

## ЛАБОРАТОРНА РОБОТА № 7

### ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ

**Мета:** використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити різні методи неконтрольованої класифікації у машинному навчанні.

### Хід роботи:

Посилання на GitHub:

### Завдання 1

Кластеризація даних за допомогою методу k-середніх.

### Програмний код:

```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')

X = np.loadtxt('data_clustering.txt', delimiter=',')
num_clusters = 5

plt.figure()
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black', s=80)
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
plt.title('Вхідні дані')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

kmeans = KMeans(init='k-means++', n_clusters=num_clusters, n_init=10)
kmeans.fit(X)

step_size = 0.01
x_vals, y_vals = np.meshgrid(np.arange(x_min, x_max, step_size), np.arange(y_min, y_max, step_size))
output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])
output = output.reshape(x_vals.shape)
```

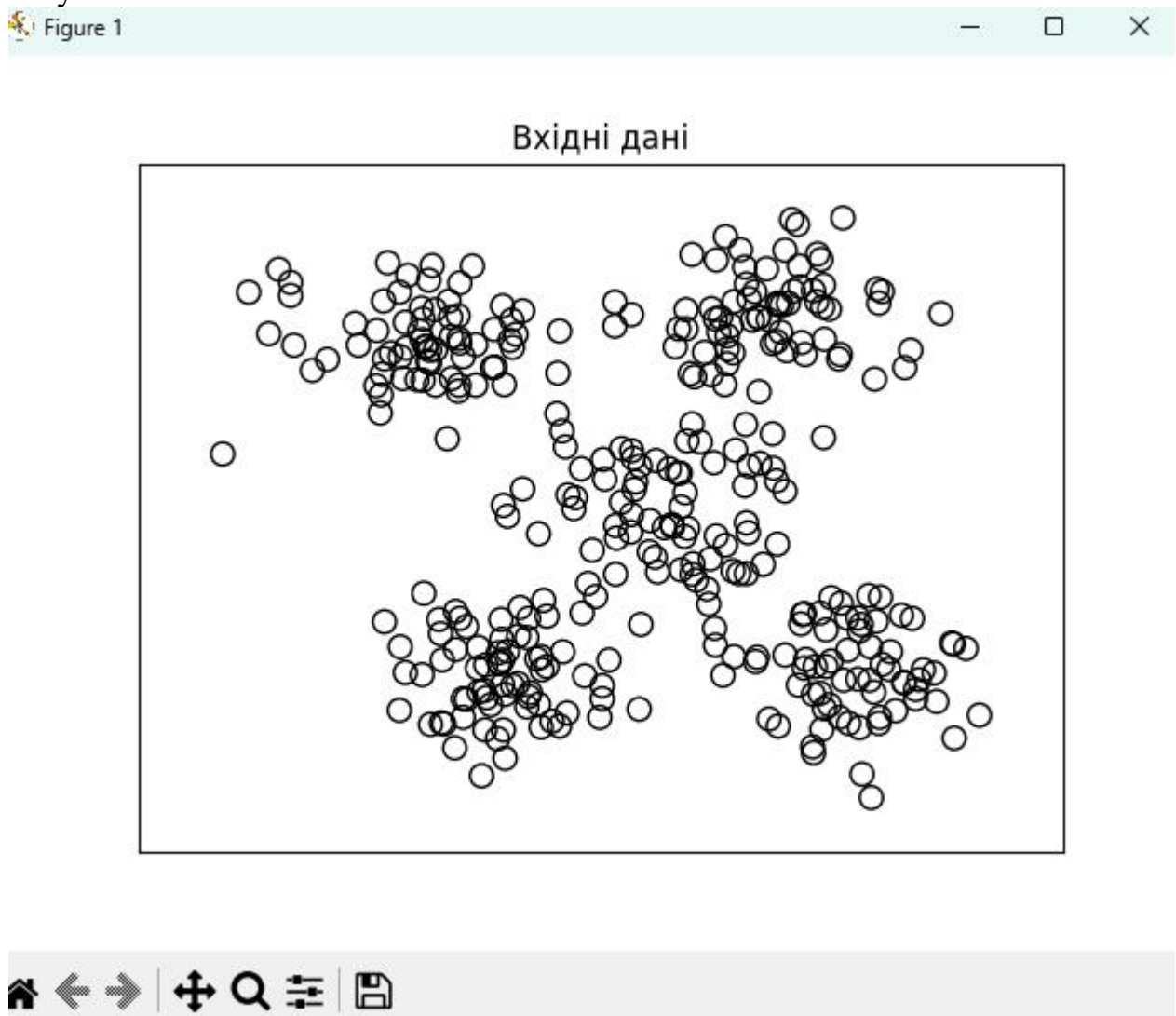
					ДУ «Житомирська політехніка».24.121.12.000 – Лр7			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Курач О.А.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Іванов Д.А.						1
Керівник							Аркушів	
Н. контр.							8	
Зав. каф.							ФІКТ Гр. ІПЗ-21-4	

```
plt.figure()
plt.clf()
plt.imshow(output, interpolation='nearest', extent=(x_vals.min(), x_vals.max(),
y_vals.min(), y_vals.max()),
cmap=plt.cm.Paired, aspect='auto', origin='lower')
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black',
s=80)

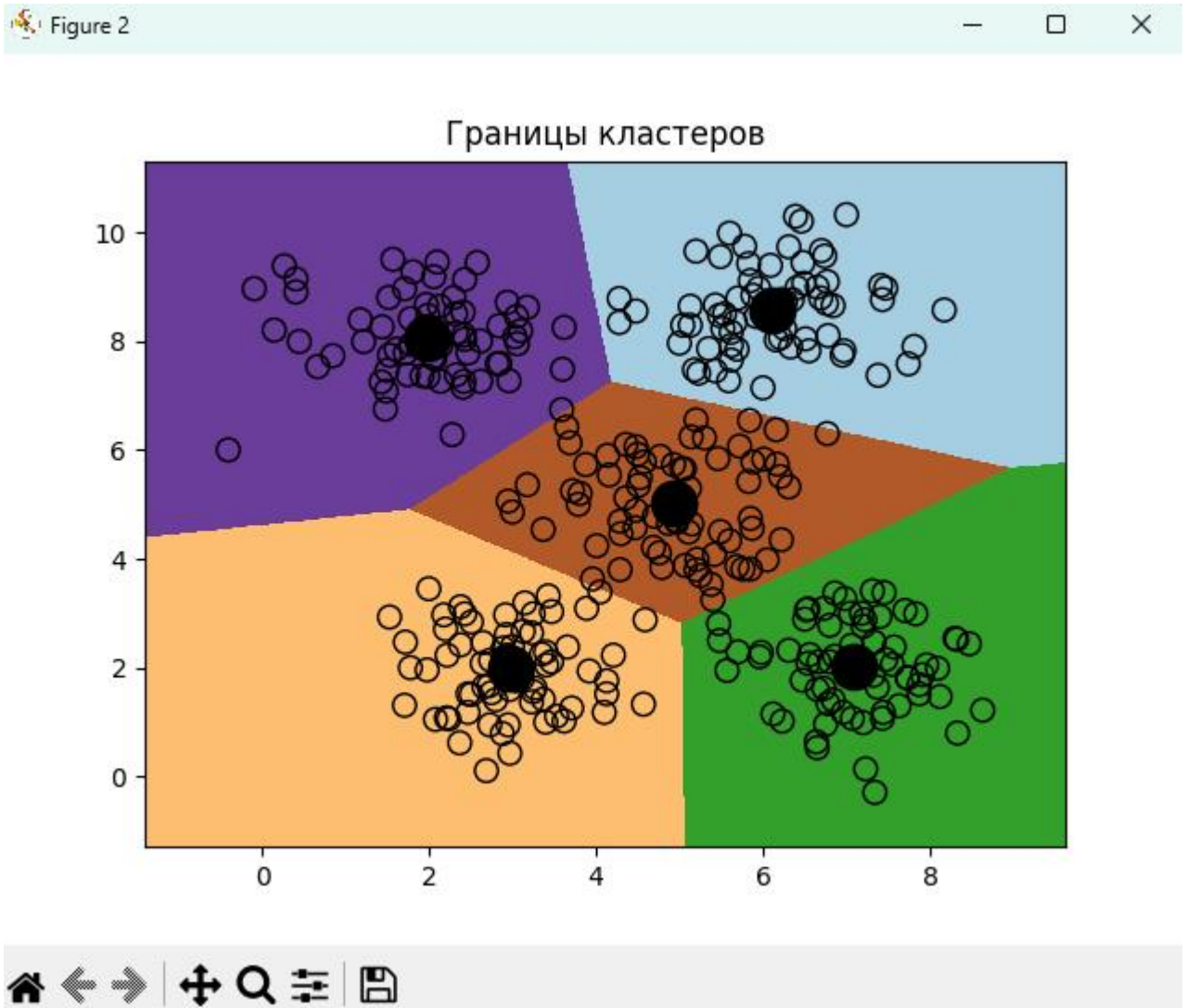
cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1], marker='o', s=210, lin-
ewidths=4, color='black', zorder=12,
facecolors='black')

plt.title('Границы кластеров')
plt.show()
```

