

ЛАБОРАТОРНА РОБОТА № 4

ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ

Мета роботи: використовуючи 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

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

# Завантаження вхідних даних
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(())

# Створення об'єкту KMeans
kmeans = KMeans(init='k-means++', n_clusters=num_clusters, n_init=10)

# Навчання моделі кластеризації KMeans
kmeans.fit(X)

# Визначення кроку сітки
step_size = 0.01

#Відображення точок сітки
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))

# Передбачення вихідних міток для всіх точок сітки
output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])

# Графічне відображення областей та виділення їх кольором
output = output.reshape(x_vals.shape)
plt.figure()
plt.clf()
```

					ДУ «Житомирська політехніка».22.121.07.806 – ІПЗк						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кияшенко А.С.			Системи штучного інтелекту Лабораторна №4			Літ.	Арк.	Аркушів	
Перевір.									1	7	
Реценз.								ФІКТ Гр. ІПЗк-19-1			
Н. Контр.											
Затверд.											

```
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')

# Відображення вхідних точок
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')

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

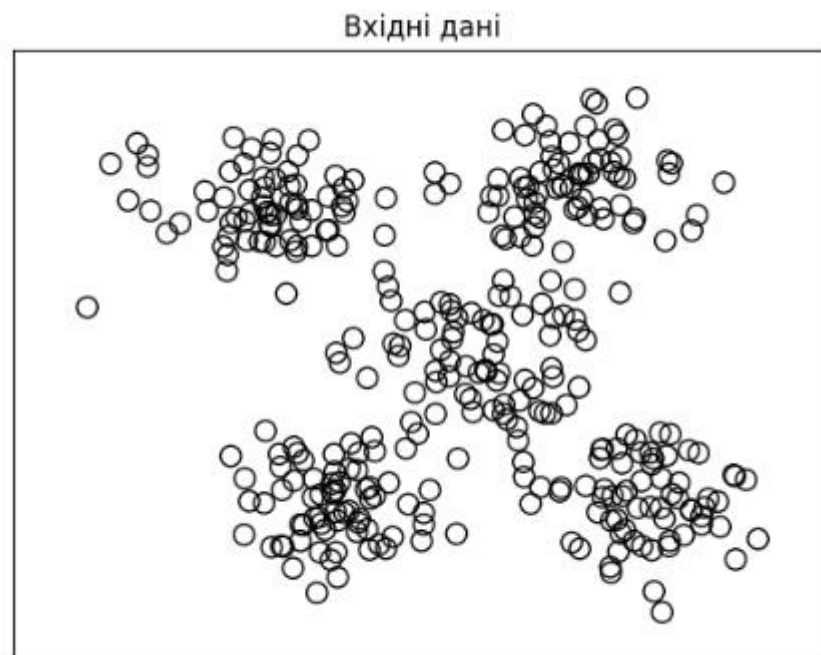


Рис. 1. Вхідні дані

		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІПЗк	Арк.
		.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

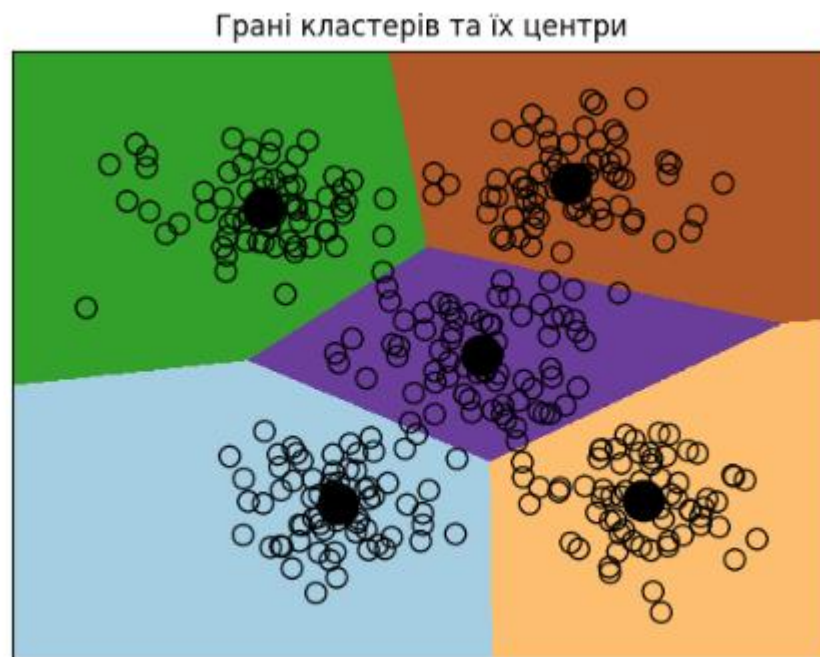


Рис. 2. Грані кластерів та їх центри

Я провів кластеризацію даних методом k-середніх, використовуючи файл вхідних даних: data_clustering.txt.

