

ЛАБОРАТОРНА РОБОТА № 7
ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ
Мета роботи: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи неконтрольованої класифікації даних у машинному навчанні.

ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ ТА МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЙОГО ВИКОНАННЯ

Завдання 2.1. Кластеризація даних за допомогою методу k-середніх
Провести кластеризацію даних методом k-середніх. Використовувати файл вхідних даних: data_clustering.txt.

Лістинг:

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

input_file = 'data_clustering.txt'
X = np.loadtxt(input_file, delimiter=',')

plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black',
s=80)
plt.title('Вхідні дані (Input Data)')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
plt.savefig('cluster_input_data.png')
plt.show()

num_clusters = 5

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(figsize=(10, 8))
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')
```

Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.	Кравченко К.М.			
Перевір.	Маєвський О.В.			
Реценз.				
Н. Контр.				
Зав.каф.				

ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 –
Пр.7

Звіт з лабораторної
роботи №7

Lіт.	Арк.	Аркушів
	1	10
ФІКТ, гр. ІПЗ-22-2		

```

plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black',
s=80, label='Data points')
cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
marker='x', s=200, linewidths=4, color='black', zorder=10,
label='Centroids')

plt.title('Результат кластеризації k-means (межі та центроїди)')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.legend()
plt.savefig('cluster_output_result.png')
plt.show()

```

Результат:

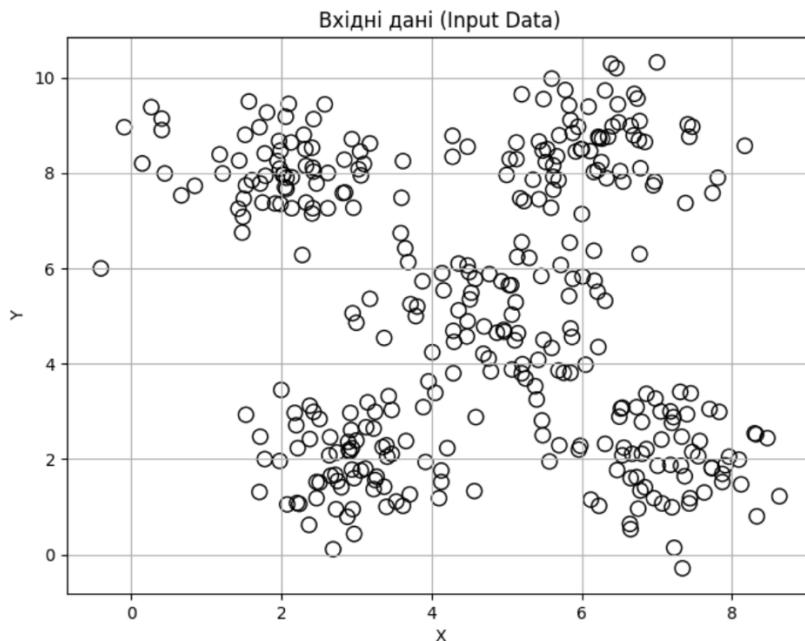


Рис.1.1. – Графік вхідних даних

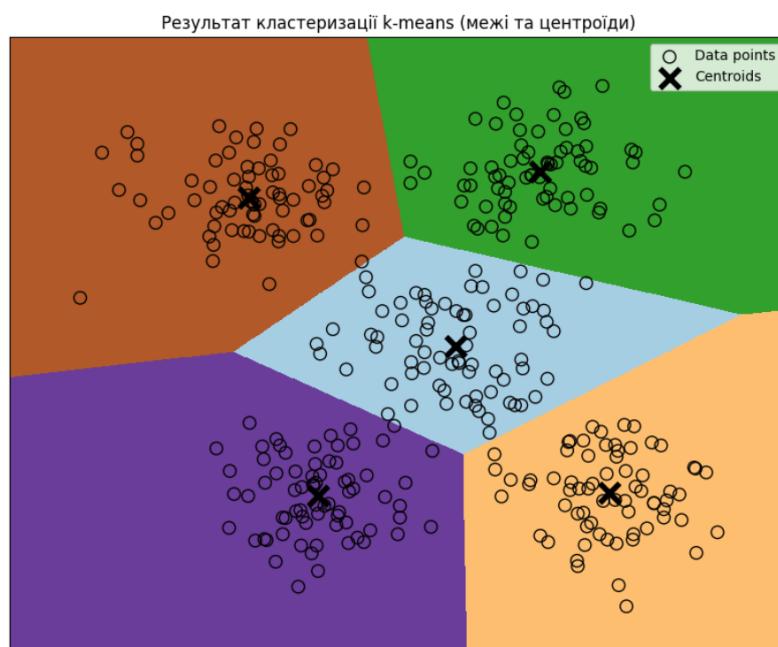


Рис.1.2. – Графік результату кластеризації

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Лр. 7

У ході виконання завдання було проведено кластеризацію двовимірного набору даних із файлу `data_clustering.txt` за допомогою алгоритму **k-середніх (k-means)**.

На етапі попереднього візуального аналізу вхідних даних було виявлено 5 чітко відокремлених груп точок, тому для навчання моделі було обрано параметр кількості кластерів **k=5**. Застосування алгоритму з ініціалізацією центроїдів методом k-means++ дозволило успішно розділити простір на відповідні зони.

Як свідчить фінальний графік, алгоритм коректно визначив структуру даних: розраховані центроїди (позначені хрестиками) розташувалися в центрах щільності груп, а межі кластерів (Voronoi cells) чітко розмежовують скучення точок. Це підтверджує ефективність методу k-means для роботи з даними, що мають виражену групову структуру.

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 i 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.

Лістинг:

```
import matplotlib.pyplot as plt
import seaborn as sns;

sns.set()
import numpy as np
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

kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)

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', label='Data Points')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5,
label='Centroids')
plt.title("K-Means (Sklearn implementation)")
plt.legend()
plt.show()

def find_clusters(X, n_clusters, rseed=2):
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

    while True:
        labels = pairwise_distances_argmin(X, centers)
```

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Лр. 7

```

new_centers = np.array([X[labels == i].mean(0)
                      for i in range(n_clusters)])

if np.all(centers == new_centers):
    break
centers = new_centers

return centers, labels
centers, labels = find_clusters(X, 3)

plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("K-Means (Manual Implementation)")
plt.show()
centers, labels = find_clusters(X, 3, rseed=0)
plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("K-Means (Manual Implementation, rseed=0)")
plt.show()

```

Результат:



Рис.2.1. – Результат кластеризації ірисів за допомогою бібліотеки Scikit-Learn ($k=3$). Сірі кола – центроїди кластерів.



Рис.2.2. – Результат роботи власної реалізації алгоритму k-means (стандартна ініціалізація).

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Лр. 7

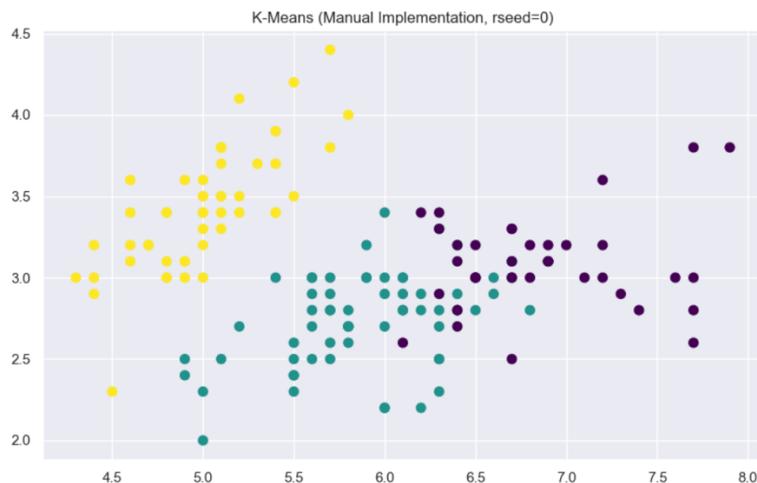


Рис.2.3. – Демонстрація чутливості алгоритму до початкової ініціалізації центроїдів (rseed=0).

У ході виконання завдання було проведено кластеризацію класичного набору даних Iris (квіти ірису) методом k-середніх. Оскільки набір даних містить три біологічні види квітів, параметр кількості кластерів було встановлено рівним 3.

1. Було порівняно роботу стандартного класу KMeans з бібліотеки scikit-learn та власної програмної реалізації алгоритму. Обидва методи показали ідентичні результати розбиття даних на групи.
2. Візуалізація (побудована за двома першими ознаками: довжина та ширина чахолистка) демонструє, що алгоритм чітко відокремив один вид ірисів (на графіках зліва), тоді як два інші види знаходяться близько один до одного і мають незначне перекриття, що є природною властивістю цього набору даних.
3. Експеримент зі зміною параметра випадковості (rseed) у власній реалізації підтверджує теоретичну особливість методу k-means: кінцевий результат кластеризації може залежати від того, де саме були випадково розміщені початкові центроїди.

