Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №3

з дисципліни «Методи та технології штучного інтелекту»

на тему

«Дослідження алгоритму нечіткої кластеризації»

Виконала:
студентка групи ІС-12
Павлова Софія

Викладач:
Шимкович В. М.

Київ – 2023

1. Постановка задачі

Мета: Вирішення практичного завдання кластеризації методами нечіткої логіки.

Завдання:

1. Необхідно сформулювати завдання в галузі обчислювальної техніки або програмування, для якої була б необхідна автоматична класифікація множини об'єктів, які задаються векторами ознак в просторі ознак.

2. Вирішити сформульовану задачу з використанням механізму кластеризації методами нечіткої логіки за допомогою програмних засобів моделювання або мови програмування високого рівня.

3. Знайти центри кластерів і побудувати графік зміни значень цільової функції.

2. Виконання

Формулювання завдання:

Нехай ми маємо дані про **500 покупців**: їх **витрати** і **частоту покупок**. Наше завдання – розділити покупців на **5 кластерів** на основі цих ознак:

High-Spenders: Покупці, які витрачають велику суму грошей та роблять покупки досить часто.

Regular Shoppers: Покупці, які не витрачають настільки багато, але вони роблять покупки з високою частотою.

Low-Frequency Big-Spenders: Покупці, які роблять покупки рідко, але кожна покупка великої вартості.

Low-Frequency Small-Spenders: Покупці, які не витрачають багато грошей і роблять покупки рідко.

Middle-Class Shoppers: Покупці, які мають помірні витрати і середню частоту покупок.

Вхідні дані:

Дані про покупців, де для кожного покупця відомі дві ознаки: "Витрати" та "Частота покупок".

Вихідні дані:

Результати кластеризації, де кожен покупець включений до одного з 5 кластерів.

Генерація датасету:

Згенеруємо датасет для нашої задачі.

Лістинг:

```
import numpy as np
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
```

```
import skfuzzy as fuzz

# Параметри для генерації датасету
n_samples = 500
n_features = 2
n_clusters = 5
names = ['фіксовані центри', 'рандомні центри']
# Вибір режиму зчитування даних
print('Оберіть режим генерації центрів:')
for i in range(len(names)):
    print(i + 1, '-', names[i])
data_mode = int(input('mode:'))
# Якщо джерело даних існує
if data_mode in range(1, len(names) + 1):
    # Фіксовані центри
    if (data_mode == 1):
        centers = [[3, 2], [4, 5], [5, 3], [7, 4], [8, 6]]
        centers = np.array(centers)
    # Рандомні центри
    else:
        centers = np.random.rand(n_clusters, n_features) * 10
random_state = 42

# Функція генерації датасету
def create_customers_dataset(n_samples, n_features, centers, random_state):
    dataset, mark = make_blobs(n_samples=n_samples, n_features=n_features,
centers=centers, random_state=random_state)
    return dataset, mark

# Функція виведення графіків точок датасету
def plot(dataset, title, clustering):
    # Якщо результати кластеризації
    if (clustering):
        plt.scatter(dataset[:, 0], dataset[:, 1], marker='x', color='black')
    # Якщо згенерований датасет
    if not clustering:
        plt.scatter(dataset[:, 0], dataset[:, 1])
    plt.title(title)
    plt.xlabel('Витрати')
    plt.ylabel('Частота покупок')
    plt.show()

# Датасет
customers, mark = create_customers_dataset(n_samples, n_features, centers, random_state)
plot(customers, 'Дані покупців', False)
```

Результат:

```
Оберіть режим генерації центрів:
1 - фіксовані центри
2 - рандомні центри
mode:1
```

Рисунок 1 – Режим генерації датасету

Для спрощення аналізу надалі будемо розглядати фіксовані центри.

