

ЛАБОРАТОРНА РОБОТА № 7

ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ

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

Посилання на гіт: <https://github.com/IPZ213mmv/Lab7.git>

Завдання 2.1. Кластеризація даних за допомогою методу k-середніх

Лістинг програми:

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

# Завантаження даних
X = np.loadtxt('data_clustering.txt', delimiter=',')

# Визначення кількості кластерів
num_clusters = 5

# Створення моделі K-means та навчання
kmeans = KMeans(n_clusters=num_clusters, random_state=0)
kmeans.fit(X)

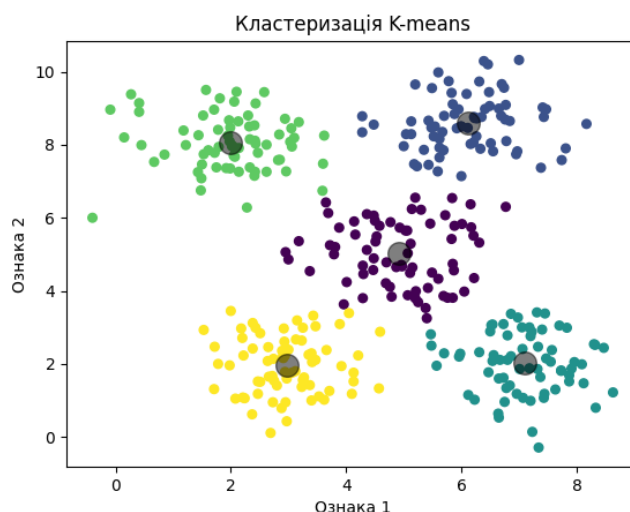
# Передбачення кластерів для кожного об'єкта
y_kmeans = kmeans.predict(X)

# Візуалізація результатів
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=30, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)

plt.xlabel('Ознака 1')
plt.ylabel('Ознака 2')
plt.title('Кластеризація K-means')
plt.show()
```

Результат:



Житомирська політехніка.24.121.12.000 – Лр7

Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.		Маліновський М.В.		
Перевір.		Голенко М.Ю.		
Керівник				
Н. контр.				
Зав. каф.				

Звіт з
Лабораторної роботи 7

Літ.	Арк.	Аркушів
	1	4
ФІКТ Гр. ІПЗ-21-3		

Завдання 2.2. Кластеризація К-середніх для набору даних Iris

Лістинг програми:

```
# Завдання 2.2. Кластеризація К-середніх для набору даних Iris
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris

# Завантаження набору даних Iris
iris = load_iris()
X = iris['data'] # Атрибути (довжина та ширина чашолистка і пелюстки)
y = iris['target'] # Цільова змінна (класи квітів)

# Ініціалізація та налаштування моделі К-середніх
# Створюємо об'єкт кластеризації з 3 кластерами (за кількістю класів у наборі да-
них Iris)
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=10, max_iter=300,
tol=0.0001, random_state=42)

# Навчання моделі на даних
kmeans.fit(X)

# Отримання передбачених кластерів
y_kmeans = kmeans.predict(X)

# Візуалізація результатів кластеризації
# Малюємо точки даних, розфарбовуючи їх відповідно до кластерів
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

# Відображення центрів кластерів
centers = kmeans.cluster_centers_ # Координати центрів кластерів
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5,
label='Centers')
plt.title('K-Means Clustering on Iris Dataset')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.legend()
plt.show()

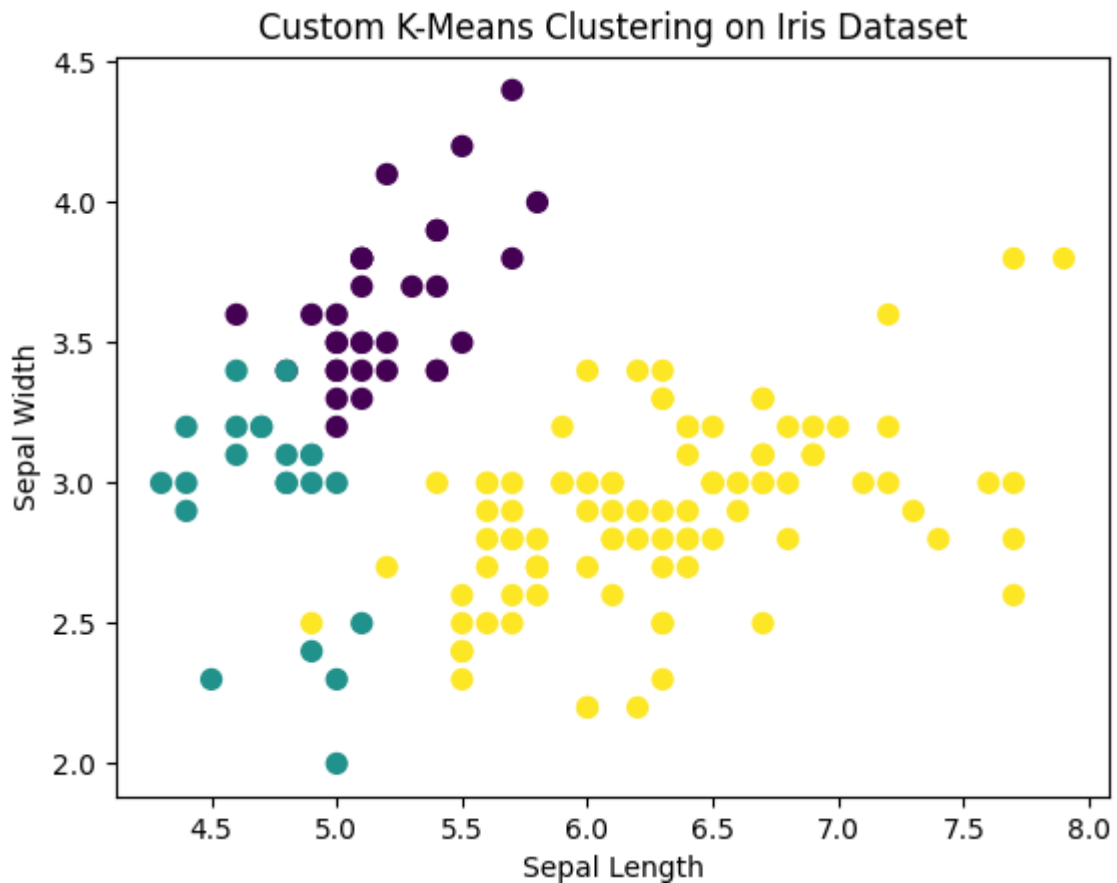
# Додаткова функція для пошуку кластерів
# Реалізація алгоритму К-середніх вручну для розуміння
from sklearn.metrics import pairwise_distances_argmin