Метод k-середніх успішно впорався з задачею групування багатовимірних даних, коректно виділивши основні скupчення об'єктів без використання інформації про їхні справжні мітки класів.

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 середньою. Для аналізу використовуйте дані, які містяться у файлі data_clustering.txt.

Лістинг:

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

input_file = 'data_clustering.txt'
X = np.loadtxt(input_file, delimiter=',')
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

bandwidth = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))
print(f"Оцінена ширина вікна: {bandwidth:.4f}")

ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
ms.fit(X)

labels = ms.labels_
cluster_centers = ms.cluster_centers_
n_clusters_ = len(np.unique(labels))

print(f"Оцінена кількість кластерів: {n_clusters_}")
print("Координати центрів кластерів:")
print(cluster_centers)

plt.figure(figsize=(10, 8))
colors = 10 * ['r', 'g', 'b', 'c', 'k', 'y', 'm']

for i in range(len(X)):
    plt.plot(X[i][0], X[i][1], colors[labels[i]] + '.', markersize=10)

plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
            marker='x', s=250, linewidths=3, color='black', zorder=10,
            label='Centroids')

plt.title(f'Mean Shift Clustering (Кількість кластерів = {n_clusters_})')
plt.legend()
plt.grid(True)
plt.show()

```

Результат:

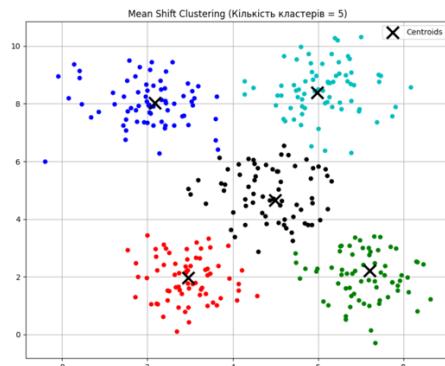


Рис.3.1. – Результат кластеризації методом зсуву середнього (Mean Shift) з автоматично визначеною кількістю кластерів ($k=5$).

Оцінена ширина вікна: 1.3045
Оцінена кількість кластерів: 5
Координати центрів кластерів:
[[2.95568966 1.95775862]
[7.20690909 2.20836364]]
[2.17603774 8.03283019]
[5.97960784 8.39078431]
[4.99466667 4.65844444]]

Рис.3.2. – Візуалізація вхідних даних для кластеризації

Автоматичне визначення кластерів: Головною перевагою методу є те, що він не потребує попереднього задання кількості кластерів. Алгоритм самостійно визначив, що оптимальна кількість кластерів для даного набору даних дорівнює 5. Це повністю підтверджує результат, отриманий у попередньому завданні (2.1), де цю кількість ми задавали вручну.

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Лр. 7

Оцінка параметрів: За допомогою функції оцінки (estimate_bandwidth) було автоматично розраховано ширину вікна пошуку (bandwidth), яка склала **1.3045**. Цей параметр став ключовим для коректного розділення груп точок.

