Stone Barrett

CSE 590: Intro to Machine Learning – Section 53

Homework 5

Problem 1: K-Means Clustering

a) **Finding Optimal Cluster Count**

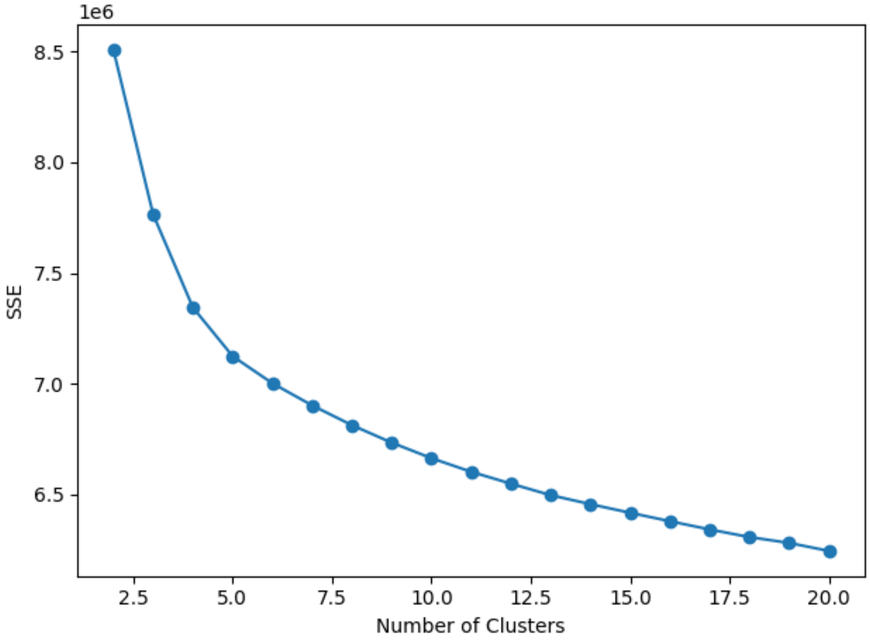


Figure 5.1

To find the optimal number of clusters to be used in K-Means Clustering, I ran the algorithm for n = 2 all the way up to n = 20 and plotted the number of clusters against the SSE. As expected, the SSE value continually decreases as the number of clusters increases. However, since we don’t want the number of clusters to be very high, I decided to employ the “elbow method” to determine the optimal value. This basically means that wherever the graph has its sharpest bend, there lies the optimal number of clusters. In this case, it would appear that n = 5 stands as the optimal value.

b) **Silhouette Plot**

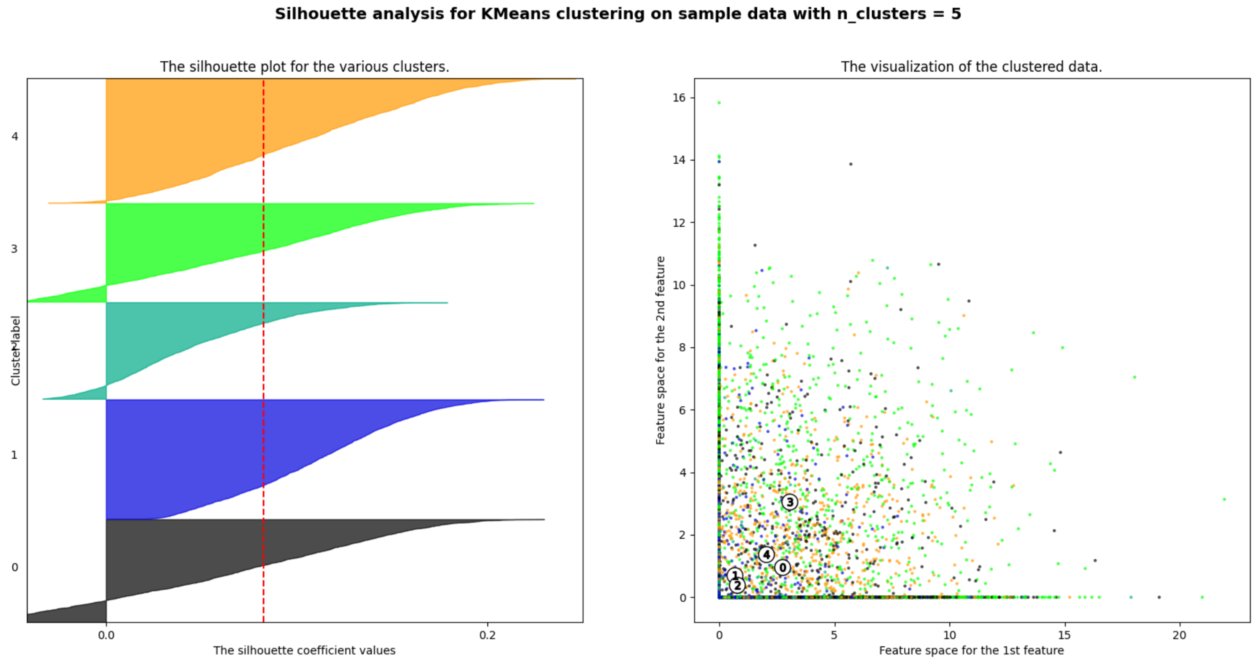
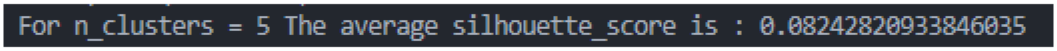


