Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

ЗВІТ

про виконання лабораторної роботи № 1

з дисципліни «Інтелектуальний аналіз даних»

Виконала:
Студентка III курсу
Групи КА-76
Оркуша А. Д.

Перевірила:
Недашківська Н. І.

Київ – 2020

1. Постановка задачі

11. Дано масив $T = \{(t_i)|t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$, $x_{ij} \in R$, де приклад t_i характеризується m ознаками. Задано кількість кластерів $2 \leq g \leq N$. Розрахувати центри кластерів за формулою:

$$c_k = \frac{\sum_{i=1}^{N} u_{ki} t_i}{\sum_{i=1}^{N} u_{ki}}, k = 1, ..., g,$$

де $U = \{(u_{ki})|k = 1, ..., g, i = 1, ..., N\}$ - випадковим чином задана матриця початкового розбиття, $u_{ki} \in \{0, 1\}$, $\sum_{k=1}^{g} u_{ki} = 1$, $\sum_{i=1}^{N} u_{ki} < N$.

Перерахувати матрицю розбиття:

$u_{ki} = 1$ якщо $d(t_i, c_k) = \min_{l=1,...,g} d(t_i, c_l)$,

$u_{ki} = 0$ в іншому випадку,

за умови, що $d(t_i, c_k)$ - евклідова відстань між векторами.

Виконати декілька ітерацій з уточнення центрів кластерів.

2. Лістинг

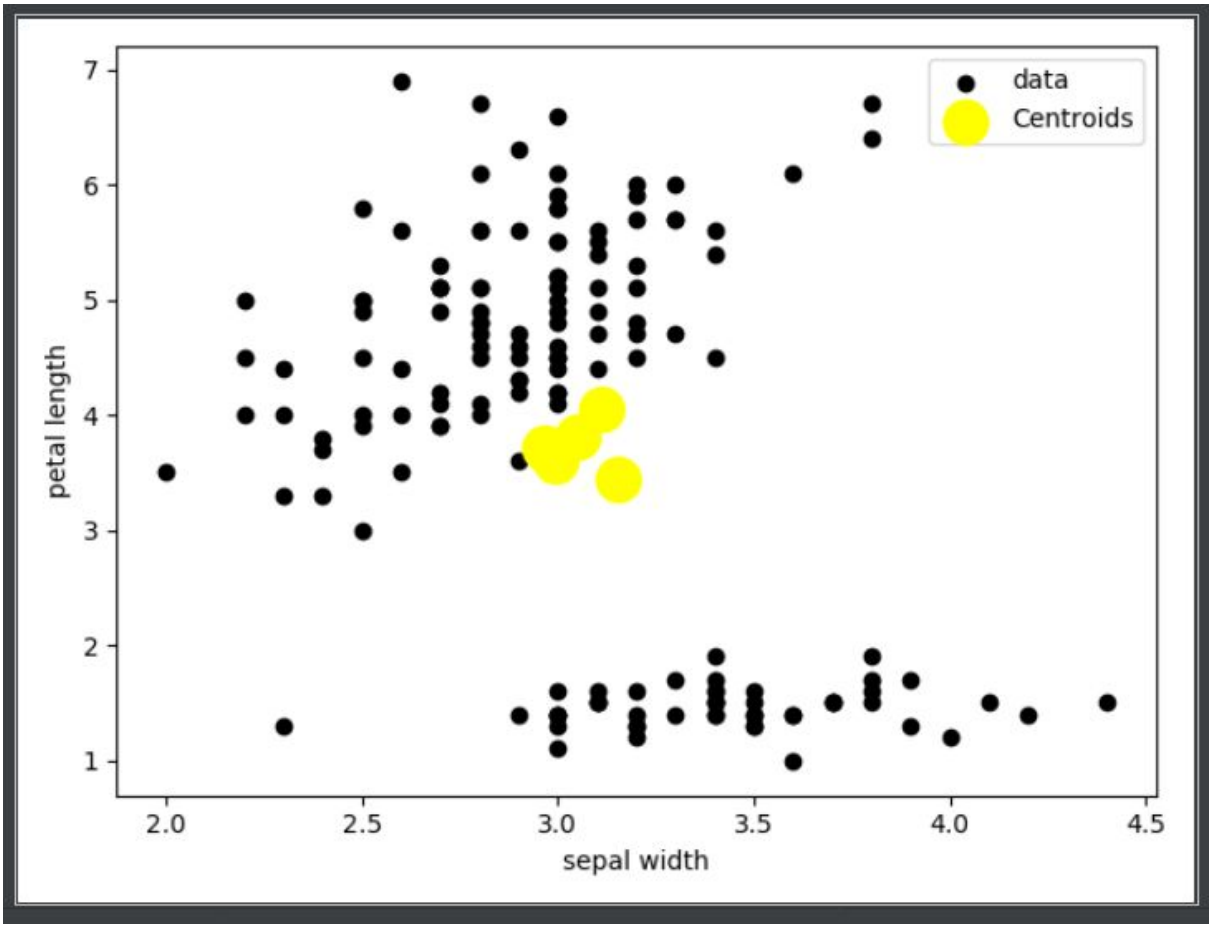
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