Завдання 2. Кластеризація К-середніх для набору даних Iris

```
import matplotlib.pyplot as plt
from pandas import read_csv
from sklearn.metrics import pairwise_distances_argmin
import numpy as np
from sklearn.cluster import KMeans

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
dataset = read_csv(url)

# Розділення датасету
array = dataset.values
X = array[:, 0:4]
y = array[:, 4]

plt.scatter(X[:, 0], X[:, 3], s=50)

# Параметри кластеризації
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=10, max_iter=300,
tol=0.0001,
verbose=0, random_state=
None, copy_x=True, algorithm='auto')

# Тренування
kmeans.fit(X)

# Передбачення найближчого кластеру
y_kmeans = kmeans.predict(X)

# Налаштування відображення
plt.scatter(X[:, 0], X[:, 3], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
```

		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІІЗк	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```
plt.scatter(centers[:, 0], centers[:, 3], c='black', s=200, alpha=0.5)
plt.show()

# Оголошення функції
def find_clusters(X, n_clusters, rseed=2):
    # Випадковий вибір кластерів
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

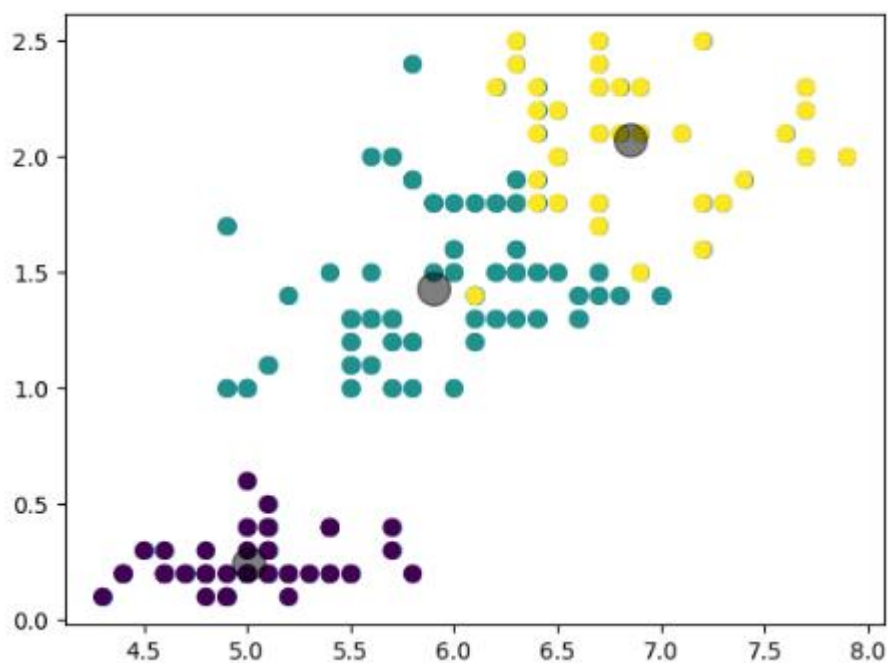
    while True:
        # Зіставити підписи на основі найближчого центру
        labels = pairwise_distances_argmin(X, centers)

        # Знайти нові центри на основі значень точок
        new_centers = np.array([X[labels == i].mean(0)
                                for i in range(n_clusters)])

