Name	ID
Nabil Sherif Nabil Ibrahim	**20200583**
Lilian Stephanos Younan	**20200404**
Aly Walid	**20200336**
Nathalie Monged	**20200586**

Phase 1: Exploratory Data Analysis and PCA Implementation

Introduction:

In this phase, the goal was to explore an unlabelled dataset and apply Principal Component Analysis (PCA) from scratch. The primary objectives included investigating the impact of different Q Matrices on the dataset, aiming to select the one that minimally changes the original feature vectors while effectively reducing dimensionality.

Dataset Exploration:

The Iris dataset is a classic dataset in machine learning and is often used for exploratory data analysis and classification tasks. It consists of samples from three species of Iris flowers (Iris setosa, Iris virginica, and Iris versicolor). Despite being a labelled dataset, for the purposes of this project, we treated it as an unlabelled dataset to demonstrate the PCA implementation.

- **Number of Features:**

The Iris dataset contains four features, representing various measurements related to the morphology of the iris flowers:

1. Sepal Length (in cm)
2. Sepal Width (in cm)
3. Petal Length (in cm)
4. Petal Width (in cm)

- **Data Types:**

All features in the Iris dataset are continuous numerical values, represented as floating-point numbers.

PCA Implementation steps:

- **Centring the Data:**

In the initial step, we calculate the mean of the data and subtract it from the original dataset. This process centers the data around the mean, ensuring that the subsequent analysis is based on a centered distribution.

- **Covariance Matrix Calculation:**

The covariance matrix is computed using the centered data. This matrix captures the relationships between different features in the dataset, providing a basis for understanding the variance and covariances within the data.

- **Eigenvalue and Eigenvector Computation:**

The eigenvalues and eigenvectors of the covariance matrix are essential components of PCA. We compute these values by constructing the characteristic polynomial matrix and finding its roots. The eigenvectors are then obtained by solving the characteristic equation.

- **Sorting and Selecting Top Eigenvectors:**

To identify the principal components, we sort the eigenvalues and corresponding eigenvectors in descending order. The top eigenvectors, determined by the specified number of components form the projection matrix. This matrix is crucial for projecting the data onto a reduced-dimensional space.

- **Compression and Decompression:**

The compression step involves projecting the centered data onto the principal components, resulting in a compressed representation. Conversely, decompression reconstructs the original data from the compressed representation, utilizing the inverse transformation of the projection matrix.

Trial Results:

we have 4 features, so we tested 4 number of components (Different Q matrix): [1, 2, 3, 4]

And we compared the results using mse between the decompressed data and the original data:

Results:

Number of Components	Mean Squared Error
1	0.0856043
2	0.0253411
3	0.00591905
4	3.09069e-31

Discussion of Findings:

- **Trade-offs between the number of components and the quality of representation:**

    - As the number of components increases, the Mean Squared Error (MSE) decreases. This decrease in MSE suggests a better reconstruction of the original data from the compressed representation.

