**ЛАБОРАТОРНА РОБОТА № 4**

**ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЬОВАНОГО НАВЧАННЯ**

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

**Хід роботи:**

**Завдання 1:**  Кластеризація даних за допомогою методу k-середніх

**Результат:**

Код:

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn import metrics

X = np.loadtxt('data\_clustering.txt', delimiter = ',')

num\_clusters = 5

plt.figure()

plt.scatter(X[:, 0], X[:, 1], marker = 'o', facecolors = 'none', edgecolor = '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(())

plt.show()

kmeans = KMeans(init = 'k-means++', n\_clusters = num\_clusters, n\_init = 10)

kmeans.fit(X)

step\_size = 0.01

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))

output = kmeans.predict(np.c\_[x\_vals.ravel(), y\_vals.ravel()])

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')

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, linewidth = 4, color = 'black', zorder = 12, facecolors = 'none')

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('Межі кластерів')

plt.xlim(x\_min, x\_max)

plt.ylim(y\_min, y\_max)

plt.xticks(())

plt.yticks(())

plt.show()

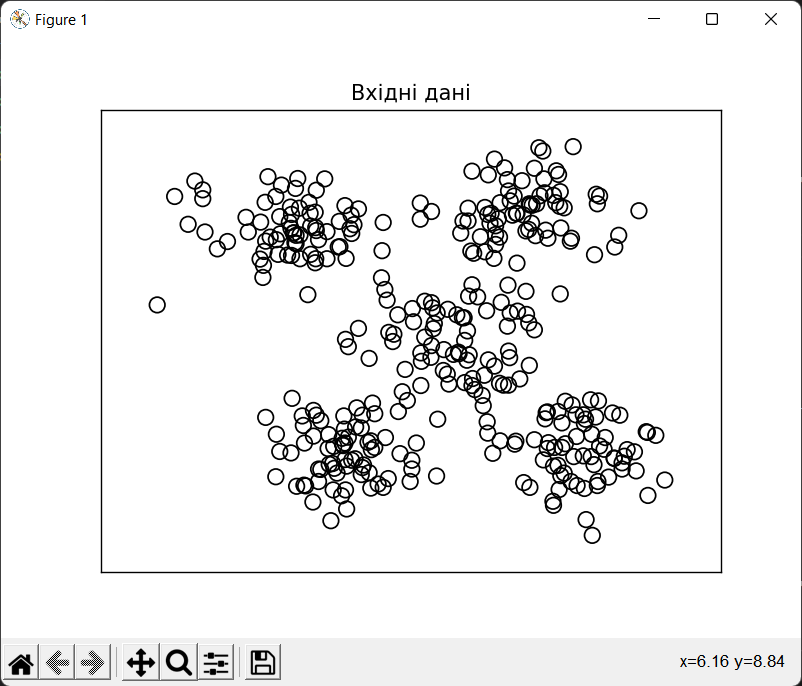


Рис. 1.1. Графік розподілу даних

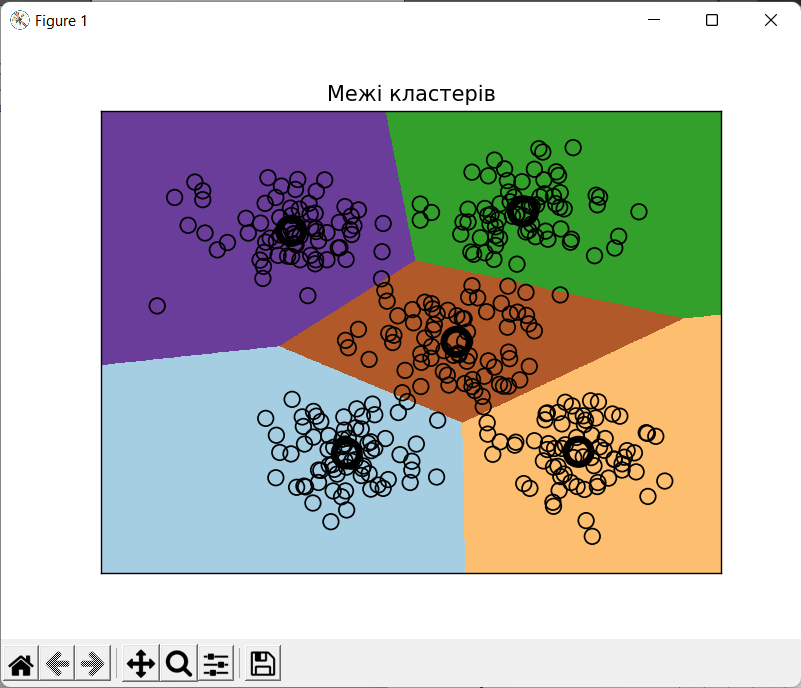


Рис. 1.2. Графік відображення результату кластеризації

**Хід роботи:**

**Завдання 2:**  Кластеризація K-середніх для набору даних Iris

**Результат:**

Код:

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

from sklearn.cluster import KMeans

iris = load\_iris()

X = iris['data']

y = iris['target']

num\_clusters = 3

kmeans = KMeans(n\_clusters = num\_clusters)

kmeans.fit(X)

y\_pred = kmeans.predict(X)

centers = kmeans.cluster\_centers\_

for i in range(X.shape[1] - 1):

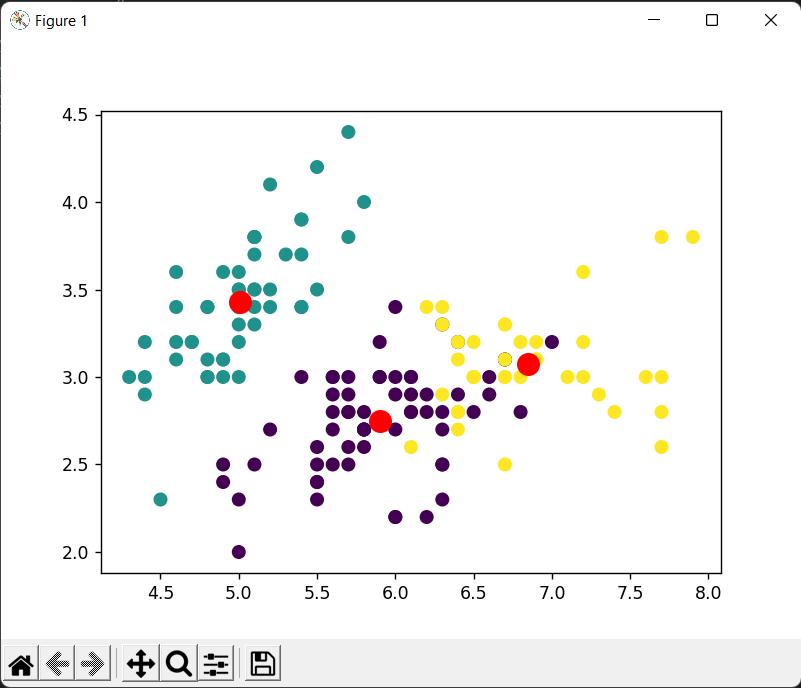
    for j in range(i + 1, X.shape[1]):

