NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
"IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Faculty of Informatics and Computer Engineering

Department of Computer Engineering

Lab 3 Report

Solving the practical task of clustering using fuzzy logic methods.
Individual tasks

Variant 12

Student, group IM-14
(group code)

in the educational and professional program

"**Software Engineering For Computer System"**

Specialty 121 "Computer Engineering"

Mehmet KULUBECİOGLU

Reviewer Associate Professor, Dr.Ph. Pavlov Valerii
(position, academic degree, academic status, surname and initials)

Kyiv – 2023

I. Purpose:

Solving the practical task of clustering using fuzzy logic methods.
Individual tasks

1. It is necessary to formulate a task in the field of computing or programming, for which
automatic classification of a set of objects, which are specified by feature vectors in the feature
space, would be necessary.

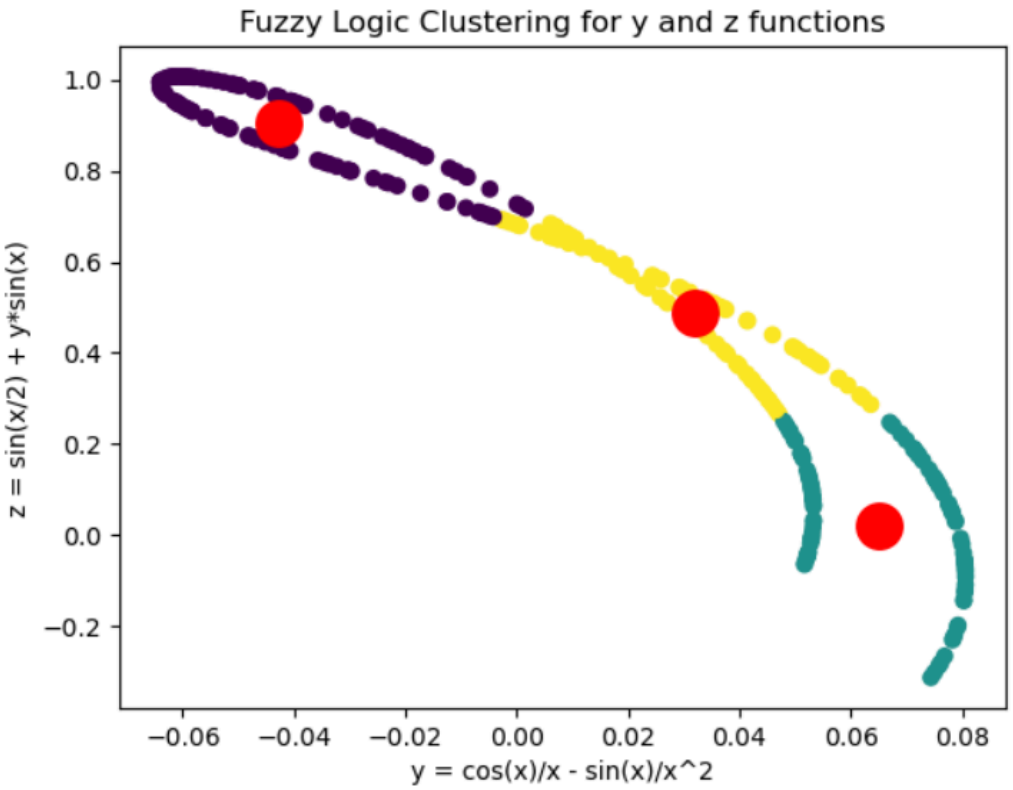
2. Solve the formulated problem using the clustering mechanism using fuzzy logic methods
using modeling software or a high-level programming language.

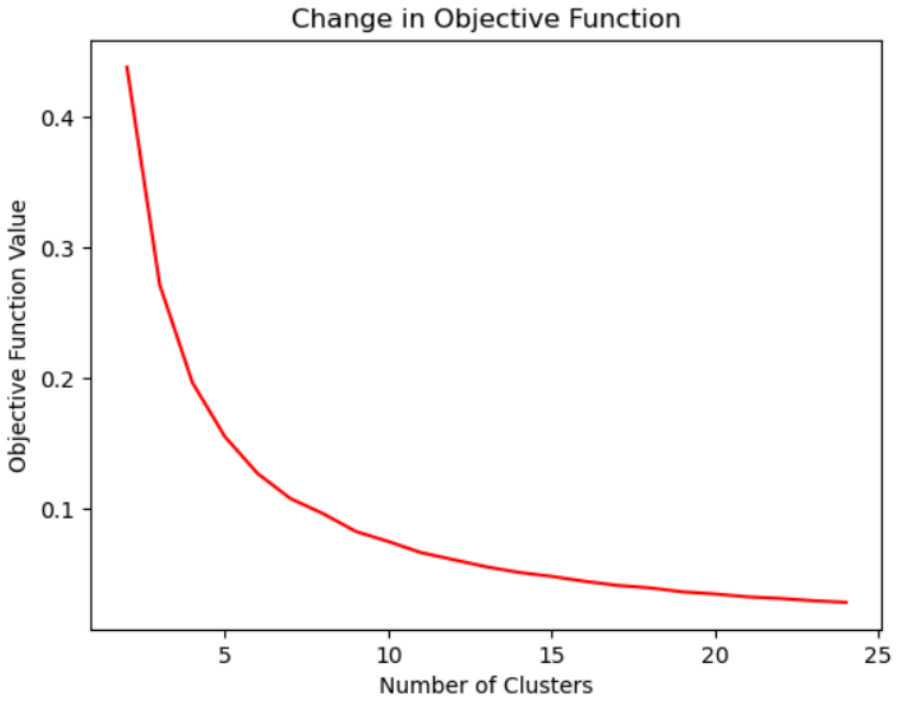
3. Find the centers of the clusters and plot the change in the values of the objective function.

4. Make a report on laboratory work.

Result

Cluster graph:





MY CODES:

```
from fcmeans import FCM
import numpy as np
from matplotlib import pyplot as plt
```

```
# Define the functions
def y_function(x):
    return np.cos(x) / x - np.sin(x) / x**2

def z_function(x, y):
    return np.sin(x/2) + y * np.sin(x)

# Generate random data points in the range [12, 19, 7]
x_values = np.random.uniform(12, 19, 300)
y_values = y_function(x_values)
z_values = z_function(x_values, y_values)

# Create feature vectors
points = np.column_stack((y_values, z_values))

# Model creation, learning, and visualization
model = FCM(n_clusters=3)
model.fit(points)

# Visualization of Clusters
plt.scatter(y_values, z_values, c=model.predict(points))
plt.scatter(model.centers[:, 0], model.centers[:, 1], s=300, c='red', marker="o", linewidths=2)
plt.title('Fuzzy Logic Clustering for y and z functions')
plt.xlabel('y = cos(x)/x - sin(x)/x^2')
plt.ylabel('z = sin(x/2) + y*sin(x)')
plt.show()

# Creating and showing the graph of changes in values of the target function
y = []
for i in range(2, 25):
    model = FCM(n_clusters=i)
    model.fit(points)
    y.append(model.partition_coefficient)

# Visualization of Objective Function Changes
plt.plot(list(range(2, 25)), y, c='red')
plt.title('Change in Objective Function')
plt.xlabel('Number of Clusters')
plt.ylabel('Objective Function Value')
plt.show()
```