Introduction to Machine Learning
Lab Session 2: Hierarchical Clustering and Linkage Measures

Group 14
Matteo Omizzolo (S0740525) & Liv Tan Ker Jin (S5778239)

September 27, 2023

1. Introduction

In this assignment, we will explore clustering analysis, which is a technique that groups similar data points based on certain features.

We are given a file containing 200 two-dimensional feature vectors. There are no labels associated with the data points. Our goal is to implement the agglomerative hierarchical clustering algorithm using simple (squared) Euclidean distance.

We will consider different linkage measures like '**Single**', '**Average**', '**Complete**', and '**Ward**' linkage functions and different choices for the number of clusters K = 2, 3, 4.

2. Methods

Agglomerative Hierarchical Clustering is a connectivity-based clustering method. It works through a bottom-up approach, initially considering each data point as a single cluster and then iteratively merging the closest clusters based on a specified linkage measure, until all data is merged into one single cluster.

This is how we will approach Agglomerative Hierarchical Clustering in this assignment:

After reading the file containing the data, the initial step involves computing the distance matrix using the squared Euclidean distance. In this step, we treat every data point as a separate cluster. The distance matrix shows us the similarity between each pair of clusters.

```
# Load the data from the CSV file into a DataFrame
data = pd.read_csv('data_clustering.csv', header=None)
```

```
# Define a function to compute the squared Euclidean distance between
   two points
def squared_euclidean_distance(point1, point2):
    return np.sum((point1 - point2) ** 2)

# Compute the distance matrix using squared Euclidean distance
distance_matrix = np.zeros((len(data), len(data)))
for i in range(len(data)):
    for j in range(len(data)):
```

```
        distance_matrix[i, j] = squared_euclidean_distance(data.values
    [i], data.values[j])
```

Next, we perform hierarchical clustering using the linkage (distance_matrix, 'single') function. It merges the closest clusters and updates the distance matrix based on the chosen linkage method ('single', in this example) until only one cluster remains. We then plot the dendrogram (figure 1), showing the merging process.

```
# Perform hierarchical clustering using single linkage
linked_single = linkage(distance_matrix, 'single')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
dendrogram(linked_single, orientation='top', distance_sort='descending'
    , show_leaf_counts=True)

# Add cut off thresholds
cut_off_thresholds = [1.05, 0.98, 0.85]
for threshold in cut_off_thresholds:
    plt.axhline(y=threshold, color='black', linestyle='--')

plt.title('Hierarchical Clustering Dendrogram (Single Linkage)')
plt.xlabel('Sample Index')
plt.ylabel('Squared Euclidean Distance')
plt.show()
```

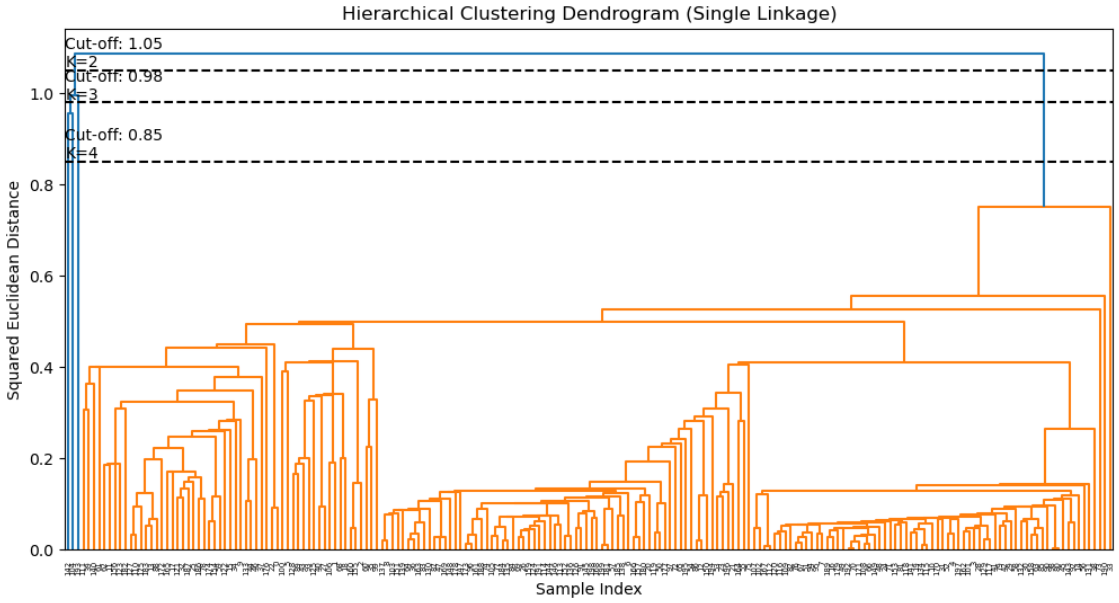

Figure 1: *Hierarchical Clustering Dendrogram (single linkage)*

The dendrogram provides a visual representation of the merging process, so we can define the number of clusters K by setting a cut-off threshold (cutting the dendrogram at a certain height). In this case, we cut off at y=[1.05, 0.98, 0.85] in order to get K= 2, 3, 4.
After that, we perform the agglomerative hierarchical clustering based on the linkage function and the determined number of clusters, K. We obtain and print the cluster labels for each datapoint.

```
# Assign data points to clusters based on the determined number of
    clusters, K
clusters = fcluster(linked_single, K, criterion='maxclust')

# Print the cluster labels
print('Cluster Labels:')
print(clusters)
```

Lastly, we compute the average silhouette score.

```
# Compute the average silhouette score automatically
average_silhouette_score = silhouette_score(data, clusters)
print('Average Silhouette Score:', average_silhouette_score)
```

We repeat the steps above for different linkage functions including 'average', 'complete', and 'ward', and for each of the clusters K = 2, 3, 4.
Different linkage measures determine how the distance between clusters is calculated during the merging step. The selection of the appropriate linkage measure is crucial and can impact the clustering outcome.
The **'single'** linkage computes the distance between the closest data points of two clusters.
The **'complete'** linkage computes the distance between the furthest data points of two clusters.
The **'average'** linkage computes the average distance between all possible pairs of data points in two clusters.
The **'ward'** linkage minimizes the sum of squared differences within all clusters. It tries to minimize the variance in the merged cluster.

3. Results

- A figure displaying the dendrogram and indicating the cut-off thresholds for different numbers of clusters using dashed horizontal lines (in total, 4 dendrograms, each with the presence of 3 dashed lines for cut-off). We choose the cut-off thresholds to get k=2,3,4.

In the clustering analysis, we use dendrograms to visualize hierarchical clustering results using different linkage methods: Single, Average, Complete, and Ward. The y-axis values for cut-off thresholds at k = [4, 3, 2] clusters are as follows:

- Single Linkage (Figure 2): y =[0.85, 0.98, 1.05]
- Average Linkage (Figure 3): y =[1.9, 2.3, 4.3]
- Complete Linkage (Figure 4): y =[3, 3.5, 4.5]
- Ward's Linkage (Figure 5): y =[9, 10, 15]

These cut-off thresholds help us decide the optimal number of clusters for each linkage method.

