

## ЛАБОРАТОРНА РОБОТА № 4

### ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО 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)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Фещенко Д.М.			Звіт з лабораторної роботи 4	Лім.	Арк.
Перевір.		Пулюко І.В.					1
Керівник						ФІКТ Гр. ПІ-59(2)	
Н. контр.							
Затверд.							

```
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()
```

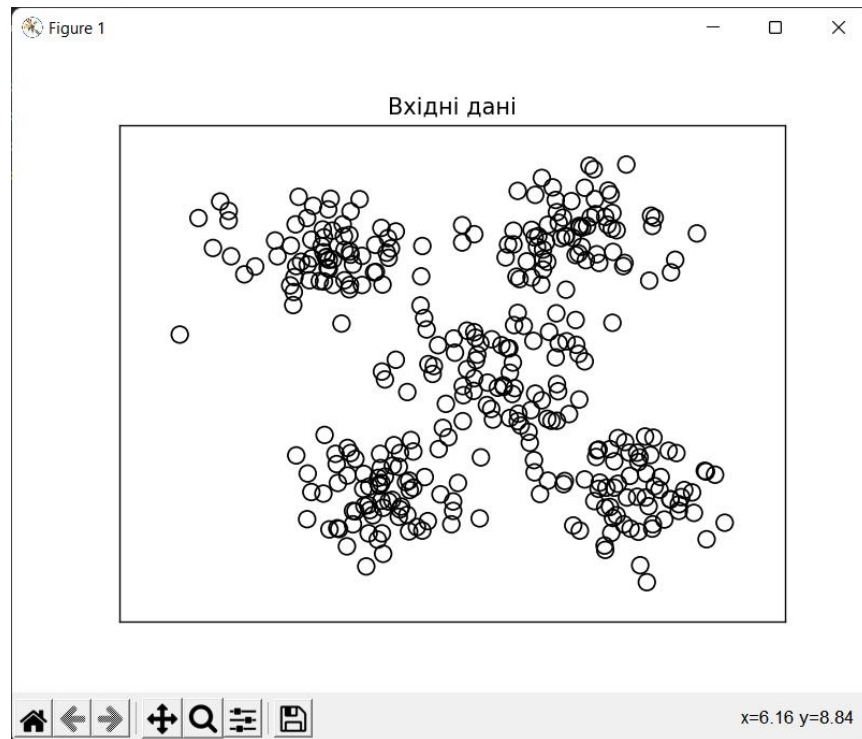


Рис. 1.1. Графік розподілу даних

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