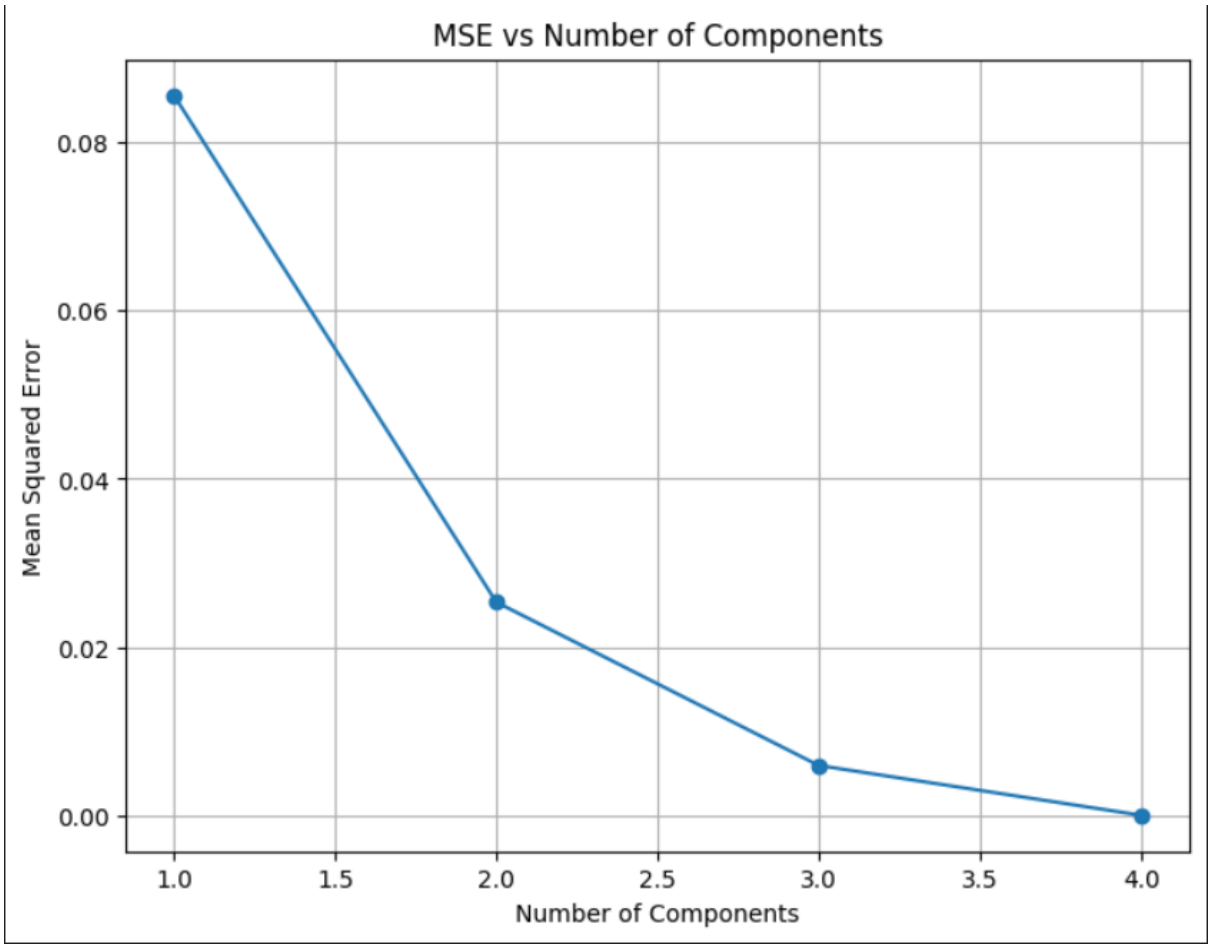
    - Lower dimensions (fewer components) reduce computational complexity but may lead to a loss of important information, as seen from the increasing MSE.

- **Lower dimensions might reduce computational complexity but could lose important information:**

    - With fewer components (lower dimensions), the MSE is higher, indicating that the compressed representation doesn't fully capture the complexity of the original data. This loss in information might impact downstream tasks that rely on the detailed information present in the data.

Using Elbow Method:

Plot the MSE values against the number of components to identify the point where the rate of MSE improvement starts to diminish, often referred to as the "elbow point." This helps in determining an optimal number of components.



Recommendation based on the MSE comparison and the Elbow Method graph:

Number of components that provides a good balance (Best Q matrix):

- If reducing complexity is important and the difference in MSE is not significant, we might choose the model with 2 components. On the other hand, if accuracy is a top priority and the computational cost is acceptable, we might choose the model with 3 components.

- So, we might choose 2 or 3.

Phase 2: PCA Impact on Unsupervised Learning

Introduction:

Building upon the groundwork laid in Phase 1, our focus in this phase is to seamlessly integrate entropy-based fuzzy clustering, a powerful unsupervised learning technique, with Principal Component Analysis (PCA).