Результат виконання:



		Курач О.А.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр7	Арк.
		Іванов Д.А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		



### Висновок:

Після виконання коду ми можемо зрозуміти, що наші дані складаються з п'яти груп.

### Завдання 2.

Кластеризація К-середніх для набору даних Iris

### Програмний код:

```
from sklearn.svm import SVC
from sklearn.metrics import pairwise_distances_argmin
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
import numpy as np

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')

iris = load_iris()
X = iris['data']
y = iris['target']
```

		Курач О.А.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр7	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.title("KMeans кластеризація")
plt.show()

def find_clusters(X, n_clusters, rseed=2):
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]
    while True:
        labels = pairwise_distances_argmin(X, centers)
        new_centers = np.array([X[labels == i].mean(0) for i in range(n_clusters)])
        if np.all(centers == new_centers):
            break
        centers = new_centers
    return centers, labels

centers, labels = find_clusters(X, 3)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.title("Ручна кластеризація")
plt.show()

centers, labels = find_clusters(X, 3, rseed=0)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.title("Ручна кластеризація (з іншим seed)")
plt.show()

labels = KMeans(3, random_state=0).fit_predict(X)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("KMeans кластеризація з 3 кластерами")
plt.show()

```

Результат виконання:

		Курач О.А.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр7	Арк.
		Іванов Д.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Figure 1

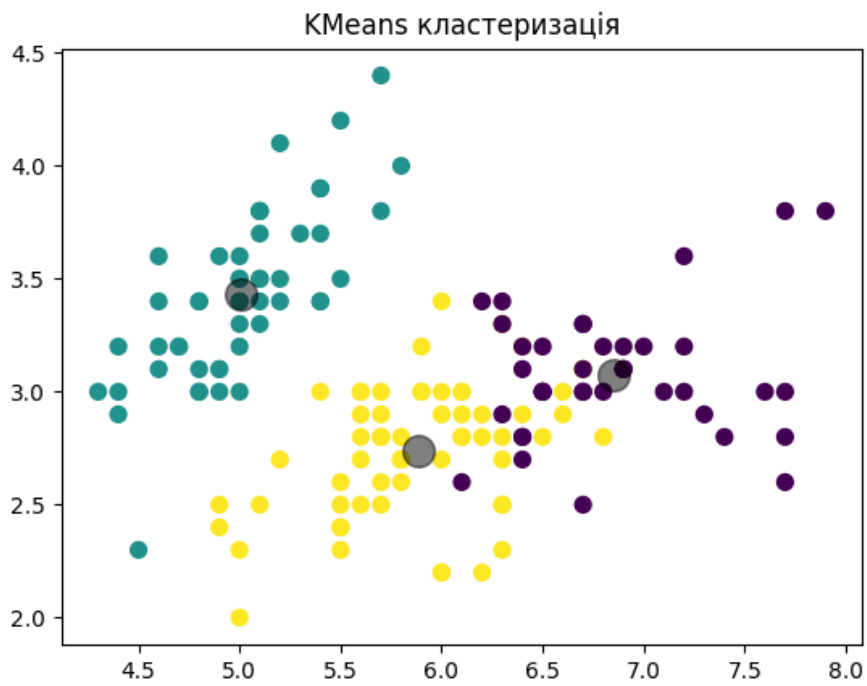


Figure 1

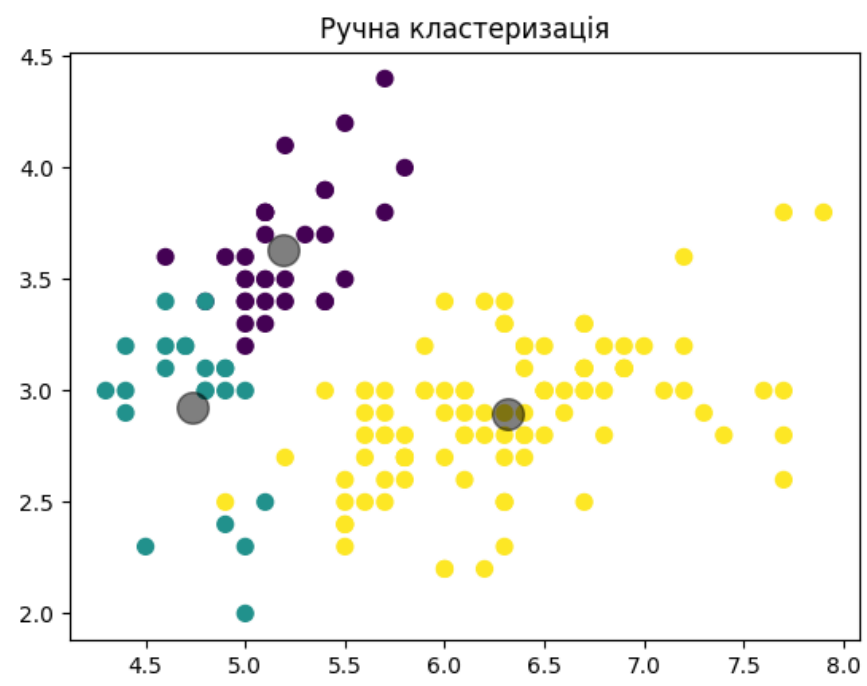


Figure 1

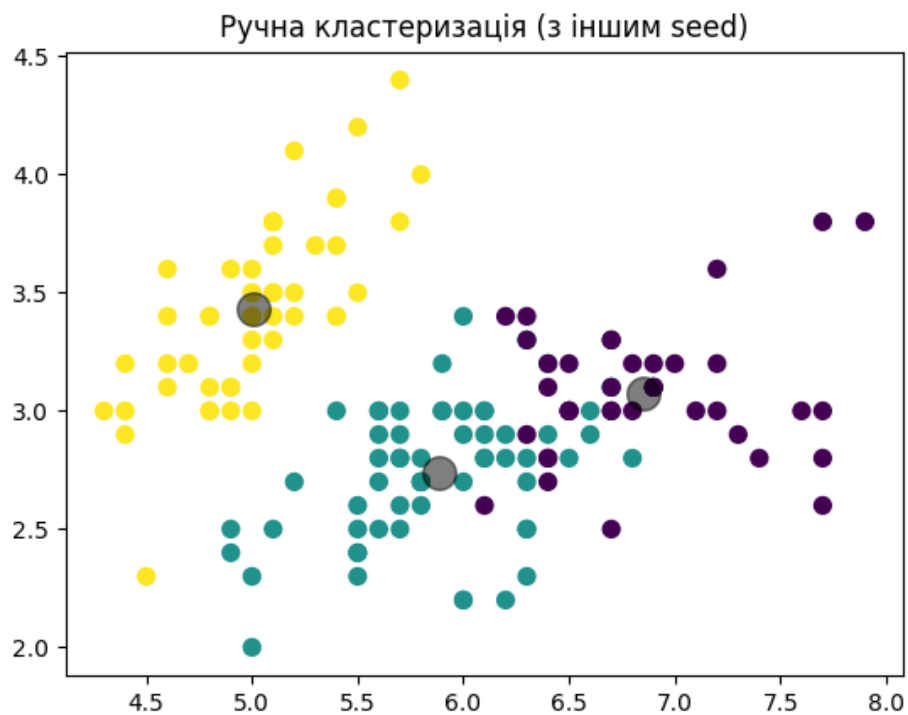
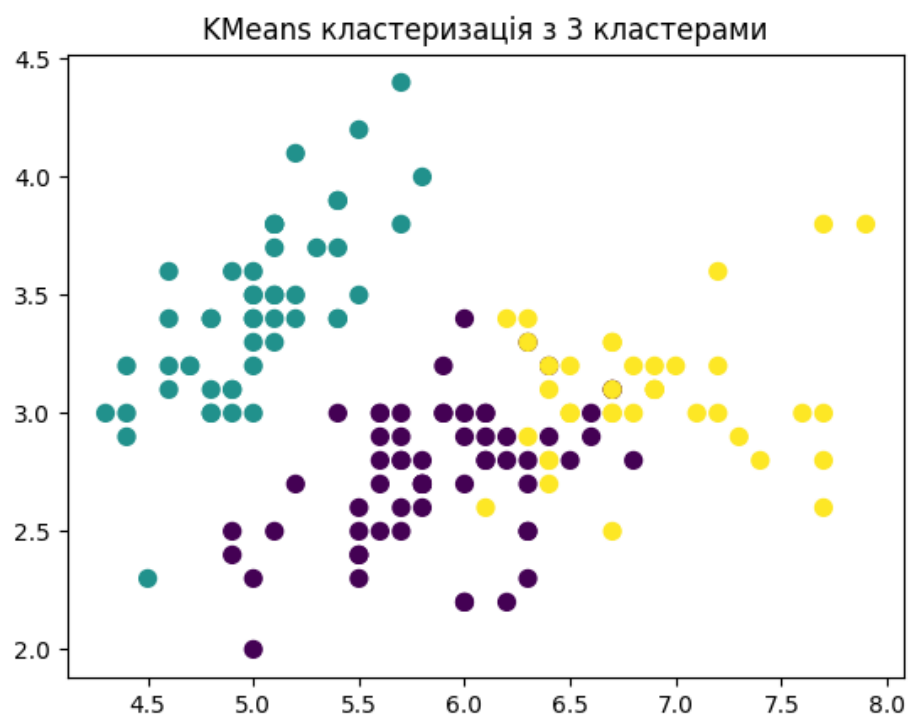


Figure 1



### Завдання 3. Оцінка кількості кластерів з використанням методу зсуву середнього

#### Програмний код:

```
import numpy as np
from sklearn.cluster import MeanShift, estimate_bandwidth
from itertools import cycle

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')

X = np.loadtxt('data_clustering.txt', delimiter=',')
bandwidth_X = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))
meanshift_model = MeanShift(bandwidth=bandwidth_X, bin_seeding=True)
meanshift_model.fit(X)
cluster_centers = meanshift_model.cluster_centers_
print('\nCenters of clusters: \n', cluster_centers)
labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)
plt.figure()
markers = 'o*xvs'
for i, marker in zip(range(num_clusters), markers):
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
                color='black')
cluster_center = cluster_centers[i]
plt.plot(cluster_center[0], cluster_center[1], marker='o',
         markerfacecolor='black', markeredgecolor='black', markersize=15)
plt.title('Кластеры')
plt.show()
```

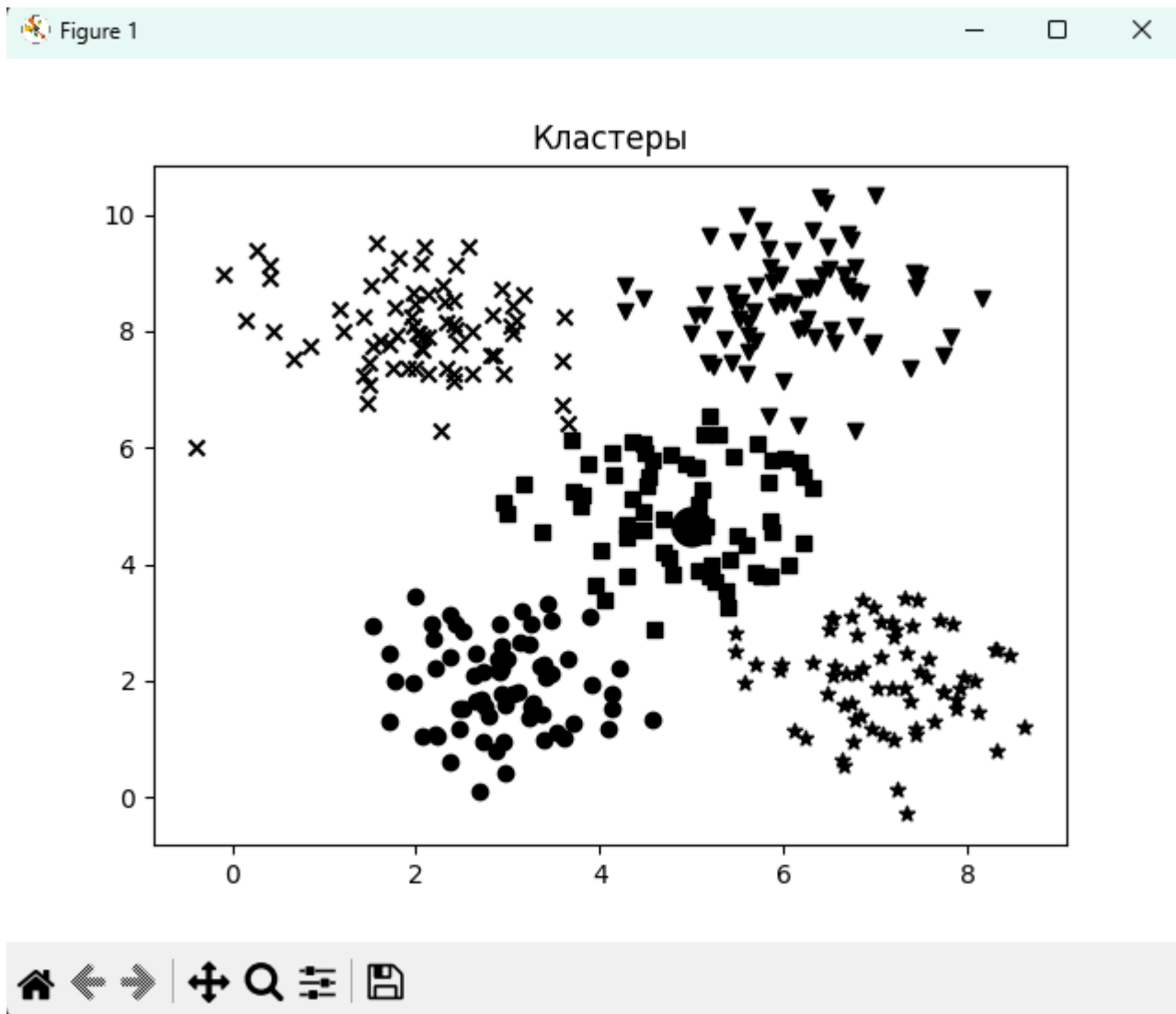
#### Результат виконання:

```
Centers of clusters:
[[2.95568966 1.95775862]
 [7.20690909 2.20836364]
 [2.17603774 8.03283019]
 [5.97960784 8.39078431]
 [4.99466667 4.65844444]]

Number of clusters in input data = 5

Process finished with exit code 0
```

		Курач О.А.			ДУ «Житомирська політехніка». 24.12.12.000 – Лр7	Арк.
		Іванов Д.А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



### Висновок:

Програма дійсно змогла обчислити кількість кластерів, як і зображено на графіку **Завдання 4**. Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності.