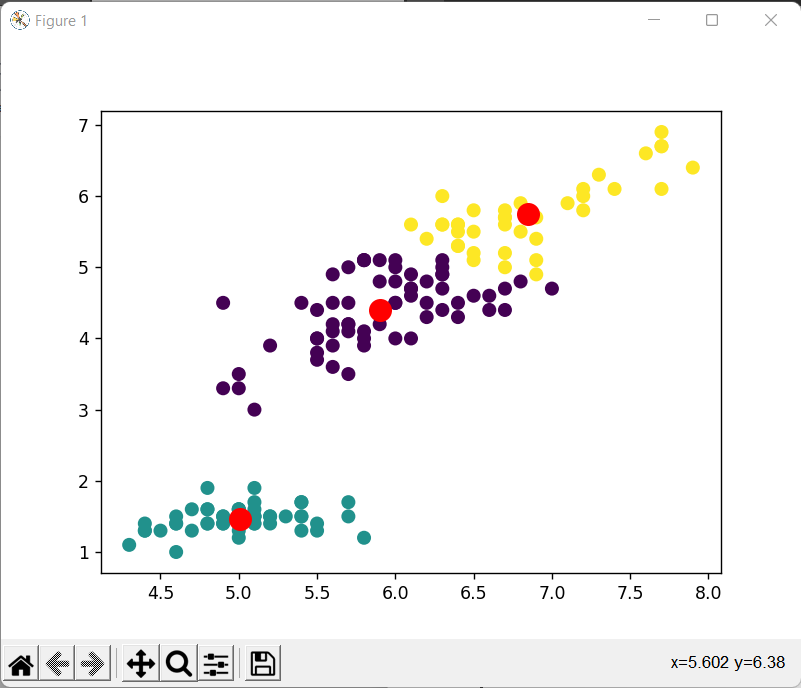
        plt.scatter(X[:, i], X[:, j], c = y\_pred, s = 50, cmap = 'viridis')

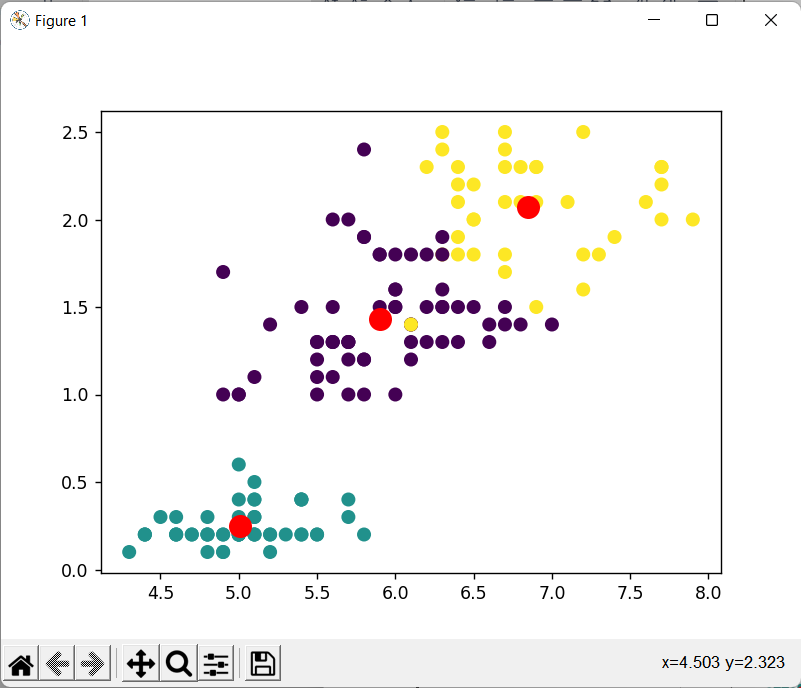
        plt.scatter(centers[:, i], centers[:, j], c = 'red', s = 150)

