Лабораторна робота №4

ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ

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

Хід роботи

Посилання на GitHub: [GitHub - Nastya3147/-]

Завдання №1:

Напишемо код для кластеризації даних за допомогою методу k-середніх:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import metrics

# Load input data
X = np.loadtxt('data_clustering.txt', delimiter=',')
num_clusters = 5

# Plot input data
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('Input data')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

# Create KMeans object
kmeans = KMeans(init='k-means++', n_clusters=num_clusters, n_init=10)

# Train the KMeans clustering model
kmeans.fit(X)

# Step size of the mesh
step_size = 0.01

# Define the grid of points to plot the boundaries
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
x_vals, y_vals = np.meshgrid(np.arange(x_min, x_max, step_size),
        np.arange(y_min, y_max, step_size))

# Predict output labels for all the points on the grid
output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])
```

```
# Plot different regions and color them
output = output.reshape(x_vals.shape)
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')

# Overlay input points
plt.scatter(X[:,0], X[:,1], marker='o', facecolors='none',
        edgecolors='black', s=80)

# Plot the centers of clusters
cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:,0], cluster_centers[:,1],
        marker='o', s=210, linewidths=4, color='black',
        zorder=12, facecolors='black')

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('Boundaries of clusters')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

Результати:

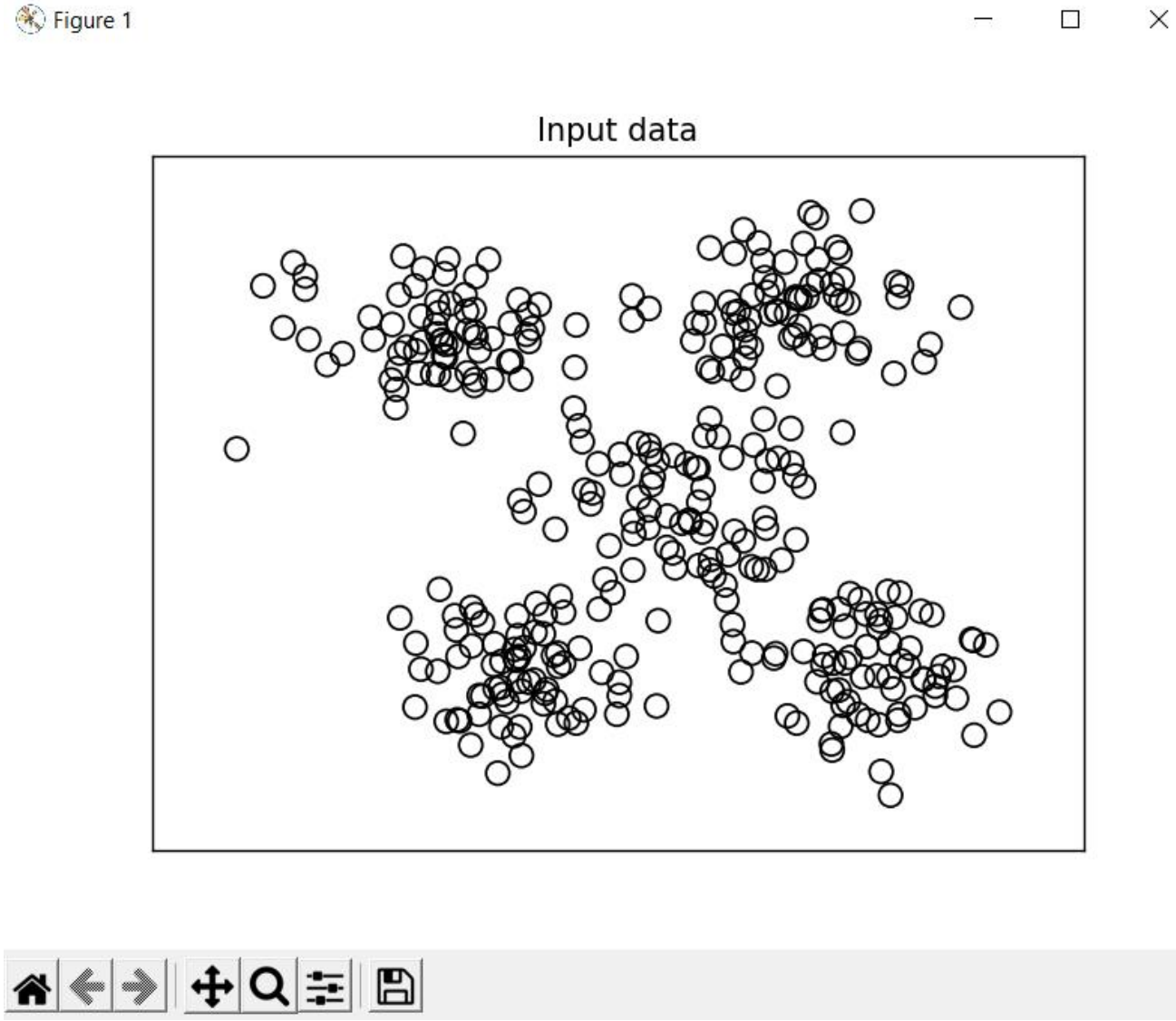

Рис. 1. Графік з вхідними даними

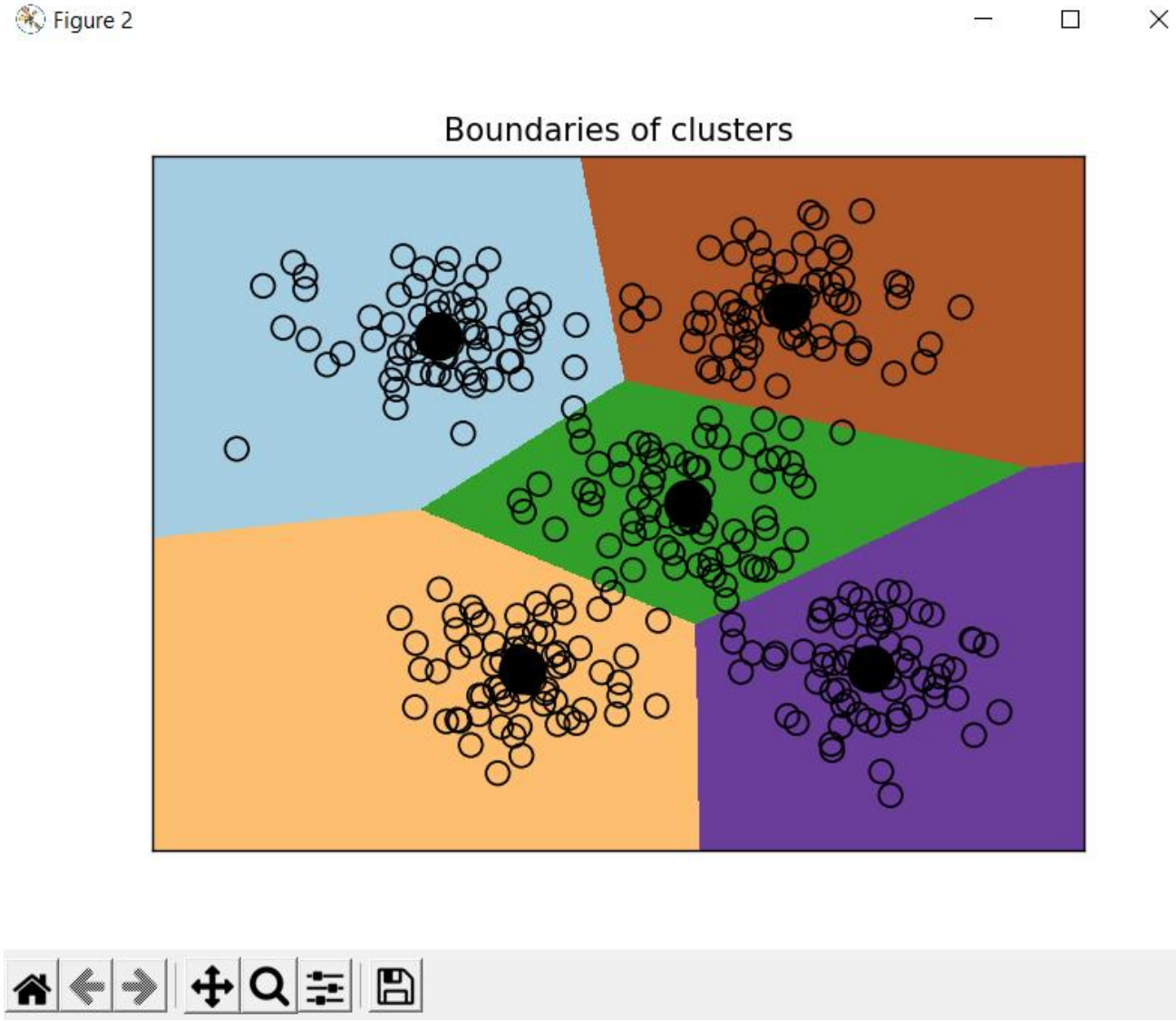

Рис. 2. Графік з кордонами після використання методу кластеризації k-середніх

Отже, судячи з графіків, маємо 5 кластерів.

Завдання №2:

Напишемо код для кластеризації набору даних Iris:

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

iris = load_iris()
X = iris["data"]
y = iris["target"]
#Створення об'єкта KMeans
kmeans = KMeans(n_clusters = 5)
#Начання моделі
kmeans.fit(X)
#Прогнозування результату
y_kmeans = kmeans.predict(X)
#Відображення вхідних точок
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)

#Функція пошуку кластерів
def find_clusters(X, n_clusters, rseed = 2):
    #Випадковий вибір кластерів
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]
    while True:
        #Призначення міток на основі найближчого до центру
        labels = pairwise_distances_argmin(X, centers)
        #Пошук нових центрів з середнього значення точок
        new_centers = np.array([X[labels == i].mean(0) for i in
range(n_clusters)])
        #Перевірка на конвергенцію(збіжність)
        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")
#Вивід глобально не оптимального результату з конвергенцією
centers, labels = find_clusters(X, 3, rseed = 0)
plt.scatter(X[:, 0], X[:, 1], c = labels,
s = 50, cmap = "viridis")
#Вивід з заданою кількістю кластерів
labels = KMeans(3, random_state = 0).fit_predict(X)
plt.scatter(X[:, 0], X[:, 1], c = labels,
s = 50, cmap = "viridis")