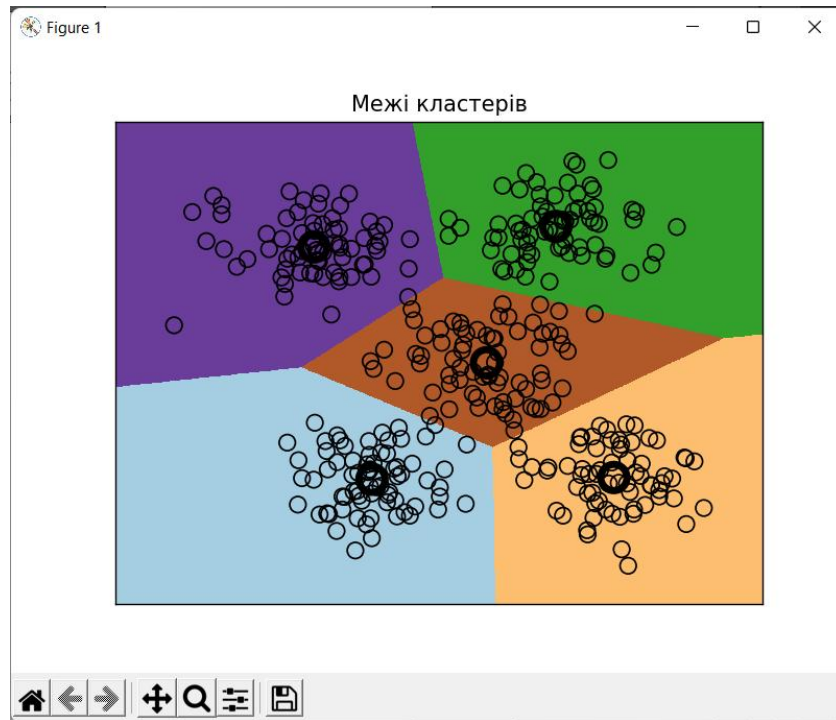


Рис. 1.2. Графік відображення результату кластеризації

**Хід роботи:**

**Завдання 2:** Кластеризація К-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

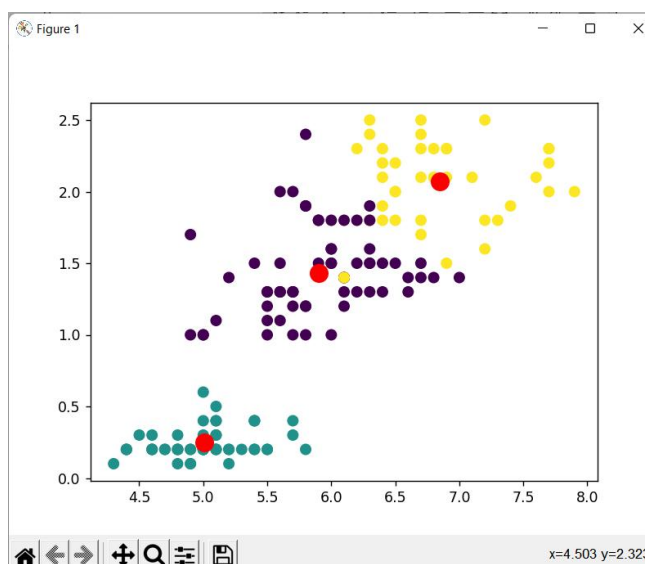
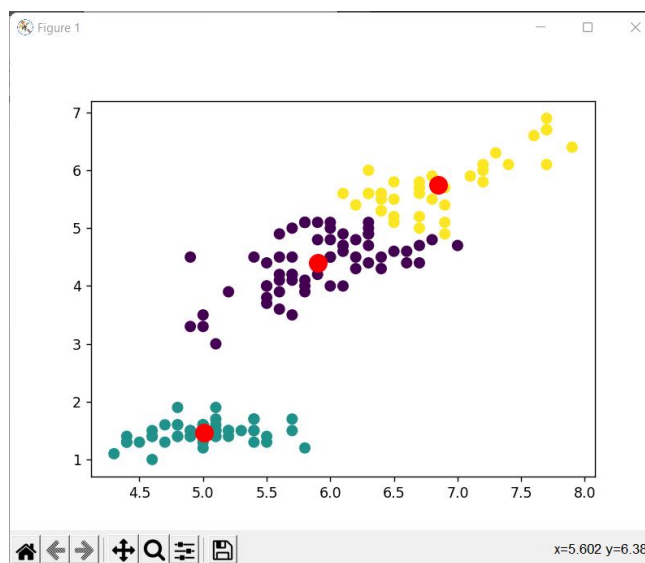
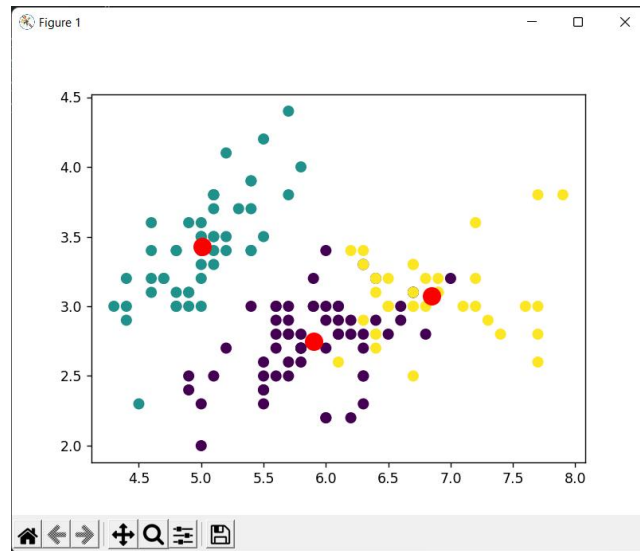
kmeans = KMeans(n_clusters = num_clusters)
kmeans.fit(X)

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()
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
		.				3
Змн.	Арк.	№ докум.	Підпис	Дата		