Having previously explored the impact of various Q Matrices on the unlabelled dataset, our objective now shifts towards applying entropy-based fuzzy clustering both before and after implementing PCA with the optimal Q Matrix.

Entropy Based Fuzzy Implementation steps:

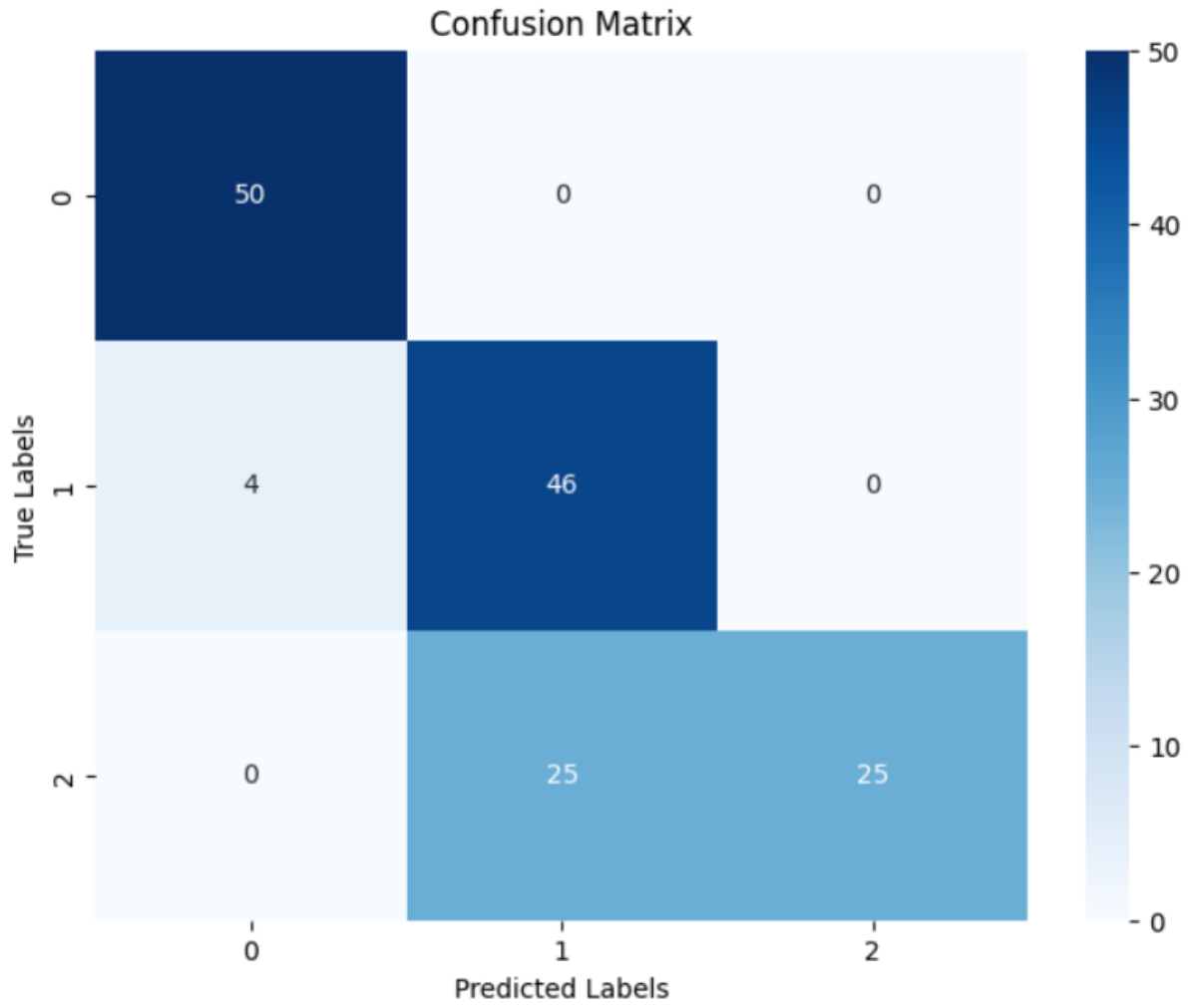
1. Compute the similarity matrix.
2. Calculate entropies for each data point.
3. Identify the data point with the minimum entropy as the seed for a new cluster.
4. Expand the cluster by adding data points with similarities above the beta threshold.
5. Remove the cluster and outliers if the cluster size is below the gamma threshold.
6. Repeat the process until no non-zero entropies remain.