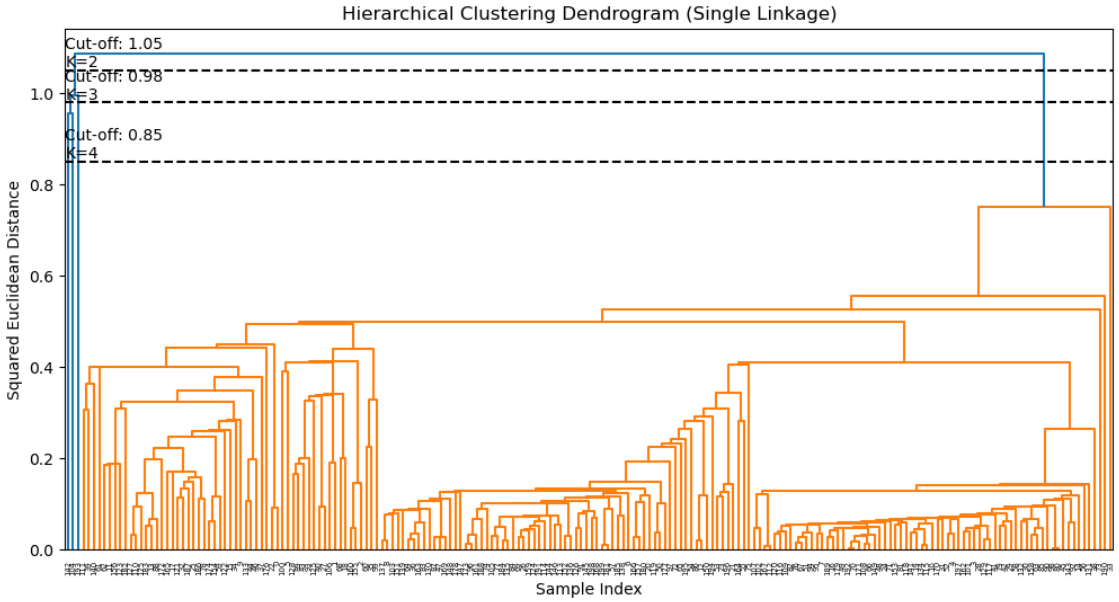

Figure 2: *Hierarchical Clustering Dendrogram (**Single Linkage**)*

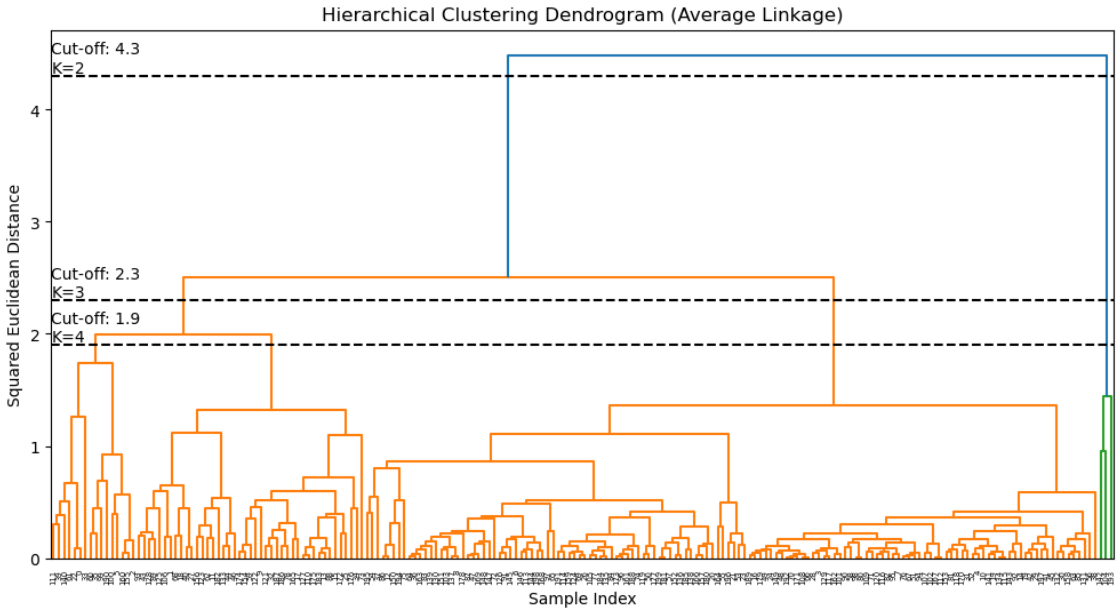

Figure 3: *Hierarchical Clustering Dendrogram (**Average Linkage**)*

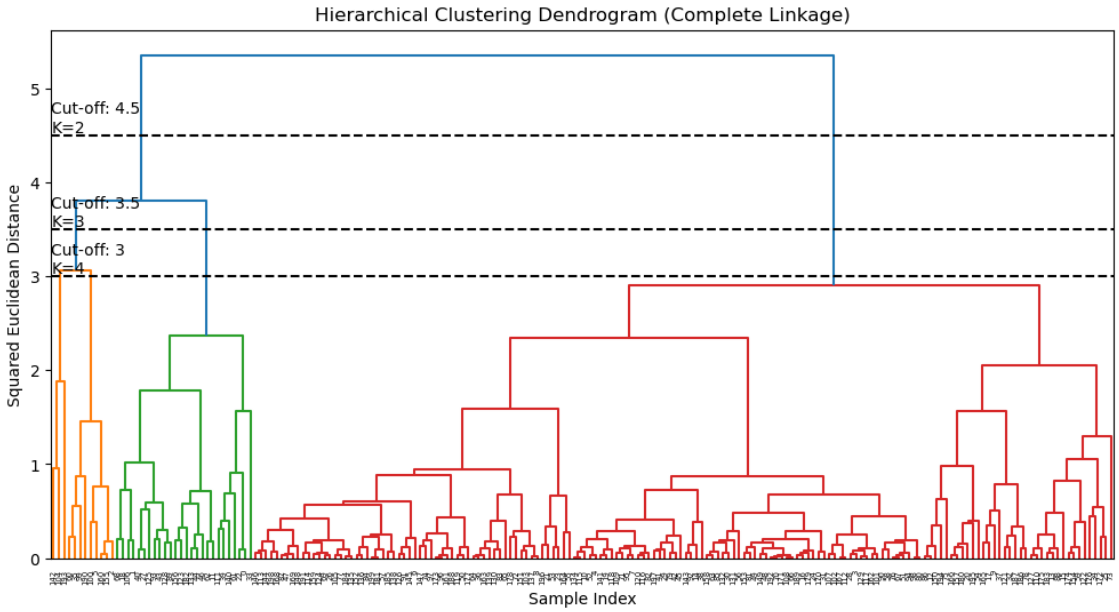

Figure 4: *Hierarchical Clustering Dendrogram (**Complete Linkage**)*

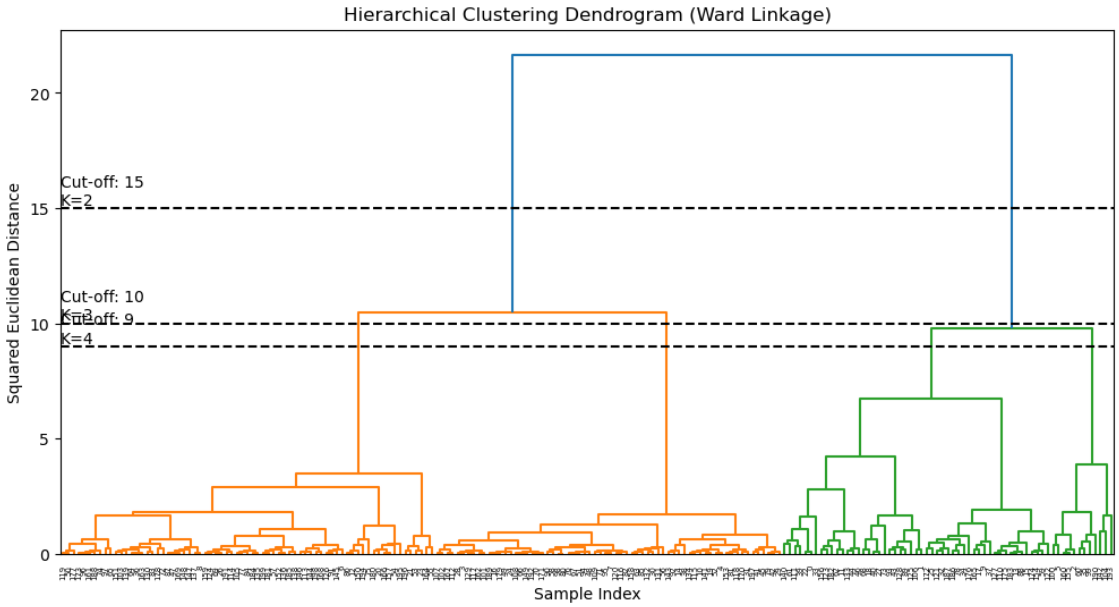

Figure 5: *Hierarchical Clustering Dendrogram (**Ward's Linkage**)*

- A figure displaying the original data points and the resulting clusters (in total, 13 figures, one for the visualization of the original data points and 12 for clustering results).

For this part, we first plotted the original data points (Figure 6). We then analyzed the resulting clusters for k = [2, 3, 4] using all the linkage methods:

- Single Linkage (Figure 7, Figure 8, Figure 9)
- Average Linkage (Figure 10, Figure 11, Figure 12)
- Complete Linkage (Figure 13, Figure 14, Figure 15)
- Ward's Linkage (Figure 16, Figure 17, Figure 18)

The scatter plot applied to the hierarchical clustering analysis is a useful representation to visualize and compare different results obtained by using different linkage approaches. Therefore, it's an important tool in the selection of the most accurate linkage method enabling a deeper understanding of the data structure.

