ЛАБОРАТОРНА РОБОТА № 4

ПОРІВНЯННЯ МЕТОДІВ НЕКОНТРОЬОВАНОГО НАВЧАННЯ

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

Хід роботи

Завдання 2.1. Кластеризація даних за допомогою методу k-середніх

Код програми:

```
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(())

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(())
```

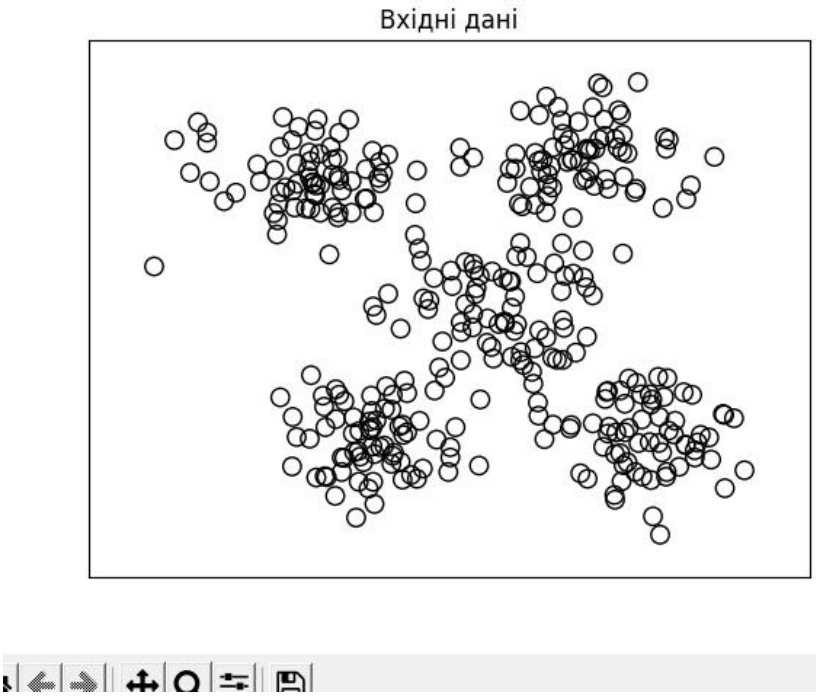


Рис. 1 Графік розподілу даних

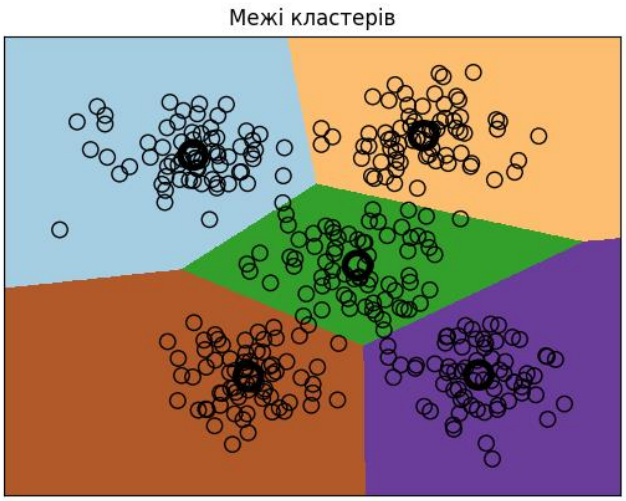


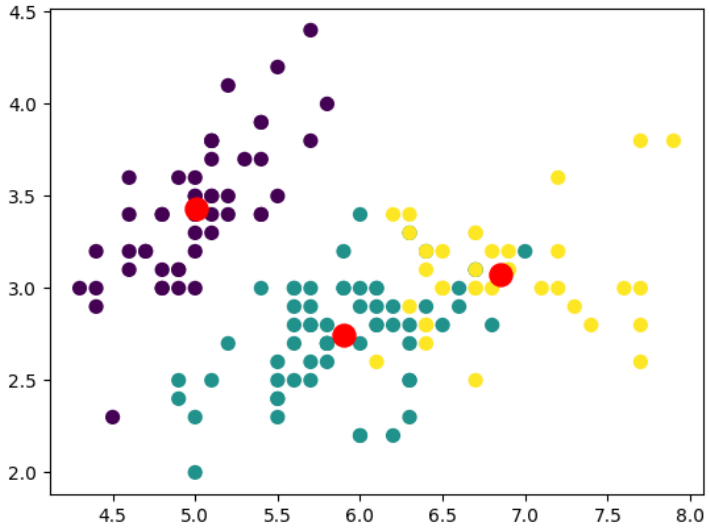
Рис. 2 Графік відображення результату кластеризації