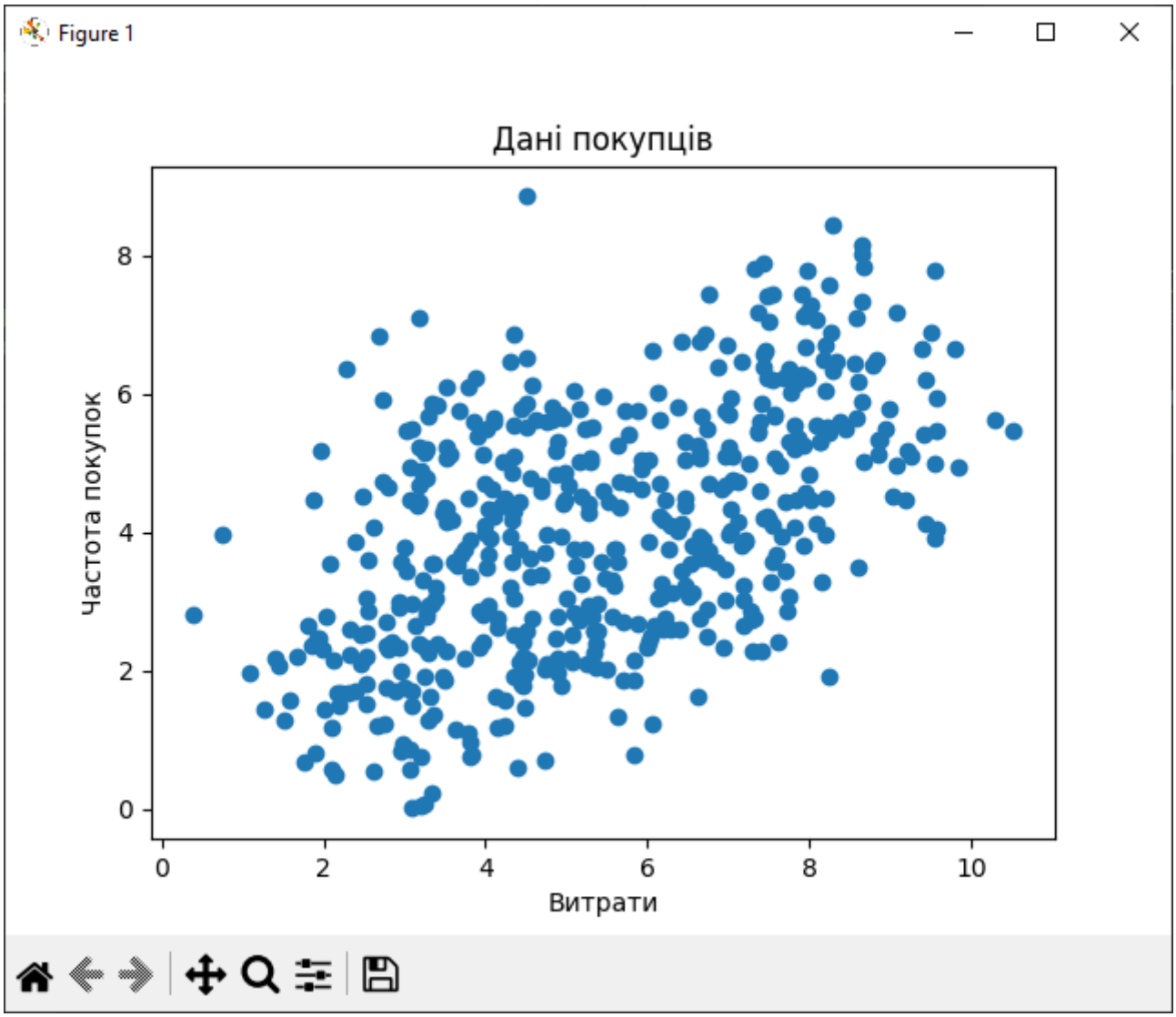

Рисунок 2 – Дані покупців за двома ознаками

Фазифікована кластеризація:

Для знаходження найкращого рішення та аналізу якості створеної моделі для розв’язання поставленої задачі, зробимо нашу програму інтерактивною. Користувач буде вводити параметри фазифікації самостійно.

