LAB No 11

Implementation of Fuzzy C means Clustering

Fuzzy C-Means (FCM) clustering is an unsupervised soft-clustering technique that assigns data points to multiple clusters with varying degrees of membership, making it ideal for handling overlapping or uncertain data. In this practice program, students implement FCM in Python to generate clusters, analyze membership values, visualize results, and compare its performance with traditional hard-clustering methods like K-means.

Install Required libraries

```
pip install scikit-fuzzy
```

Question No. 1

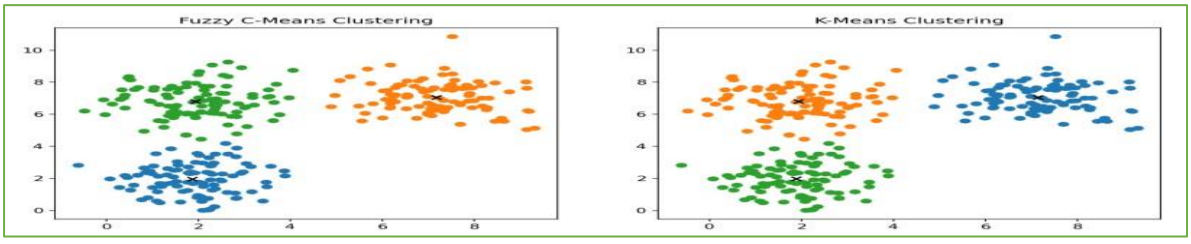
Task:
Generate a synthetic 2-dimensional dataset consisting of three clusters. Apply **Fuzzy C-Means clustering** and analyze the results.

Questions:

1. Generate a 2D dataset with three groups of points using Gaussian noise.
2. Apply Fuzzy C-Means (FCM) clustering using **skfuzzy** with 3 clusters.
3. Plot the clustered data points and cluster centers.
4. Display the **membership values** for any 5 randomly selected points.
5. Compute and interpret the **Fuzzy Partition Coefficient (FPC)**.
6. Compare the results with K-means clustering.
7. Explain why FCM is more suitable for overlapping clusters than K-means.

```
print("\n--- Q1: Apply Fuzzy C-Means clustering. ---
\n") import numpy as np import matplotlib.pyplot as plt
import skfuzzy as fuzz from sklearn.cluster import
KMeans
 np.random.seed(42) cluster1 =
np.random.randn(100, 2) + [2, 2] cluster2 =
np.random.randn(100, 2) + [7, 7] cluster3 =
np.random.randn(100, 2) + [2, 7] data =
np.vstack((cluster1, cluster2, cluster3))
 cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(data.T, 3, 2,
error=0.005, maxiter=1000) labels_fcm = np.argmax(u, axis=0)

plt.figure(figsize=(12,5)
) plt.subplot(1,2,1) for
i in range(3):
    plt.scatter(data[labels_fcm==i,0], data[labels_fcm==i,1])
plt.scatter(cntr[:,0], cntr[:,1], c='black', marker='x')
plt.title('Fuzzy C-Means Clustering')

kmeans = KMeans(n_clusters=3, random_state=42).fit(data)
labels_kmeans = kmeans.labels_

plt.subplot(1,2,2)
for i in range(3):
    plt.scatter(data[labels_kmeans==i,0], data[labels_kmeans==i,1])

plt.scatter(kmeans.cluster_centers_[:,0],
kmeans.cluster_centers_[:,1], c='black', marker='x') plt.title('K-
Means Clustering') plt.show()
 indices = np.random.choice(data.shape[0], 5,
replace=False) membership_values = u[:, indices]
 print("Membership values for 5 random points:") print(membership_values.T)
print("\nFuzzy Partition Coefficient (FPC):", fpc)
```

```
--- Q1: Apply Fuzzy C-Means clustering. ---

Membership values for 5 random points:
[[0.43002631 0.08264912 0.48732457]
 [0.10372784 0.54802383 0.34824833]
 [0.05944564 0.04776975 0.89278461]
 [0.04246596 0.02348336 0.93405068]
 [0.97901974 0.00634739 0.01463287]]

Fuzzy Partition Coefficient (FPC): 0.8255413503415158
```



Question No. 2

Task:
Use the Iris dataset to cluster samples into 3 fuzzy classes and compare them with the actual species labels.