plt.show()
```

Результат:

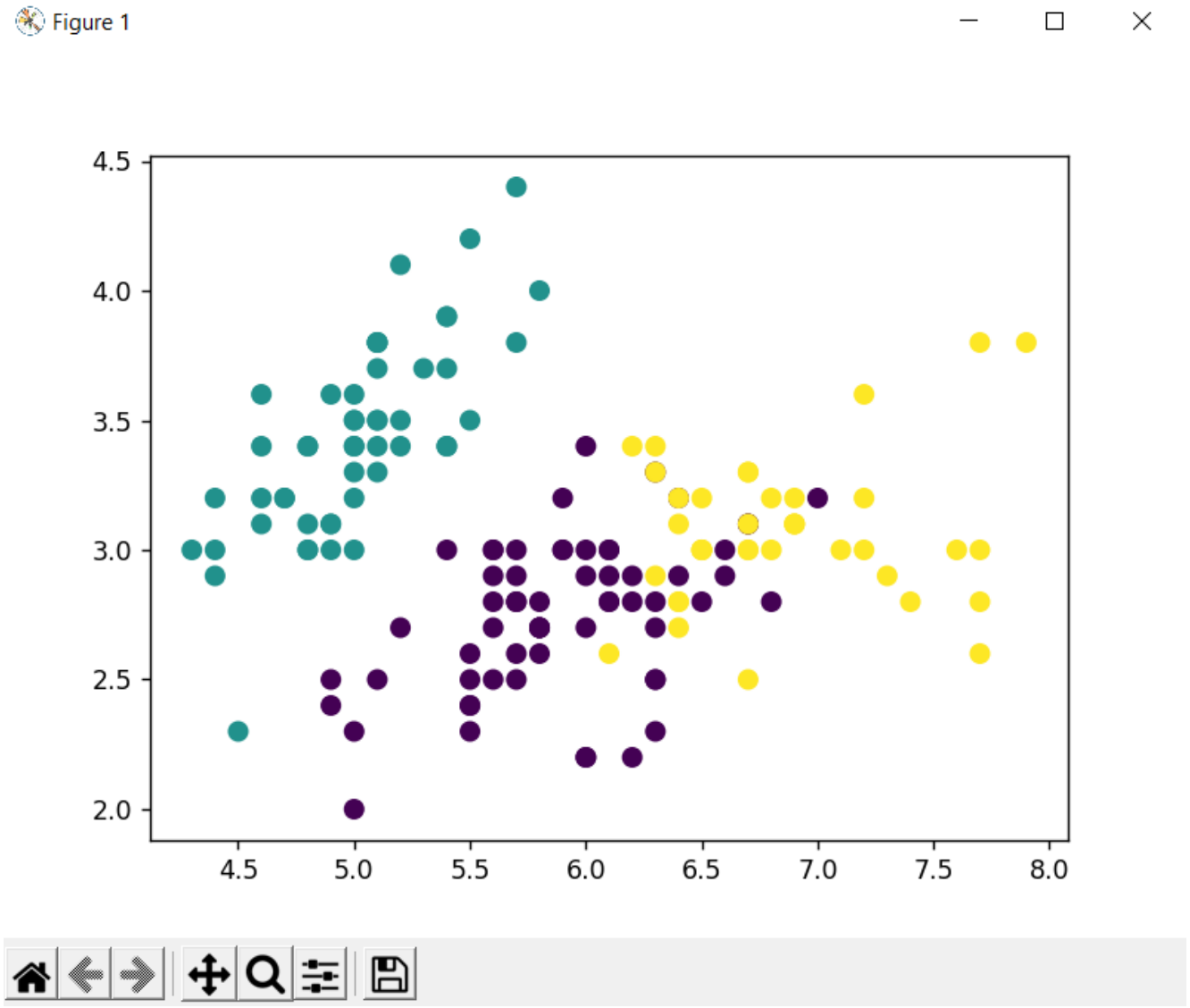

Рис. 3. Результат кластеризації набору даних Iris

Отже, набір даних Iris було поділено на 3 кластери, які позначено на графіку трьома кольорами.

Завдання №3:

Код оцінки кількості кластерів з використанням методу зсуву середнього:

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

# Load data from input file
X = np.loadtxt('data_clustering.txt', delimiter=',')

# Estimate the bandwidth of X
bandwidth_X = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))

# Cluster data with MeanShift
meanshift_model = MeanShift(bandwidth=bandwidth_X, bin_seeding=True)
meanshift_model.fit(X)

# Extract the centers of clusters
cluster_centers = meanshift_model.cluster_centers_
```

```
print('\nCenters of clusters:\n', cluster_centers)

# Estimate the number of clusters
labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)

# Plot the points and cluster centers
plt.figure()
markers = 'o*xvs'
for i, marker in zip(range(num_clusters), markers):
    # Plot points that belong to the current cluster
    plt.scatter(X[labels==i, 0], X[labels==i, 1], marker=marker, color='black')

    # Plot the cluster center
    cluster_center = cluster_centers[i]
    plt.plot(cluster_center[0], cluster_center[1], marker='o',
            markerfacecolor='black', markeredgecolor='black',
            markersize=15)

plt.title('Clusters')
plt.show()
```