Лістинг:

```
# Параметри для фазифікованої кластеризації
print('Оберіть ступінь фазифікації (2):')
q = float(input('q = '))
print('Оберіть зміну центрів кластерівміж ітераціями (0.00001):')
error = float(input('error = '))
print('Оберіть максимальну кількість ітерацій (100):')
max_iter = int(input('max_iter = '))

# Функція фазифікованої кластеризації
def fuzzy_clustering(dataset, n_clusters, q, error, max_iter):
    centers, u, _, _, jm, _, _ = fuzz.cluster.cmeans(dataset.T, n_clusters, q, error,
max_iter)
    return centers, u, jm

# Функція виведення графіку кластеризації
def plot_clustering(dataset, membership, title):
    max_membership = np.argmax(membership, axis=0)
    # Візуалізація значень в кластерах
    for cluster in range(len(centers)):
        cluster_points = dataset[max_membership == cluster]
        plt.scatter(cluster_points[:, 0], cluster_points[:, 1])
    # Візуалізація центрів кластерів
    plot(centers, title, True)

# Кластеризація
customers_centers, membership, jm = fuzzy_clustering(customers, n_clusters, q, error,
max_iter)
plot_clustering(customers, membership, 'Отримані кластери та їх центри')
```

Результат:

```
Оберіть ступінь фазифікації (2):
q = 2
Оберіть зміну центрів кластерівміж ітераціями (0.00001):
error = 0.00001
Оберіть максимальну кількість ітерацій (100):
max_iter = 20
```