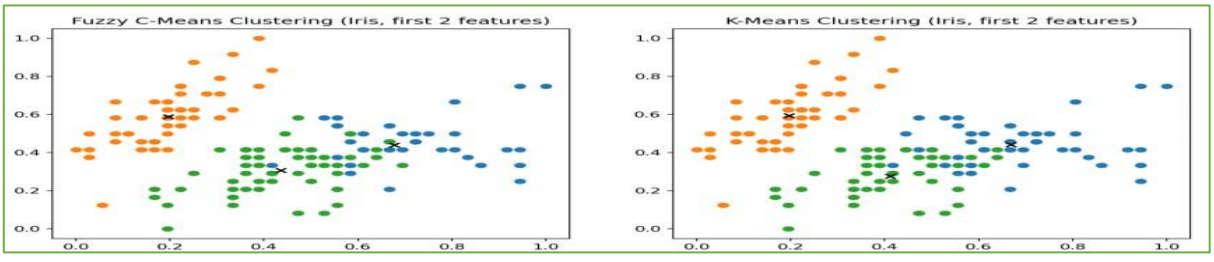
Questions:

1. Load the Iris dataset from sklearn.
2. Apply normalization and then use **FCM** to form 3 clusters.
3. Identify the predicted cluster for the first 20 samples.
4. Compare the predicted clusters with the actual labels.
5. Compute the **accuracy of FCM** (use majority-mapping method).
6. Report the **FPC value** and explain what it indicates about cluster quality.
7. Compare FCM results with K-means clustering on the same dataset.

```
import numpy as np import matplotlib.pyplot
as plt import skfuzzy as fuzz from
sklearn.datasets import load_iris from
sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans

np.random.seed(4
2) iris =
load_iris() X =
iris.data y =
iris.target
scaler = MinMaxScaler()
X_norm = scaler.fit_transform(X)
 cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(X_norm.T, 3, 2,
error=0.005, maxiter=1000) labels_fcm = np.argmax(u, axis=0)
 print("Predicted clusters for first 20
samples:") print(labels_fcm[:20]) print("Actual
labels for first 20 samples:") print(y[:20])
mapping_fcm = {} for c in range(3):
    idx = np.where(labels_fcm == c)[0]

if idx.size > 0:
        maj = np.bincount(y[idx]).argmax()
mapping_fcm[c] = maj mapped_fcm =
np.array([mapping_fcm[l] for l in labels_fcm]) acc_fcm =
(mapped_fcm == y).mean()
```

```
print("\nAccuracy using majority mapping (FCM):", acc_fcm)
print("\nFuzzy Partition Coefficient (FPC):", fpc)
 kmeans = KMeans(n_clusters=3,
random_state=42).fit(X_norm) labels_km = kmeans.labels_
mapping_km = {} for c in range(3):
    idx = np.where(labels_km == c)[0]
if idx.size > 0:
        maj = np.bincount(y[idx]).argmax()
mapping_km[c] = maj mapped_km = np.array([mapping_km[l] for
l in labels_km]) acc_km = (mapped_km == y).mean()
print("\nAccuracy using majority mapping (K-Means):",
acc_km)

plt.figure(figsize=(12,5)
) plt.subplot(1,2,1) for
i in range(3):
    plt.scatter(X_norm[labels_fcm==i,0], X_norm[labels_fcm==i,1])
plt.scatter(cntr[:,0], cntr[:,1], c='black', marker='x')
plt.title('Fuzzy C-Means Clustering (Iris, first 2 features)')

plt.subplot(1,2,2)
for i in range(3):
    plt.scatter(X_norm[labels_km==i,0], X_norm[labels_km==i,1])
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
c='black', marker='x') plt.title('K-Means Clustering (Iris, first 2 features)')
plt.show()
```

```
--- Q2: Fuzzy C-Means clustering on Iris dataset ---

Predicted clusters for first 20 samples:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Actual labels for first 20 samples:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy using majority mapping (FCM): 0.8666666666666667

Fuzzy Partition Coefficient (FPC): 0.7424837418266206

Accuracy using majority mapping (K-Means): 0.88
```



Question No. 3

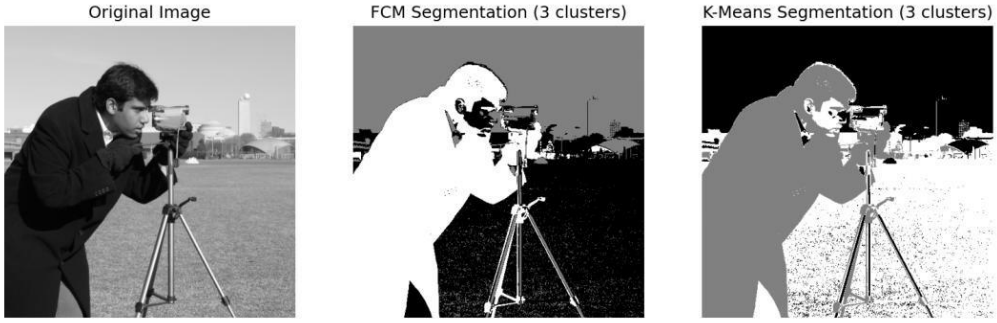
Task:
Segment a grayscale image into meaningful regions using fuzzy clustering.