Точність: Розраховані координати центрів кластерів (наприклад, [2.96, 1.96], [7.21, 2.21] та ін.) точно відповідають центрам найщільніших скупчень даних на графіку.

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 входних даних фондового ринку, що доступний в бібліотеці matplotlib. Прив'язки символічних позначень компаній до повних назв містяться у файлі company_symbol_mapping.json.

Лістинг:

```
import datetime
import json
import sys
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import yfinance as yf
from sklearn import covariance, cluster

symbol_file = 'company_symbol_mapping.json'
try:
    with open(symbol_file, 'r') as f:
        symbol_dict = json.loads(f.read())
except FileNotFoundError:
    print(f"Помилка: Файл {symbol_file} не знайдено.")
    sys.exit(1)

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

print("Завантаження даних котирувань з Yahoo Finance...")
start_date = "2023-01-01"
end_date = "2024-01-01"

quotes = []
valid_symbols = []
valid_names = []

data = yf.download(list(symbols), start=start_date, end=end_date, progress=False)

opening_quotes = data['Open']
closing_quotes = data['Close']
opening_quotes = opening_quotes.dropna(axis=1)
closing_quotes = closing_quotes.dropna(axis=1)

available_symbols = opening_quotes.columns.tolist()
valid_names = [symbol_dict[s] for s in available_symbols]
valid_symbols = available_symbols

open_data = opening_quotes.T.to_numpy()
close_data = closing_quotes.T.to_numpy()
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

variation = close_data - open_data

std_dev = variation.std(axis=1)
variation /= std_dev[:, np.newaxis]

print(f"Дані підготовлено. Розмірність: {variation.shape} (Компаній x Днів)")

print("Навчання моделі графа (GraphicalLasso)...")
edge_model = covariance.GraphicalLassoCV()

edge_model.fit(variation.T)

print("Кластеризація методом Affinity Propagation...")
_, labels = cluster.affinity_propagation(edge_model.covariance_, random_state=42)
n_labels = labels.max()

print("\n--- Результати кластеризації фондового ринку ---")
for i in range(n_labels + 1):
    cluster_members_indices = np.where(labels == i)[0]
    cluster_names = [valid_names[idx] for idx in cluster_members_indices]

    if len(cluster_names) > 0:
        print(f"Кластер {i + 1}: {', '.join(cluster_names)}")

```

company_symbol_mapping.json:

```
{
    "ADM": "Archer-Daniels-Midland",
    "AIG": "American International Group",
    "AMZN": "Amazon",
    "AAPL": "Apple",
    "AXP": "American Express",
    "BA": "Boeing",
    "C": "Citigroup",
    "CAJ": "Canon",
    "CAT": "Caterpillar",
    "COP": "ConocoPhillips",
    "CSCO": "Cisco Systems",
    "CVX": "Chevron",
    "DD": "DuPont",
    "DELL": "Dell",
    "F": "Ford",
    "GE": "General Electric",
    "GOOG": "Google",
    "GS": "Goldman Sachs",
    "HMC": "Honda",
    "HPQ": "Hewlett-Packard",
    "IBM": "IBM",
    "INTC": "Intel",
    "JPM": "JPMorgan Chase",
    "K": "Kellogg",
    "KO": "Coca-Cola",
    "MSFT": "Microsoft",
    "NAV": "Navistar",
    "NOC": "Northrop Grumman",
    "PEP": "Pepsi",
    "TM": "Toyota",
    "TOT": "Total",
    "TWX": "Time Warner",
    "TXN": "Texas Instruments",
    "VLO": "Valero Energy",
    "WFC": "Wells Fargo",
    "XOM": "Exxon",
    "XRX": "Xerox",
    "YHOO": "Yahoo"
}
```