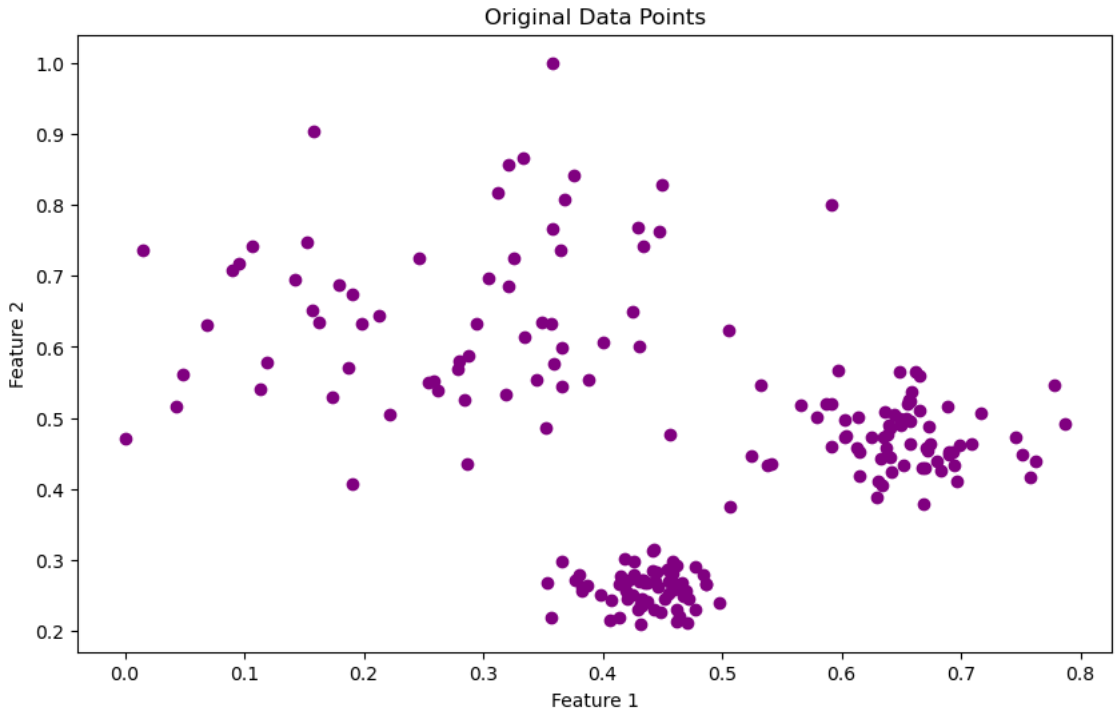

Figure 6: *Original Data Points*

SINGLE LINKAGE

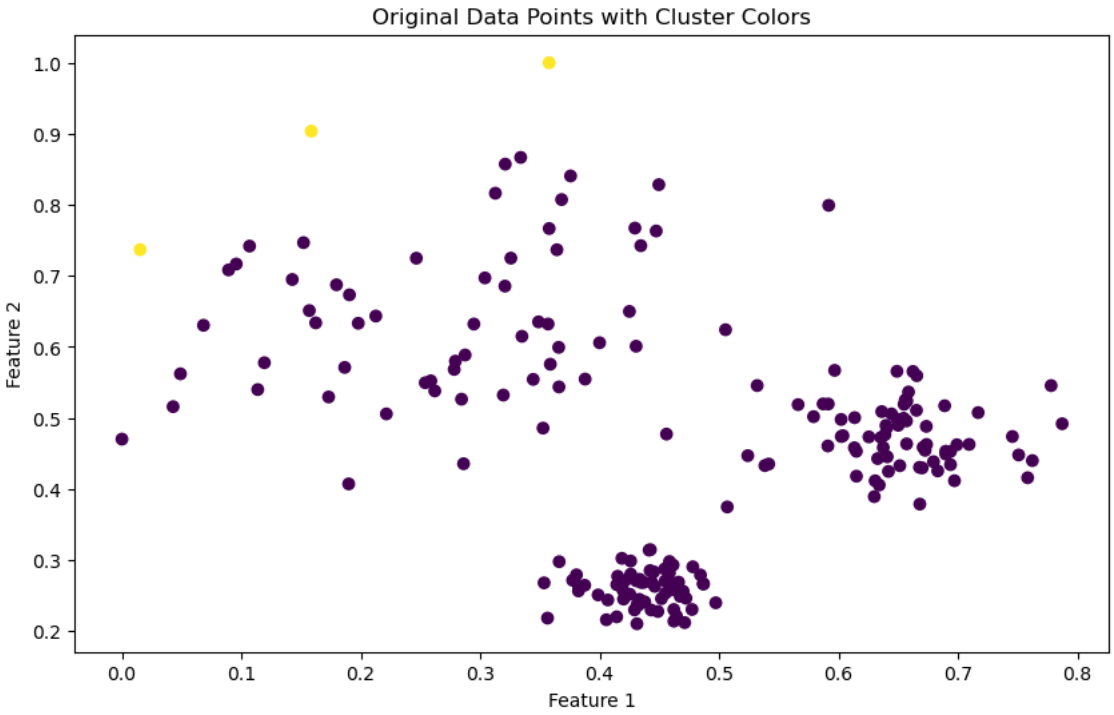

Figure 7: *Original Data Points with Cluster Colors (k=2)*

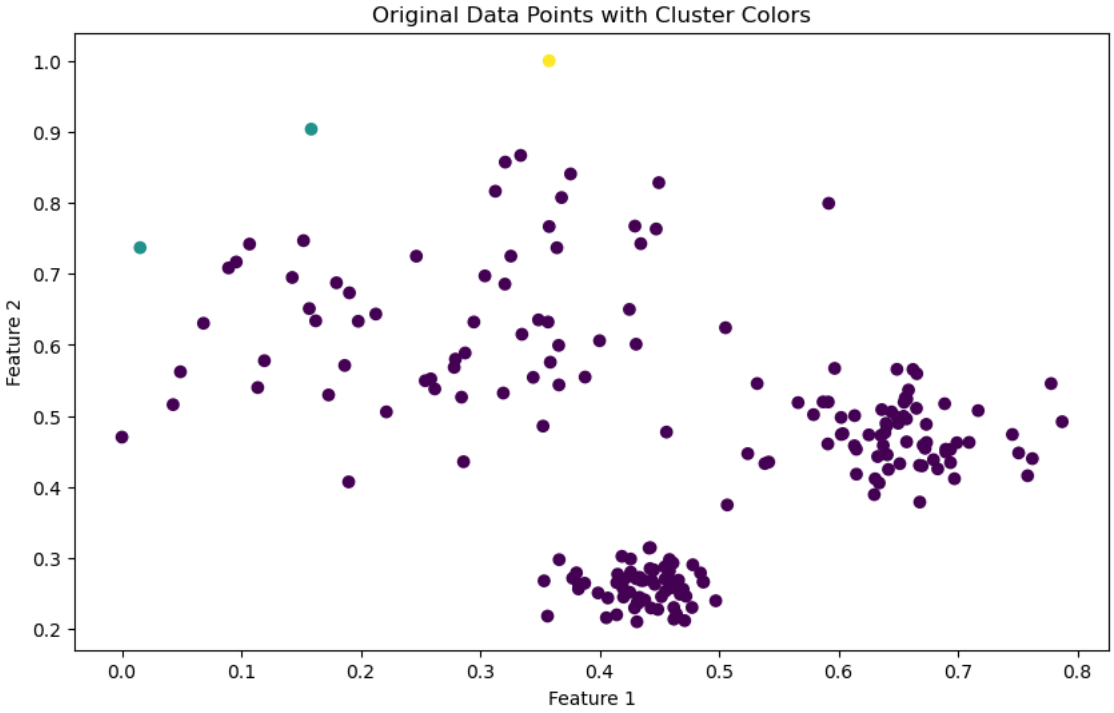

Figure 8: *Original Data Points with Cluster Colors (k=3)*

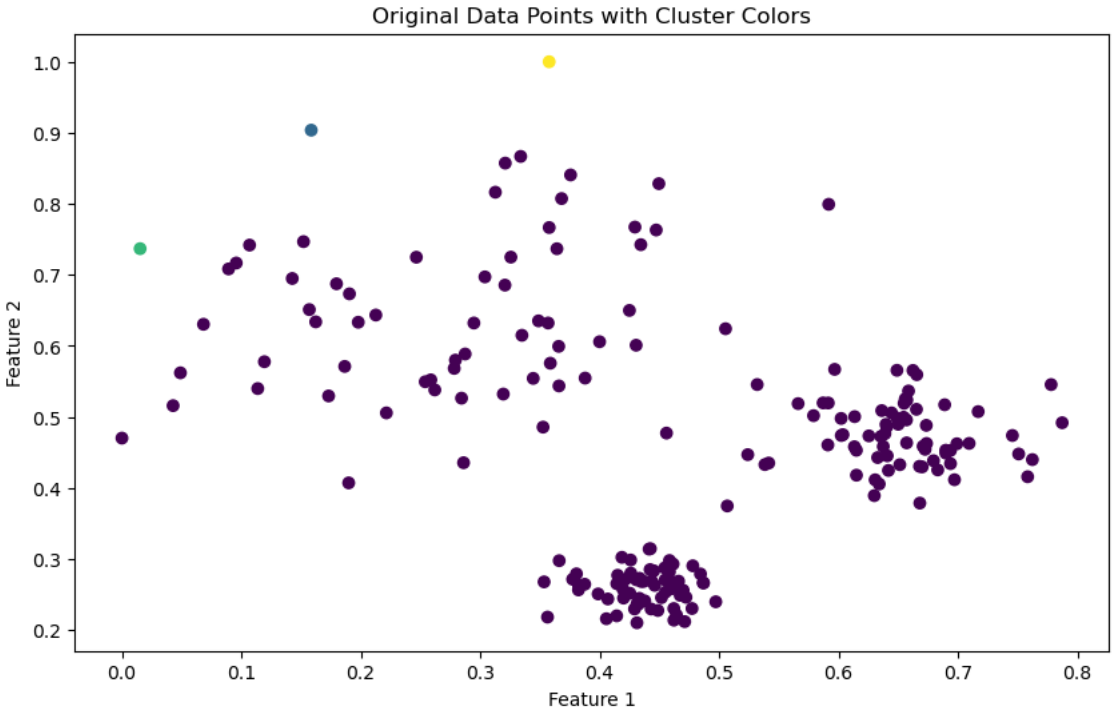

Figure 9: *Original Data Points with Cluster Colors (k=4)*

AVERAGE LINKAGE

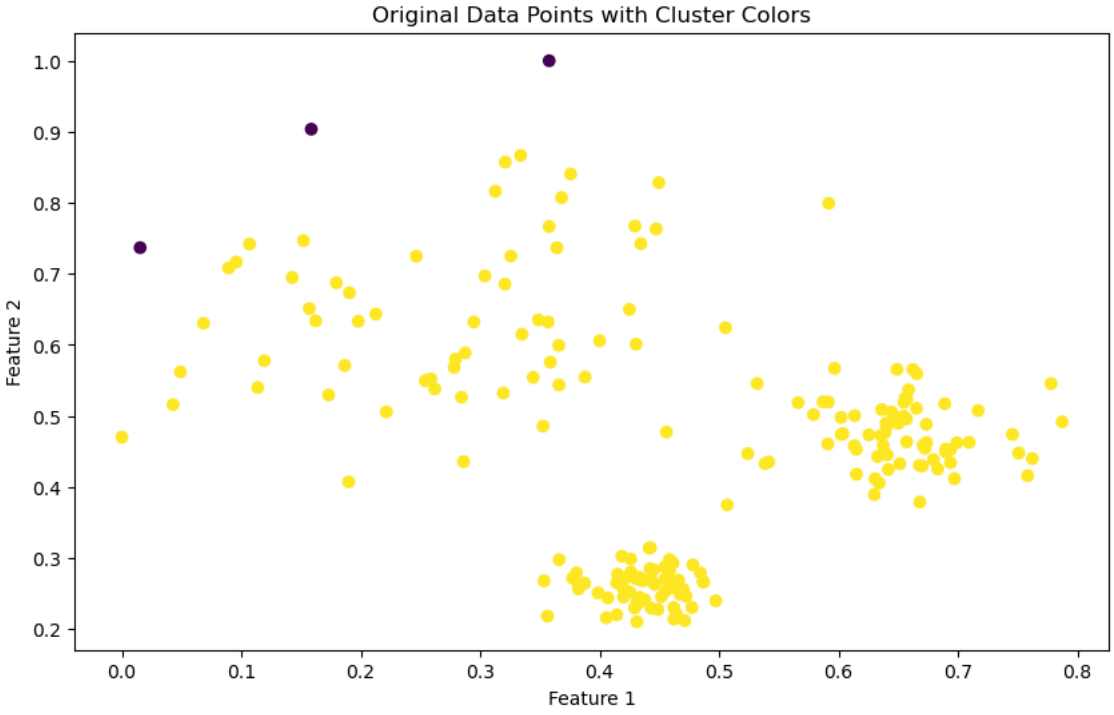

Figure 10: *Original Data Points with Cluster Colors (k=2)*

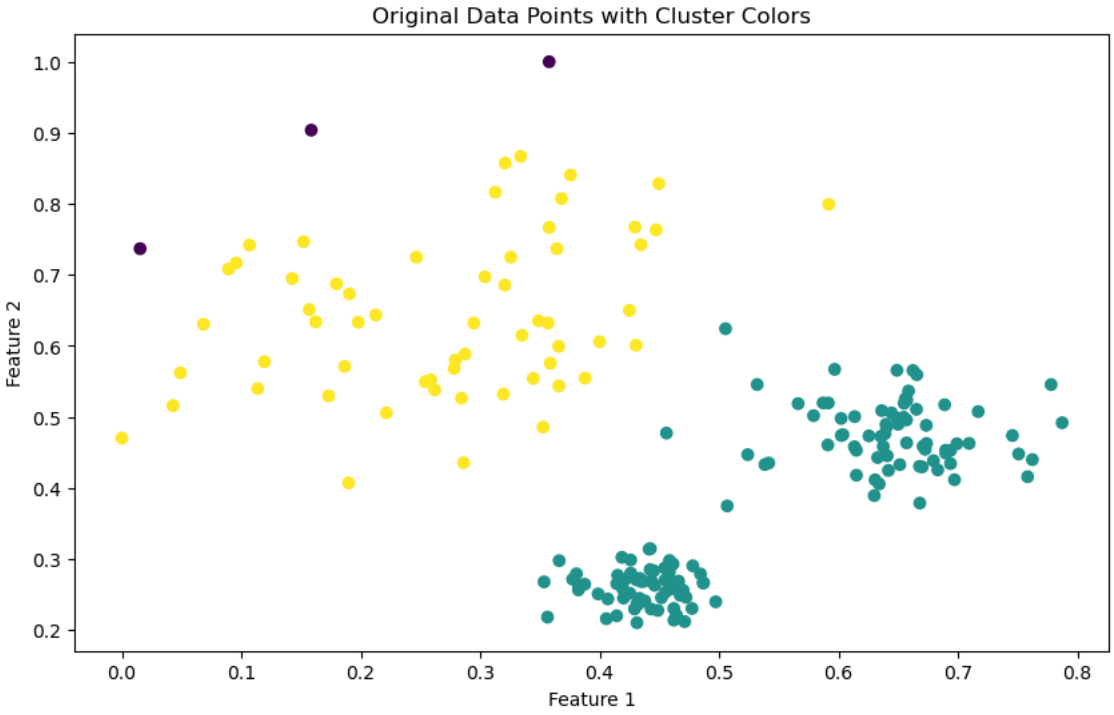

Figure 11: *Original Data Points with Cluster Colors (k=3)*

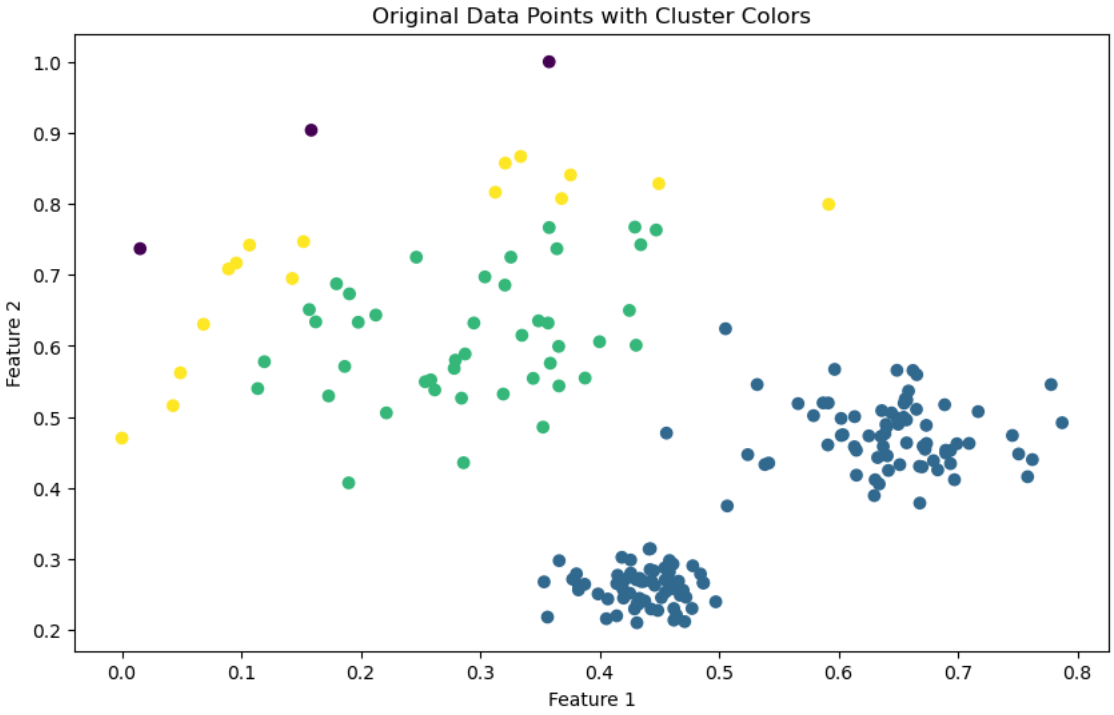

Figure 12: *Original Data Points with Cluster Colors (k=4)*

COMPLETE LINKAGE

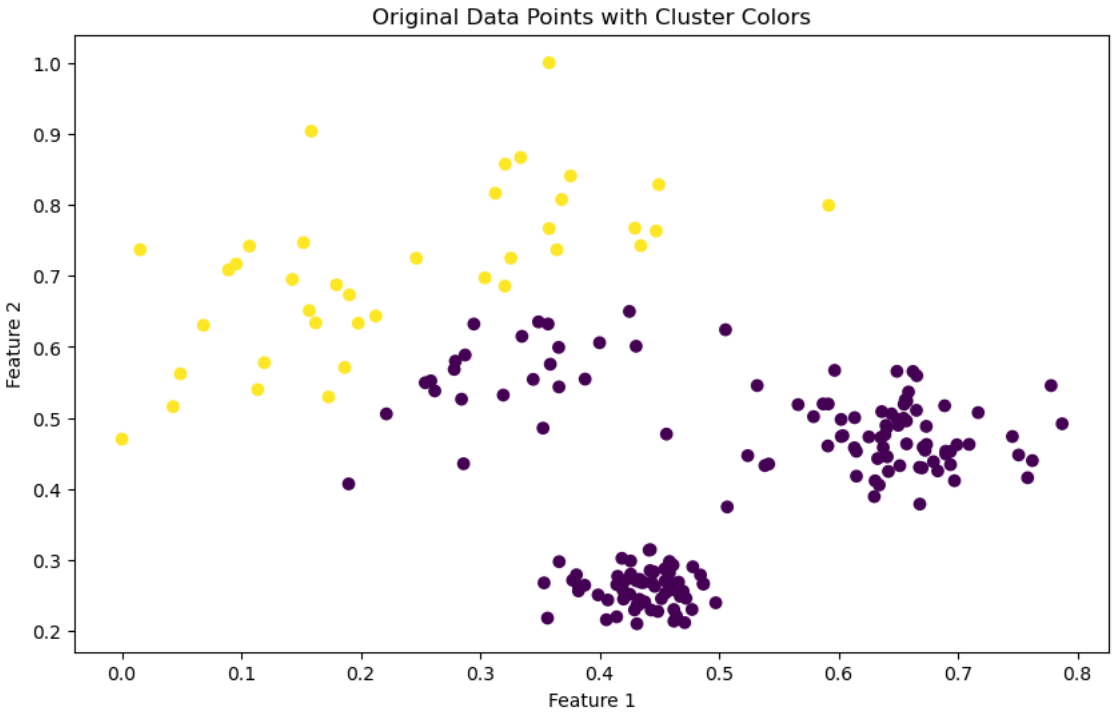

Figure 13: *Original Data Points with Cluster Colors (k=2)*

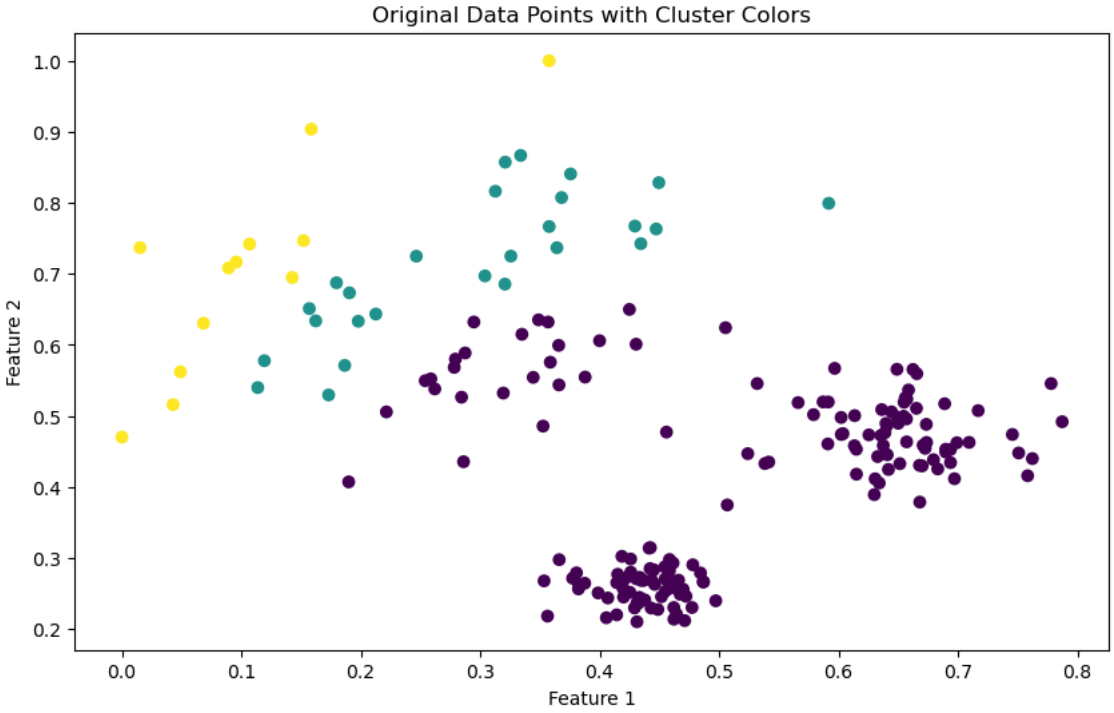

Figure 14: *Original Data Points with Cluster Colors (k=3)*

