Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Звіт

Про виконання лабораторної роботи №5

З курсу «“Системи опрацювання даних**»**

«Кластеризація даних»

Виконав:

Студент групи ФеС-21

Шавало Андрій

Львів-2025

Мета роботи: Ознайомитися з методами кластеризації даних, реалізувати та порівняти алгоритми кластеризації (KMeans, DBSCAN, Agglomerative Clustering) на реальному датасеті, а також оцінити якість отриманих клас

Теоретичні відомості: Ознайомитись з алгоритмом кластеризації к-means та прикладом його реалізації https://www.kaggle.com/kushal1996/customer-segmentation-k-meansanalysis/notebook

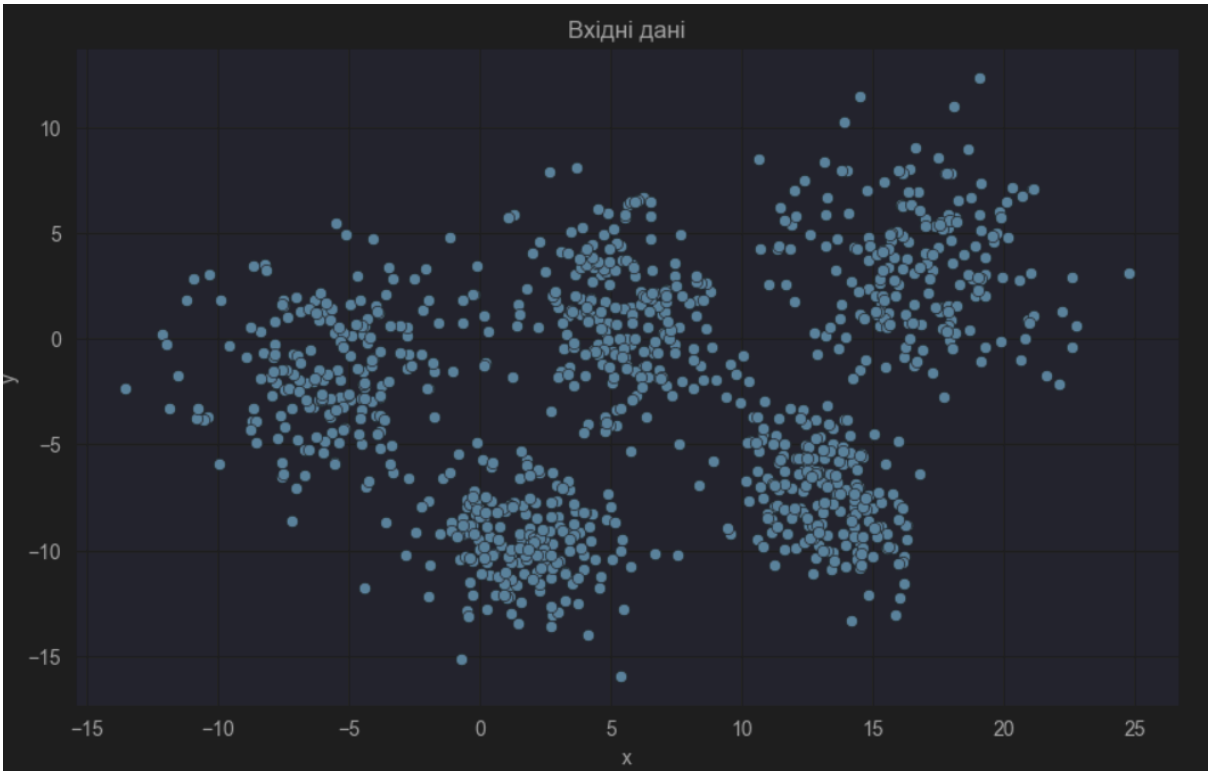
Хід роботи

Завдання 1. Підготовка даних о Завантажити набір даних о Виконати первинний аналіз даних: розмірність, типи змінних, наявність пропущених значень. о Провести масштабування (стандартизацію або нормалізацію) числових змінних.

```python
import ...

data = pd.read_csv('size_1000_n_5_sepval_0.1.csv')

print("Розмірність даних:", data.shape)
print("\nПерші 5 рядків:\n", data.head())
print("\nОпис даних:\n", data.describe())

plt.figure(figsize=(10, 6))
sns.scatterplot(x='x', y='y', data=data)
plt.title('Вхідні дані')
plt.show()

scaler = StandardScaler()
scaled_data = scaler.fit_transform(data[['x', 'y']])
[15]
```

