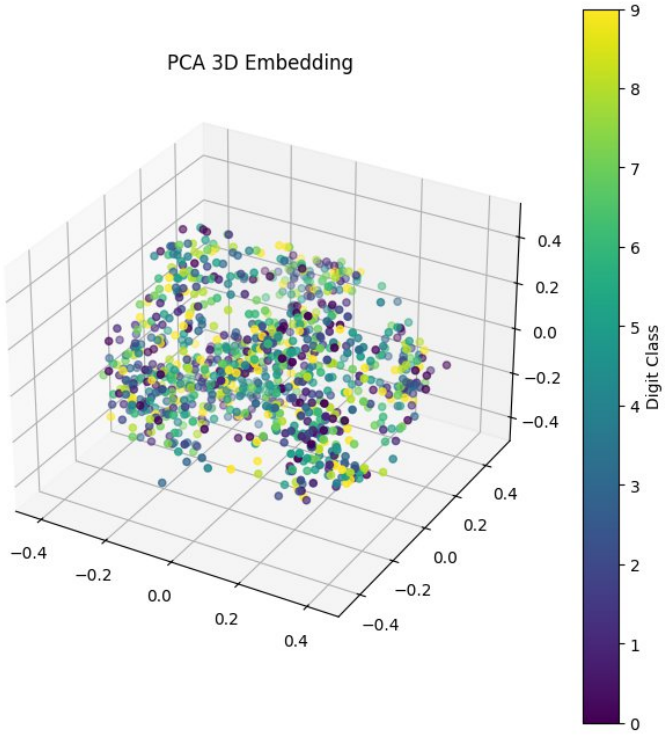
1) Візуалізація даних з якими будемо працювати.