Рисунок 3 – Праметри фазифікації за замовчуванням

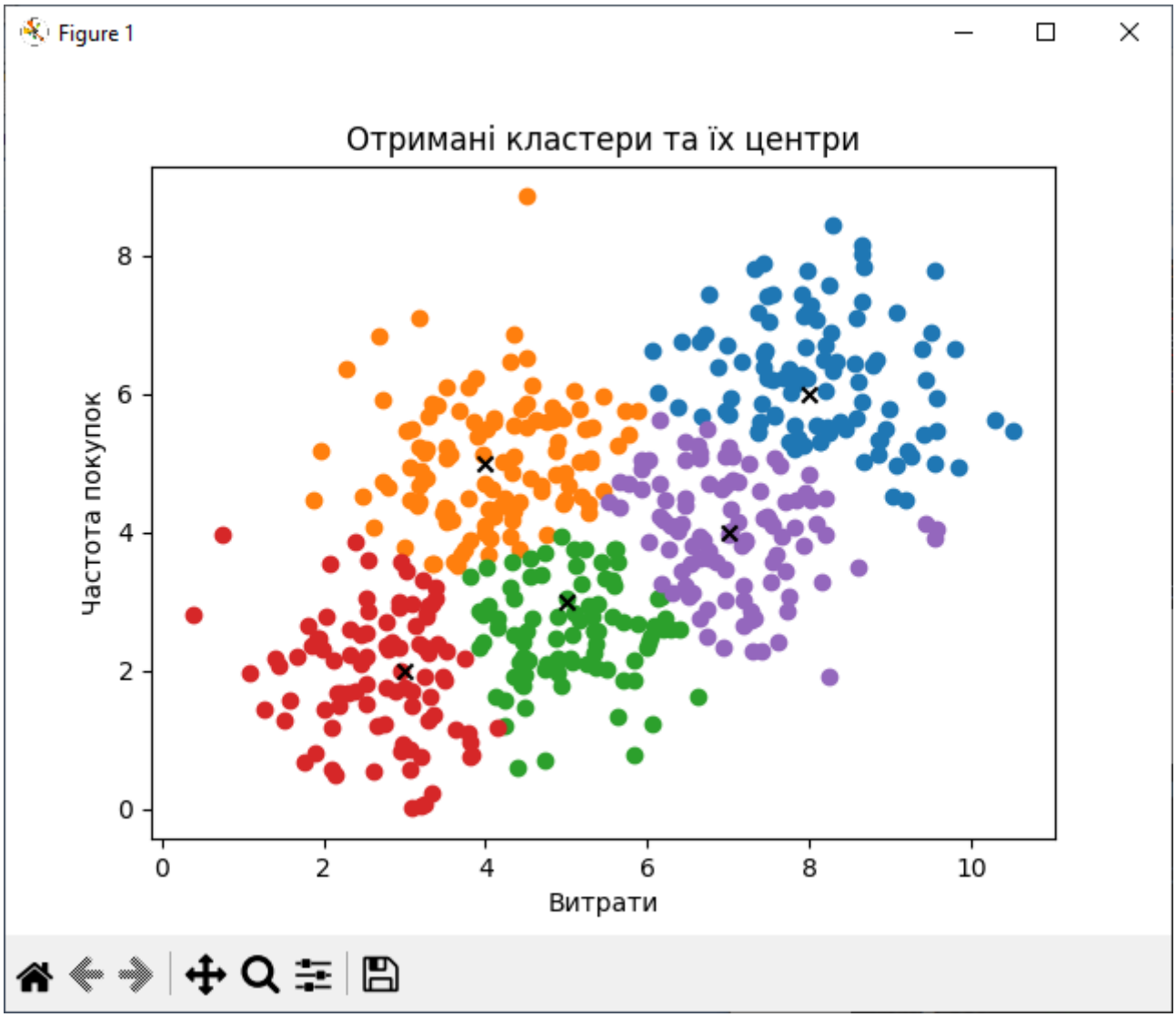

Рисунок 4 – Результати кластеризації (параметри за замовчуванням)

Центри кластерів:

Виведемо центри кластерів, знайдені під час фазифікованої кластеризації.

Лістинг:

```
# Функція виведення центрів кластеризації
def print_centers(centers, title):
    print(title)
    print(centers)

# Центри
print_centers(centers, '\nЗгенеровані центри')
print_centers(customers_centers, '\nОтримані центри')
```

Результат:

```
Згенеровані центри
[[3 2]
 [4 5]
 [5 3]
 [7 4]
 [8 6]]

Отримані центри
[[8.11171607 6.28836001]
 [4.04427162 4.98806627]
 [5.09094121 2.60972565]
 [2.72084451 1.95531806]
 [7.05695451 4.11272791]]
```

Рисунок 5 – Згенеровані та отримані центри кластерів

Бачимо, що пари точок наближено співпадають, хоча їх порядок змінений.

Цільова функція:

Виведемо графік зміни значень цільової функції, щоб отримати більш точне представлення про результати роботи моделі кластеризації на заданих параметрах (за замовчуванням).