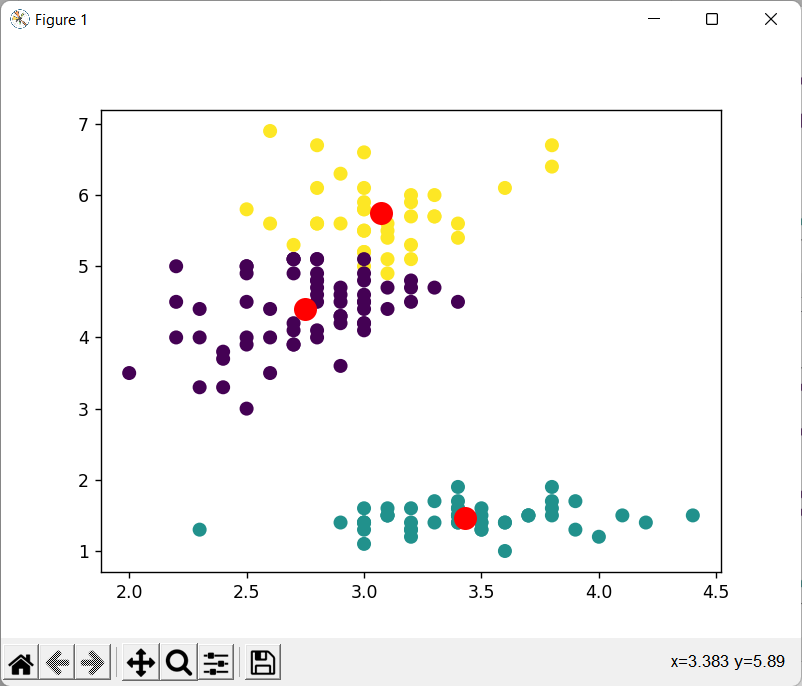
        plt.figure()

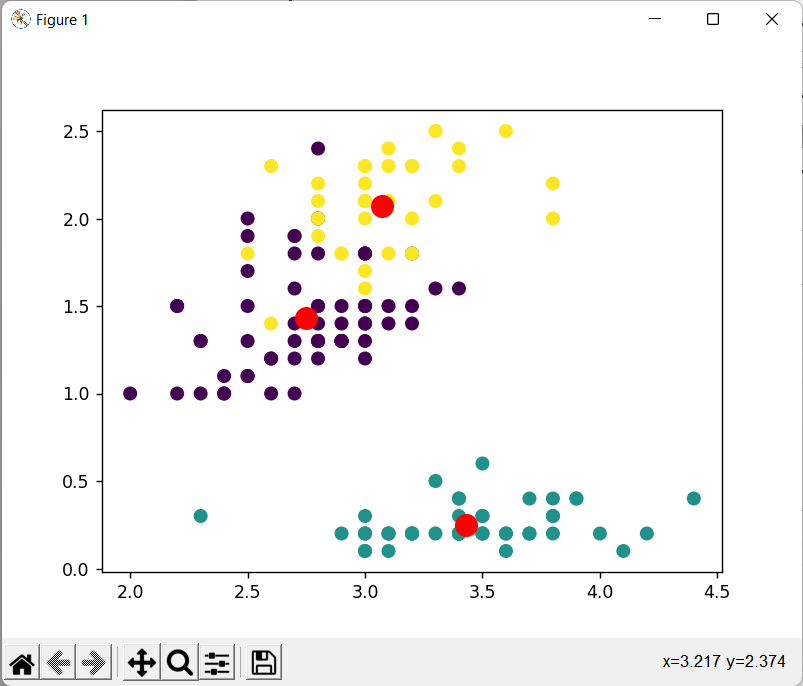
        plt.show()