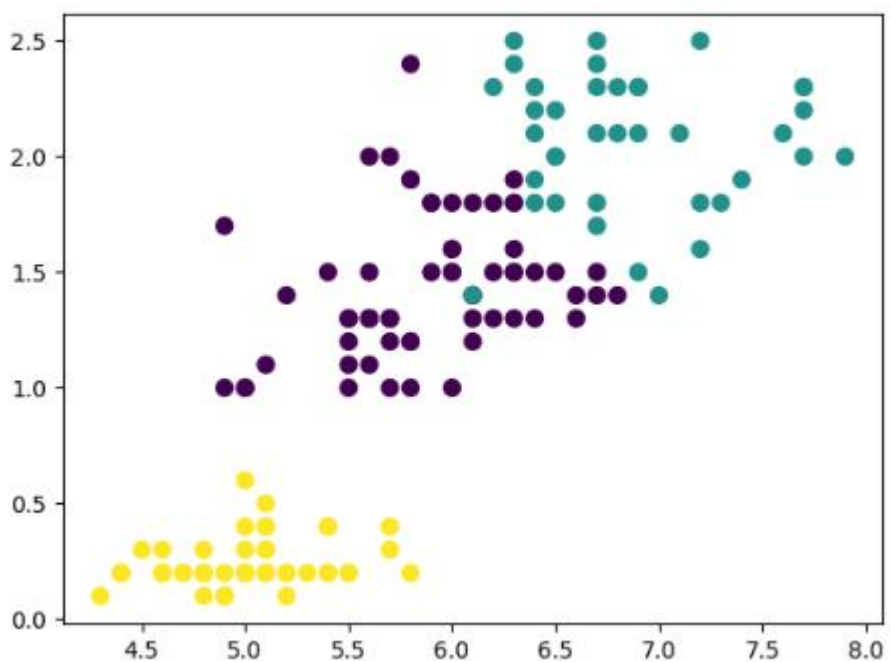
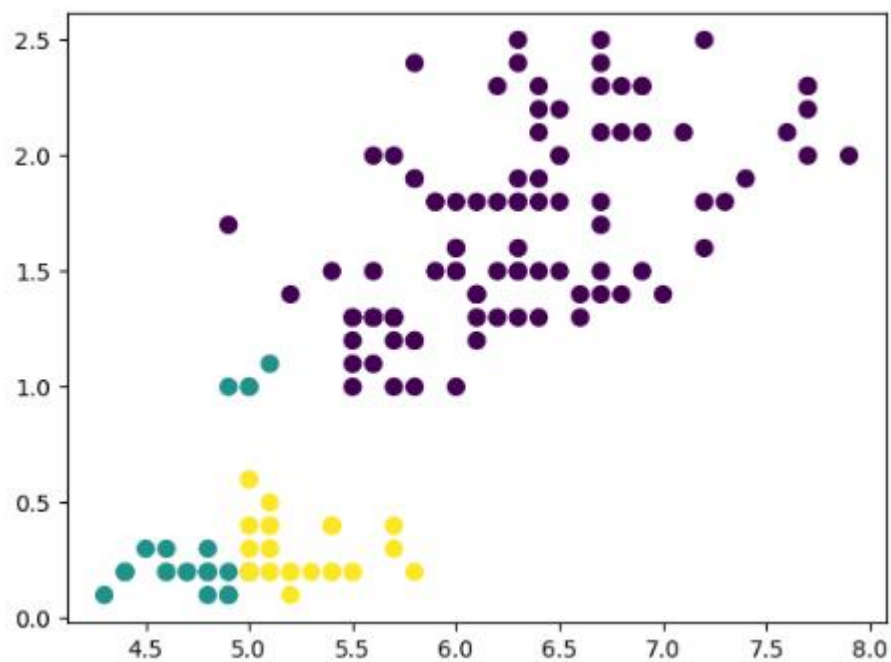
        # Перевірка на зіставлення
        if np.all(centers == new_centers):
            break
        centers = new_centers