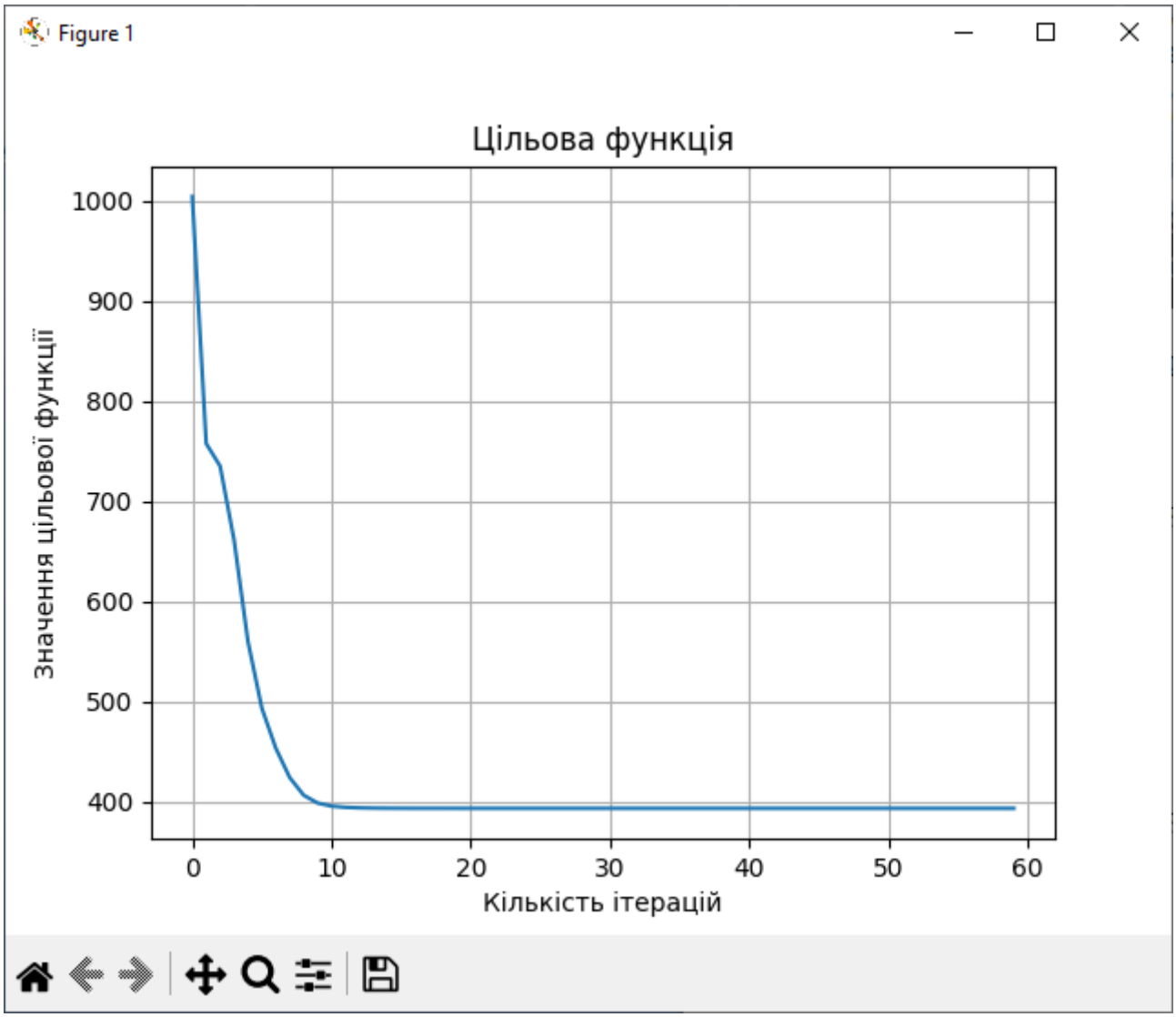

Рисунок 6 – Графік цільової функції (параметри за замовчуванням)

Бачимо, що на заданих параметрах (за замовчуванням) модель виконує кластеризацію за 60 ітерацій.

Аналіз впливу параметрів фазифікації:

Спробуємо змінити параметр **ступеню фазифікації q**.

