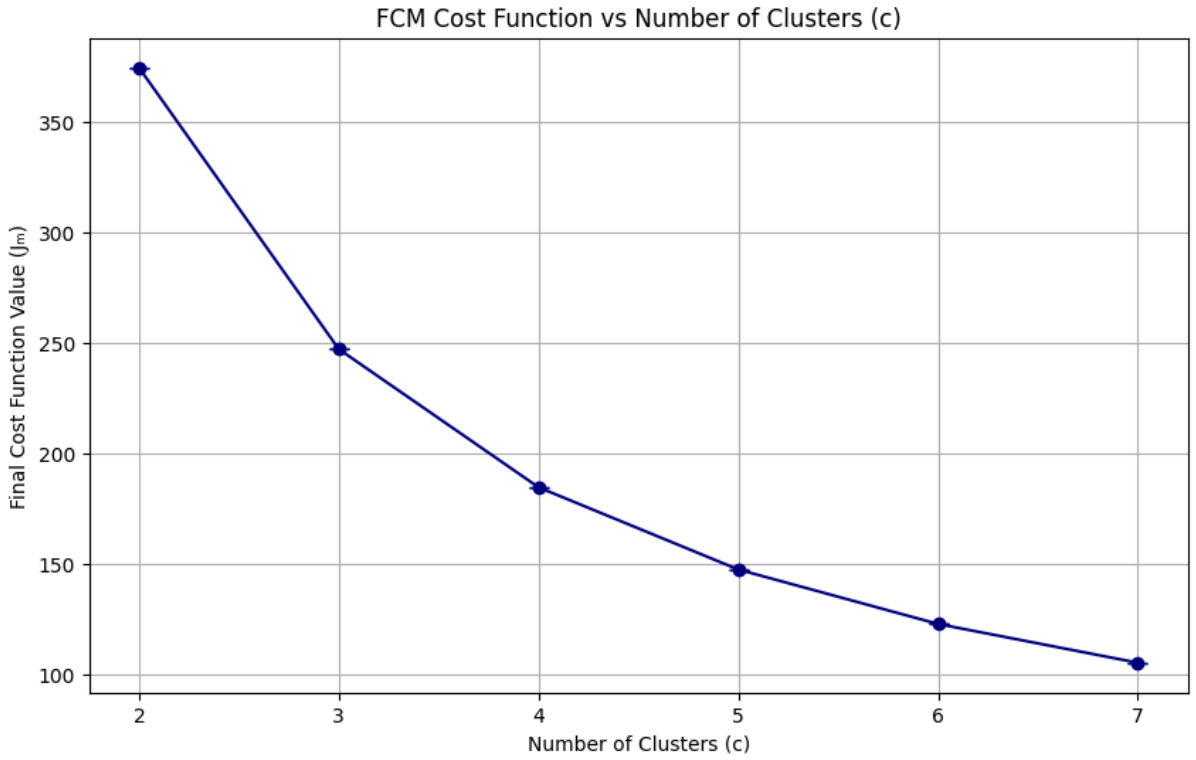
**FUZZY CLUSTERING**




From the FCM clustering criterion analysis, we observe a clear “elbow” between **c = 2** and **c = 3,** where the cost function (J□) experiences its largest drop. A smaller but still noticeable decrease occurs between **c = 3** and **c = 4**. However, after **c = 4**, the reductions in cost become increasingly marginal, indicating diminishing returns in terms of clustering quality.

In fuzzy c-means, a decrease in the cost function suggests that the algorithm is finding tighter and more well-defined clusters, with less fuzzy overlap between them. This is a desirable outcome, as it typically reflects more coherent groupings within the data. However, if the cost keeps decreasing only slightly as we increase the number of clusters, it often signals that the algorithm is fitting noise or over-segmenting the data.