Figure 5.2: Silhouette and Scatter Plot for K-Means Clustering

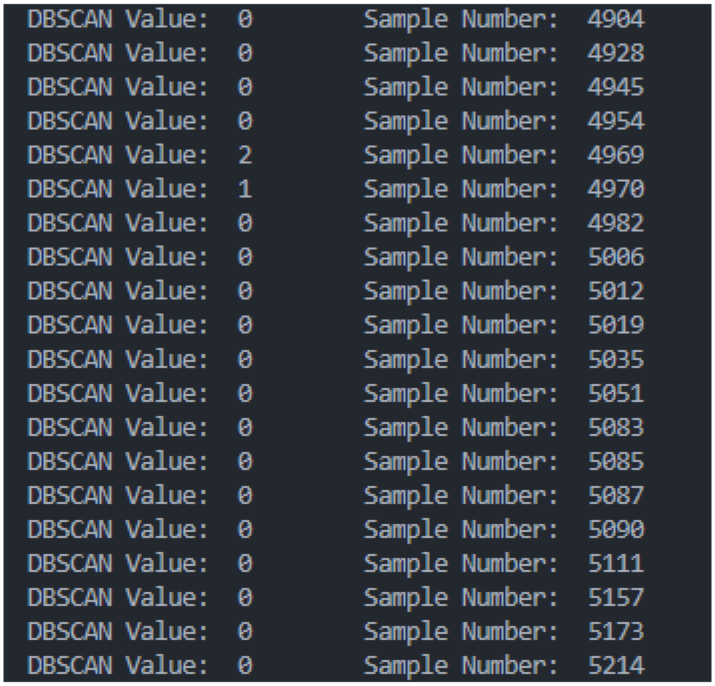


```
For n_clusters = 5 The average silhouette_score is : 0.08242820933846035
```

Figure 5.3: Silhouette Score for K-Means Clustering

Here I was able to generate a silhouette plot using n = 5. In Figure 5.2, we can see on the right that this dataset is not easily clustered. Just visually speaking, the data is spread very erratically and there are no easily identifiable blobs. Nevertheless, the silhouette score is relatively high in Figure 5.3.