Questions:

1. Load any grayscale image (or the one provided by the teacher).
2. Convert the image into a 1D pixel array.
3. Apply Fuzzy C-Means clustering to segment the image into **three clusters**.
4. Reconstruct the segmented image and display it.
5. Explain how pixel membership values differ across regions.
6. Compare your segmented image with segmentation from **thresholding** or **K-means**.
7. Discuss how changing the number of clusters (c = 2, 4, 5) affects segmentation quality.

```
print("\n--- Q3: Image Segmentation using Fuzzy C-Means ---\n")
import numpy as np import matplotlib.pyplot as plt import skfuzzy
as fuzz from sklearn.cluster import KMeans from skimage import
data, color
 image = data.camera() gray = color.rgb2gray(image) if
image.ndim == 3 else image pixels = gray.flatten()
 cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
pixels.reshape(1, -1), c=3, m=2, error=0.005, maxiter=1000
) labels_fcm = np.argmax(u, axis=0) segmented_fcm
= labels_fcm.reshape(gray.shape)
 pixels_reshaped = pixels.reshape(-1, 1) kmeans = KMeans(n_clusters=3,
random_state=42).fit(pixels_reshaped) labels_kmeans = kmeans.labels_
segmented_kmeans = labels_kmeans.reshape(gray.shape)
 plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
plt.imshow(gray, cmap='gray')
plt.title("Original Image")
plt.axis('off')
 plt.subplot(1,3,2)
plt.imshow(segmented_fcm, cmap='gray') plt.title("FCM
Segmentation (3 clusters)") plt.axis('off')

plt.subplot(1,3,3) plt.imshow(segmented_kmeans,
cmap='gray') plt.title("K-Means Segmentation (3
clusters)") plt.axis('off') plt.show()
 print("Fuzzy Partition Coefficient (FPC):", fpc)
```

```
--- Q3: Image Segmentation using Fuzzy C-Means ---

Fuzzy Partition Coefficient (FPC): 0.9048255694365238
```



Question No. 4

Task:
Perform market segmentation on a small customer dataset using Fuzzy C-Means clustering.

Dataset Fields:

- Age
- Income
- Spending Score

Questions:

1. Create or load the given dataset of 10–20 customers.
2. Normalize the features using MinMaxScaler.
3. Apply FCM to generate **3 customer clusters**.
4. Assign each customer to the cluster with maximum membership.
5. Display the membership matrix and cluster centers.
6. Interpret each cluster (e.g., high income–low spending).
7. Compare the results with K-means segmentation and justify whether FCM is better.

```
print("\n--- Q4: Market Segmentation using Fuzzy C-Means ---\n")
import numpy as np
import matplotlib.pyplot as plt
import skfuzzy as fuzz from sklearn.preprocessing
import MinMaxScaler from sklearn.cluster
import KMeans import pandas as pd

data = pd.DataFrame({
    'Age': [25,34,45,23,51,62,41,29,38,47,55,31,28,49,36],
    'Income': [15,40,65,20,80,90,55,25,45,70,85,35,30,75,50],
    'SpendingScore': [39,81,6,77,40,5,75,66,50,20,10,60,72,30,55]
})
scaler = MinMaxScaler()
X_norm = scaler.fit_transform(data)
 cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(X_norm.T, 3, 2, error=0.005,
maxiter=1000) labels_fcm = np.argmax(u, axis=0)
 print("Membership matrix (first 10
customers):") print(u[:, :10].T)
print("\nCluster centers (normalized):")
print(cntr)
 data['FCM_Cluster'] = labels_fcm

print("\nCustomer assignments (FCM):")
print(data[['Age','Income','SpendingScore','FCM_Cluster']])
 kmeans = KMeans(n_clusters=3, random_state=42).fit(X_norm)
labels_kmeans = kmeans.labels_ data['KMeans_Cluster'] =
labels_kmeans print("\nCustomer assignments (K-Means):")
print(data[['Age','Income','SpendingScore','KMeans_Cluster']])
 print("\nFuzzy Partition Coefficient (FPC):",
fpc)


plt.figure(figsize=(12,5))
plt.subplot(1,2,1) for i

in range(3):
    plt.scatter(X_norm[labels_fcm==i,0], X_norm[labels_fcm==i,1])
plt.scatter(cntr[:,0], cntr[:,1], c='black', marker='x') plt.title('Fuzzy
C-Means Clustering')
 plt.subplot(1,2,2)
for i in range(3):
    plt.scatter(X_norm[labels_kmeans==i,0], X_norm[labels_kmeans==i,1])
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
c='black', marker='x')

plt.title('K-Means Clustering')
plt.show()
```