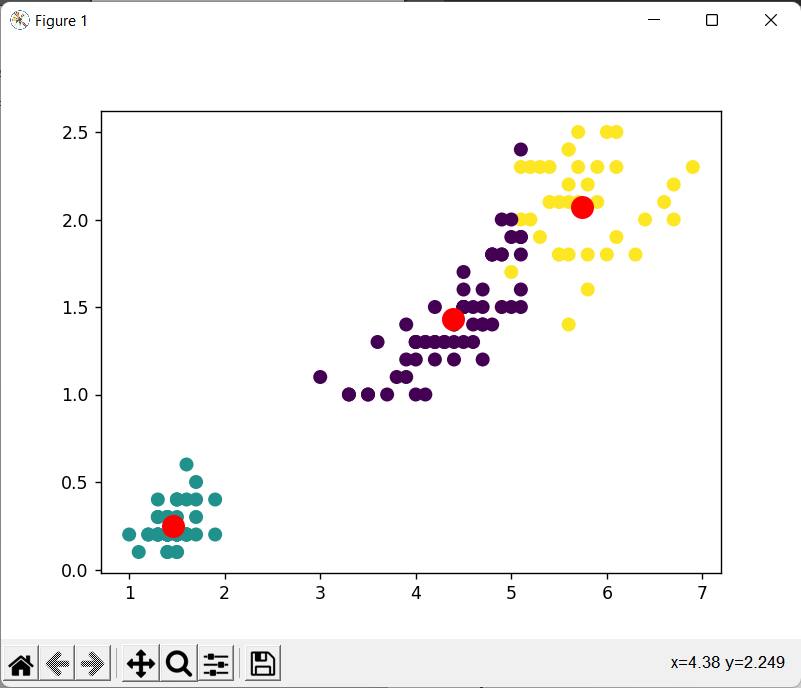


Рис. 2.1. Графік результату кластеризації

**Хід роботи:**

**Завдання 3:**  Оцінка кількості кластерів з використанням методу зсуву середнього

**Результат:**

Код:

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import MeanShift, estimate\_bandwidth

from itertools import cycle

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('Centers of cluster:', cluster\_centers)

labels = meanshift\_model.labels\_

num\_clusters = len(labels)

print('Number 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('Кластери')

plt.show()

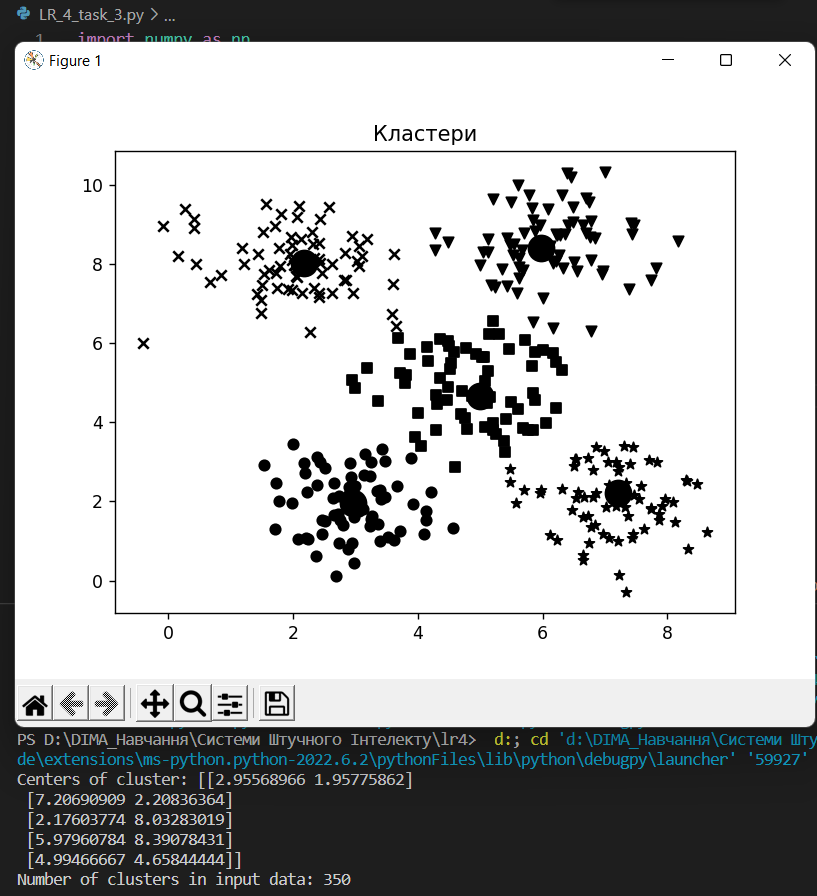


Рис. 3.1. Результат використання методу зсуву середнього

**Хід роботи:**

**Завдання 4:**  Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності

**Результат:**

Код:

import datetime

import json

import numpy as np

import matplotlib.pyplot as plt

from sklearn import covariance, cluster

import yfinance as yf

input\_file = 'company\_symbol\_mapping.json'

with open(input\_file, 'r') as f:

    company\_symbols\_map = json.loads(f.read())

symbols, names = np.array(list(company\_symbols\_map.items())).T

start\_date = datetime.datetime(2003, 7, 3)

end\_date = datetime.datetime(2007, 5, 4)

quotes = []

for symbol in symbols:

    try:

        quote = yf.download(symbols[1],start = start\_date, end = end\_date, progress = False)

        quotes.append(quote)

    except:

        continue

opening\_quotes = np.array([quote['Open'] for quote in quotes]).astype(np.float)

closing\_quotes = np.array([quote['Close'] for quote in quotes]).astype(np.float)

quotes\_diff = closing\_quotes - opening\_quotes

X = quotes\_diff.copy().T

X /= X.std(axis = 0)

edge\_model = covariance.GraphicalLassoCV()

with np.errstate(invalid = 'ignore'):

    edge\_model.fit(X)

\_, labels = cluster.affinity\_propagation(edge\_model.covariance\_)

num\_labels = labels.max()

for i in range(num\_labels + 1):

    print('Cluster', i + 1, '==>', ', '.join(names[labels == i]))

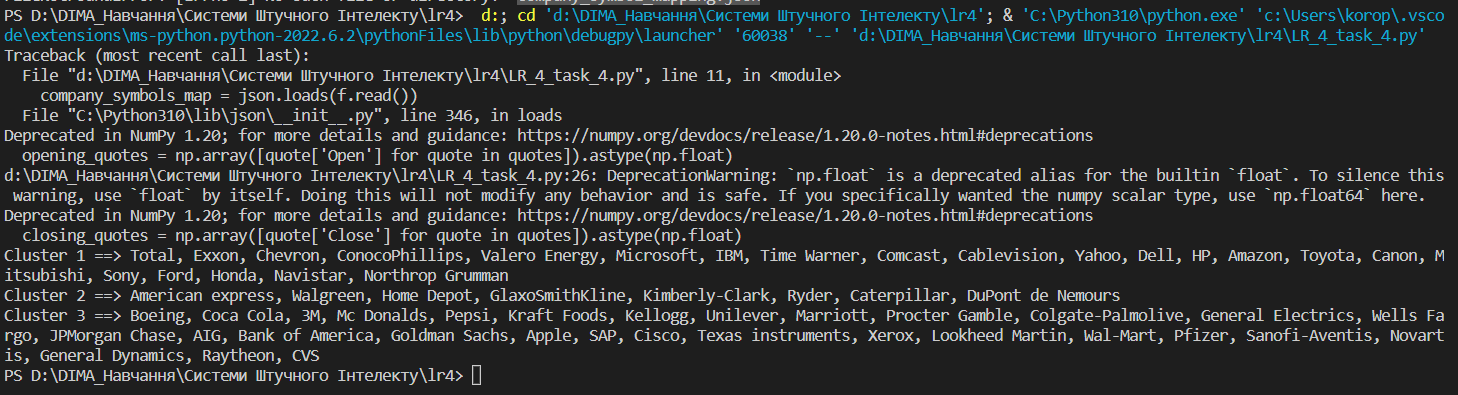


Рис. 1.4. Результат знаходження підгруп (3 кластери)

**Висновок:** на цій лабораторній роботі ми ввикористовуючи спеціалізовані бібліотеки та мову програмування Python дослідили методи неконтрольованої класифікації даних у машинному навчанні.