Trial Results:

To compare the results of the three methods involving Entropy Based Fuzzy Clustering on the Iris dataset, let's interpret and analyze the outcomes:

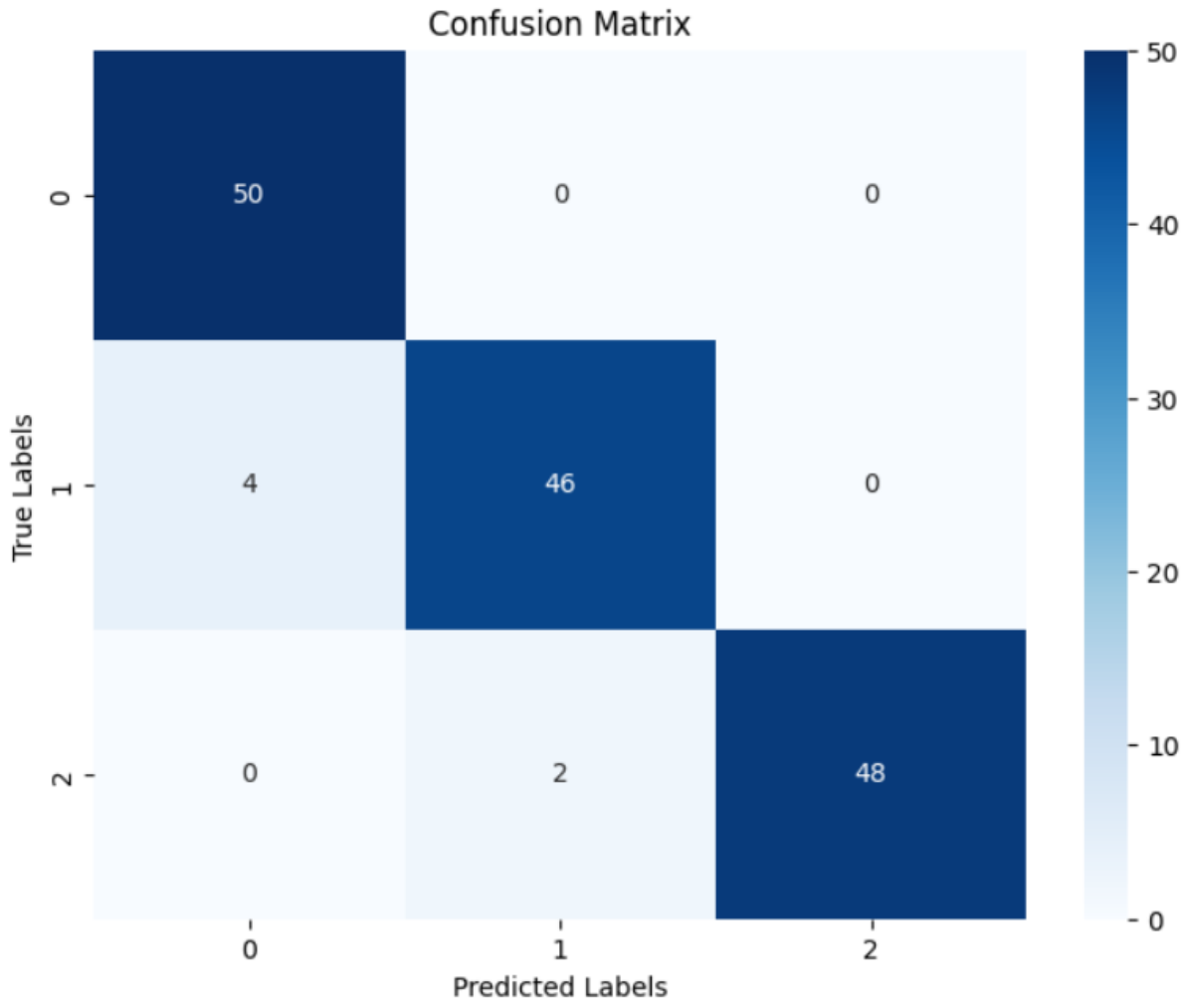
Method 1: Entropy Based Fuzzy Clustering (without PCA)