```
import numpy as np
import pandas as pd
from sklearn.cluster import AffinityPropagation
from sklearn.preprocessing import normalize
import matplotlib.pyplot as plt

# Завантаження даних із файлу data_clustering.txt
data = np.loadtxt('data_clustering.txt', delimiter=',') # Передбачається, що
# числа розділені комами

# Обчислення різниць між координатами (у даному разі залежить від формату даних)
# Якщо це двовимірні дані, їх можна кластеризувати без обчислення різниць
normalized_data = normalize(data)

# Створення моделі кластеризації
affinity_model = AffinityPropagation(random_state=0)
affinity_model.fit(normalized_data)

# Отримання результатів
cluster_centers_indices = affinity_model.cluster_centers_indices_
labels = affinity_model.labels_
```

		Курач О.А.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр7	Арк.
		Іванов Д.А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

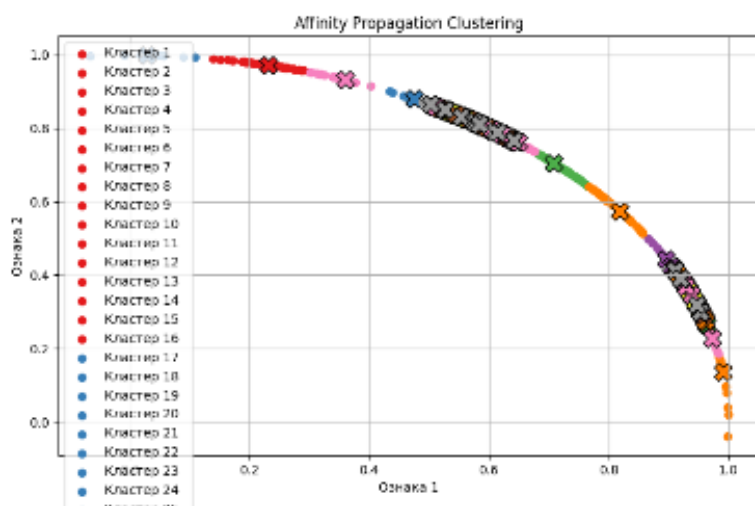


```
# Виведення результатів
n_clusters = len(cluster_centers_indices)
print(f"Кількість кластерів: {n_clusters}")
print("Кластери та їх центри:")
for i, center in enumerate(cluster_centers_indices):
    print(f"Кластер {i+1}: Центр - {normalized_data[center]}")

# Візуалізація результатів
plt.figure(figsize=(10, 6))
colors = plt.cm.Set1(np.linspace(0, 1, n_clusters))

for k, col in zip(range(n_clusters), colors):
    class_members = labels == k
    cluster_center = normalized_data[cluster_centers_indices[k]]
    plt.scatter(normalized_data[class_members, 0], normalized_data[class_members, 1], color=col, label=f'Кластер {k+1}')
    plt.scatter(cluster_center[0], cluster_center[1], s=200, color=col, edgecolor='k', marker='X')

plt.title('Affinity Propagation Clustering')
plt.xlabel('Ознака 1')
plt.ylabel('Ознака 2')
plt.legend()
plt.grid(True)
plt.show()
```



**Висновок:** Під час лабораторної роботи я навчилася використовувати спеціалізовані бібліотеки та мови програмування Python та дослідила методи неконтрольованої класифікації даних у машинному навчанні.

		Курач О.А.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр7	Арк.
		Іванов Д.А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		