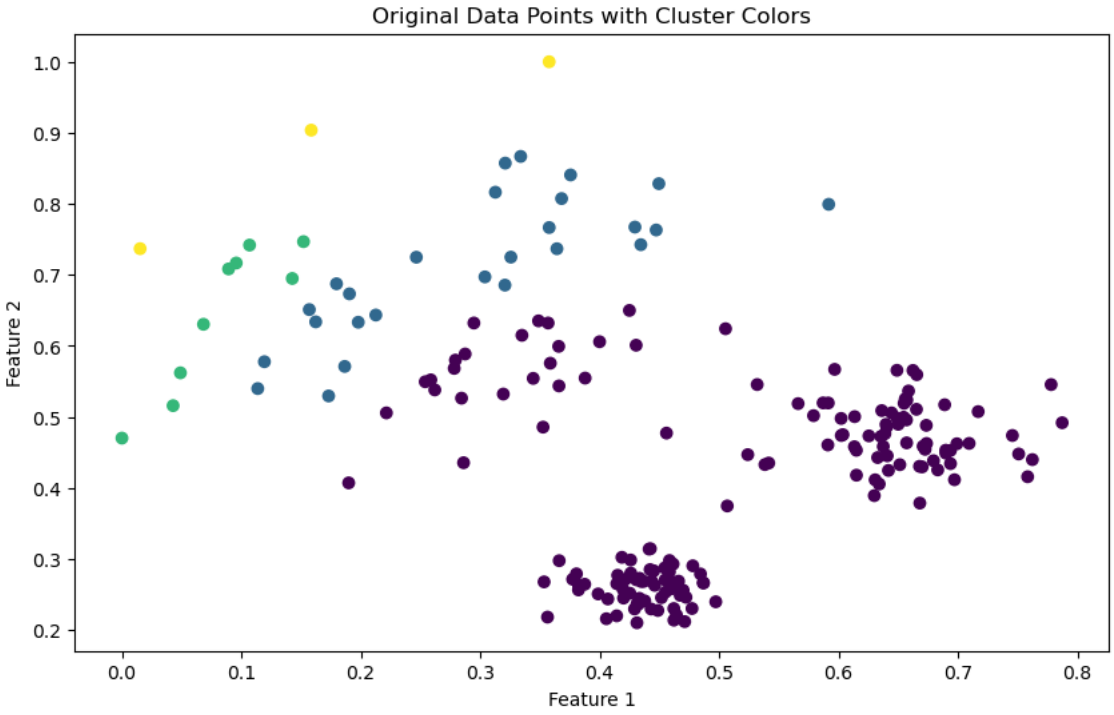

Figure 15: *Original Data Points with Cluster Colors (k=4)*

WARD'S LINKAGE

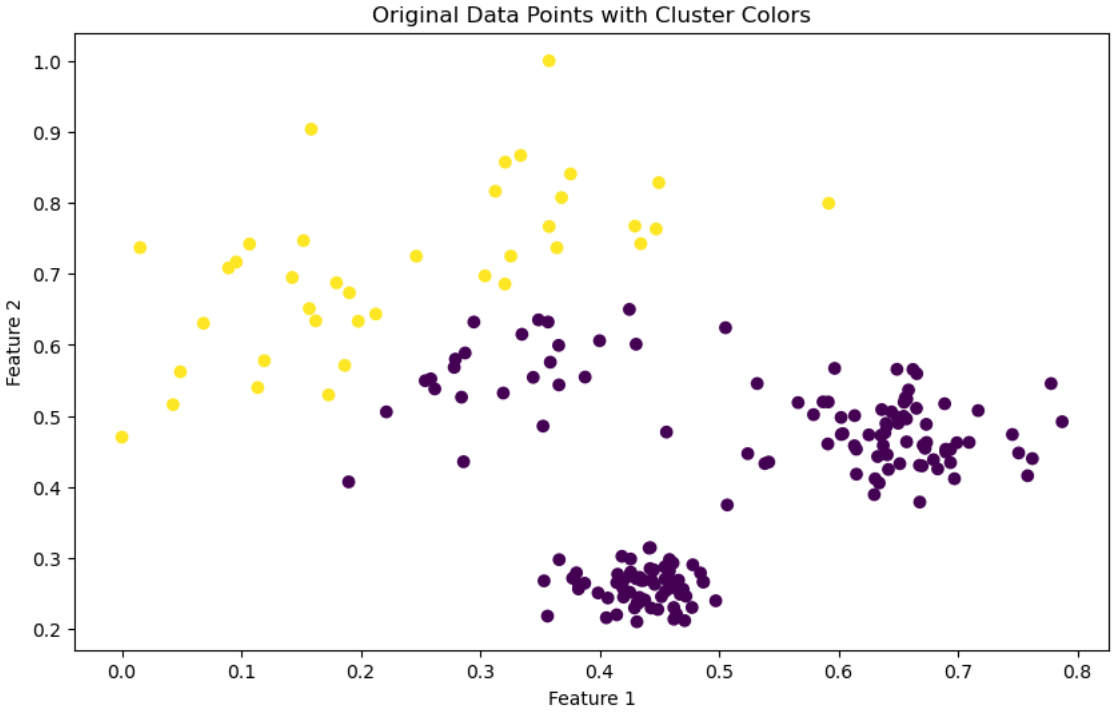

Figure 16: *Original Data Points with Cluster Colors (k=2)*

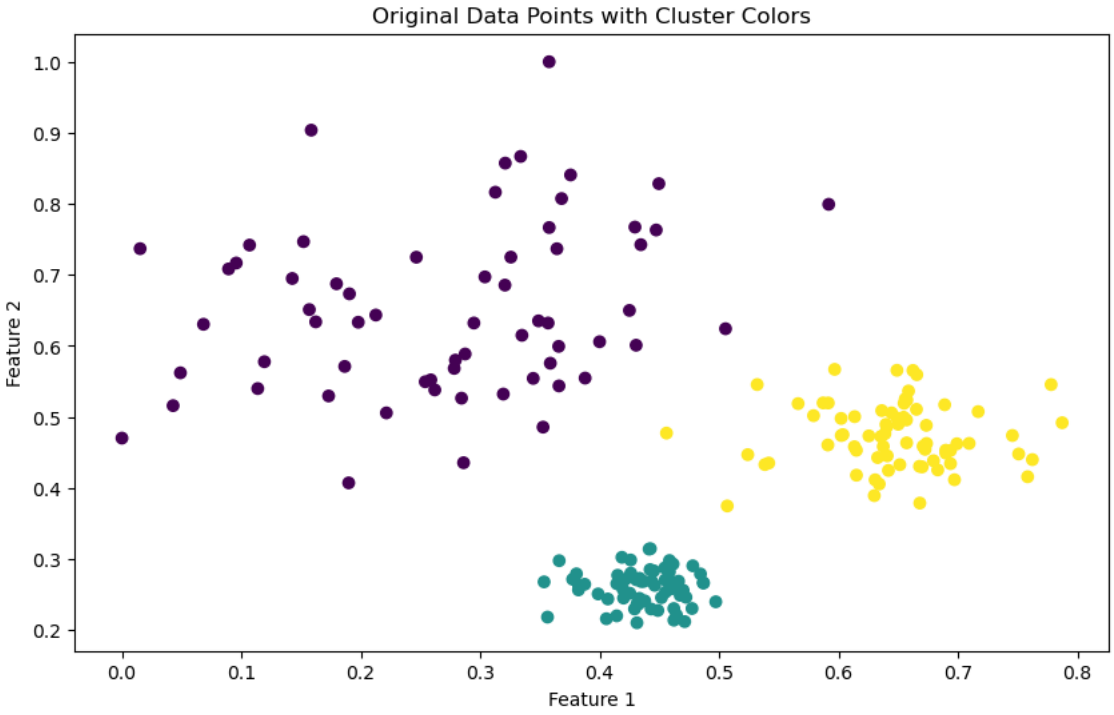

Figure 17: *Original Data Points with Cluster Colors (k=3)*

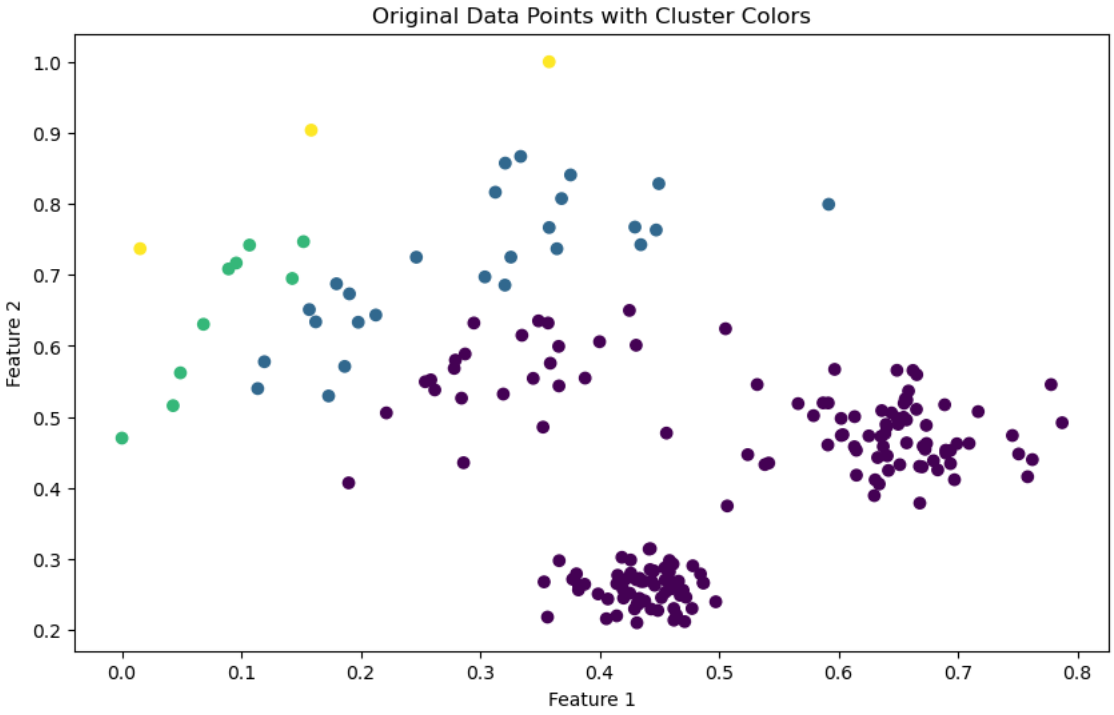

Figure 18: *Original Data Points with Cluster Colors (k=4)*

- For the quantitative results, you should provide a table listing the computed silhouette score for all different linkage measures and different choices of the number of clusters.

The Silhouette Score is a metric used to calculate the goodness of a clustering technique and its value ranges from -1 to 1. If it's equal to 1 that means that the clusters are clearly distinguished from each other, if it's equal to 0 the distance between the cluster is not significant and if it's equal to -1 clusters are being computed in the wrong way.