c) **Identifying Core and Boundary Samples**



```
DBSCAN Value:  0        Sample Number:  4904
DBSCAN Value:  0        Sample Number:  4928
DBSCAN Value:  0        Sample Number:  4945
DBSCAN Value:  0        Sample Number:  4954
DBSCAN Value:  2        Sample Number:  4969
DBSCAN Value:  1        Sample Number:  4970
DBSCAN Value:  0        Sample Number:  4982
DBSCAN Value:  0        Sample Number:  5006
DBSCAN Value:  0        Sample Number:  5012
DBSCAN Value:  0        Sample Number:  5019
DBSCAN Value:  0        Sample Number:  5035
DBSCAN Value:  0        Sample Number:  5051
DBSCAN Value:  0        Sample Number:  5083
DBSCAN Value:  0        Sample Number:  5085
DBSCAN Value:  0        Sample Number:  5087
DBSCAN Value:  0        Sample Number:  5090
DBSCAN Value:  0        Sample Number:  5111
DBSCAN Value:  0        Sample Number:  5157
DBSCAN Value:  0        Sample Number:  5173
DBSCAN Value:  0        Sample Number:  5214
```

Figure 5.4: DBSCAN Output Classifying Noise and Non-Noise

I was able to have DBSCAN classify each datapoint as noise, core, or boundary and then list their indices. The full list is obviously too large to show in a Figure, but from this output I was able to locate 5 core samples: 5090, 5111, 5157, 5173, 5214. I was also able to locate 2 double-boundary samples: 4969, 5322.

To be completely honest, I’m unsure of how to display the original images that the dataset was pulled from.

Problem 2: Agglomerative Clustering

a) Finding Optimal Cluster Count

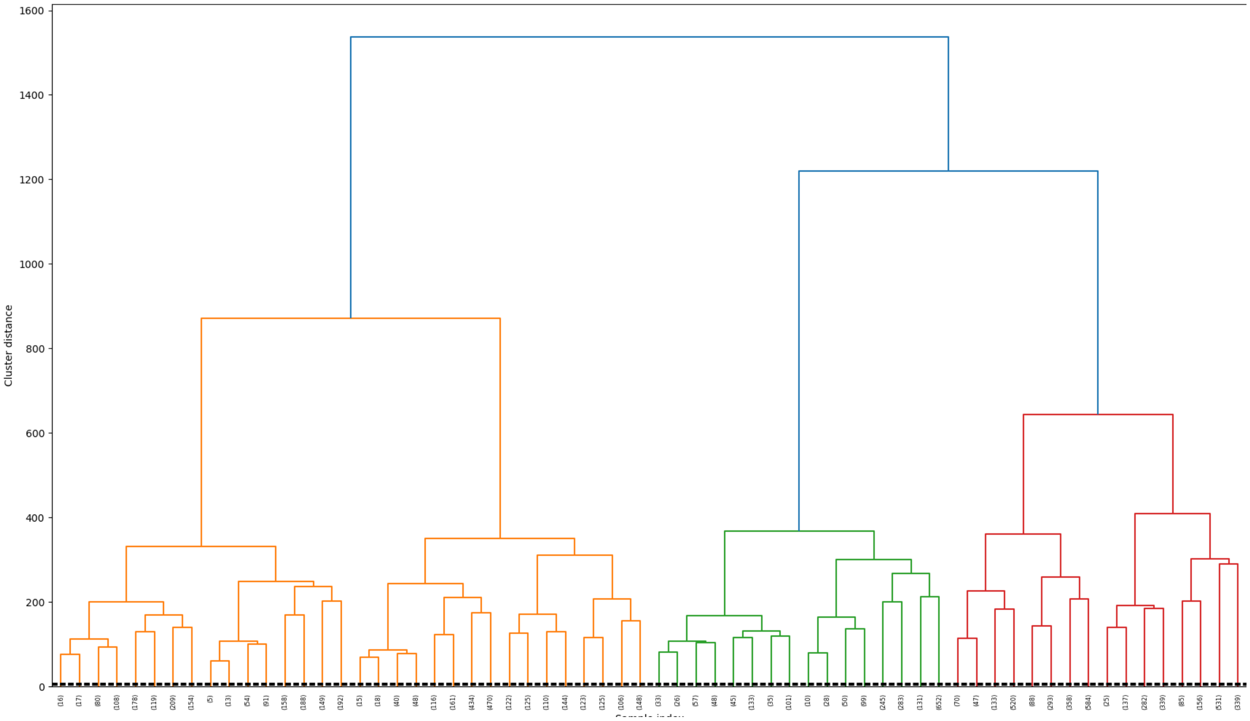


Figure 5.5: Dendrogram

Figure 5.5 shows us a dendrogram that I limited to 5 layers. Any more than 5 resulted in a graph that when displayed was completely incomprehensible due to the sheer size and density of the dataset.

