

Лабораторна роботи №7

Дослідження методів неконтрольованого навчання

Мета роботи: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи неконтрольованої класифікації (кластеризації) даних у машинному навченні.

Хід роботи

Завдання 2.1. Кластеризація даних за допомогою методу k-середніх

Провести кластеризацію даних методом k-середніх. Використовувати файл вхідних даних: `data_clustering.txt`.

Код програми (`LR_7_task_1.py`):

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

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(())
plt.show()

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))
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.8.000 – Пр.6		
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи		
Розроб.	Вєщиков О.М.				Lіт.	Арк.	Аркушів
Перевір.	Маєвський О.В					1	24
Керівник					ФІКТ, гр. ІПЗ-22-2		
Н. контр.							
Затверд.							

```

output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])
output = output.reshape(x_vals.shape)

# Графічне відображення областей та виділення їх кольором
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)

# Відображення центрів кластерів
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')

plt.title('Границі кластерів')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

```

Результати:

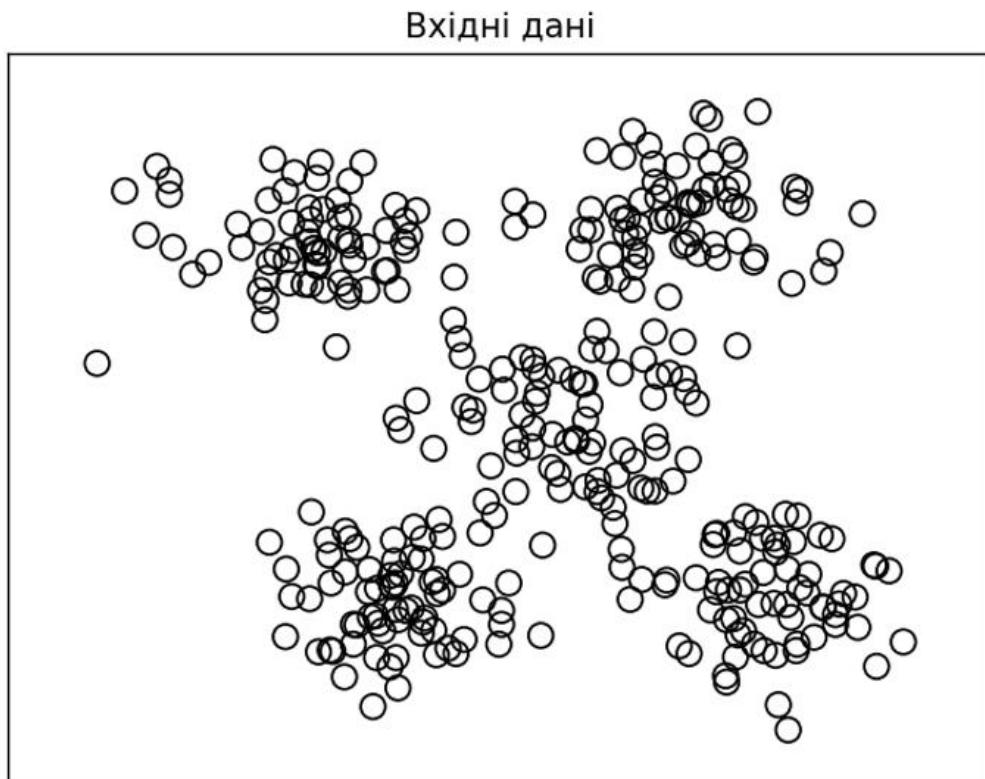


Рис. 1. Вхідні дані для кластеризації

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 24. 121.8.000 – Лр. 7	Арк.
						2