Although the elbow method should be used cautiously and not as a standalone criterion, this behavior suggests that **c = 2, 3, and 4** are the most informative cluster counts. Beyond **c = 4**, the cost function plateaus, reinforcing the idea that increasing the number of clusters further yields minimal gain in clustering performance.

| Clusters | **Xie-Beni** | **Silhouette** | **ARI** |
|---|---|---|---|
| **2** | 1.3664 | 0.2071 | **0.0604** |
| **3** | **0.6995** | **0.2501** | 0.0595 |
| **4** | 0.9068 | 0.1548 | 0.0536 |
| **5** | 28.3104 | 0.1032 | 0.0530 |
| **6** | 20.3117 | 0.0960 | 0.0518 |

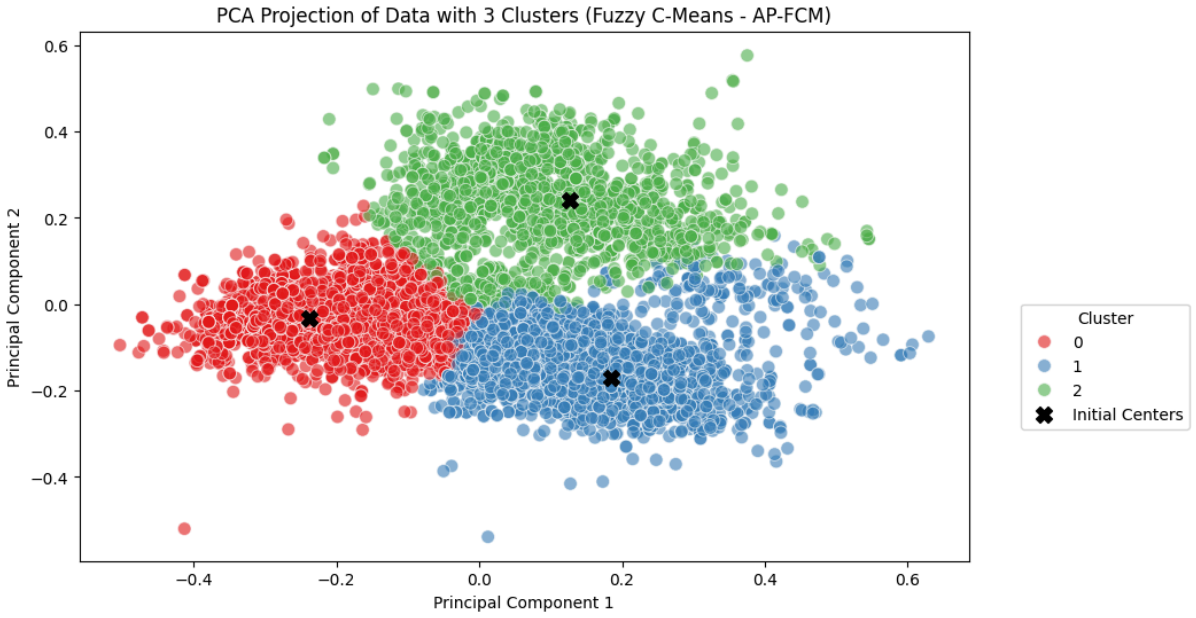
The validation indices confirm the trend observed in the cost function analysis, diminishing returns after c = 4. Notably, the only cluster counts that yield meaningful values across all three validation metrics (Xie-Beni Index, Silhouette Score, and ARI) are **c = 2, 3,** and **4**. Beyond **c = 4**, the Xie-Beni Index sharply increases (jumping to values as high as 20 or more), and the scores become unstable, indicating potential overfitting or incoherent clusters.

Among the cluster counts analyzed, **c = 3** emerges as the most balanced and optimal choice, achieving the best Xie-Beni score and Silhouette score, while also being a close second in ARI. Given this consistent performance, we select **c = 3** as the optimal number of clusters for subsequent analysis, particularly for the Anomalous Pattern (AP) detection task.