def dist(a, b):
    return round(np.sqrt(np.sum((a - b) ** 2)), 4)


def calc(a, b, c):
    Outp = np.empty((G, M))
    multipl = np.dot(a, b)
    temp = np.sum(a, axis=1)
    for g in range(c):
        Outp[:, g] = multipl[:, g] / temp
    return Outp


dataset = pd.read_csv('iris.csv')
dataset.describe()
```

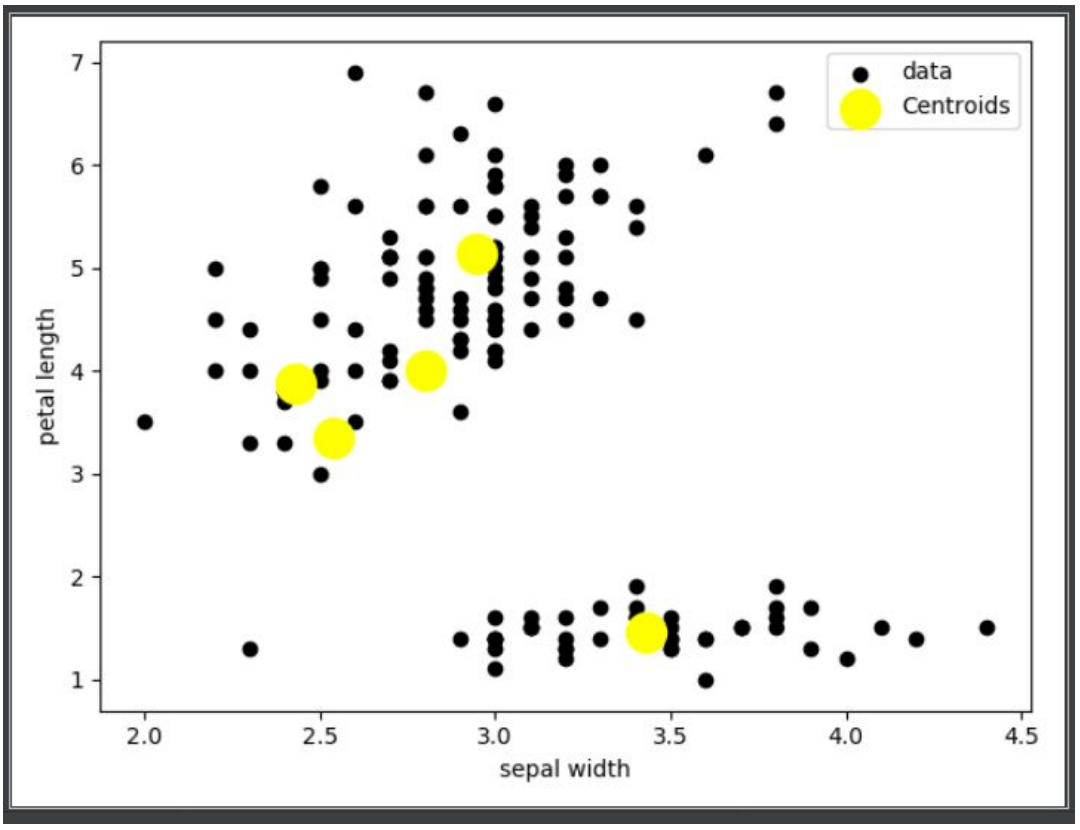
```
T = dataset.iloc[:, [1, 2]].values
N = T.shape[0]  # number of training examples
M = T.shape[1]  # number of features. Here M=2
n_iter = 5
G = 5  # number of clusters
U = np.zeros((G, N), dtype=float)  # GxN