Linkage measures	Number of clusters K	Silhouette Score
Single	2	0.40
	3	0.35
	4	0.32
Average	2	0.40
	3	0.47
	4	0.42
Complete	2	0.47
	3	0.39
	4	0.37
Ward	2	0.47
	3	0.65
	4	0.37

4. Discussion

After a thorough analysis of the qualitative and quantitative results, we can formulate valuable observations and conclusions regarding the optimal linkage methods and the appropriate number of clusters.

The **Single Linkage** approach, as depicted in the figures displaying the original data points and the resulting clusters, does not effectively differentiate the clusters, often leading to chain-like structures. This means that several clusters may be joined together simply because of the proximity of points from different clusters. A characteristic of this method is that it performs poorly in the presence of noise, resulting in the grouping of outliers as other clusters (as observed with the three points at the top left of Figure 8 and Figure 9). Additionally, the Silhouette Score for various cluster numbers tends to approach 0 rather than 1, suggesting suboptimal results in the clustering process. The optimal number of clusters for this approach is 2.

The **Average Linkage** and **Complete Linkage** appear to offer improved clustering approaches compared to the one previously analyzed. These methods provide a better representation of the data points and give a more accurate reflection of the distance between different clusters. However, it's important to note that the Silhuotte Score did not reach values indicative of highly accurate clustering, suggesting an alternative to achieve higher accuracy in the clustering process. In fact, the optimal number of clusters for these two approaches is 3 for Average Linkage and 2 for Complete Linkage, resulting in a Silhouette Score of only 0.47 for both.