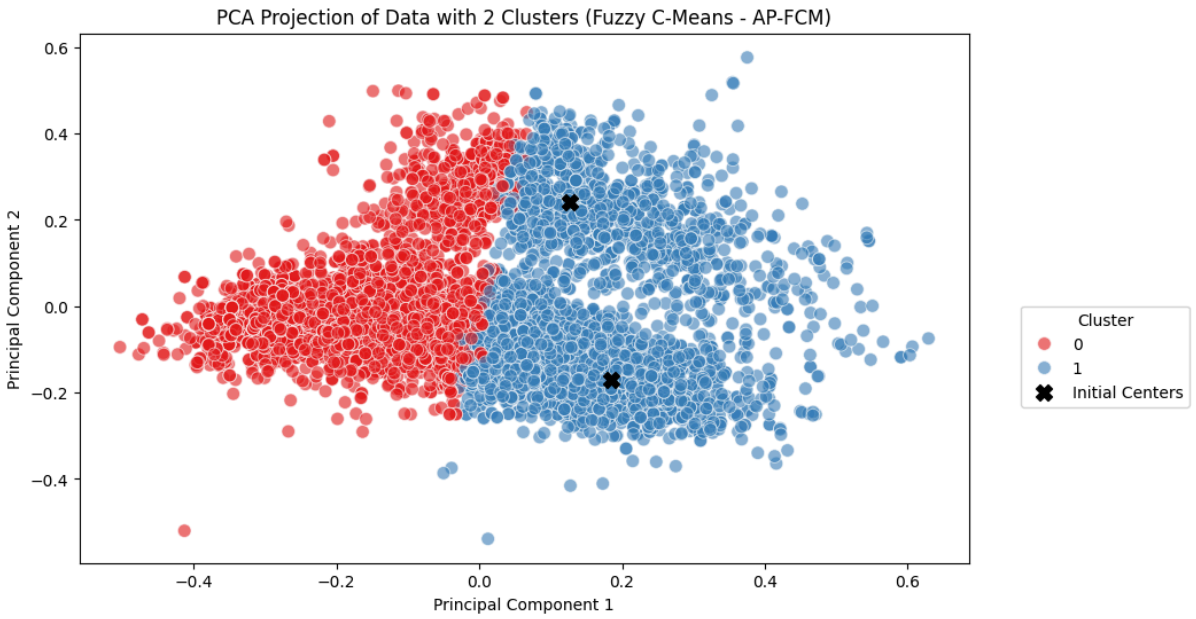
Additionally, all tests were conducted across multiple seeds: **1, 42, 70, 100, and 200**. Interestingly, the results across all validation indices remained identical for **c = 2 to c = 4**, and only began to diverge for **c ≥ 5**. This stability across seeds aligns with an earlier observation made by my teacher, who described the dataset as **"well-behaved"**, meaning it yields consistent clustering results regardless of initialization. As a result, we do not expect significant variability when running the Anomalous Pattern Clustering Algorithm.