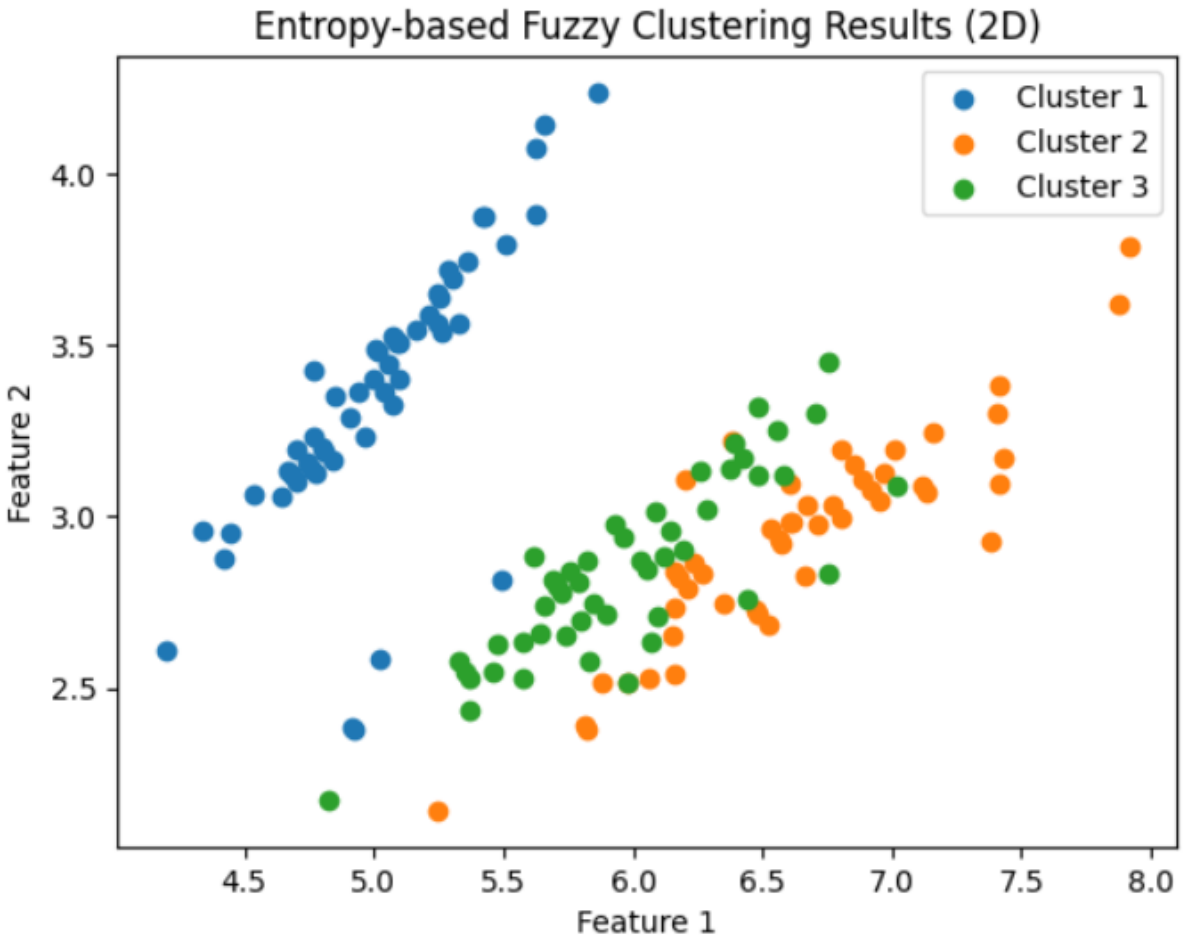
- **Class 0**: Correctly classified all 50 samples.
- **Class 1**: 46 out of 50 correct, resulting in 4 misclassified samples.
- **Class 2**: 25 out of 50 correct, resulting in 25 misclassified samples.
- **Parameters**: beta = 0.5, gamma = 0
- **Accuracy metrics**:
    - Precision: 0.8066666666666666
    - Recall: 0.8579377499565294
    - F1 Score: 0.7961785689058415



