

Національний університет “Одеська Політехніка”

Інститут комп’ютерних систем

Кафедра інформаційних систем

Лабораторна робота № 2

з дисципліни “Інтелектуальний аналіз даних”

Підготувала студентка

Групи АІ – 212

Козуб К.О

Перевірили:

Антощук С.Г

Кошутіна Д.В

Одеса 2023

Тема: Дослідження геометричних мір наближення об'єктів та класів у системах розпізнавання

Мета роботи: Дослідження геометричних мір наближення об'єктів та класів у системах розпізнавання

Google Colab:

<https://colab.research.google.com/drive/1zB3WLb6FHJaO8VM0fuFsxCiWWaGm-h8Q?usp=sharing>

Завдання:

1. Виберіть попередньо оброблені дані, які вже були виконані в лабораторній роботі 1, відповідно до ваших варіантів.
2. Розробіть алгоритм прийняття рішення в детермінованій системі розпізнавання на основі використання відомих геометричних мір наближення.
3. Виконати програмне впровадження розробленого алгоритму, налаштувати програму.
4. Виконати контрольне розпізнавання невідомих об'єктів за векторами їх ознак (встановлюються самостійно).
5. Порівняйте ефективність прийняття рішень щодо призначення невідомих об'єктів даним класам для різних методів обчислення відстані між об'єктом та класом. Визначте найкраще поєднання запропонованих.

Результат:

```
Ознака 1:  
29  
Ознака 2:  
6  
Невідомий об'єкт належить до Класу 1
```

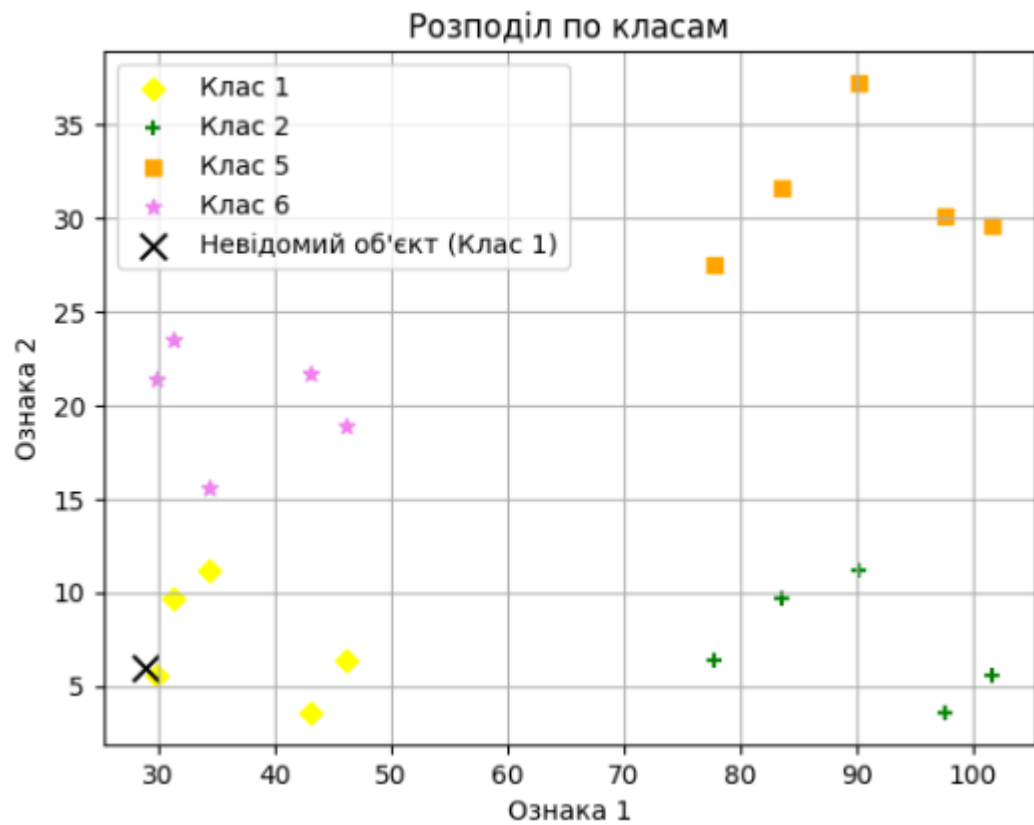


Рисунок 1 – Розподіл об'єктів по класам

Висновок: На лабораторній роботі № 2. Я допомогою методів обчислення ступеня близькості та обчислення відстані дослідили розподіл об'єктів по класам

Додаток А

```
import numpy as np
import matplotlib.pyplot as plt

class1 = np.array([[ 29.835, 5.61],[ 34.425, 11.22],[ 46.155, 6.375],[ 31.365,
9.69],[ 43.095, 3.57]])

class2 = np.array([[ 101.75, 5.61],[ 90.27, 11.22],[ 77.775, 6.375],[ 83.64, 9.69],[
97.665, 3.57]])

class5 = np.array([[ 101.75, 29.58],[ 90.27, 37.23],[ 77.777, 27.54],[ 83.64,
31.62],[ 97.665, 30.09]])

class6 = np.array([[ 29.835, 21.42],[ 34.425, 15.56],[ 46.155, 18.87],[ 31.365,
23.46],[ 43.095, 21.68]])

# Функція для розрахунку відстані Мінковського
def minkowski_distance(a, b, p):
    return (np.sum(np.abs(a - b) ** p)) ** (1 / p)

# Функція для обчислення евклідової відстані
def euclidean_distance(a, b):
    return np.sqrt(np.sum((a - b) ** 2))

# Функція для визначення відстані до найближчого об'єкта в класі
def distance_to_closest(obj, cls_data):
    distances = [euclidean_distance(obj, cls_obj) for cls_obj in cls_data]
    return np.min(distances)

# Побудуйте точки даних для кожного класу
for cls_data, label, marker, color in zip([class1, class2, class5, class6], ['Клас 1',
'Клас 2', 'Клас 5', 'Клас 6'], ['D', '+', 's', '*'], ['yellow', 'green', 'orange', 'violet']):
    cls_data = np.array(cls_data)
```

```

plt.scatter(cls_data[:, 0], cls_data[:, 1], label=label, marker=marker, c=color)
# Встановіть мітки та назву для ділянки
plt.xlabel('Ознака 1')
plt.ylabel('Ознака 2')
plt.title('Розподіл по класам')
plt.legend()
plt.grid(True)
# Запропонувати користувачеві ввести значення для невідомого об'єкта
print("Ознака 1:")
first_feature = float(input())
print("Ознака 2:")
second_feature = float(input())
# Створіть масив для невідомого об'єкта
obj = np.array([first_feature, second_feature])
# Обчисліть відстані до найближчого об'єкта в кожному класі для невідомого об'єкта
distances = []
for cls_data in [class1, class2, class5, class6]:
    dist = distance_to_closest(obj, cls_data)
    distances.append(dist)
# Визначте клас невідомого об'єкта на основі найближчої відстані
assigned_class = np.argmin(distances) + 1
# Нанесіть невідомий об'єкт на графік чорним маркером «x».
plt.scatter(obj[0], obj[1], label=f'Невідомий об'єкт (Клас {assigned_class})',
            marker='x', c='black', s=100)
plt.legend()

print(f'Невідомий об'єкт належить до Класу {assigned_class}')
plt.show()

```