# Anomalous Clustering

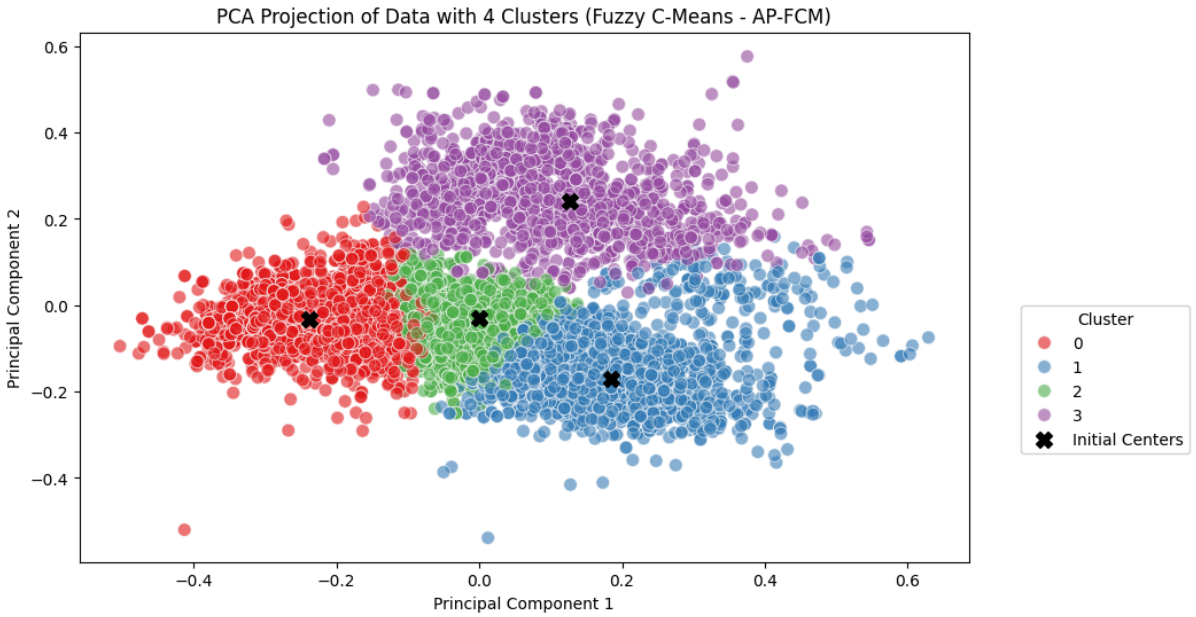
As predicted, after implementing the Anomalous Clustering Algorithm as the Initialization for our Fuzzy C-Means (FCM) clustering, we observed no significant differences in the resulting clusters. The chosen threshold was **25**, which, given our dataset size of **6,497** samples, ensures that no resulting cluster will have fewer than 25 data points. This represents a minimum cluster size of approximately 0.38% of the total data, which is acceptable for this dataset, especially considering the natural density and distribution observed in earlier exploratory data analysis.




From the PCA visualization with 3 clusters, our choice of c, backed up by the previous analysis of our validation indices and cost function, it becomes clear that the initial centroids discovered via Anomalous Pattern Clustering are already closely aligned with the natural centers of the three clusters. This further reinforces the notion that the dataset has an inherent cluster tendency, a desirable trait in clustering applications. In well-behaved datasets, initial centroids, even those selected through advanced strategies like Anomalous Clustering, do not significantly outperform random seeds.




However, an interesting observation arises with c = 2, where one of the initial centroids is noticeably distant from the eventual center of the red cluster (cluster 0). This suggests that forcing the data into only two clusters results in a more complex optimization path for centroid movement. This longer path to convergence indicates that **c = 2** might not sufficiently capture the underlying structure, leading to a lower Silhouette Score and high Xie-Beni Index score (compared to c=3 and 4), as previously observed.