    return centers, labels

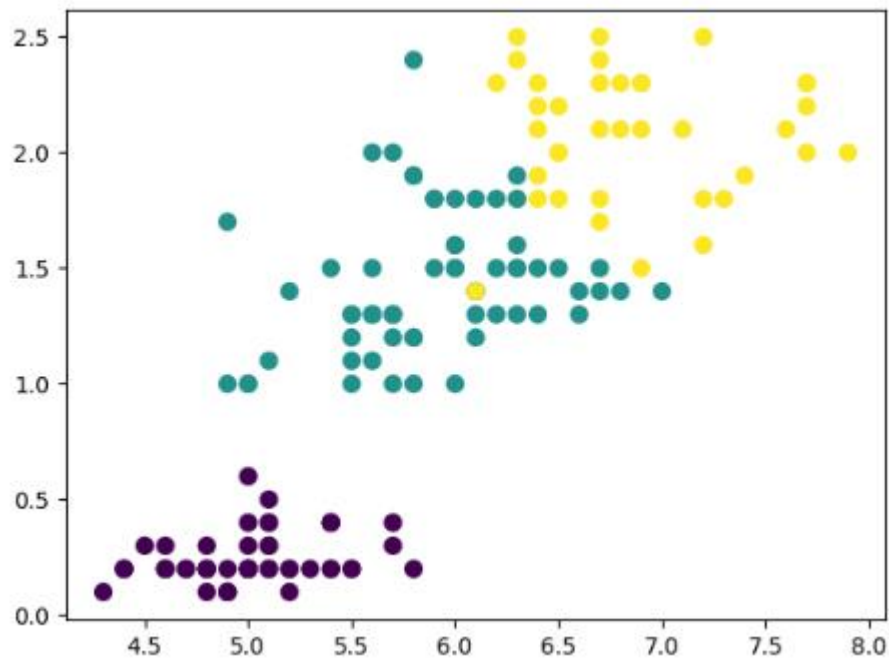
centers, labels = find_clusters(X, 3)
plt.scatter(X[:, 0], X[:, 3], c=labels,
            s=50, cmap='viridis')
plt.show()
centers, labels = find_clusters(X, 3, rseed=0)
plt.scatter(X[:, 0], X[:, 3], c=labels,
            s=50, cmap='viridis')
plt.show()
labels = KMeans(3, random_state=0).fit_predict(X)
plt.scatter(X[:, 0], X[:, 3], c=labels,
            s=50, cmap='viridis')
plt.show()
```



		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІПЗк	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4



		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІПЗк	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		



Завдання 3. Оцінка кількості кластерів з використанням методу зсуву середнього

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

# Завантаження
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('\nCenters of clusters:\n', cluster_centers)

# Оцінка кількості кластерів
labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)

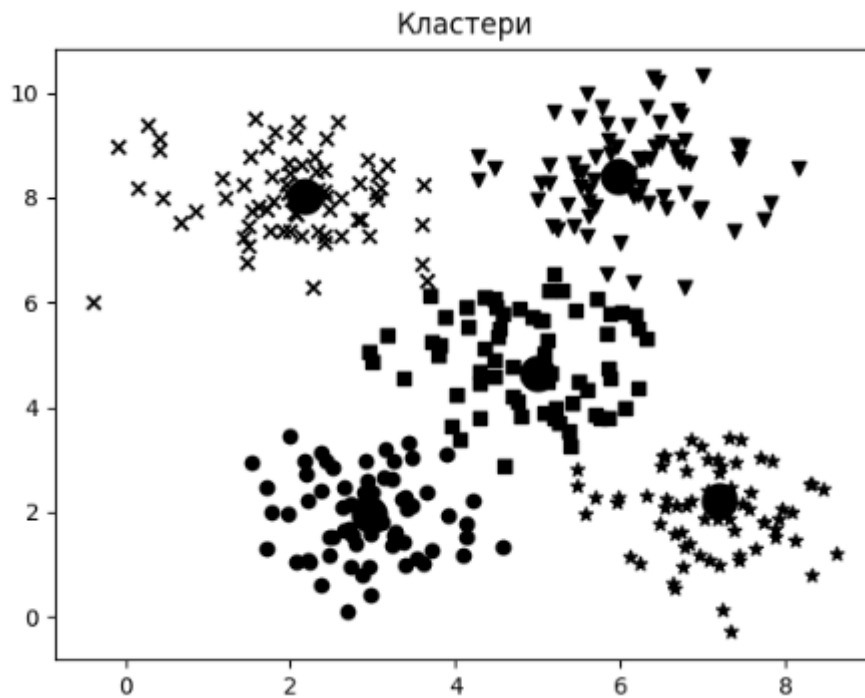
# Відображення на графіку точок та центрів кластерів
plt.figure()
markers = 'o*xvs'
for i, marker in zip(range(num_clusters), markers):
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
                color='black')

# Відображення на графіку центру кластера
cluster_center = cluster_centers[i]
```

		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІІЗк	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
plt.plot(cluster_center[0], cluster_center[1], marker='o',
         markerfacecolor='black', markeredgecolor='black',
         markersize=15)

plt.title('Кластери')
plt.show()
```



Посилання на Git: <https://github.com/Grum74/AI>

Висновок

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

		Кияшенко А.С.			ДУ «Житомирська політехніка».22.121.07. 806– ІПЗк	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		