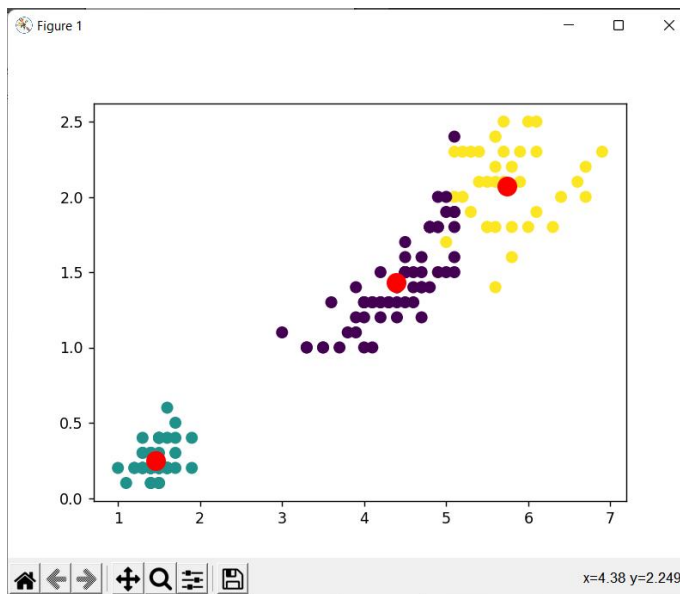
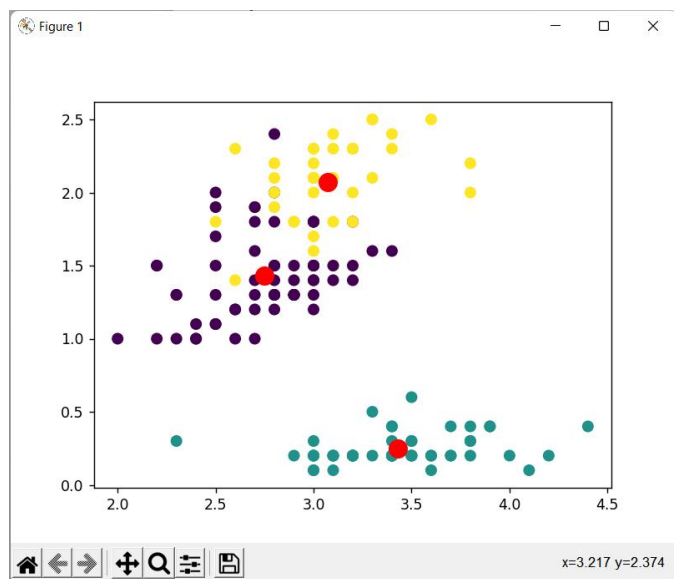
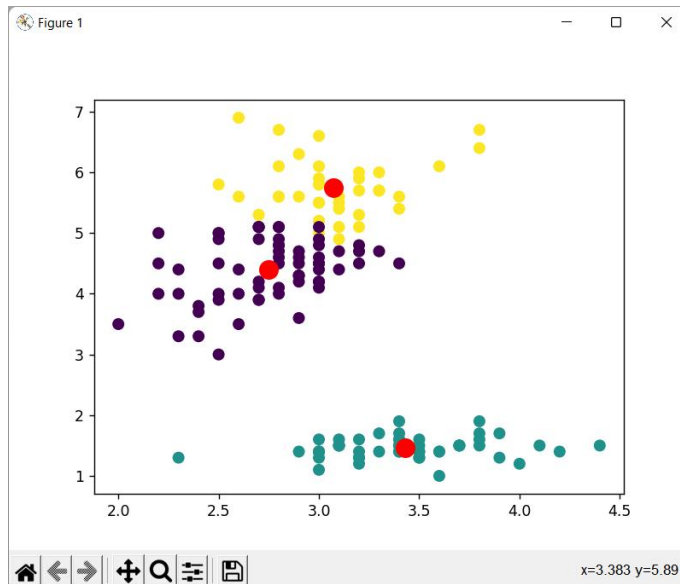


Рис. 2.1. Графік результату кластеризації

Хід роботи:

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 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()
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
		.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