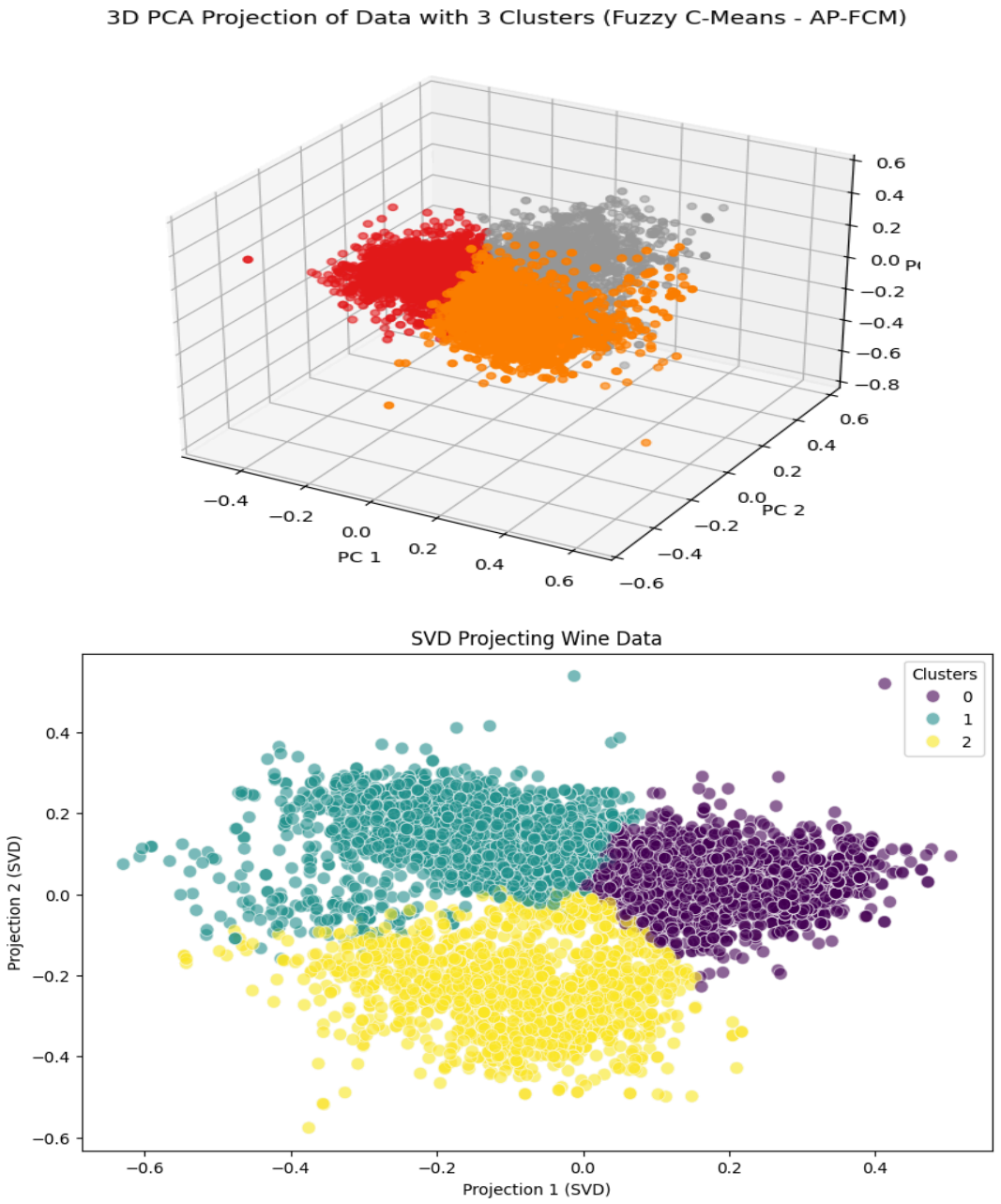



On the other hand, for **c = 4**, the initial centroids remain well-centered, and the overall clustering quality, while slightly lower in terms of validation metrics compared to **c = 3**, still reflects the dataset's robustness.

in all tested configurations (c = 2, 3, 4), the validation indices remained consistent with the earlier experiments using standard FCM. This further supports the assessment that the Wine

Quality Dataset is well-behaved, a term typically used to describe datasets with high inter-cluster separation and low intra-cluster variance. In such cases, different clustering initializations and parameter settings tend to converge to similar local optima. Consequently, all validation indices yield stable results, and advanced initialization techniques (like Anomalous Clustering) act more as validation tools than necessary improvements.

Below are the 3D PCA and SVD visualizations for the configuration with 3 clusters, the optimal number of clusters identified in our analysis, using Anomalous Clustering as the initialization method for Fuzzy C-Means.



## Normalization and Comparative Analysis

For the clustering experiments, we tested both Range Normalization and Z-score Normalization. After evaluating the results from both methods, Range Normalization was selected as the best fit for our dataset. This decision was supported by the fact that the points seem to “cluster together” more effectively, as shown visually in the PCA and SVD visualizations. Specifically, the Range Normalization produced a clustering structure where the data points within each cluster are more closely grouped, exhibiting stronger cohesion and clearer separation between clusters compared to Z-score Normalization.