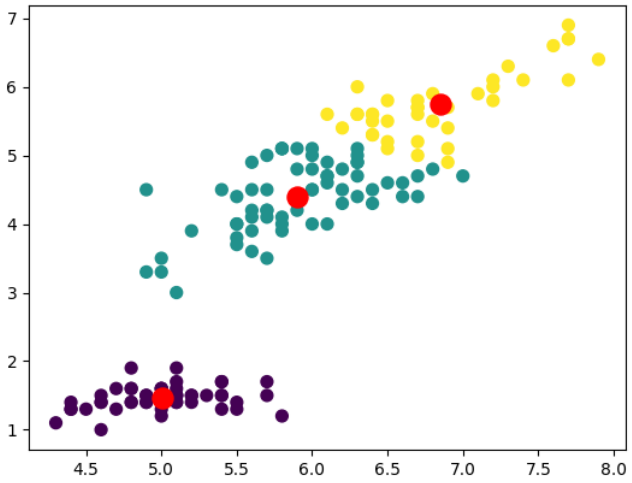
В результаті виконання даного завдання ми змогли створити кластери для вхідних даних й наочно побачити їх на графіку, де кожний кластер представлений іншим кольором.

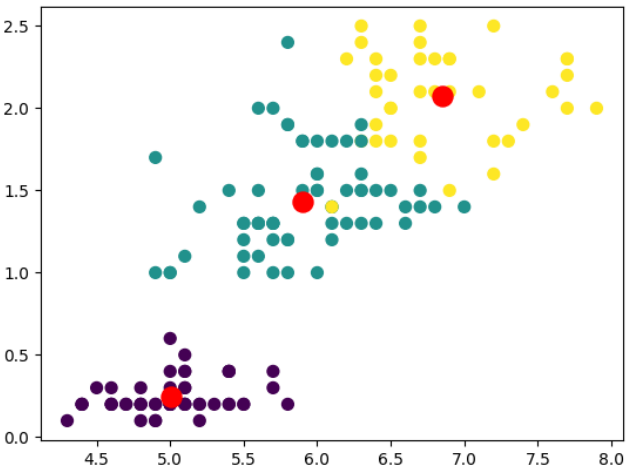
Завдання 2.2. Кластеризація K-середніх для набору даних Iris

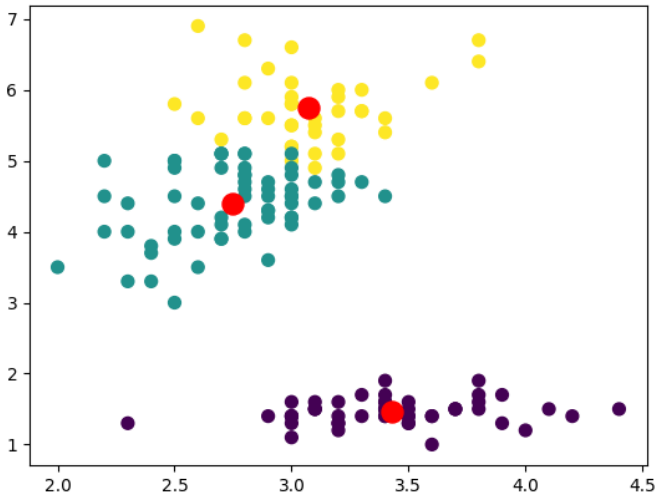
Код програми:

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans

# завантажуємо iris dataset
iris = load_iris()

# задаємо змінні
X = iris['data']
y = iris['target']

# кількість кластерів
num_clusters = 3

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

# отримуємо передбачені labels
y_pred = kmeans.predict(X)

# зберігаємо центри кластерів у змінну
centers = kmeans.cluster_centers_

# відобрамаємо попарно характеристики ірису
for i in range(X.shape[1] - 1):
  for j in range(i + 1, X.shape[1]):
    # зображуємо екземпляри
    plt.scatter(X[:, i], X[:, j], c = y_pred, s = 50, cmap = 'viridis')
    # зображуємо центри кластерів
    plt.scatter(centers[:, i], centers[:, j], c = 'red', s = 150)
    # створюємо новий графік та виводимо його на екран
    plt.figure()
    plt.show()
```




7
6
5
4
3
2
1
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


2.5
2.0
1.5
1.0
0.5
0.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5

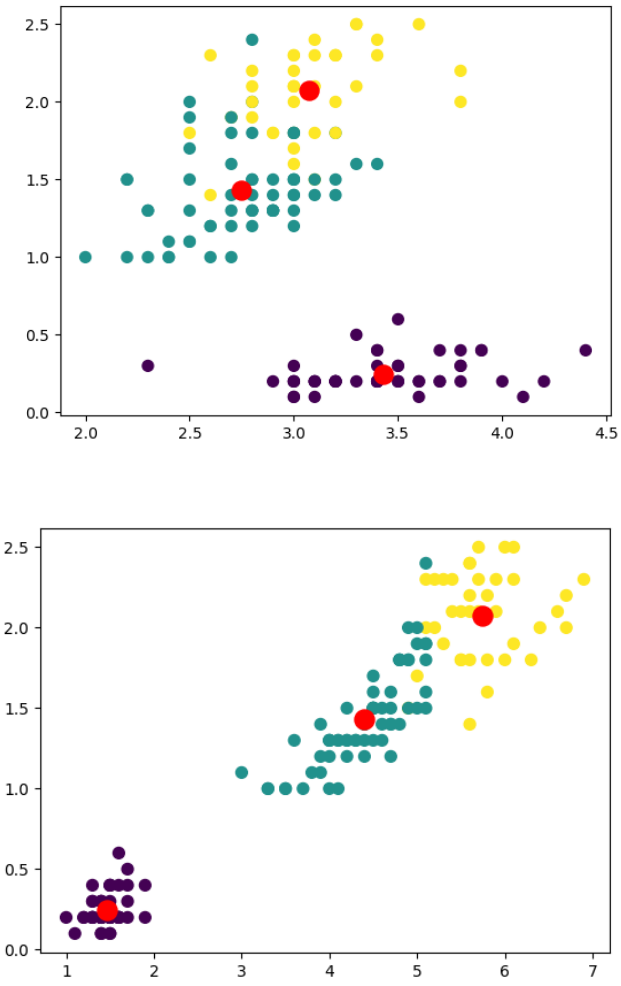


Рис. 3 Графік результату кластеризації

В результаті ми побачили як формуються кластери характеристик квіток ірису.

Завдання 2.3. Оцінка кількості кластерів з використанням методу зсуву середнього

Код програми:

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

# Завантажимо вхідні дані.
X = np.loadtxt('data_clustering.txt', delimiter=',')
# Оцінка ширини вікна для X
bandwidth_X = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))

# Кластеризація даних методом зсуву середнього
meanshift_model = MeanShift(bandwidth=bandwidth_X, bin_seeding=True)
meanshift_model.fit(X)

# Витягування центрів кластерів
cluster_centers = meanshift_model.cluster_centers_
print('Centers of cluster:', cluster_centers)

# Оцінка кількості кластерів
```

```
labels = meanshift_model.labels_
num_clusters = len(labels)
print('Number of clusters in input data:', num_clusters)

# Відображення на графіку точок та центрів кластерів
plt.figure()
markers = 'o*xvs'
# обхід циклом кластерів для відображення на графіку
for i, marker in zip(range(num_clusters), markers):
    # відображення даних
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
color='black')
    cluster_center = cluster_centers[i]
    # відображення центру кластера
    plt.plot(cluster_center[0], cluster_center[1], marker='o', markerface-
color='black',
             markeredgecolor='black', markersize=15)

plt.title('Кластери')
plt.show()
```

