```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#==Install Pacage===#
#!pip install scikit-fuzzy
import skfuzzy as fuzz

from mpl_toolkits.mplot3d import Axes3D
from sklearn.datasets import make_moons
from sklearn.datasets import make_blobs
#Generate moons dataset
#X, $y=$ make_moons(n_samples 300, noise $=0.1,$
#random state=42)
#Generate synthetic data
X, y_true = make_blobs(n_samples =300, centers =4,
cluster_std $=0.60$, random_state=0)

D=pd.DataFrame(X)

D.columns=['x', 'Y']
plt.figure(figsize=(5,5))
plt.plot(D['x'], D['Y'], 'o', color='b')
plt.xlabel("Value of X")
plt.ylabel("Value of Y")
plt.title('Plot for X vs Y')
plt.savefig('D:/Fuzzy C-Means_Scatter plot for Data.jpg',dpi $=1000$)
plt.show()
```

```
np.random.seed(104729)
FPC=[]
K=[]
for k in range(2,11):
    #Perform fuzzy c-means clustering
    cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans (
    D.T, k ,m=2, error =0.0005, maxiter =1000, init=None)
    FPC.append(fpc)
    K.append(k)

k=pd.DataFrame(FPC).idxmax()+2
k
k[0]

plt.figure(figsize=(6, 5))
plt.plot(range(2, 11), np.round(FPC, 3), ls='-',
linewidth =2, color='red', marker='o',
markerfacecolor='blue', markersize=7)
plt.axvline(x=k[0], color='b', ls='--', linewidth =1.5)
plt.title('Optimal number of cluster for Fuzzy C-means')
plt.xlabel('Number of clusters for Fuzzy C-means')
plt.ylabel('Fuzzy partition coefficient (FPC)')
plt.savefig('D:/Fuzzy C-Means_Optimal number of Cluster.jpg',dpi=700)
plt.show()

# Perform fuzzy c-means clustering
np.random.seed(199999)
cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans (
```

```
D.T, k[0] ,m=2, error =0.0005, maxiter =1000, init=None)

#Print results
print("Cluster centers:\n", cntr)
print("\nWithin-cluster sum of squares: \n", jm)
print("\nFinal partition matrix:\n", np.round(u,3))
print("\nNumber of iterations: \n", p)
print("\nThe fuzzy partition coefficient (FPC):\n", fpc)

# Predict cluster membership for each data point
cluster_membership = np.argmax(u, axis=0)
cluster_membership
#Print the cluster membership for each data point
print('Cluster Membership:', cluster_membership)

# Create a List of colors for each cluster
colors = ['r', 'g', 'b', 'm','c', 'y', 'k']
#Create a figure with specified size
plt.figure(figsize=(5,5))
#Plot the data points with their respective cluster colors
for i in range(len(D)):
    plt.plot(D.iloc[i, 0], D.iloc[i, 1], 'o',
    color=colors[cluster_membership[i]])
#Set the x-axis and y-axis Labels
plt.xlabel("Value of X")
plt.ylabel("Value of Y")
plt.title('Plot for X vs Y')
plt.savefig('D:/Fuzzy C-Means_Scatter plot for Solution.jpg', dpi=1000)
```

```
plt.show()
```