This choice was further confirmed by the stability of the clustering results across different seeds, as well as the consistent performance of the validation indices.

Below is the table showing the clustering results for the Z-score normalized data:

| c | Cost | Xie-Beni | Silhouette | ARI |
|---|---|---|---|---|
| 2 | 35729.45 | 22.6983 | 0.1740 | 0.0019 |
| 3 | 23786.37 | 20.3566 | 0.2158 | 0.0362 |
| 4 | 17668.91 | 321.5452 | 0.1576 | 0.0353 |
| 5 | 14109.28 | 698.5475 | 0.1244 | 0.0351 |

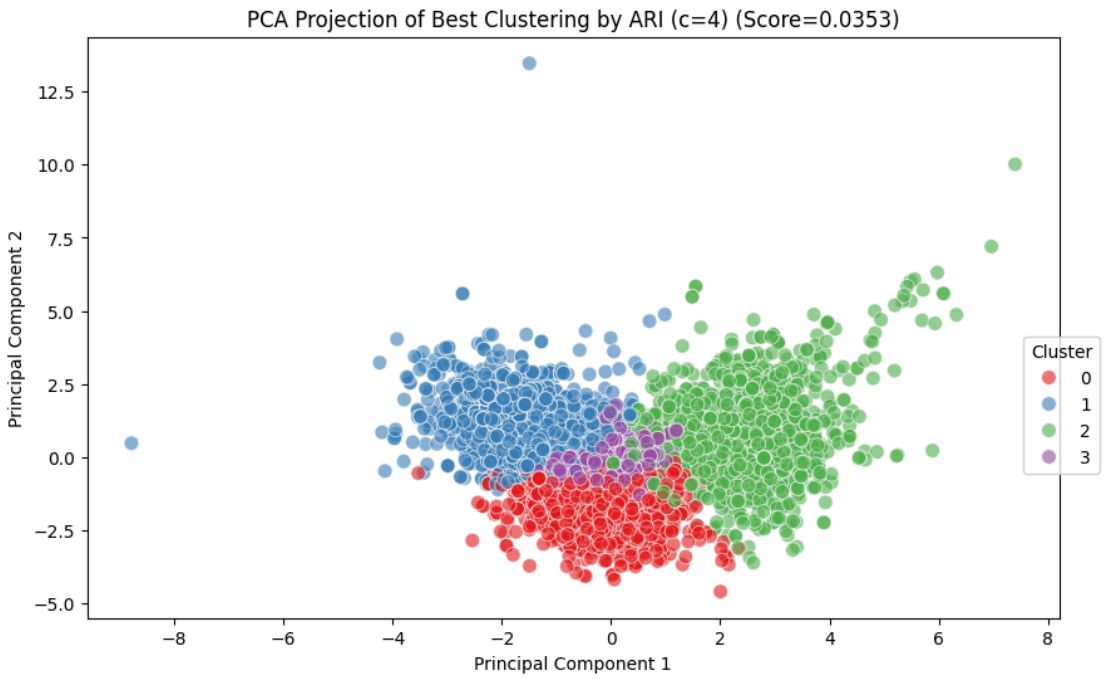
When comparing the results from Range Normalization with those obtained using Z-score Normalization, the differences are significant. The Xie-Beni index, which measures compactness and separation, shows a dramatic increase in values with z-score Normalization, jumping from **0.6995** at c=3 in range normalization to **22.6983** for c=2 in Z-score Normalization. This highlights a significant deterioration in clustering quality.

Similarly, the Silhouette score drops from **0.2501** for c=3 with Range normalization to **0.2158** in Z-score Normalization, which suggests that the clusters have a less distinct boundary and are less cohesive. Additionally, the ARI (Adjusted Rand Index), which measures the agreement between the clusters and the ground truth, also suffers a decline from **0.0604** (Range) to **0.0362** (Z-score), further confirming the suboptimal performance of Z-score when compared to Range Normalization.