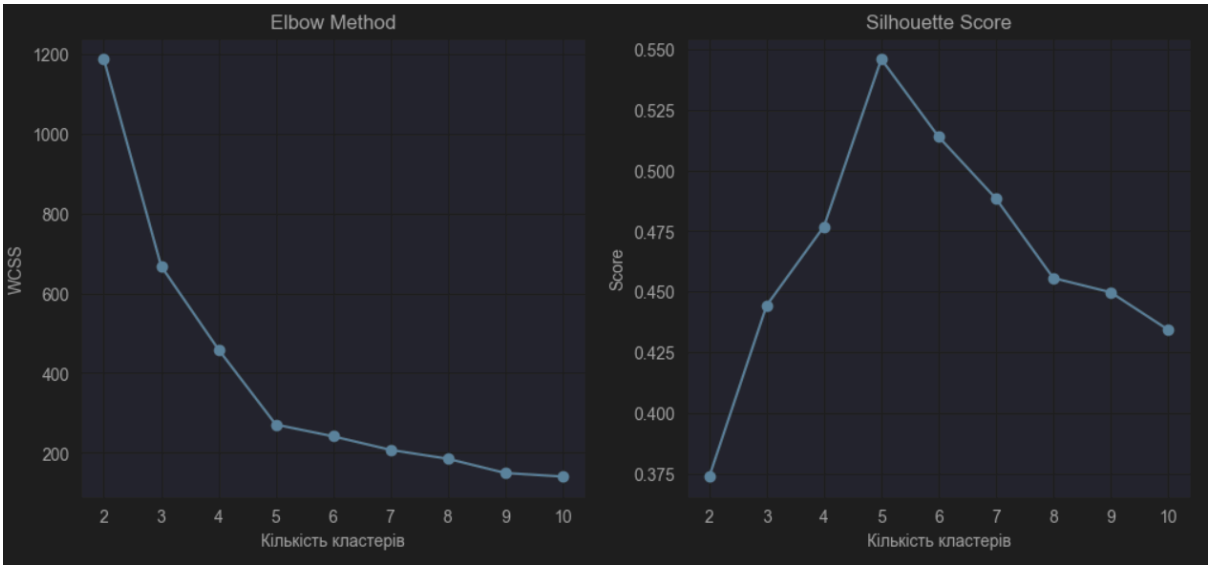
```
Розмірність даних: (1000, 4)

Перші 5 рядків:
   Unnamed: 0         x          y      run
0           1 -7.535642   1.787421  oupvCgP
1           2  4.351434  -0.506274  oupvCgP
2           3 -0.776635 -10.419561  oupvCgP
3           4 -4.529642   0.695028  oupvCgP
4           5 -5.107399   4.915450  oupvCgP
```

```
Опис даних:
        Unnamed: 0            x            y
count  1000.000000  1000.000000  1000.000000
mean    500.500000     6.207532    -2.752120
std     288.819436     8.245608     5.545930
min       1.000000   -13.548163   -15.939657
25%     250.750000     0.214693    -7.864758
50%     500.500000     5.446603    -2.192122
75%     750.250000    13.611509     1.606741
max    1000.000000    24.747021    12.345398
```