Результати:

```
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
```

Рис. 4. Результати знаходження центрів кластерів

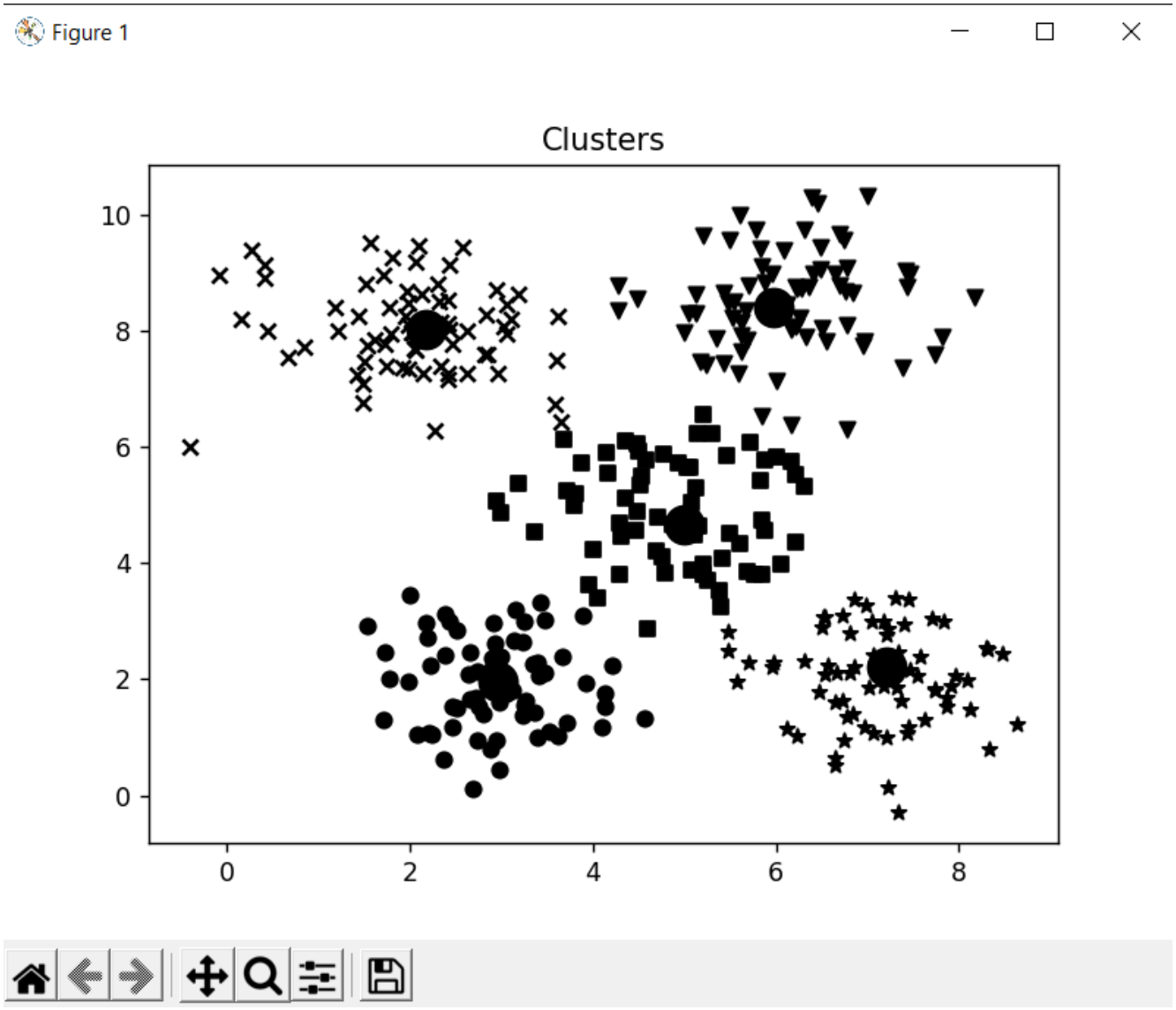

Рис. 5. Графік з кластерами

Отже, на графіку ми можемо побачити 5 кластерів, які позначені різними фігурками та мають свої центри.

Висновок: на цій лабораторній роботі я, використовуючи спеціалізовані бібліотеки та мову програмування Python, дослідив методи неконтрольованої класифікації даних у машинному навчанні.