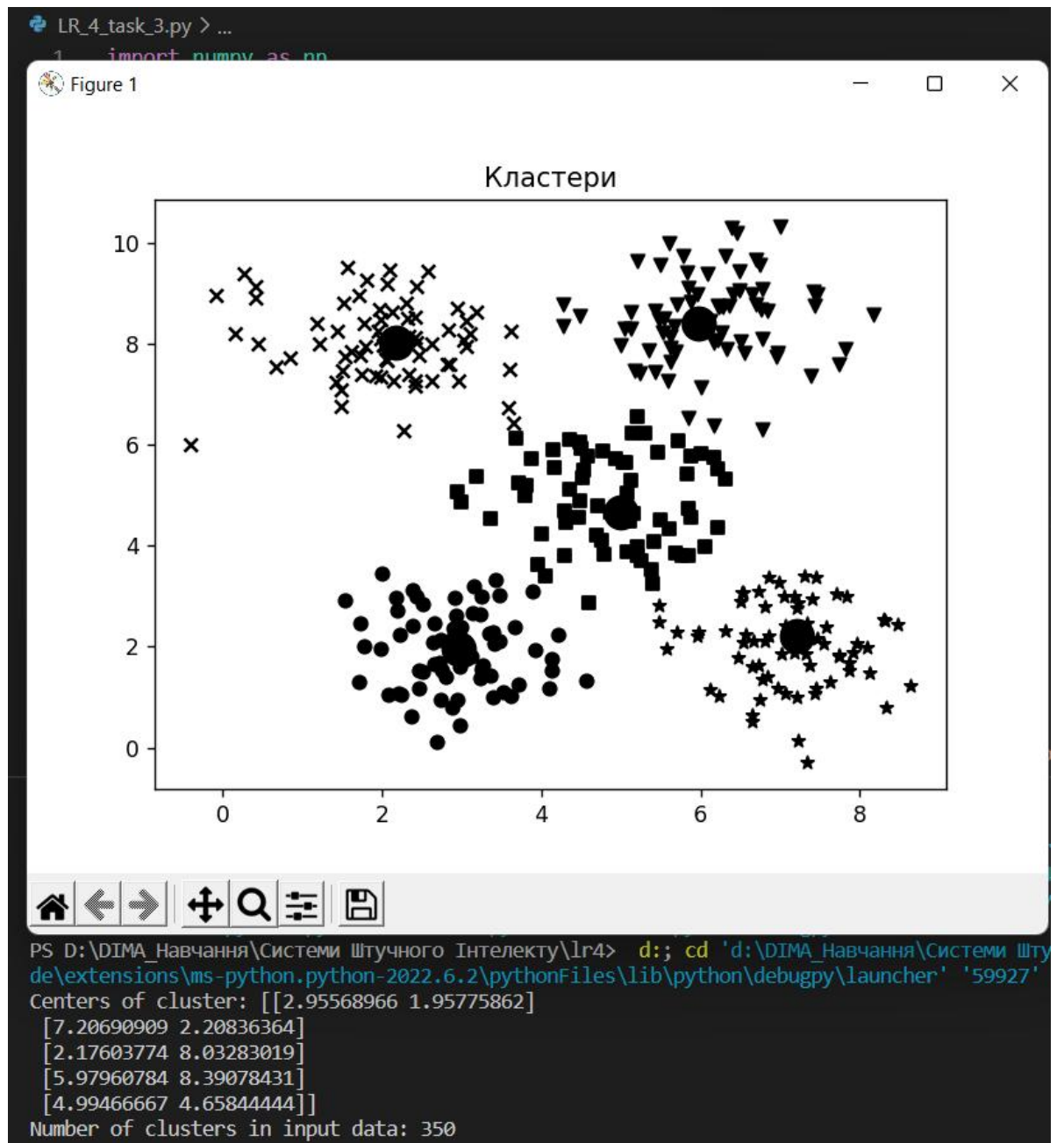


Рис. 3.1. Результат використання методу зсуву середнього

#### 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
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```
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]))
```

```
PS D:\DIMA Навчання\Системи Штучного Інтелекту\lr4> d; cd 'd:\DIMA Навчання\Системи Штучного Інтелекту\lr4'; & 'C:\Python310\python.exe' 'c:\Users\korop\.vscode\extensions\ms-python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher' '60038' '...' 'd:\DIMA Навчання\Системи Штучного Інтелекту\lr4\LR_4_task_4.py'
Traceback (most recent call last):
  File "d:\DIMA Навчання\Системи Штучного Інтелекту\lr4\LR_4_task_4.py", line 11, in <module>
    company_symbols_map = json.loads(f.read())
  File "c:\Python310\lib\json\_init_.py", line 346, in loads
    deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
opening_quotes = np.array([quote['Open'] for quote in quotes]).astype(np.float)
d:\DIMA Навчання\Системи Штучного Інтелекту\lr4\LR_4_task_4.py:26: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.
    closing_quotes = np.array([quote['Close'] for quote in quotes]).astype(np.float)
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
Cluster 1 => Total, Exxon, Chevron, ConocoPhillips, Valero Energy, Microsoft, IBM, Time Warner, Comcast, Cablevision, Yahoo, Dell, HP, Amazon, Toyota, Canon, Mitsubishi, Sony, Ford, Honda, Navistar, Northrop Grumman
Cluster 2 => American express, Walgreen, Home Depot, GlaxoSmithKline, Kimberly-Clark, Ryder, Caterpillar, DuPont de Nemours
Cluster 3 => Boeing, Coca Cola, 3M, McDonalds, Pepsi, Kraft Foods, Kellogg, Unilever, Marriott, Procter Gamble, Colgate-Palmolive, General Electrics, Wells Fargo, JPMorgan Chase, AIG, Bank of America, Goldman Sachs, Apple, SAP, Cisco, Texas Instruments, Xerox, Lockheed Martin, Wal-Mart, Pfizer, Sanofi-Aventis, Novartis, General Dynamics, Raytheon, CVS
PS D:\DIMA Навчання\Системи Штучного Інтелекту\lr4>
```

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.

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР04	Арк.
		.				8
Змн.	Арк.	№ докум.	Підпис	Дата		