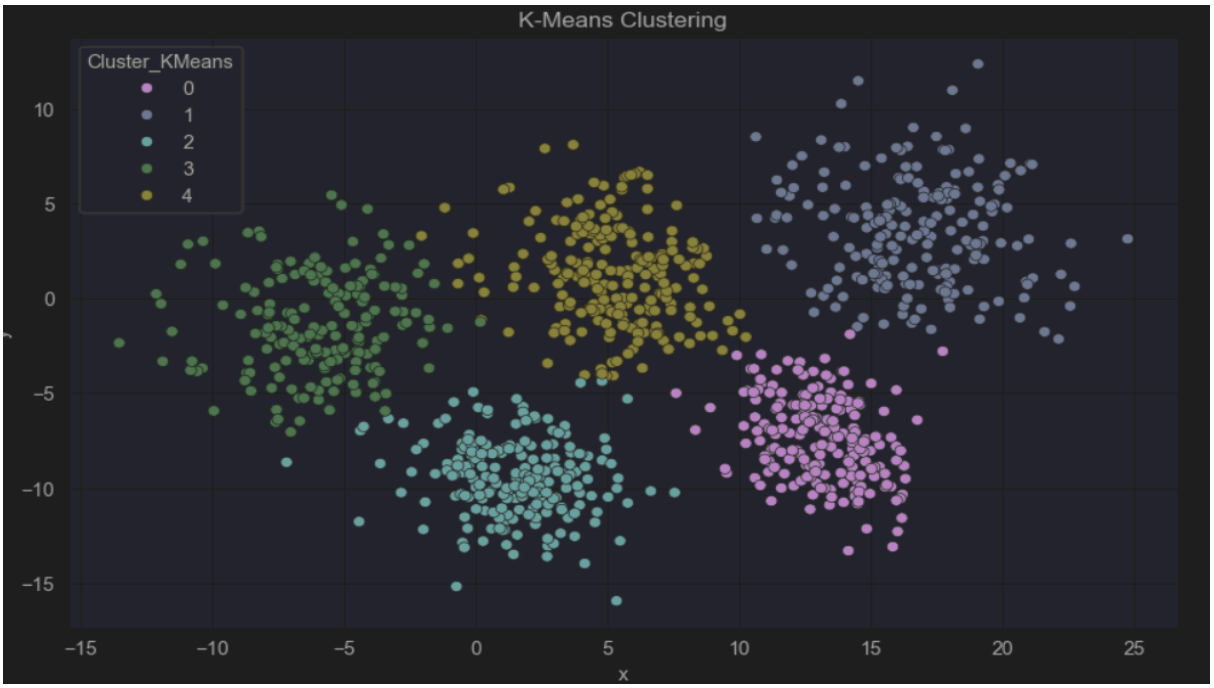
2. Реалізація кластеризації о Визначити оптимальну кількість кластерів за допомогою методу "Elbow Method" або Silhouette Score.

```python
wcss = []
silhouette_scores = []
for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(scaled_data, kmeans.labels_))

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(range(2, 11), wcss, marker='o')
plt.title('Elbow Method')
plt.xlabel('Кількість кластерів')
plt.ylabel('WCSS')

plt.subplot(1, 2, 2)
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.title('Silhouette Score')
plt.xlabel('Кількість кластерів')
plt.ylabel('Score')
plt.show()
```