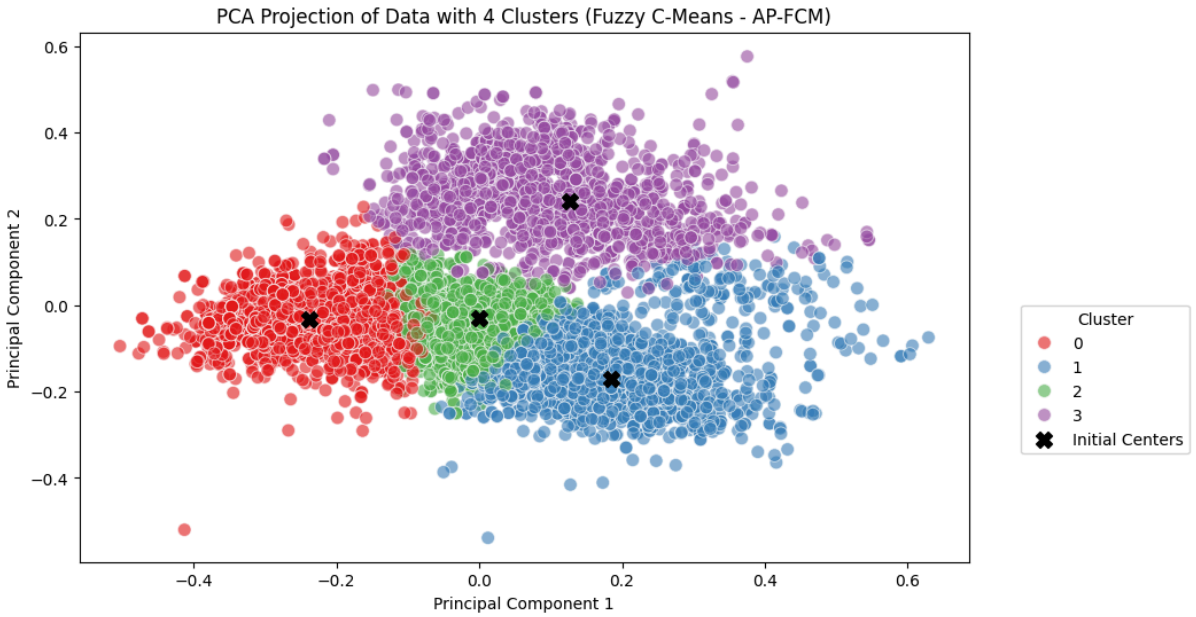
The cost function values are significantly higher with Range Normalization (e.g., 35729.45 for c=2), which, although expected due to the scale of the data, still indicates a higher level of within-cluster variance compared to the results obtained with Z-score normalization.

Visually, the clusters in the Z-score Normalization case appear more overlapped, with reduced separation between them. Below, there are two figures, each showing clustering results for 4 clusters: the first using Z-score normalization and the second using Range normalization.

**Z-score Normalization with Fuzzy C-Means for 4 Clusters**




**Range Normalization with Fuzzy C-Means for 4 Clusters**




In the Z-score normalization plot, we observe a decrease in inter-cluster separation, with the clusters appearing more similar to each other. This suggests higher intra-cluster similarity, which is undesirable as it indicates that the clustering algorithm struggles to clearly distinguish between the clusters. In contrast, with Range Normalization, the clusters are more distinctly separated, with clear boundaries between them. This separation is particularly noticeable when compared to the Z-score Normalization case, where one of the

clusters is almost indistinguishable from the others. In fact, in the Z-score case, this cluster appears to be positioned in the middle of the other three, making it difficult to identify clearly.

In summary, Range Normalization provided a better fit for the dataset, allowing the clustering algorithm to perform more effectively. It resulted in much clearer and more meaningful clustering outcomes, both visually and quantitatively (via validation indices). Z-score normalization, on the other hand, led to poorer clustering performance, with higher intra-cluster similarity and reduced inter-cluster separation, making it less suitable for this dataset.