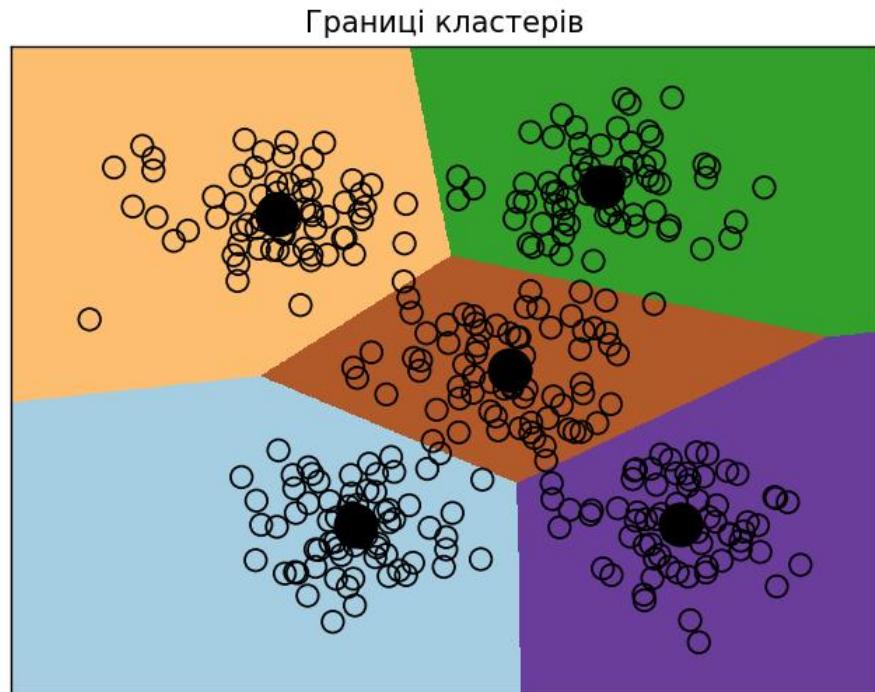


Рис. 2. Результат кластеризації методом K-Means (5 кластерів)

Висновок до завд. 2.1: Метод k-середніх успішно розділив дані на 5 кластерів. Центри кластерів розташовані в центрах щільних груп даних. Границі кластерів чітко розділяють простір.

Завдання 2.2. Кластеризація K-середніх для набору даних Iris

Виконайте кластеризацію K-середніх для набору даних Iris, який включає три типи (класи) квітів ірису (Setosa, Versicolour і Virginica) з чотирма атрибутами: довжина чашолистка, ширина чашолистка, довжина пелюстки та ширина пелюстки. У цьому завданні використовуйте `sklearn.cluster.KMeans` для пошуку кластерів набору даних Iris.

Код програми (LR_7_task_2.py):

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

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

Змн.	Арк.	№ докум.	Підпис	Дата

```

# n_clusters=3, оскільки ми знаємо, що є 3 види ірисів
kmeans = KMeans(n_clusters=3, n_init=10)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

plt.figure(figsize=(10, 6))
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('K-Means кластеризація Iris')
plt.show()

plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='viridis')
plt.title('Реальні класи Iris')
plt.show()

```

Результати:

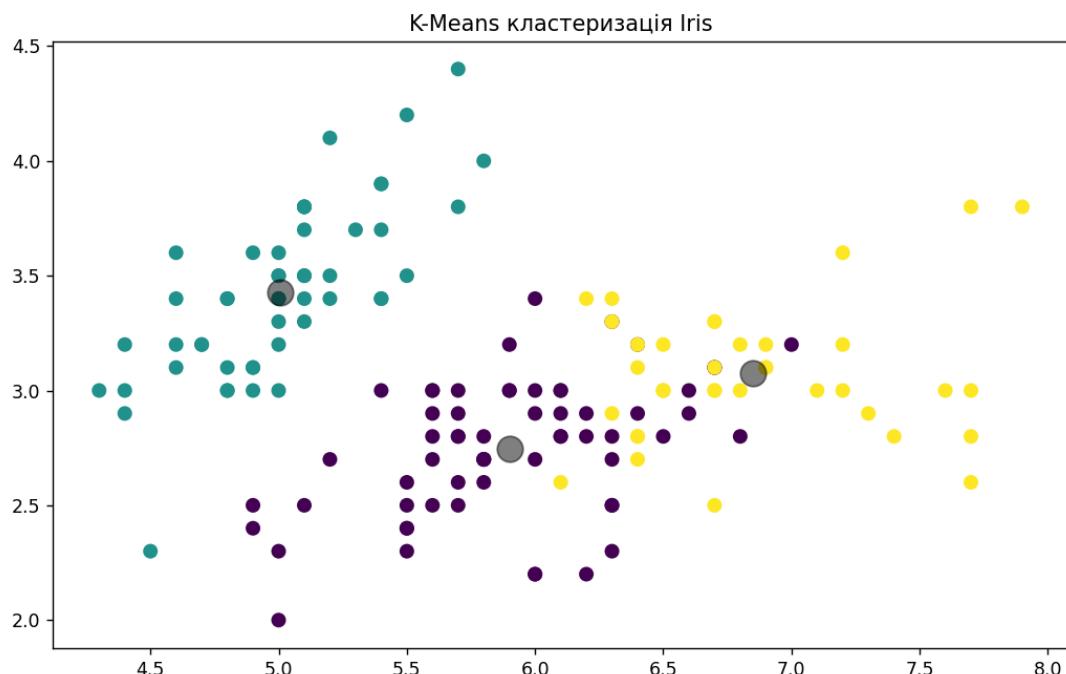


Рис. 3. Кластери, знайдені алгоритмом K-Means

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 24. 121.8.000 – Лр. 7	Арк.
						4

