**ЛАБОРАТОРНА РОБОТА № 4**

**ДОСЛІДЖЕННЯ МЕТОДІВ НЕКОНТРОЬОВАНОГО 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 КМеаns  
kmeans = KMeans(init='k-means++', n\_clusters=num\_clusters, n\_init=10)  
  
# Навчання моделі кластеризації КМеаns  
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()

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()

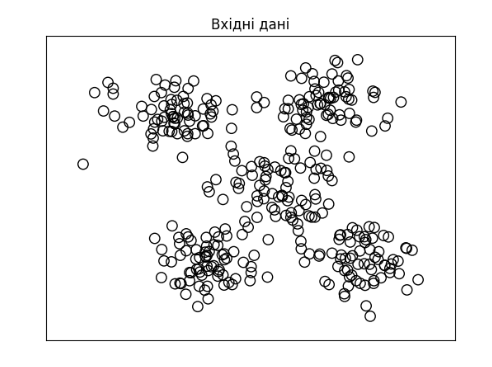


Рис. 1. Вхідні дані

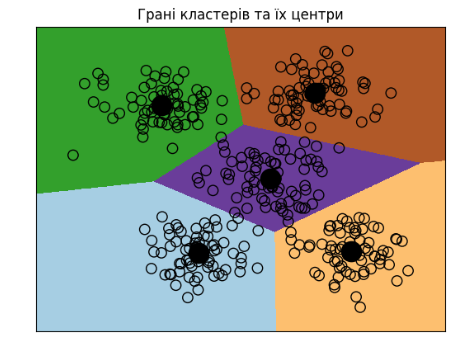


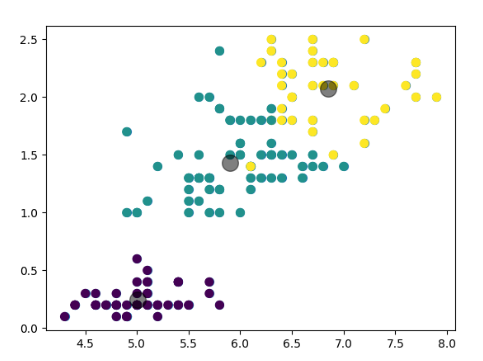
Рис. 2. Грані кластерів та їх центри

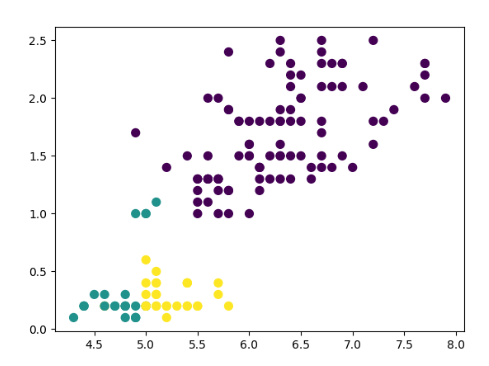
Я провів кластеризацію даних методом k-середніх, використовуючи файл вхідних даних: data\_clustering.txt.

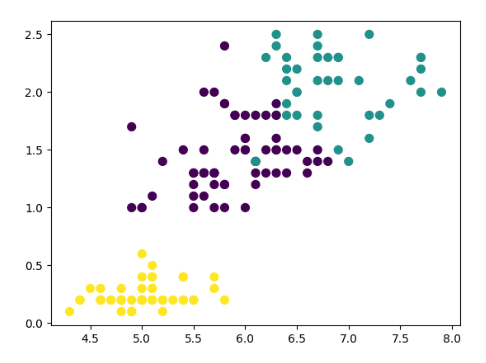
**Завдання 2. Кластеризація K-середніх для набору даних Iris**

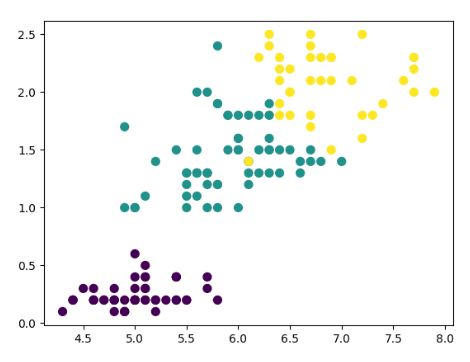
import matplotlib.pyplot as plt  
from pandas import read\_csv  
from sklearn.metrics import pairwise\_distances\_argmin  
import numpy as np  
from sklearn.cluster import KMeans  
  
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"  
dataset = read\_csv(url)  
  
# Розділення 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\_

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()





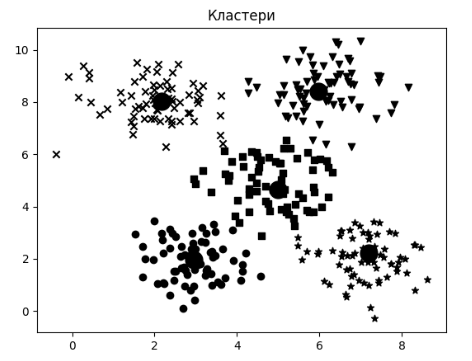




**Завдання 3. Оцінка кількості кластерів з використанням методу зсуву середнього**

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 Х  
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]

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.