Method 2: Entropy Based Fuzzy Clustering (using PCA with 2 components)

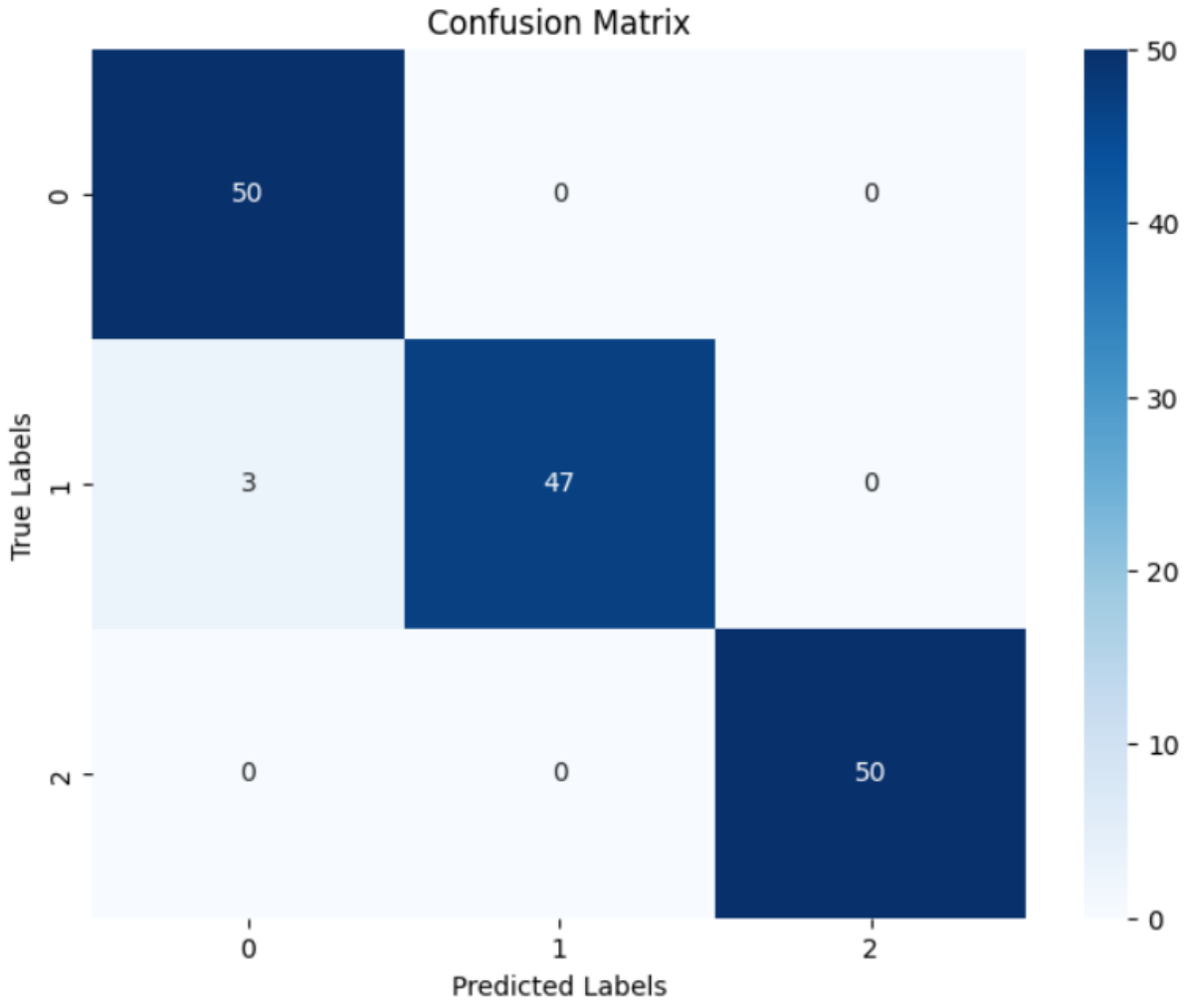
- **Class 0**: Correctly classified all 50 samples.
- **Class 1**: 46 out of 50 correct, resulting in 4 misclassified samples.
- **Class 2**: 48 out of 50 correct, resulting in 2 misclassified samples.
- **Parameters**: beta = 0.5, gamma = 0.2
- **Accuracy metrics**:
  - Precision: 0.96
  - Recall: 0.9614197530864198
  - F1 Score: 0.9599686028257457