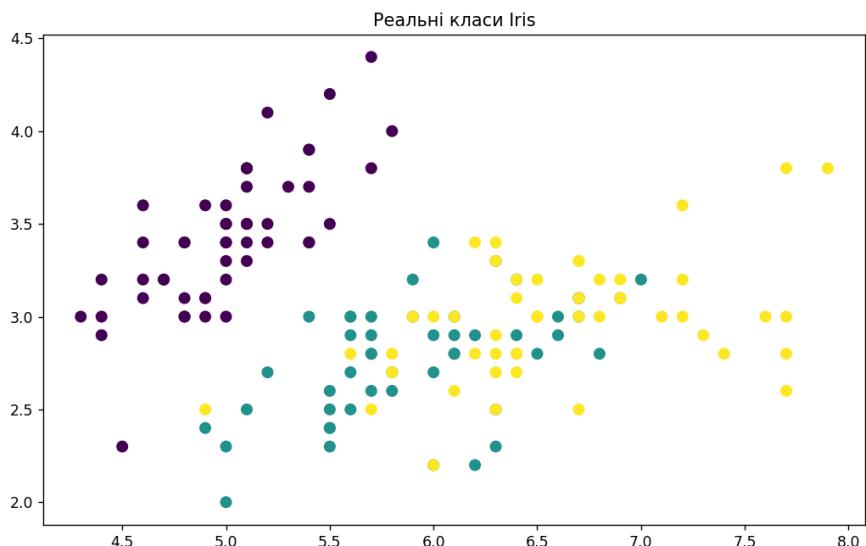


Рис. 4. Реальні класи квітів Iris

Висновок до завд. 2.2: Алгоритм K-Means зміг згрупувати дані на 3 кластери. Порівняння з реальними мітками показує, що один клас (Setosa) відокремлюється ідеально, тоді як два інші (Versicolour та Virginica) мають деяке перекриття, що ускладнює їх ідеальне розділення без учителя.

Завдання 2.3. Оцінка кількості кластерів з використанням методу зсуву середнього

Відповідно до рекомендацій, напишіть програму та оцініть максимальну кількість кластерів у заданому наборі даних за допомогою алгоритму зсуву середнього. Для аналізу використовуйте дані, які містяться у файлі data_clustering.txt.

Код програми (LR_7_task_3.py):

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

X = np.loadtxt('data_clustering.txt', delimiter=',')
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('\nCenters of clusters:\n', cluster_centers)

labels = meanshift_model.labels_
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)

plt.figure()
markers = 'o*xvs'
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']

for i in range(num_clusters):
    # Відображення точок поточного кластера
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=markers[i % len(markers)], color=colors[i % len(colors)])

    # Відображення центру кластера
    cluster_center = cluster_centers[i]
    plt.plot(cluster_center[0], cluster_center[1], marker='o',
             markerfacecolor='k', markeredgecolor='k', markersize=15)

plt.title('Кластери (Mean Shift)')
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. 5. Результати роботи Mean Shift у консолі

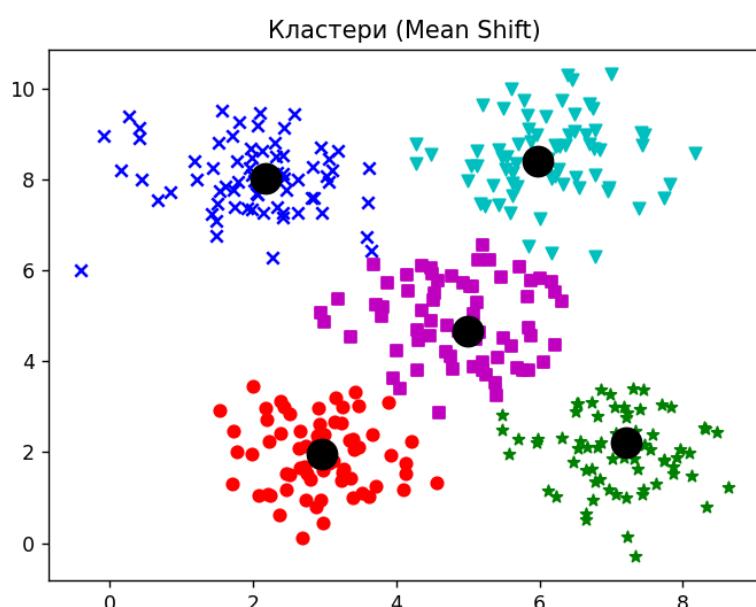


Рис. 6. Кластеризація методом Mean Shift

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.8.000 – Лр.7	Арк.
						6

Висновок до завд. 2.3: Метод Mean Shift автоматично визначив кількість кластерів без необхідності задавати цей параметр вручну. Це робить його зручним для аналізу даних, про структуру яких ми нічого не знаємо заздалегідь.