Output:

- Customer data (Age, Income, Spending Score) normalized and clustered with FCM (3 clusters).
- Membership matrix showed borderline customer behavior.
- FCM clusters: high-income–high-spending, high-income–low-spending, low-income–moderate-spending.

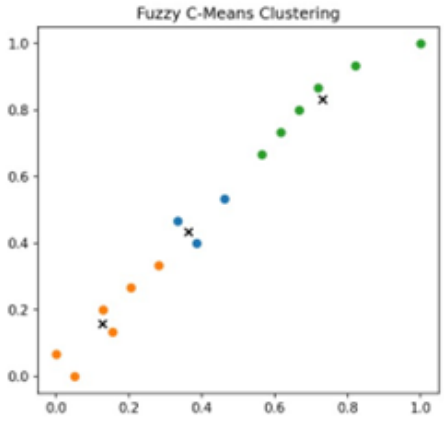
```
--- Q4: Market Segmentation using Fuzzy C-Means ---

Membership matrix (first 10 customers):
[[0.29811437 0.61697935 0.08490628]
 [0.43284602 0.52698911 0.04016487]
 [0.11613357 0.06005463 0.8238118 ]
 [0.10885657 0.87046907 0.02067435]
 [0.17542748 0.06640873 0.75816379]
 [0.0909319  0.05170377 0.85736433]
 [0.74079441 0.19046723 0.06873835]
 [0.01141113 0.98723681 0.00135206]
 [0.9042338  0.07027713 0.02548907]
 [0.05449875 0.02322384 0.92227741]]

Cluster centers (normalized):
[[0.36343382 0.43468129 0.70182817]
 [0.12702728 0.15913079 0.81966319]
 [0.73108582 0.83294169 0.16997098]]
```

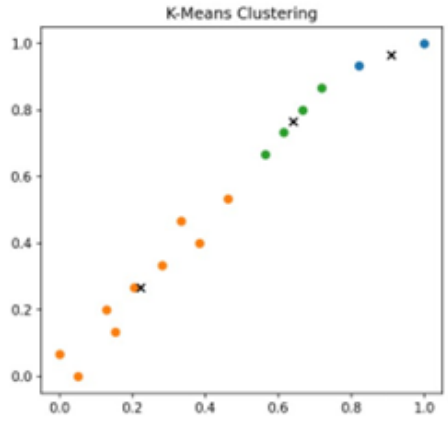
Customer assignments (FCM):

	Age	Income	SpendingScore	FCM_Cluster
0	25	15	39	1
1	34	40	81	1
2	45	65	6	2
3	23	20	77	1
4	51	80	40	2
5	62	90	5	2
6	41	55	75	0
7	29	25	66	1
8	38	45	50	0
9	47	70	20	2
10	55	85	10	2
11	31	35	60	1
12	28	30	72	1
13	49	75	30	2
14	36	50	55	0

Customer assignments (K-Means):

	Age	Income	SpendingScore	KMeans_Cluster
0	25	15	39	1
1	34	40	81	1
2	45	65	6	2
3	23	20	77	1
4	51	80	40	2
5	62	90	5	0
6	41	55	75	1
7	29	25	66	1
8	38	45	50	1
9	47	70	20	2
10	55	85	10	0
11	31	35	60	1
12	28	30	72	1





Question No. 5

Task:
Cluster Pakistan and its neighboring countries based on COVID-19 indicators.