This dendrogram tells us (by showing 4 different clusters) that the optimal number of clusters to be used with Agglomerative Clustering is 4.

b) Silhouette Plot

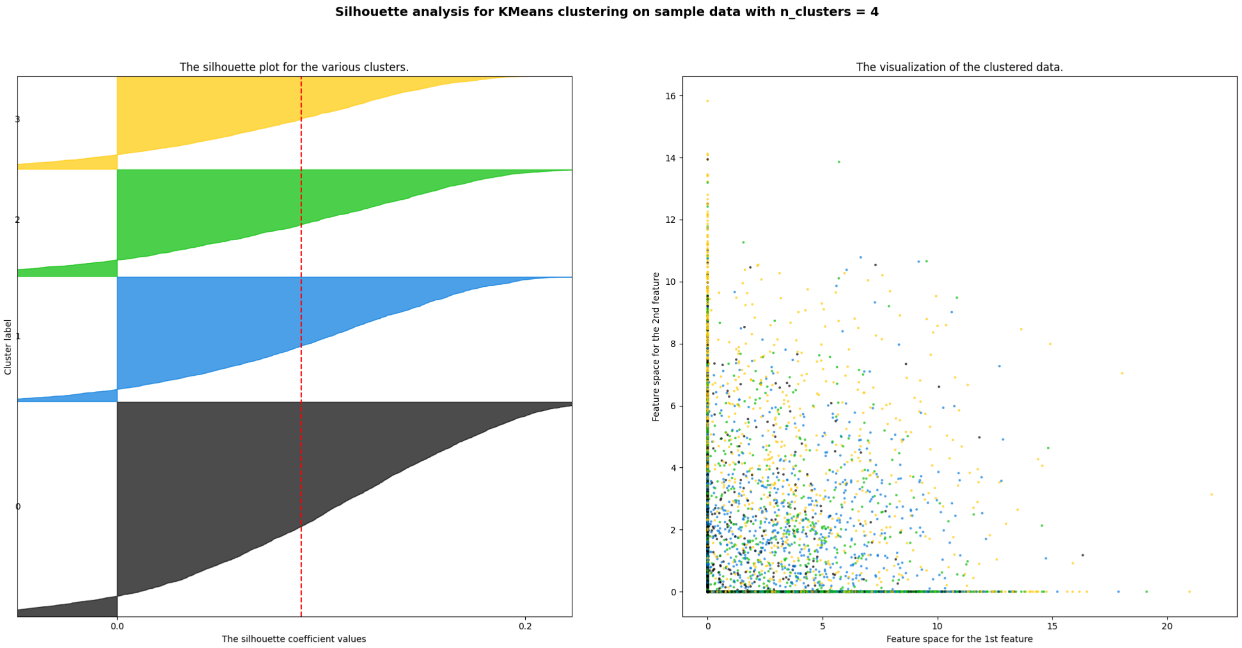
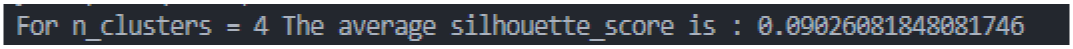


Figure 5.6: Silhouette and Scatter Plot for Ward-Link Agglomerative Clustering



```
For n_clusters = 4 The average silhouette_score is : 0.09026081848081746
```

Figure 5.7: Silhouette Score for Ward-Link Agglomerative Clustering

Here we can see the silhouette plot and score for this algorithm with n = 4. This uses Ward's method for linkage and returns a slightly higher silhouette score than K-Means.