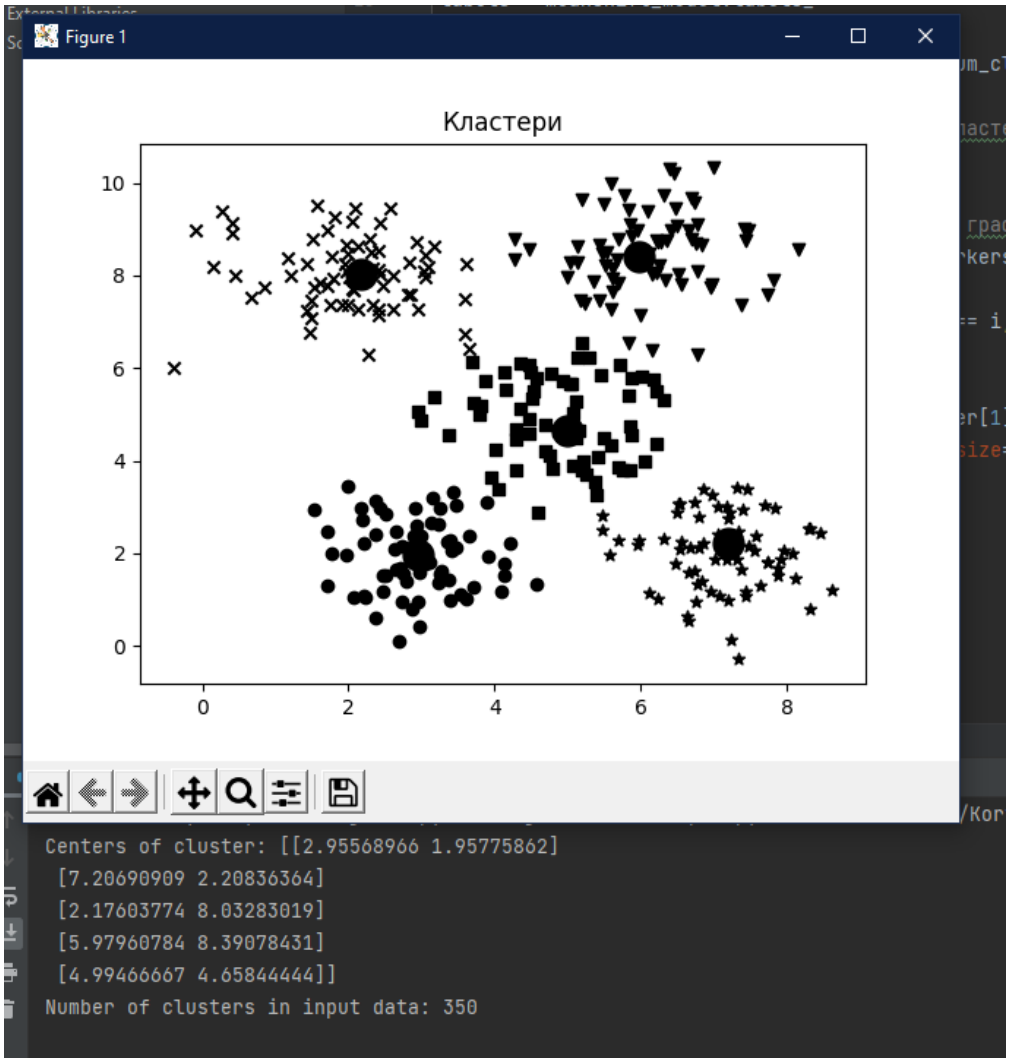


Рис. 4 Результат використання методу зсуву середнього

В результаті, можна отримати інформацію про кількість кластерів, їх центри та переглянути наочне представлення на графіку.

Завдання 2.4. Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності

Код програми:

```
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]))
```

```
Cluster 1 ==> Total, Exxon, Chevron, ConocoPhillips, Valero Energy, Microsoft, IBM, Time Warner, Comcast, Cablevision, Yahoo, Dell, HP, Ama
Cluster 2 ==> American express, Walgreen, Home Depot, GlaxoSmithKline, Kimberly-Clark, Ryder, Caterpillar, DuPont de Nemours
Cluster 3 ==> Boeing, Coca Cola, 3M, Mc Donalds, Pepsi, Kraft Foods, Kellogg, Unilever, Marriott, Procter Gamble, Colgate-Palmolive, Genera
```

Рис. 5 Результат знаходження підгруп

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