МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ФУНДАМЕНТАЛЬНИХ НАУК

Кафедра прикладної математики

**ЗВІТ**

про виконання лабораторної роботи №3 із дисципліни:

“**Математичні основи штучного інтелекту”**

Виконав:

студент групи ПМ-31

Матолін Марко

Прийняв:

доцент кафедри ПМ

Пабирівський В.В.

Львів ‒ 2023

**Тема:** «Кластеризація даних»

# **Завдання**

Розробити програмну реалізацію алгоритму кластеризації та перевірити роботу алгоритму на тестовій множині даних.

Етапи виконання завдання

1. Вивчити із використанням запропонованих літературних джерел зміст задачі кластеризації та методи кластеризації даних.
2. Згенерувати тестову послідовність з N значень (для визначеності, можна покласти N≥1000), що є парами дійсних чисел на одиничному квадраті.
3. Реалізувати допоміжну функцію для обчислення міри віддалі.
4. Реалізувати допоміжну функцію для виконання алгоритму кластеризації за методом К-середніх (англ. K-means).
5. Реалізувати допоміжну функцію для виконання алгоритму кластеризації

за будь-яким іншим (за вибором) методом (ієрархічний, Cсередніх, мінімального покриваючого дерева, по-шарової кластеризації, ART1, тощо).

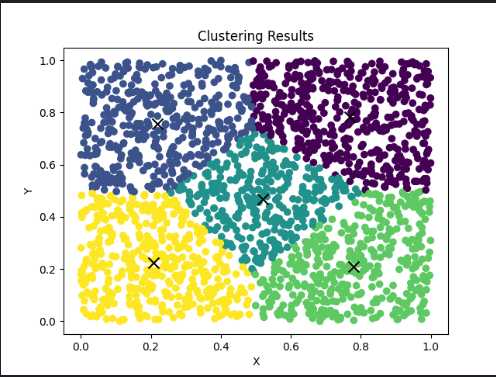
1. Безпосередньо реалізувати кластеризацію даних двома методами та порівняти результати кластеризації.
2. Порівняйте кількість кластерів та якість кластеризації (можна просто оцінити середньо-зважене розмірів утворених кластерів згідно заданої міри віддалі для кожного із методів).

Варіант -3

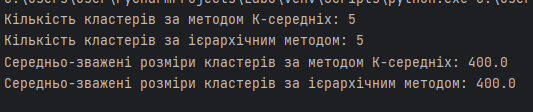
Код програми:

import numpy as np  
from sklearn.cluster import KMeans, AgglomerativeClustering  
from sklearn.metrics import pairwise\_distances  
import matplotlib.pyplot as plt  
  
# Функція для генерації тестової послідовності на одиничному квадраті  
def generate\_data(n):  
 return np.random.rand(n, 2)  
  
# Функція для обчислення міри віддалі (в даному випадку Евклідова відстань)  
def distance\_measure(x, y):  
 return np.linalg.norm(x - y)  
  
# Функція для виконання кластеризації за методом К-середніх  
def kmeans\_clustering(data, k):  
 kmeans = KMeans(n\_clusters=k, n\_init=10)  
 kmeans.fit(data)  
 labels = kmeans.labels\_  
 cluster\_centers = kmeans.cluster\_centers\_  
 return labels, cluster\_centers  
  
# Функція для виконання кластеризації за ієрархічним методом  
def hierarchical\_clustering(data, k):  
 clustering = AgglomerativeClustering(n\_clusters=k)  
 labels = clustering.fit\_predict(data)  
 return labels  
  
# Функція для обчислення середньо-зваженого розміру утворених кластерів за мірою віддалі  
def compute\_cluster\_sizes(data, labels):  
 sizes = []  
 for cluster\_label in np.unique(labels):  
 cluster\_points = data[labels == cluster\_label]  
 cluster\_size = len(cluster\_points)  
 sizes.append(cluster\_size)  
 return sizes  
  
# Функція для візуалізації результатів кластеризації  
def plot\_clusters(data, labels, centers=None):  
 plt.scatter(data[:, 0], data[:, 1], c=labels)  
 if centers is not None:  
 plt.scatter(centers[:, 0], centers[:, 1], marker='x', c='black', s=100)  
 plt.xlabel('X')  
 plt.ylabel('Y')  
 plt.title('Clustering Results')  
 plt.show()  
  
# Генерування тестової послідовності з 1000 значень  
data = generate\_data(2000)  
  
# Виконання кластеризації за методом К-середніх  
k = 5 # кількість кластерів  
k1 = 5  
kmeans\_labels, kmeans\_centers = kmeans\_clustering(data, k)  
  
# Визуалізація результатів кластеризації за методом К-середніх  
plot\_clusters(data, kmeans\_labels, kmeans\_centers)  
  
# Виконання кластеризації за ієрархічним методом  
hierarchical\_labels = hierarchical\_clustering(data, k1)  
  
# Визуалізація результатів кластеризації за ієрархічним методом  
plot\_clusters(data, hierarchical\_labels)  
  
# Обчислення середньо-зваженого розміру утворених кластерів за мірою віддалі  
kmeans\_cluster\_sizes = compute\_cluster\_sizes(data, kmeans\_labels)  
hierarchical\_cluster\_sizes = compute\_cluster\_sizes(data, hierarchical\_labels)  
  
# Порівняння результатів кластеризації  
num\_kmeans\_clusters = len(np.unique(kmeans\_labels))  
num\_hierarchical\_clusters = len(np.unique(hierarchical\_labels))  
mean\_kmeans\_cluster\_size = np.mean(kmeans\_cluster\_sizes)  
mean\_hierarchical\_cluster\_size = np.mean(hierarchical\_cluster\_sizes)  
  
print("Кількість кластерів за методом К-середніх:", num\_kmeans\_clusters)  
print("Кількість кластерів за ієрархічним методом:", num\_hierarchical\_clusters)  
print("Середньо-зважені розміри кластерів за методом К-середніх:", mean\_kmeans\_cluster\_size)  
print("Середньо-зважені розміри кластерів за ієрархічним методом:", mean\_hierarchical\_cluster\_size)

**Результат виконання:**







**Висновок:**

На лабораторній роботі я ознайомилась з методами кластеризації даних та розробила програмну реалізацію кількох видів алгоритму на тестовій множині даних.