c) Single- and Complete-Link

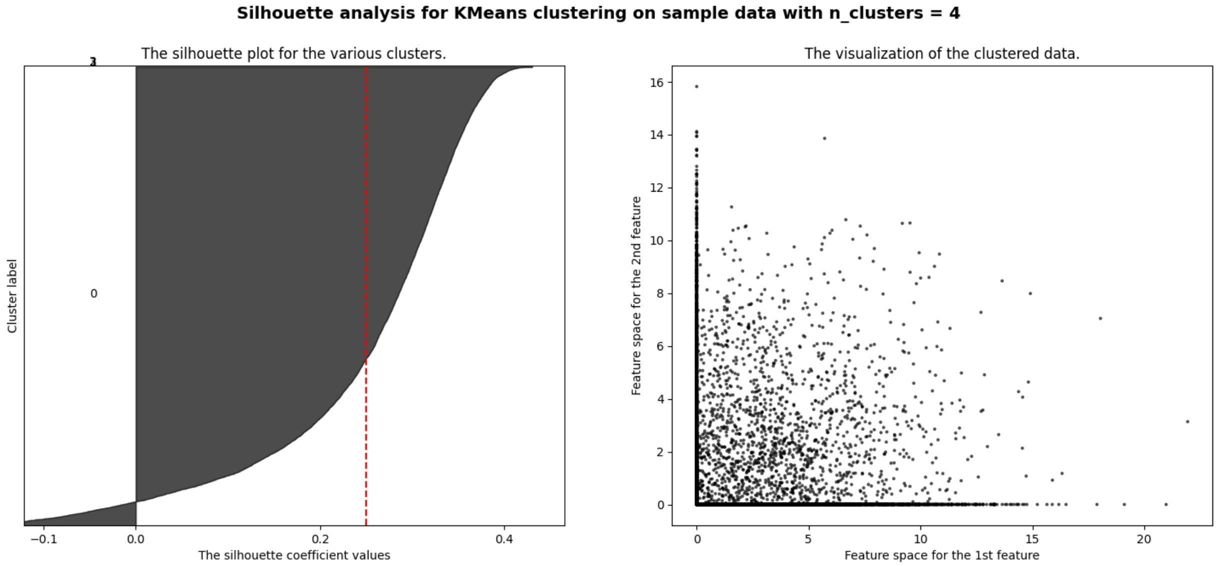
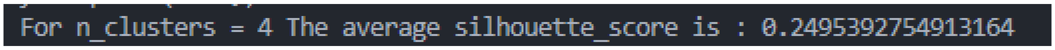


Figure 5.8: Silhouette and Scatter Plot for Single-Link Agglomerative Clustering



```
For n_clusters = 4 The average silhouette_score is : 0.2495392754913164
```