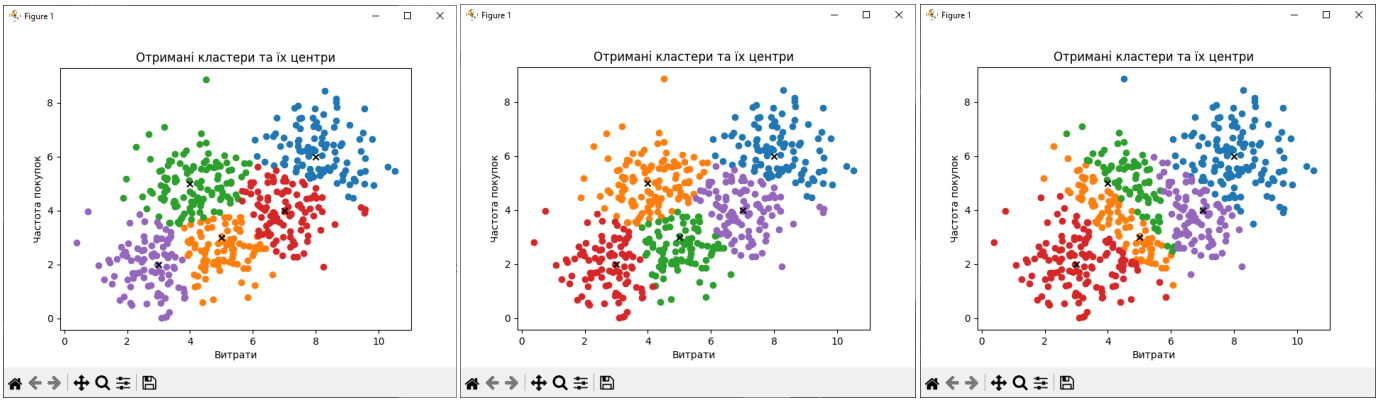

Рисунок 7 – Результати кластеризації **q = 1.1, q = 2, q = 20**

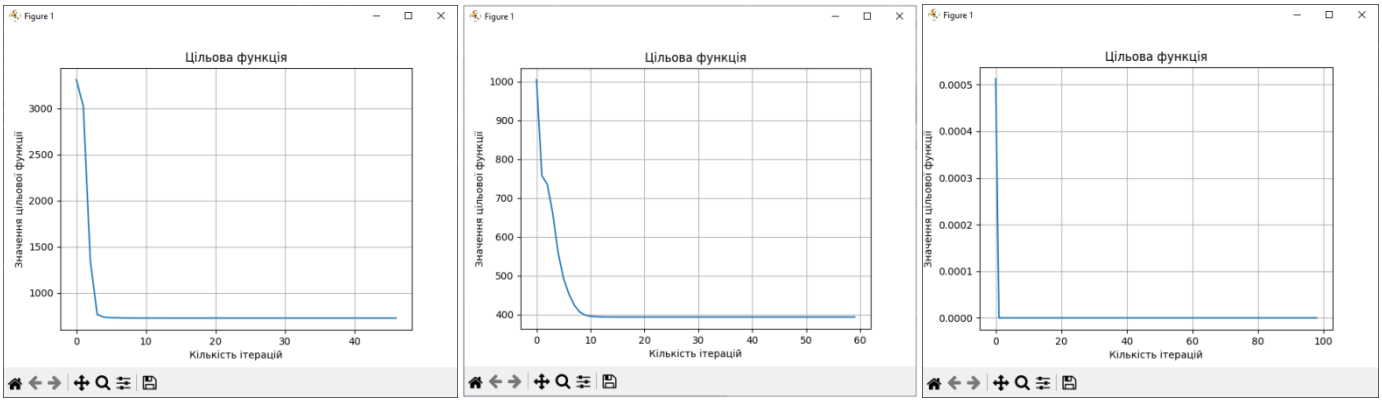

Рисунок 8 – Цільова функція **q = 1.1, q = 2, q = 20**

Бачимо, що при зменшенні параметра q до мінімально допустимого, результати кластеризації майже не змінились, натомість при збільшенні q, видно, що модель кластеризує покупців з похибкою.

Це добре видно на графіку цільової функції.

Можемо зробити висновок, що **q = 2** – призводить до найкращих результатів.

Спробуємо змінити параметр **зміни центрів кластерів між ітераціями error**.