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

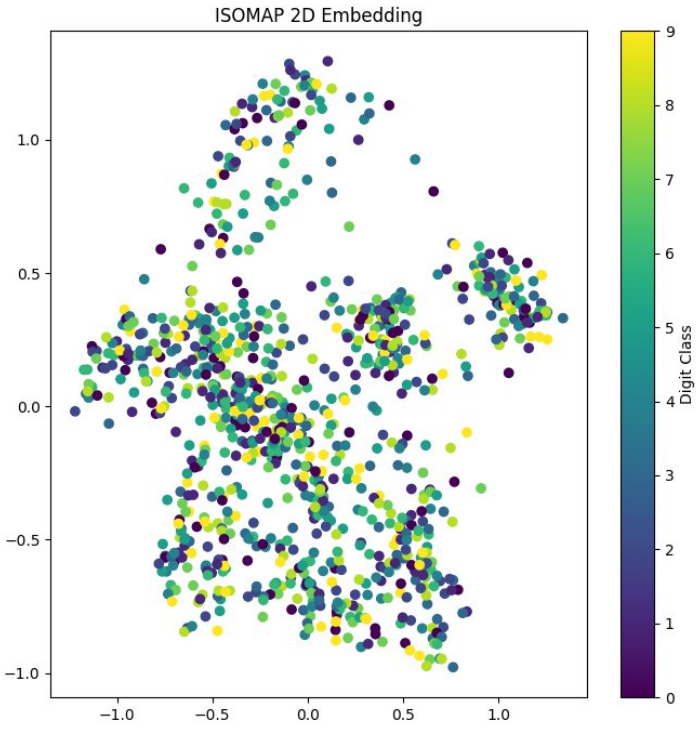


Візуалізація у 2-вимірному просторі

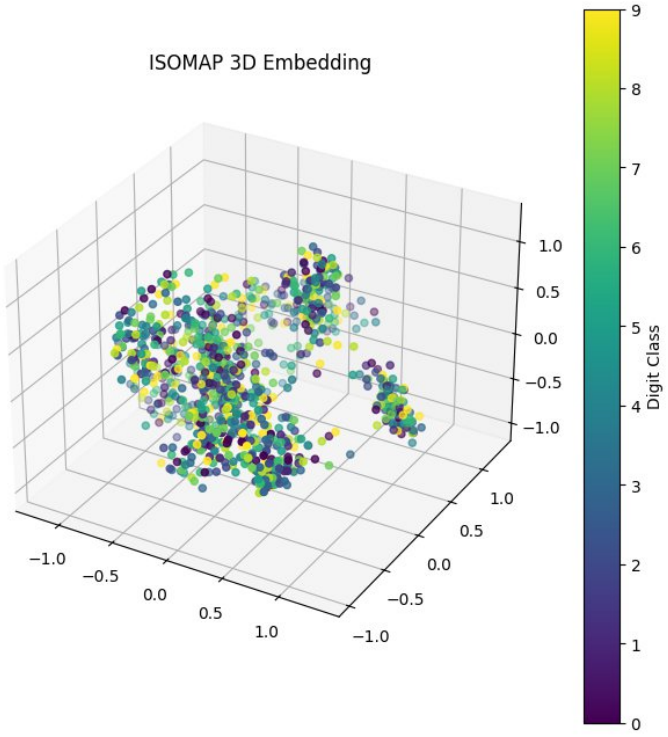


Візуалізація у 3-вимірному просторі

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

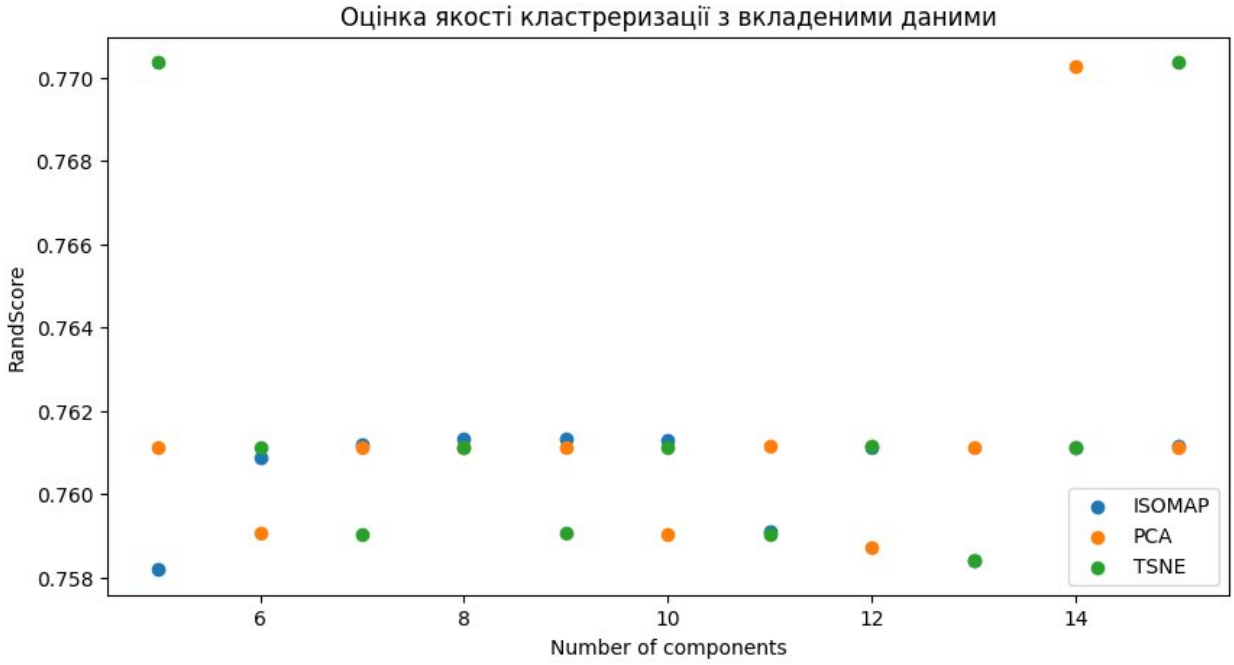


Візуалізація у 2-вимірному просторі



Візуалізація у 3-вимірному просторі

4) 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 екран 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}')
```