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

Використовуючи модель поширення подібності, знайти підгрупи серед учасників фондового ринку. У якості керуючих ознак будемо використовувати варіацію котирувань між відкриттям і закриттям біржі.

Використовувати файл вхідних даних фонового ринку, що доступний в бібліотеці matplotlib. Прив'язки символічних позначень компаній до повних назв містяться у файлі company_symbol_mapping.json.

Код програми (LR_7_task_4.py):

```
import datetime
import json
import numpy as np
import pandas as pd
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.load(f)

names, symbols = np.array(list(company_symbols_map.items())).T

start_date = "2003-07-03"
end_date = "2007-05-04"

quotes = []
valid_names = []

print("Завантаження котирувань...")
for symbol, name in zip(symbols, names):
    try:
        data = yf.download(symbol, start=start_date, end=end_date,
                           progress=False, auto_adjust=True)

        if not data.empty:
            quotes.append(data)
            valid_names.append(name)
        else:
            print(f"Немає даних для {name} ({symbol})")
    except Exception as e:
        print(f"Помилка для {name}: {e}")

names = np.array(valid_names)

if len(quotes) == 0:
    print("Помилка: Не вдалося завантажити жодних даних. Перевірте інтернет або символи.")
```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

```

    exit()

quotes_aligned = pd.concat(quotes, axis=1, keys=names)

# Витягуємо Open/Close для всіх компаній
opening_quotes = quotes_aligned.loc[:, (slice(None), 'Open')].values
closing_quotes = quotes_aligned.loc[:, (slice(None), 'Close')].values

quotes_diff = closing_quotes - opening_quotes

# Нормалізація та очищенння
X = quotes_diff.copy()
X = np.nan_to_num(X) # замінює NaN та inf
X = X[~np.all(X == 0, axis=1)]
X /= X.std(axis=0)

# Створення моделі графа
edge_model = covariance.GraphicalLassoCV()

with np.errstate(invalid='ignore'):
    edge_model.fit(X)

# Кластеризація
_, labels = cluster.affinity_propagation(edge_model.covariance_, random_state=0)
num_labels = labels.max()

print("\nРезультати кластеризації:")
for i in range(num_labels + 1):
    cluster_members = names[labels == i]
    print(f"Cluster {i + 1} ==> {', '.join(cluster_members)}")

```

Результат:

company_symbol_mapping.json:

```
{
    "Amazon": "AMZN", "Apple": "AAPL", "Wal-Mart": "WMT",
    "Microsoft": "MSFT", "Boeing": "BA", "IBM": "IBM",
    "Coca-Cola": "KO", "Google": "GOOG", "Chevron": "CVX",
    "Exxon": "XOM", "Novartis": "NVS", "Toyota": "TM",
    "Ford": "F", "Bank of America": "BAC", "Pepsi": "PEP"
}
```

Завантаження котирувань...

Результати кластеризації:

```

Cluster 1 ==> Apple, Google
Cluster 2 ==> Coca-Cola, Pepsi
Cluster 3 ==> Chevron, Exxon
Cluster 4 ==> Amazon, Wal-Mart, Microsoft, Boeing, IBM, Novartis, Toyota, Ford, Bank of America

```

Рис. 7. Групи компаній, сформовані алгоритмом

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.8.000 – Лр.7	Арк.
						8

Висновок до завд. 2.4: Використовуючи алгоритм Affinity Propagation, вдалося згрупувати компанії за схожістю динаміки їхніх акцій.

Загальний висновок: у ході лабораторної роботи я дослідив методи неконтрольованого навчання:

K-Means: Ефективний для простих кластерів, але вимагає знання кількості кластерів.

Mean Shift: Дозволяє автоматично знаходити кількість кластерів, шукаючи "центр мас" розподілу даних.

Affinity Propagation: Потужний метод для виявлення структур у складних даних не вимагає задання кількості кластерів, але може бути обчислювально складним.

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 24. 121.8.000 – Лр. 7	Арк.
						9