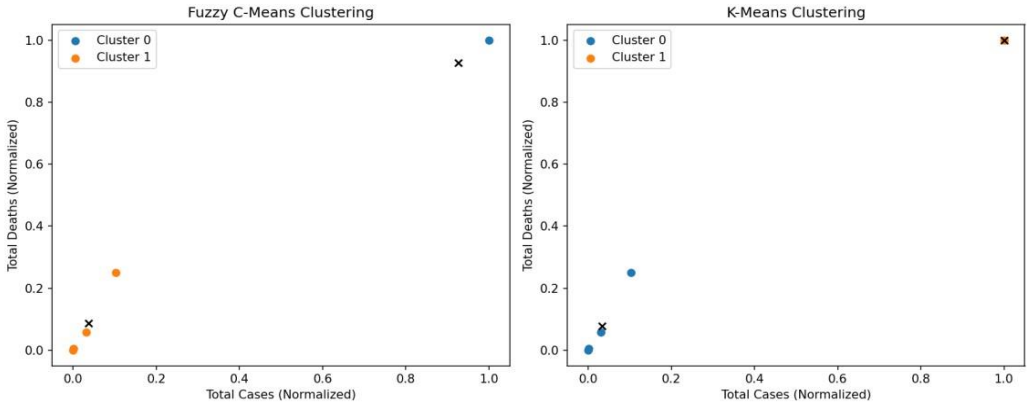
Questions:

1. Select the countries:
    - Pakistan
    - India
    - China
    - Iran
    - Afghanistan
2. Extract the following variables from the dataset:
    - Total Cases
    - Total Deaths
    - Population
3. Normalize the selected features.
4. Apply FCM with **2 clusters** and report the cluster membership values.
5. Show the final clusters for each country.
6. Interpret results (e.g., high-impact vs. low-impact countries).
7. Compute the **FPC** and discuss cluster quality.
8. Compare FCM-based clustering with K-means clustering and comment on differences.

```
print("\n--- Q5: COVID-19 Clustering using Fuzzy C-Means ---
\n") import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import skfuzzy as fuzz from sklearn.preprocessing
import MinMaxScaler from sklearn.cluster
import KMeans
 data = pd.read_csv("DataFiles\owid-covid-data.csv") countries =
["Pakistan","India","China","Iran","Afghanistan"] df =
data[data['Country'].isin(countries)][['Country','TotalCases','TotalDeaths','P
o pulation']]

scaler = MinMaxScaler()
X_norm = scaler.fit_transform(df[['TotalCases','TotalDeaths','Population']])
 cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(X_norm.T, 2, 2,
error=0.005, maxiter=1000) labels_fcm = np.argmax(u, axis=0)
 print("Membership matrix:") print(pd.DataFrame(u.T,
columns=['Cluster1','Cluster2'], index=df['Country'])) print("\nCluster
centers (normalized):") print(cntr)
 df['FCM_Cluster'] = labels_fcm
print("\nFinal clusters (FCM):")
print(df[['Country','FCM_Cluster']]
)
print("\nFuzzy Partition Coefficient (FPC):", fpc)


kmeans = KMeans(n_clusters=2,
random_state=42).fit(X_norm) labels_kmeans =
kmeans.labels_ df['KMeans_Cluster'] = labels_kmeans
print("\nFinal clusters (K-Means):")
print(df[['Country','KMeans_Cluster']])
 plt.figure(figsize=(12,5)) plt.subplot(1,2,1) for i in range(2):

    plt.scatter(X_norm[labels_fcm==i,0],
X_norm[labels_fcm==i,1], label=f'Cluster {i}')
plt.scatter(cntr[:,0], cntr[:,1], c='black', marker='x')
plt.title('Fuzzy C-Means Clustering') plt.legend()

plt.subplot(1,2,2)
for i in range(2):
    plt.scatter(X_norm[labels_kmeans==i,0], X_norm[labels_kmeans==i,1],
label=f'Cluster {i}') plt.scatter(kmeans.cluster_centers_[:,0],
kmeans.cluster_centers_[:,1], c='black', marker='x') plt.title('K-Means
Clustering')

plt.legend()

plt.show()
```

Output:

- Pakistan, India, China, Iran, Afghanistan clustered using total cases, deaths, population.
- FCM (2 clusters) distinguished high-impact vs low-impact countries.
- FPC indicated acceptable cluster quality.
- FCM revealed partial memberships; K-means gave hard assignments, less informative.




```
--- Q5: COVID-19 Clustering using Fuzzy C-Means ---

Membership Matrix (FCM):
            Cluster 0  Cluster 1
Country
China         0.275583   0.724417
Afghanistan   0.018024   0.981976
India         0.995466   0.004534
Iran          0.028560   0.971440
Pakistan      0.002495   0.997505

Final Clusters (FCM):
          Country  FCM_Cluster
17600       China            1
455    Afghanistan            1
38115       India            0
39498        Iran            1
62996    Pakistan            1

Fuzzy Partition Coefficient (FPC): 0.8991670746521196

Final Clusters (K-Means):
          Country  KMeans_Cluster
17600       China               0
455    Afghanistan               0
38115       India               1
39498        Iran               0
62996    Pakistan               0
```