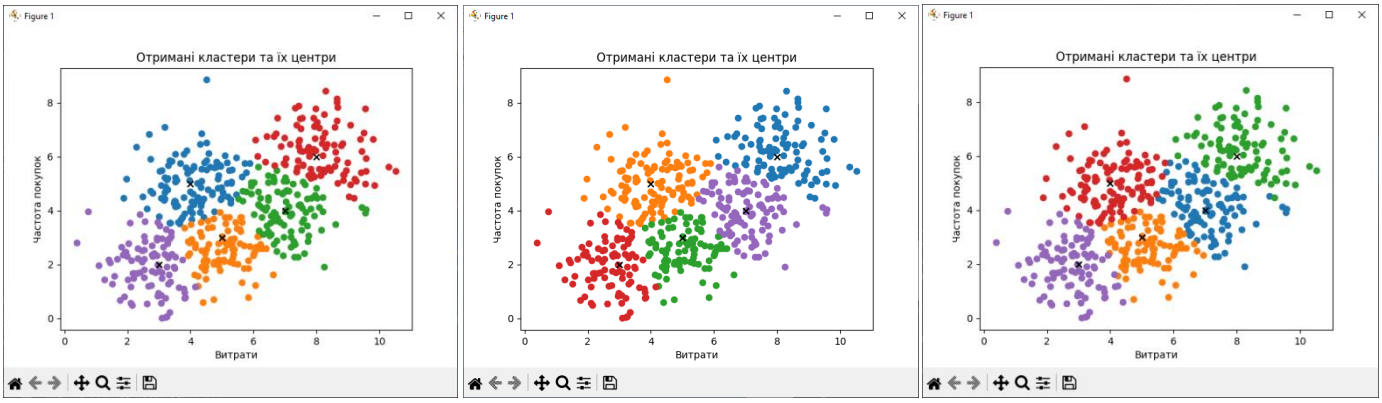

Рисунок 9 – Результати кластеризації **error = 10^{-9}, error = 10^{-5}, error = 10^{-1}**

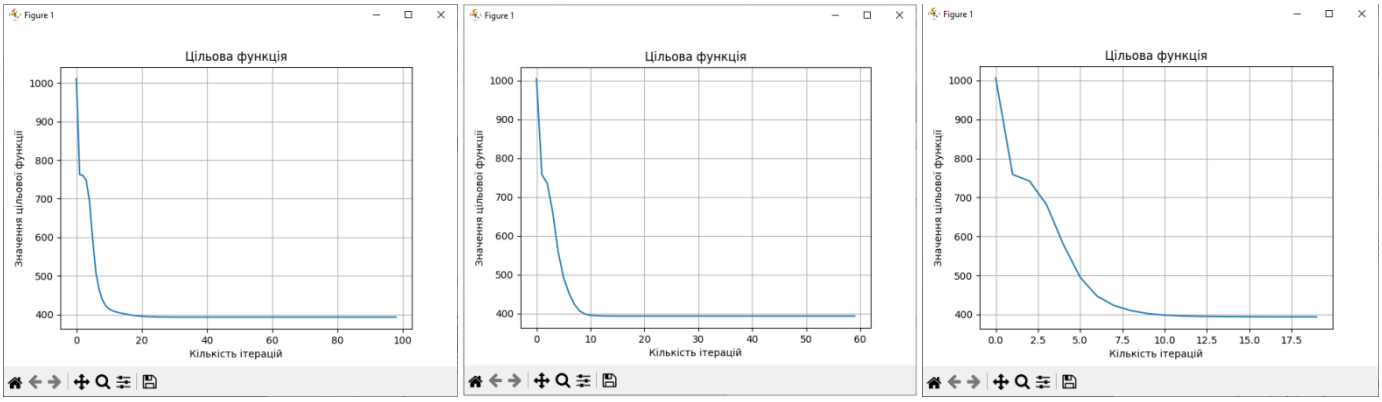

Рисунок 10 – Цільова функція **error = 10^{-9}, error = 10^{-5}, error = 10^{-1}**

Бачимо, що при зміні параметра error, результати кластеризації майже не змінились, натомість цільова функція при зменшенні error стає більш костурбатою, а при збільшенні error – більш згладженою.

Можемо зробити висновок, що **error = 10^{-9}** – призводить до найкращих результатів.

Спробуємо змінити параметр **максимальної кількості ітерацій max_iter**.