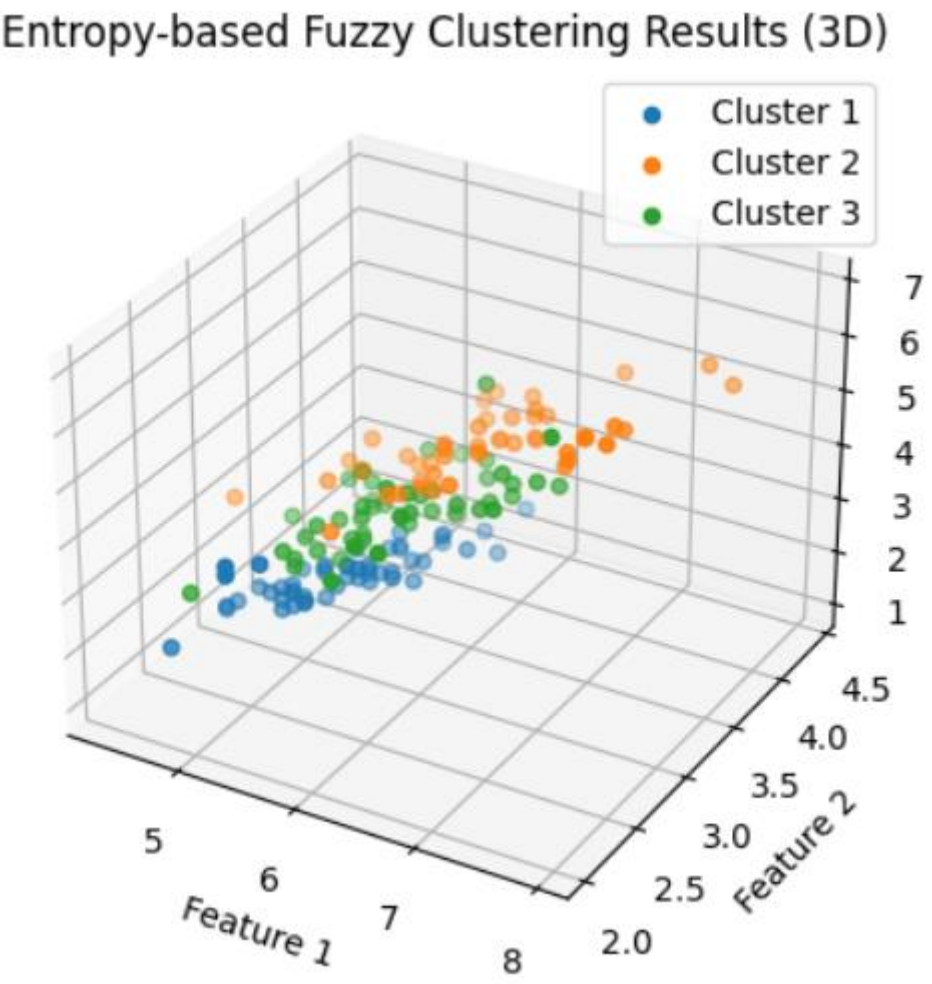



Entropy-based Fuzzy Clustering Results (2D)
Cluster 1
Cluster 2
Cluster 3
Feature 2
Feature 1
4.0
3.5
3.0
2.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Method 3: Entropy Based Fuzzy Clustering (using PCA with 3 components)

- **Class 0**: Correctly classified all 50 samples.
- **Class 1**: 47 out of 50 correct, resulting in 3 misclassified samples.
- **Class 2**: Correctly classified all 50 samples.
- **Parameters**: beta = 0.5, gamma = 0.2
- **Accuracy metrics**:
  - Precision: 0.98
  - Recall: 0.9811320754716982
  - F1 Score: 0.9799819837854069





Observations and Analysis:

- **Method 1**: Without using PCA, this method resulted in higher misclassification, especially for Class 2.

- **Method 2**: Utilizing PCA with 2 components improved the results, reducing misclassifications for Class 2.

- **Method 3**: Employing PCA with 3 components showed slight improvements in performance.

Key findings:

1. Class 0 was accurately classified by all methods.

2. Without PCA, classification accuracy was lower than with PCA especially for Class 2.

3. PCA, especially with 3 components, improved overall accuracy.

4. Method 1 is eliminated, and now we compare PCA effectiveness between methods 2 and 3.

Comparison of Entropy Based Fuzzy Clustering with PCA (2 components) and PCA (3 components):

Entropy Based Fuzzy Clustering with PCA (2 components):

- Achieved improved results compared to clustering without PCA.
- Shows reasonably close performance to PCA (3 components) but with simpler dimensionality (lower number of components).

Entropy Based Fuzzy Clustering with PCA (3 components):

- Showed slight improvements over PCA (2 components) in the classification of one of the classes but didn't have a significant impact overall.
- Offers a slightly more detailed representation of the data due to the inclusion of an additional component.

Discussion of Findings:

- **Entropy Based Fuzzy Clustering without PCA**: Considered the least effective method among the three approaches, resulting in higher misclassifications across classes.

- **Entropy Based Fuzzy Clustering with PCA (2 components):** Demonstrated notably improved performance compared to clustering without PCA, **providing a good balance between simplicity (lower dimensionality) and classification accuracy.**

- **Entropy Based Fuzzy Clustering with PCA (3 components):** Displayed slight enhancements over PCA (2 components) in certain class classifications **but didn't substantially outperform the 2-component PCA approach.**

Decision:

Given the marginal difference in performance between PCA (2 components) and PCA (3 components) in terms of classification accuracy and the simplicity offered by 2 components, choosing PCA with 2 components for Entropy Based Fuzzy Clustering appears optimal. It strikes a balance between reducing dimensionality and achieving a reasonably good classification performance.

The decision may vary based on specific application needs, computational complexity, and the significance of the slight performance improvement provided by 3 components. However, considering a trade-off between complexity and performance, **PCA with 2 components seems to be a practical and effective choice for the clustering analysis.**