Overall, it's evident that the most effective measure is the **Ward's Linkage**. The clusters displayed in the plots are well apart from each other, suggesting a robust performance of this approach in capturing the stricture of the data. The optimal number of clusters for this linkage approach is 3 with a Silhouette Score of 0.65 which is closer to 1 than the other values for different numbers of clusters.

To conclude, it's important to emphasize how different linkage measures resulted in different results. Understanding these differences allows a better interpretation of the clusters and helps in selecting the optimal linkage measure for specific data, increasing the effectiveness and reliability of the clustering process.

5. Bonus

Small variations might exist due to differences in floating-point arithmetic or handling of edge cases. We tried multiple times but the result won't match.

```
# Compute the average silhouette score manually
def calculate_silhouette_score(data, labels, distance_matrix):
    num_samples = len(data)
    silhouette_scores = np.zeros(num_samples)

    for i in range(num_samples):
        # Calculate a_i (average intra-cluster distance)
        a_i = np.sum([distance_matrix[i, j] for data in cluster_points
   ]) / max(len(cluster_points) - 1, 1)

        # Calculate b_i (average inter-cluster distance)
        b_i = min([np.mean([distance_matrix[i, j] for j in range(
   num_samples) if labels[j] == k])
                   for k in set(labels) if k != labels[i]])

        # Calculate silhouette score
        silhouette_scores[i] = (b_i - a_i) / max(a_i, b_i)

    # Compute the mean silhouette score
    average_silhouette_score = np.mean(silhouette_scores)

    return average_silhouette_score

print("Silhouette Score manual:", average_silhouette_score)
```