for i in range(N):
    rand = np.random.randint(0, G)
    U[rand, i] = 1.

C = calc(U, T, M)
plt.scatter(T[:, 0], T[:, 1], c='black', label='data')
plt.scatter(C[:, 0], C[:, 1], s=300, c='yellow', label='Centroids')
plt.xlabel('sepal width')
plt.ylabel('petal length')
plt.legend()
plt.show()

for it in range(n_iter):
    EuclidianDistance = np.empty((N, G))
    for j in range(N):
        for k in range(G):
            EuclidianDistance[j, k] = dist(T[j, :], C[k, :])
    arg = np.min(EuclidianDistance, axis=1)
    # print(EuclidianDistance)

    for j in range(N):
        U[EuclidianDistance[j, :] != arg[j], j] = 0.0
        U[EuclidianDistance[j, :] == arg[j], j] = 1.0

    C = calc(U, T, M)
    plt.scatter(T[:, 0], T[:, 1], c='black', label='data')
    plt.scatter(C[:, 0], C[:, 1], s=300, c='yellow', label='Centroids')
    plt.xlabel('sepal width')
    plt.ylabel('petal length')
    plt.legend()
    plt.show()
```

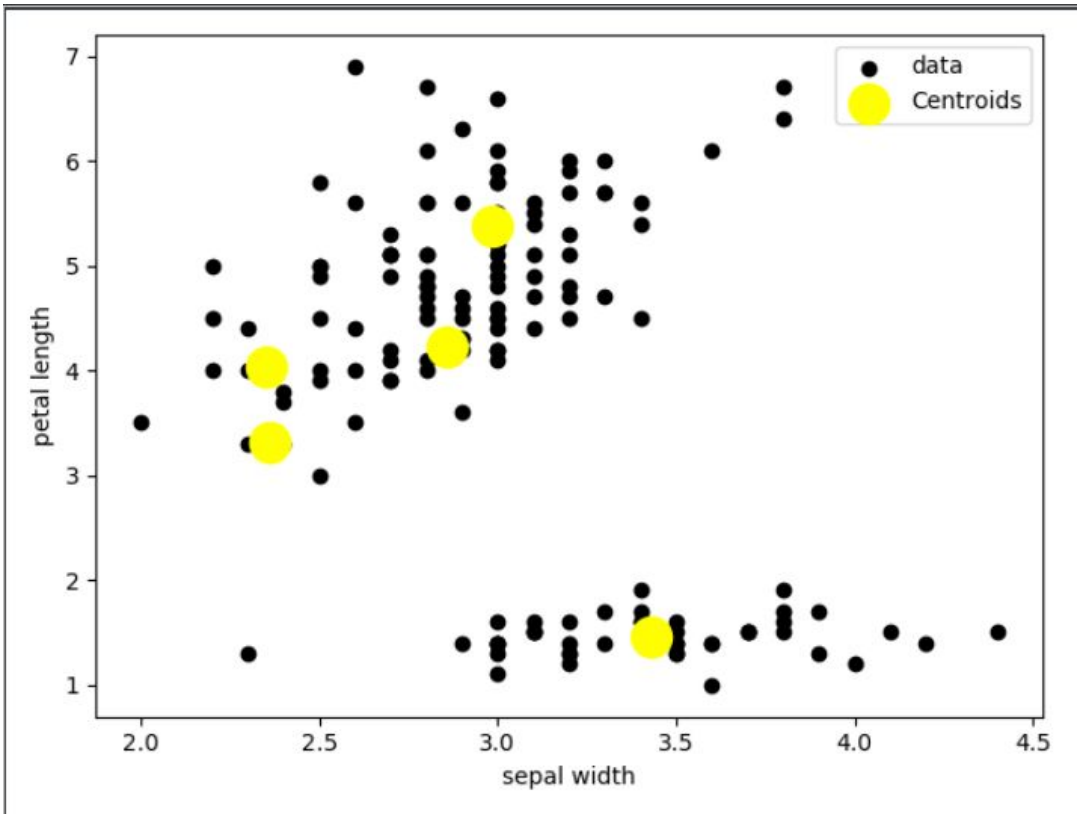
```
print(C)
```

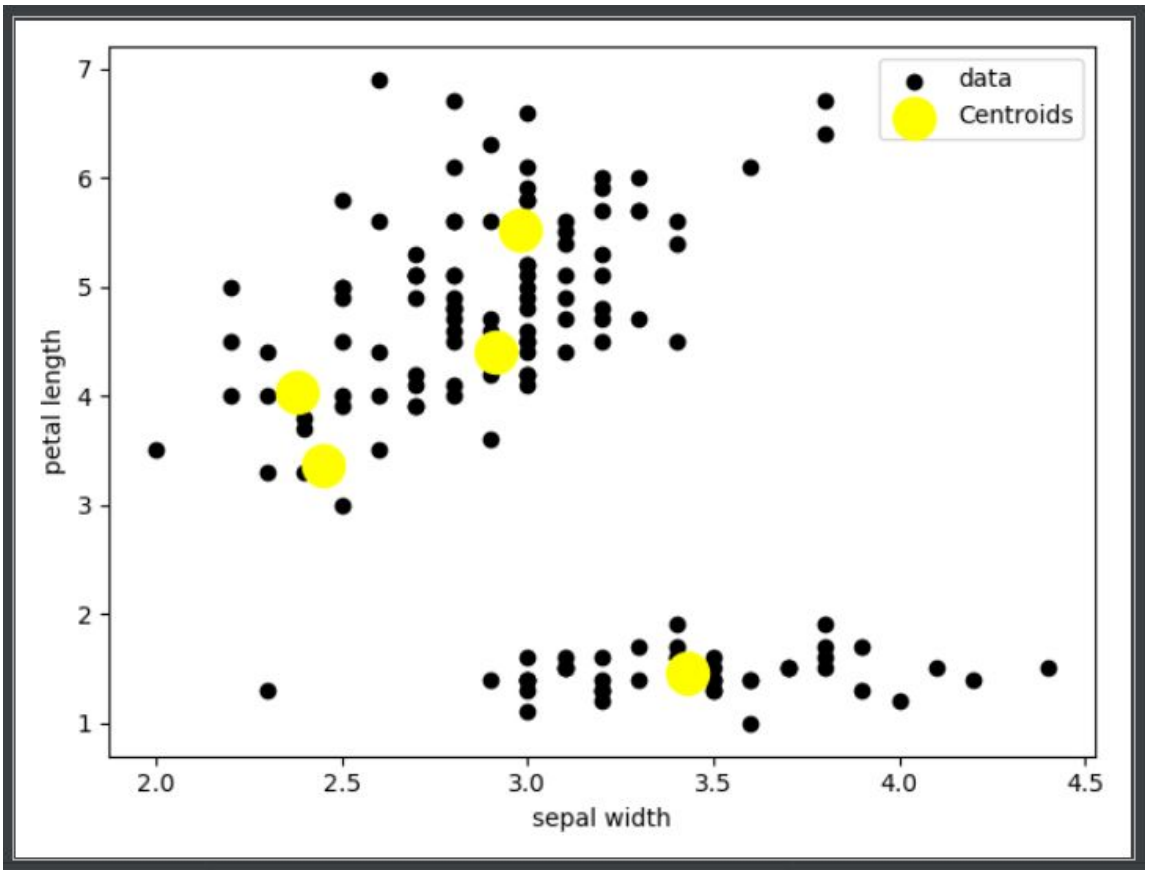
3. Результати виконання

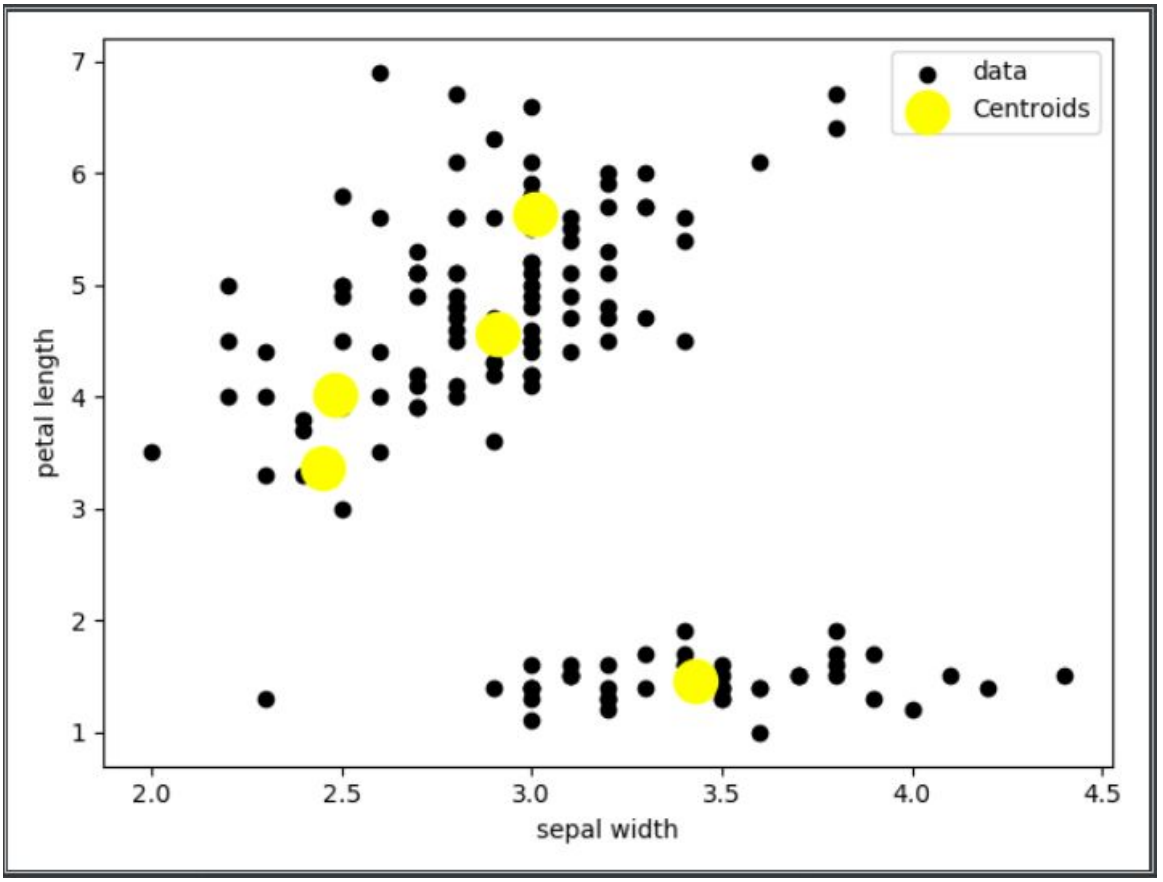
```
[[3.073  5.7676]
 [2.881  4.6786]
 [2.45   3.3667]
 [2.52   4.0333]
 [3.428  1.462 ]]

Process finished with exit code 0
```




data
Centroids
petal length
sepal width
7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5


data
Centroids
petal length
sepal width
7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5


data
Centroids
petal length
sepal width
7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5


data
Centroids
petal length
sepal width
7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5