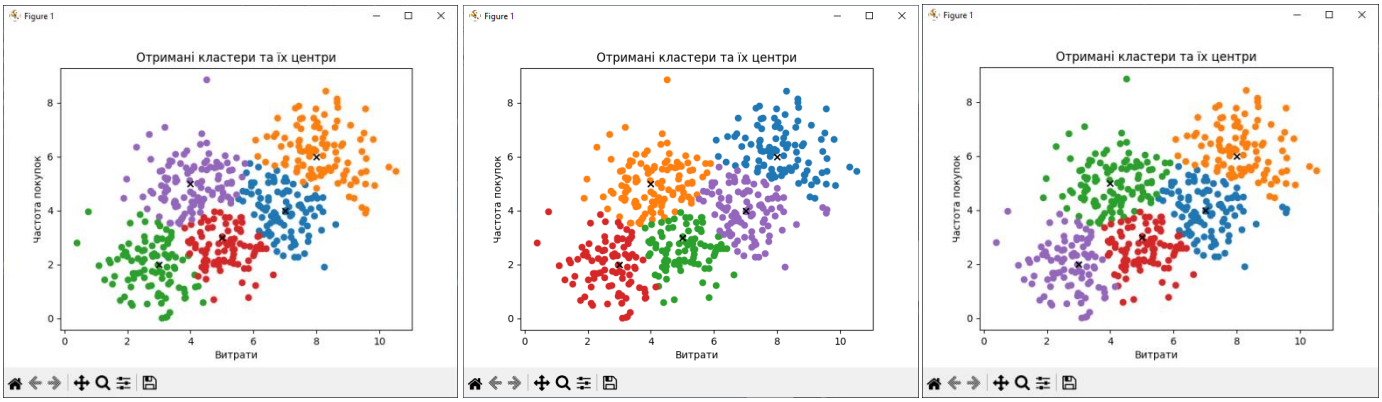

Рисунок 11 – Результати кластеризації **max_iter = 10, max_iter = 100, max_iter = 500**

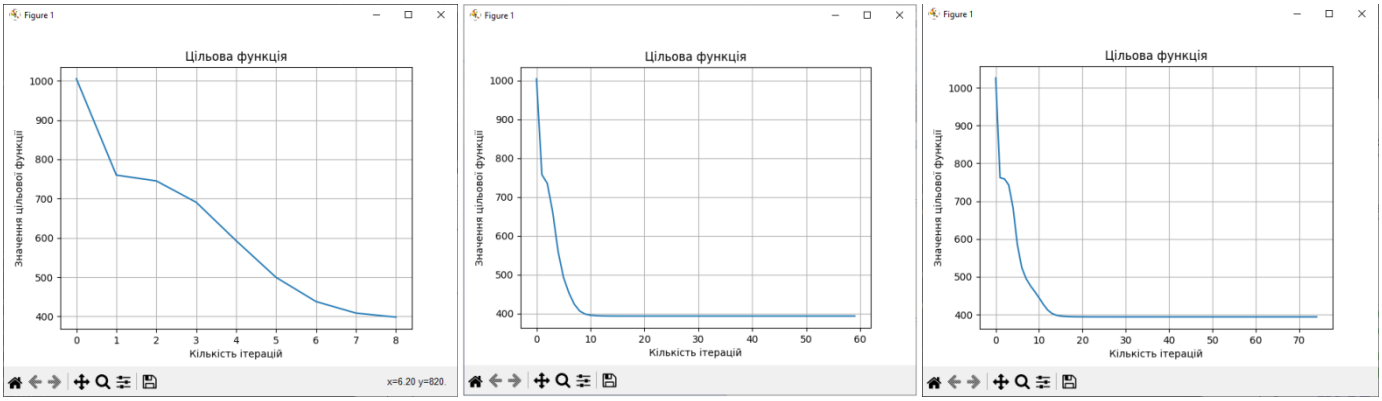

Рисунок 12 – Цільова функція **max_iter = 10, max_iter = 100, max_iter = 500**

Бачимо, що при зміні параметра max_iter, результати кластеризації майже не змінились, натомість цільова функція при зменшенні max_iter стає більш згладженою, а при збільшенні max_iter більш костурбатою.

Можемо зробити висновок, що **max_iter = 500** – призводить до найкращих результатів.

Висновок:

У даній лабораторній роботі я ознайомилась з методами кластеризації даних за допомогою засобів нечіткої логіки, навчилась кластеризувати набори даних за допомогою алгоритму Fuzzy C-Means.

Розробила алгоритм на мові програмування python для вирішення задачі нечіткої кластеризації покупців у магазині.

Провела дослідження впливу параметрів фазифікації на якість моделі і виявила, що на результат впливають ступені фазифікації (найкраще q = 2), змінf центрів кластерів між ітераціями (найкраще error = 10^{-9}), максимальна кількість ітерацій (найкраще max_iter = 500).