Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

Лабораторна робота № 4

З дисципліни: "Інтелектуальний аналіз даних"

Тема: "Кластеризація, розбиття об'єктів на класи в системах розпізнавання образів «без вчителя»”

Підготувала студентка

Групи АІ – 212

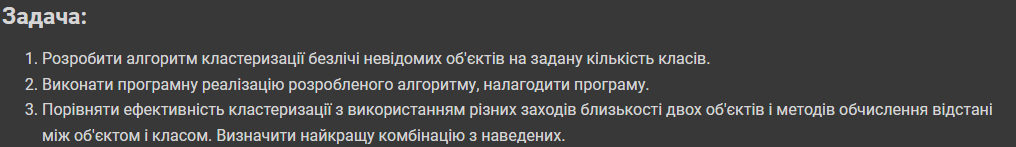
Козуб К.О

Перевірили:

Антощук С.Г

Кошутіна Д.В

Одеса 2023



<https://colab.research.google.com/drive/1OW41du9oy12lezMikL3OiWRFFODYXOJc?usp=sharing>

**Результат:**

Середня відстань до класу: 10.291280261759312



Рисунок 1 – Кластеризація К-середніх

Висновок: На лабораторній роботі 4. Я виконала завдання алгоритм K-середніх

Додаток А

import numpy as np

import matplotlib.pyplot as plt

# Визначення функції для алгоритму K-середніх

def k\_means(data, k, distance\_func=np.linalg.norm, max\_iters=100):

    # Ініціалізація випадкових центрів кластерів

    centroids = data[np.random.choice(range(len(data)), k, replace=False)]

    # Ініціалізація міток для точок даних

    labels = np.zeros(len(data), dtype=int)

    # Виконання алгоритму K-середніх протягом максимальної кількості ітерацій

    for \_ in range(max\_iters):

        # Обчислення відстаней від кожної точки даних до кожного центру

        distances = np.array([distance\_func(data - centroid, axis=1) for centroid in centroids])

        # Призначення точок даних до найближчого центру

        labels = np.argmin(distances, axis=0)

        # Обчислення нових центрів як середнє значення точок даних в кожному кластері

        new\_centroids = np.array([data[labels == i].mean(axis=0) if np.sum(labels == i) > 0 else centroid for i, centroid in enumerate(centroids)])

        # Перевірка, чи збіглися центри кластерів

        if np.all(centroids == new\_centroids):

           break

        centroids = new\_centroids

    return labels, centroids

# Визначення функції для обчислення середньої відстані до класу

def compute\_average\_distance\_to\_class(data, labels, centroids, distance\_func=np.linalg.norm):

    distances = np.array([distance\_func(data[labels == i] - centroid) for i, centroid in enumerate(centroids) if np.sum(labels == i) > 0])

    return distances.mean()

# Визначення функції для візуалізації результатів K-середніх

def plot\_k\_means(data, labels, centroids):

    colors = ['g', 'b', 'y', 'c']

    for i in range(len(data)):

        plt.scatter(data[i, 0], data[i, 1], c=colors[labels[i]], marker='o')

    plt.scatter(centroids[:, 0], centroids[:, 1], c='black', marker='x', s=100, label='Центри кластерів')

    plt.title('Кластеризація K-середніх')

    plt.xlabel('Ознака 1')

    plt.ylabel('Ознака 2')

    plt.grid()

    plt.legend()

    plt.show()

# Випадкове насіння для відтворюваності результатів

np.random.seed(735)

data = np.random.rand(100, 2) \* 10

k = 4

labels\_combined, centroids\_combined = k\_means(data, k)

average\_distances = compute\_average\_distance\_to\_class(data, labels\_combined, centroids\_combined)

print("\nСередня відстань до класу:", average\_distances)

print("")

plot\_k\_means(data, labels\_combined, centroids\_combined)