def find_clusters(X, n_clusters, rseed=2):
    """Знаходження кластерів вручну"""
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]
    while True:
        # Визначення міток на основі найближчих центрів
        labels = pairwise_distances_argmin(X, centers)
        # Обчислення нових центрів як середнього значення точок у кожному кластері
        new_centers = np.array([X[labels == i].mean(0) for i in
range(n_clusters)])
        # Перевірка на зупинку, якщо центри більше не змінюються
        if np.all(centers == new_centers):
            break
        centers = new_centers
    return centers, labels
```

		Малиновський М.В.			Житомирська політехніка.24.121.12.000 – Лр7	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```
# Використання функції для знаходження 3 кластерів
centers, labels = find_clusters(X, 3)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title('Custom K-Means Clustering on Iris Dataset')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```

Результат:



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

Лістинг програми:

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.metrics import adjusted_rand_score

# Завантаження набору даних Iris
iris = load_iris()
X = iris.data # Атрибути (довжина/ширина чашолистка та пелюстки)
y = iris.target # Цільова змінна (класи квітів)

# Визначення параметрів K-середніх
n_clusters = 3 # Кількість кластерів

# Створення та навчання моделі K-середніх
kmeans = KMeans(n_clusters=n_clusters, random_state=0)
kmeans.fit(X)
```

		Малиновський М.В.			Житомирська політехніка.24.121.12.000 – Лр7	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```
# Передбачення кластерів для даних
y_pred = kmeans.predict(X)

# Візуалізація результатів у 3D
fig = plt.figure(1, figsize=(8, 6)) # Збільшено розмір фігури для кращого вигляду
ax = fig.add_subplot(111, projection='3d') # Створення 3D-графіка
ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=y_pred, edgecolor='k', cmap='viridis') # Використання кольорової карти для кластерів

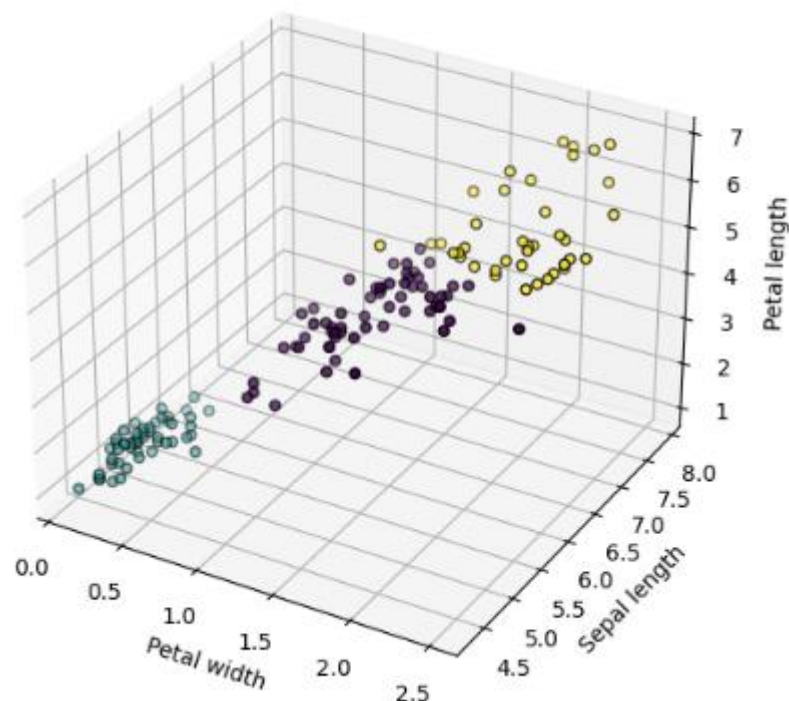
# Налаштування підписів осей
ax.set_xlabel('Petal width')
ax.set_ylabel('Sepal length')
ax.set_zlabel('Petal length')

# Додавання заголовка
plt.title("K Means clustering on Iris dataset")
plt.show()

# Оцінка якості кластеризації за Adjusted Rand Index
print("Adjusted Rand Index:", adjusted_rand_score(y, y_pred))
```

Результат:

K Means clustering on Iris dataset



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

		Маліновський М.В.			Житомирська політехніка.24.121.12.000 – Лр7	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		4