Результат:

```

Кластеризація методом Affinity Propagation...

--- Результати кластеризації фондового ринку ---|
Кластер 1: Archer-Daniels-Midland, ConocoPhillips, Chevron, Valero Energy, Exxon
Кластер 2: Caterpillar, DuPont, Ford, General Electric, Honda, Toyota, Texas Instruments
Кластер 3: Dell, Hewlett-Packard, IBM, Xerox
Кластер 4: Apple, Amazon, Cisco Systems, Google, Intel, Microsoft
Кластер 5: Northrop Grumman
Кластер 6: Kellogg, Coca-Cola, Pepsi
Кластер 7: American International Group, American Express, Boeing, Citigroup, Goldman Sachs, JPMorgan Chase, Wells Fargo

```

Рис.4.1. – Результати кластеризації фондового ринку

У ході виконання завдання було проведено аналіз фондового ринку та пошук підгруп компаній за допомогою алгоритму **Affinity Propagation** (Поширення подібності).

1. Підготовка даних: Для аналізу було використано історичні дані котирувань за період 2023–2024 рр., отримані через бібліотеку *yfinance*. Оскільки деякі компанії зі старого списку (Yahoo, Canon, Navistar) були делістинговані або змінили тікери, скрипт автоматично відфільтрував їх і продовжив роботу з 33 активними компаніями. В якості ознак для кластеризації використовувалася щоденна варіація цін (різниця між ціною закриття та відкриття).

2. Результати кластеризації: Алгоритм *Affinity Propagation* автоматично (без попереднього задання кількості груп) виділив **7 кластерів**, які чітко відображають реальні економічні сектори:

- **Кластер 1 (Енергетика):** Об'єднав нафтогазові гіганти (*Chevron, Exxon, ConocoPhillips, Valero*) та агропромислову корпорацію *Archer-Daniels-Midland*. Це свідчить про сильну кореляцію цін на енергоносії.
- **Кластер 4 (Технології):** Згрупував лідерів ІТ-індустрії (*Apple, Amazon, Google, Microsoft, Intel, Cisco*).
- **Кластер 6 (Споживчі товари):** Чітко виділив виробників продуктів харчування та напоїв (*Coca-Cola, Pepsi, Kellogg*).
- **Кластер 7 (Фінанси):** Об'єднав найбільші банки та фінансові установи (*JPMorgan, Goldman Sachs, Citigroup, Wells Fargo, American Express*).
- **Інші кластери:** Виділили групи важкої промисловості/автопрому (*Ford, Toyota, Caterpillar*) та "старої" техніки (*IBM, Dell, HP*).

Використання моделі **GraphicalLasso** для побудови графа кореляцій та алгоритму **Affinity Propagation** дозволило успішно виявити приховану структуру ринку. Модель змогла самостійно згрупувати компанії за їхньою галузевою приналежністю виключно на основі подібності у коливаннях їхніх акцій, довівши свою ефективність для фінансового аналізу.

Висновок: У ході виконання лабораторної роботи №4 було проведено комплексне дослідження методів неконтрольованого машинного навчання засобами мови Python. На першому етапі було реалізовано алгоритм k-середніх (k-means) для класифікації набору даних Iris, що підтвердило його високу ефективність у швидкому розбитті об'єктів на групи за умови попередньо відомої кількості кластерів, причому результати власної програмної реалізації повністю співпали з бібліотечним рішенням. Наступним кроком стало застосування непараметричного методу зсуву середнього (Mean Shift) до синтетичних даних, де алгоритм продемонстрував свою головну перевагу — здатність автоматично

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Пр.7

визначати оптимальну кількість кластерів на основі аналізу локальної щільності точок без втручання користувача. Завершальним етапом став аналіз реального фондового ринку за допомогою алгоритму поширення подібності (Affinity Propagation), який успішно виявив приховані економічні закономірності та згрупував компанії у чіткі галузеві сектори (енергетика, технології, фінанси) виключно на основі подібності динаміки цін їхніх акцій. Робота засвідчила, що вибір методу кластеризації залежить від наявності априорних знань про дані: k-means є оптимальним для фіксованої кількості груп, тоді як Mean Shift та Affinity Propagation ефективніші для розвідувального аналізу складних структур.

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Пр. 7	Арк.