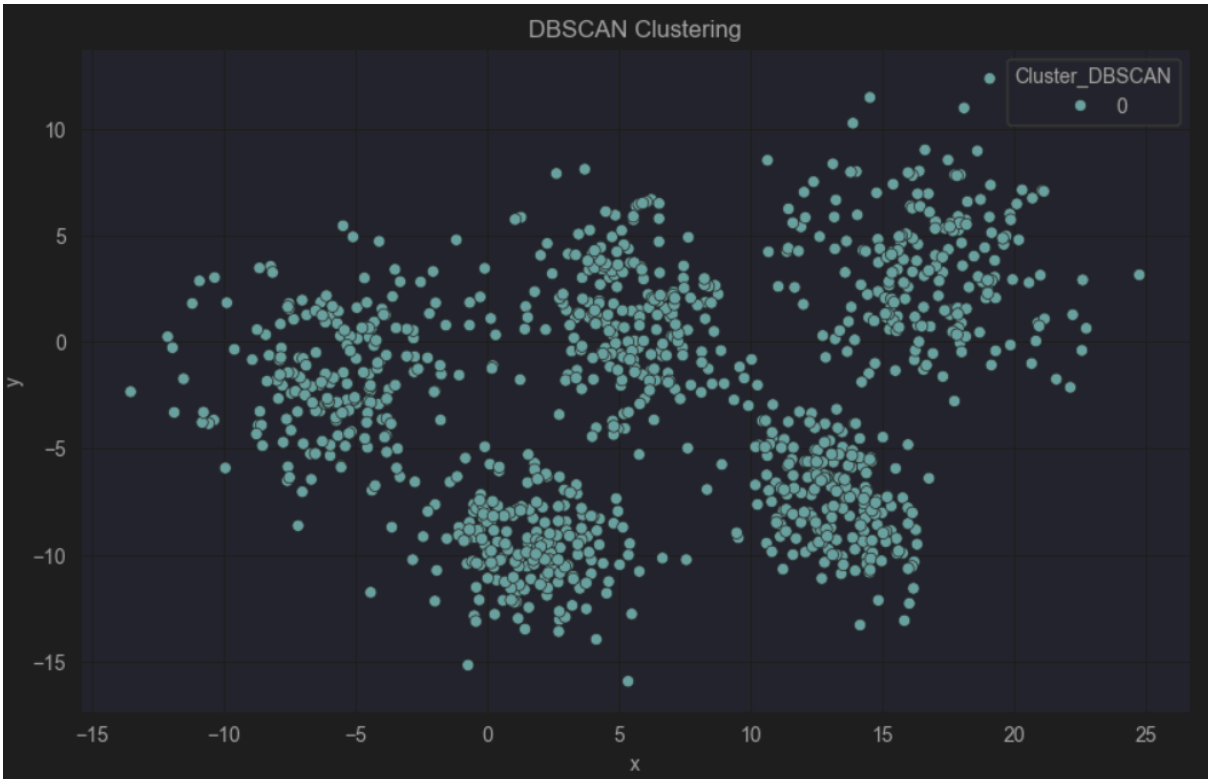
Виконати кластеризацію за допомогою K-Means.

```python
optimal_k = 5
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters_kmeans = kmeans.fit_predict(scaled_data)

data['Cluster_KMeans'] = clusters_kmeans

plt.figure(figsize=(10, 6))
sns.scatterplot(x='x', y='y', hue='Cluster_KMeans', data=data, palette='viridis')
plt.title('K-Means Clustering')
plt.show()
```



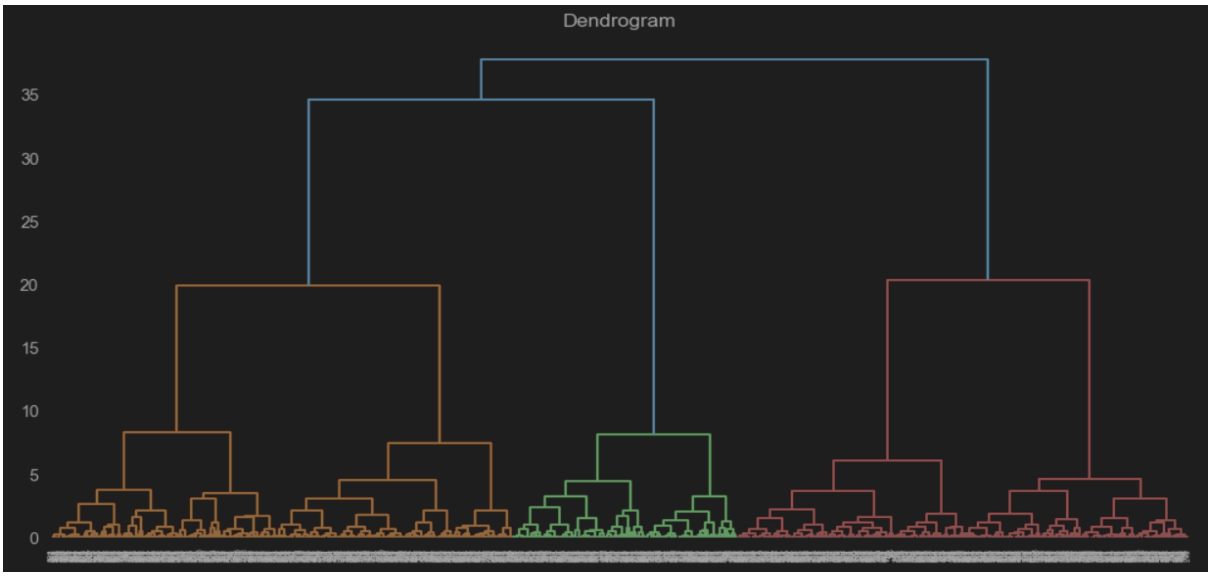
Реалізувати та застосувати DBSCAN для виявлення аномалій.