Figure 5.9: Silhouette Score for Single-Link Agglomerative Clustering

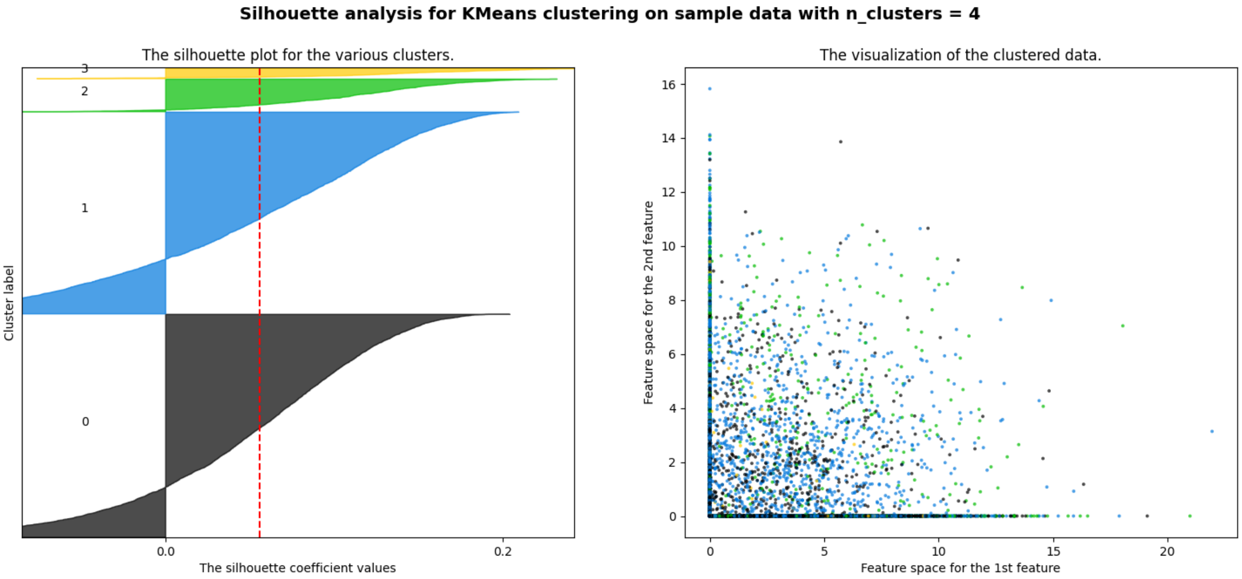
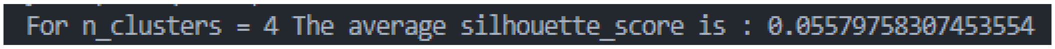


Figure 5.10: Silhouette and Scatter Plot for Complete-Link Agglomerative Clustering

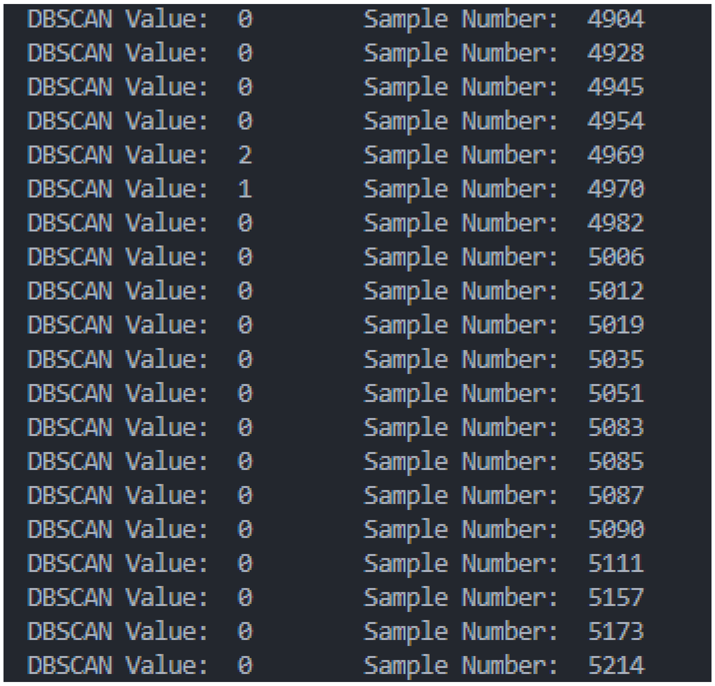


```
For n_clusters = 4 The average silhouette_score is : 0.05579758307453554
```

Figure 5.11: Silhouette Score for Complete-Link Agglomerative Clustering

Figures 5.8 through 5.11 show the silhouette plots and scores for the Single and Complete methods for linkage. Both of these methods return significantly worse scores than Ward's method, meaning that Ward's method is the optimal choice for this situation.

d) **Identifying Core and Boundary Samples**



```
DBSCAN Value:  0        Sample Number:  4904
DBSCAN Value:  0        Sample Number:  4928
DBSCAN Value:  0        Sample Number:  4945
DBSCAN Value:  0        Sample Number:  4954
DBSCAN Value:  2        Sample Number:  4969
DBSCAN Value:  1        Sample Number:  4970
DBSCAN Value:  0        Sample Number:  4982
DBSCAN Value:  0        Sample Number:  5006
DBSCAN Value:  0        Sample Number:  5012
DBSCAN Value:  0        Sample Number:  5019
DBSCAN Value:  0        Sample Number:  5035
DBSCAN Value:  0        Sample Number:  5051
DBSCAN Value:  0        Sample Number:  5083
DBSCAN Value:  0        Sample Number:  5085
DBSCAN Value:  0        Sample Number:  5087
DBSCAN Value:  0        Sample Number:  5090
DBSCAN Value:  0        Sample Number:  5111
DBSCAN Value:  0        Sample Number:  5157
DBSCAN Value:  0        Sample Number:  5173
DBSCAN Value:  0        Sample Number:  5214
```