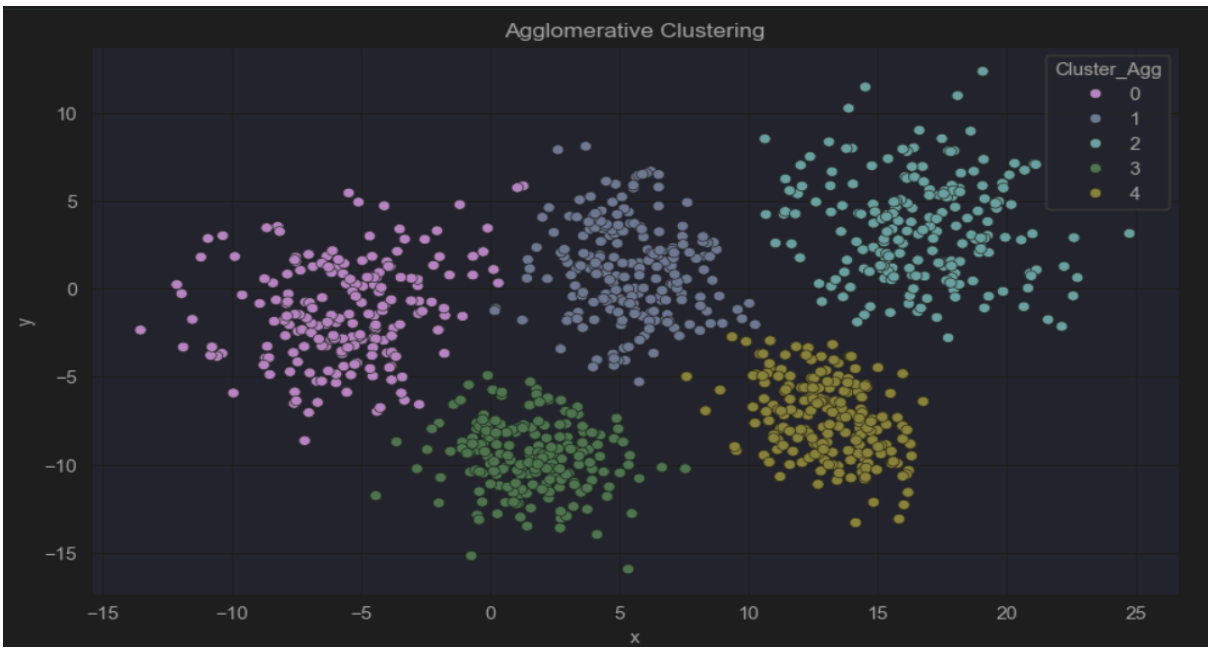
```python
dbscan = DBSCAN(eps=0.5, min_samples=5)
clusters_dbscan = dbscan.fit_predict(scaled_data)

data['Cluster_DBSCAN'] = clusters_dbscan

plt.figure(figsize=(10, 6))
sns.scatterplot(x='x', y='y', hue='Cluster_DBSCAN', data=data, palette='viridis')
plt.title('DBSCAN Clustering')
plt.show()

n_noise = list(clusters_dbscan).count(-1)
print(f"Кількість аномалій (шум): {n_noise}")
```



Використати Agglomerative Clustering (ієрархічну кластеризацію) та побудувати дендрограму.

```python
agg = AgglomerativeClustering(n_clusters=5)
clusters_agg = agg.fit_predict(scaled_data)
data['Cluster_Agg'] = clusters_agg

plt.figure(figsize=(12, 6))
linked = linkage(scaled_data, method='ward')
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.title('Dendrogram')
plt.show()

plt.figure(figsize=(10, 6))
sns.scatterplot(x='x', y='y', hue='Cluster_Agg', data=data, palette='viridis')
plt.title('Agglomerative Clustering')
plt.show()
```