Figure 5.12: DBSCAN Output Classifying Noise and Non-Noise

I was able to have DBSCAN classify each datapoint as noise, core, or boundary and then list their indices. The full list is obviously too large to show in a Figure, but from this output I was able to locate 5 core samples: 6, 8, 29, 44, 47. I was also able to locate 2 double-boundary samples: 8066, 7277.

To be completely honest, I’m unsure of how to display the original images that the dataset was pulled from.

Adjusted Rank Index and Conclusion

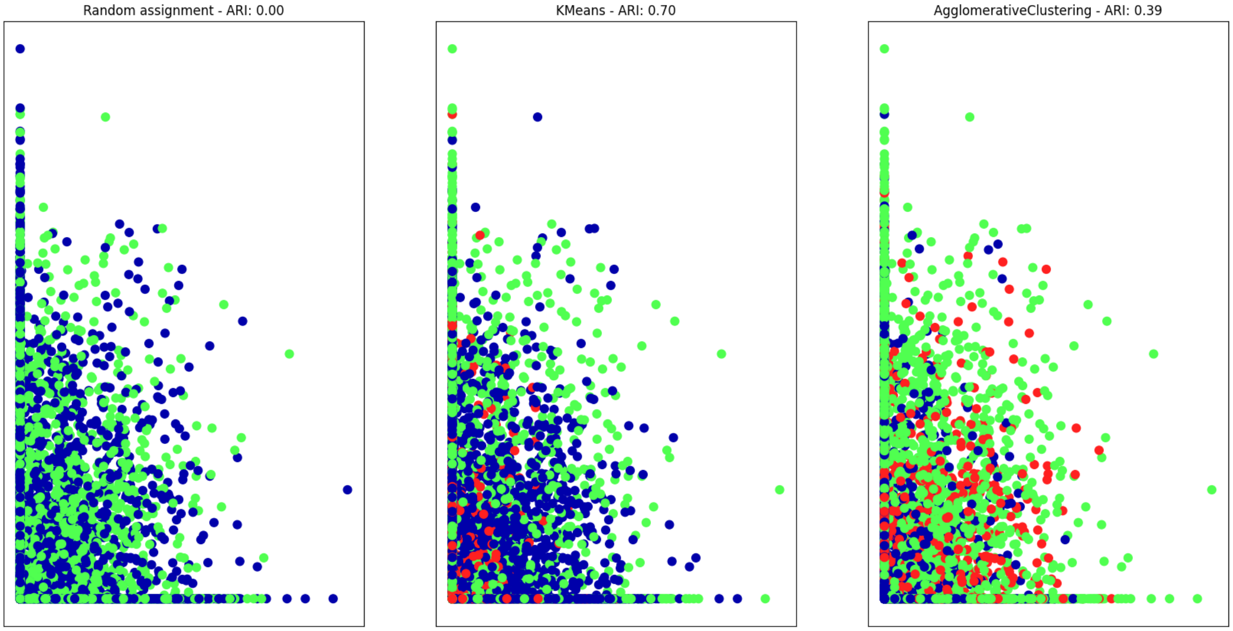


Figure 5.13: ARI Scores and Scatter Plots

Here we can see the adjusted rank index graphs and scores for K-Means and Agglomerative (both with their optimal parameters set). Agglomerative scores significantly worse here, and K-Means was only 2 points lower in silhouette scores. Taking these scores into consideration, it is likely to be the case that K-Means Clustering is the better choice for analyzing the provided dataset.