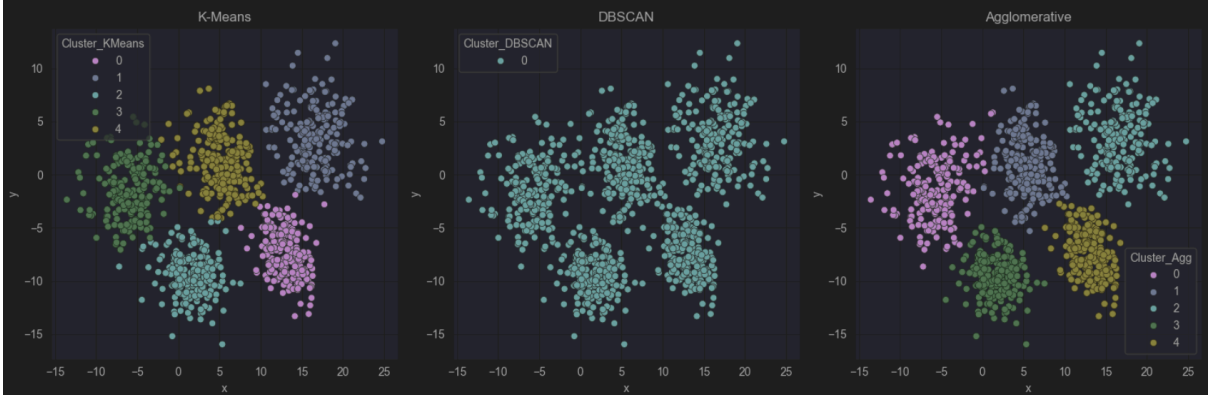




3. Оцінка якості кластеризації о Порівняти результати кластеризації за допомогою метрики Silhouette Score о Візуалізувати отримані кластери (наприклад, scatter plot, t-SNE, PCA). о Охарактеризувати кластери. (Наприклад, кластер тих хто часто робить недорогі покупки)

```python
print("Silhouette Scores:")
print(f"K-Means: {silhouette_score(scaled_data, clusters_kmeans)}")

unique_dbscan = np.unique(clusters_dbscan)
n_dbscan_clusters = len(unique_dbscan) - (1 if -1 in unique_dbscan else 0)

if n_dbscan_clusters > 1:
    print(f"DBSCAN: {silhouette_score(scaled_data, clusters_dbscan)}")
else:
    print("DBSCAN: не можна обчислити (виявлено лише 1 кластер)")

print(f"Agglomerative: {silhouette_score(scaled_data, clusters_agg)}")
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.scatterplot(x='x', y='y', hue='Cluster_KMeans', data=data, palette='viridis')
plt.title('K-Means')

plt.subplot(1, 3, 2)
sns.scatterplot(x='x', y='y', hue='Cluster_DBSCAN', data=data, palette='viridis')
plt.title('DBSCAN')

plt.subplot(1, 3, 3)
sns.scatterplot(x='x', y='y', hue='Cluster_Agg', data=data, palette='viridis')
plt.title('Agglomerative')

plt.tight_layout()
plt.show()
```

```
Silhouette Scores:
K-Means: 0.5458695136189783
DBSCAN: не можна обчислити (виявлено лише 1 кластер)
Agglomerative: 0.5383626078765015
```



Висновок:

У цій лабораторній роботі я ознайомився з методами кластеризації даних та реалізував три алгоритми: K-Means, DBSCAN і Agglomerative Clustering. Я виконав попередню обробку даних, включаючи масштабування, визначив оптимальну кількість кластерів за допомогою "Elbow Method" та Silhouette Score, а також провів аналіз результатів кластеризації.

Порівняння алгоритмів показало, що K-Means добре працює на чітко відокремлених групах даних, DBSCAN виявляє аномалії, але чутливий до параметрів, а Agglomerative Clustering дає гнучкіший розподіл, проте може страждати на великому обсязі даних. Візуалізація кластерів дозволила оцінити якість отриманих розподілів.