CSE 5243 Introduction to Data Mining

Lab 4: Cluster Analysis
(Executive Summary)

Bharat Suri (suri.40@osu.edu)
Nithin Senthil Kumar (senthilkumar.16@osu.edu)
Taught By: Michael Burkhardt
Date: 11/8/2019

Introduction

In this lab, different clustering algorithms were evaluated against three datasets with 2-dimensional features. The clusters were of different densities, shapes and distributions. The performance of each clustering algorithm was evaluated and their ability and inability to detect clusters of varying forms were observed and described. The clustering algorithms are:

1. Prototype based clustering - K-means (sklearn.cluster.KMeans)
2. Hierarchical clustering - Agglomerative Clustering (sklearn.cluster.AgglomerativeClustering)
3. Density-based clustering - DBScan (sklearn.cluster.DBSCAN)
4. Spectral Clustering (sklearn.cluster.SpectralClustering)

1. Exploratory Data Analysis

All 3 datasets used in this lab were synthetic. Therefore, it was clear from a standard EDA report that there was no need to modify the original features. The arrangement of data points in the original feature space was interesting and it was studied to identify the clusters. Following is a summary of the data exploration including checking for null values, missing values, and summary statistics along with the plots of the datasets.

- No values were missing
- All values were non-null
- 2 out of 3 datasets had well-formed clusters visible in the plot
- There were no outliers
- No data quality issues were found

1.1 Summary Statistics

```
=> Dataset 3

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399 entries, 0 to 398
Data columns (total 3 columns):
x        399 non-null float64
y        399 non-null float64
class    399 non-null int64
dtypes: float64(2), int64(1)
memory usage: 9.5 KB

=> Dataset 4

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 788 entries, 0 to 787
Data columns (total 3 columns):
x        788 non-null float64
y        788 non-null float64
class    788 non-null int64
dtypes: float64(2), int64(1)
memory usage: 18.6 KB

=> Dataset 5

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 3 columns):
x        500 non-null float64
y        500 non-null float64
class    500 non-null int64
dtypes: float64(2), int64(1)
memory usage: 11.8 KB
```

=> Dataset 3

	x	y	class
count	399.000000	399.000000	399.000000
mean	22.215038	13.970677	3.543860
std	9.736752	4.743516	1.581125
min	7.150000	5.750000	1.000000
25%	14.100000	9.775000	2.000000
50%	18.950000	14.150000	4.000000
75%	32.725000	18.125000	5.000000
max	42.900000	22.750000	6.000000

=> Dataset 4

	x	y	class
count	788.000000	788.000000	788.000000
mean	19.566815	14.171764	3.770305
std	9.922042	8.089683	1.596305
min	3.350000	1.950000	1.000000
25%	11.150000	7.037500	2.000000
50%	18.225000	11.725000	4.000000
75%	30.700000	21.962500	5.000000
max	36.550000	29.150000	7.000000

=> Dataset 5

	x	y	class
count	500.000000	500.000000	500.000000
mean	-1.762312	1.651048	0.998000
std	3.076568	3.566898	0.816903
min	-7.779000	-8.029000	0.000000
25%	-4.657750	-0.872750	0.000000
50%	-1.232000	0.942500	1.000000
75%	-0.174500	5.706500	2.000000
max	7.674000	7.083000	2.000000

1.2 Visualizations

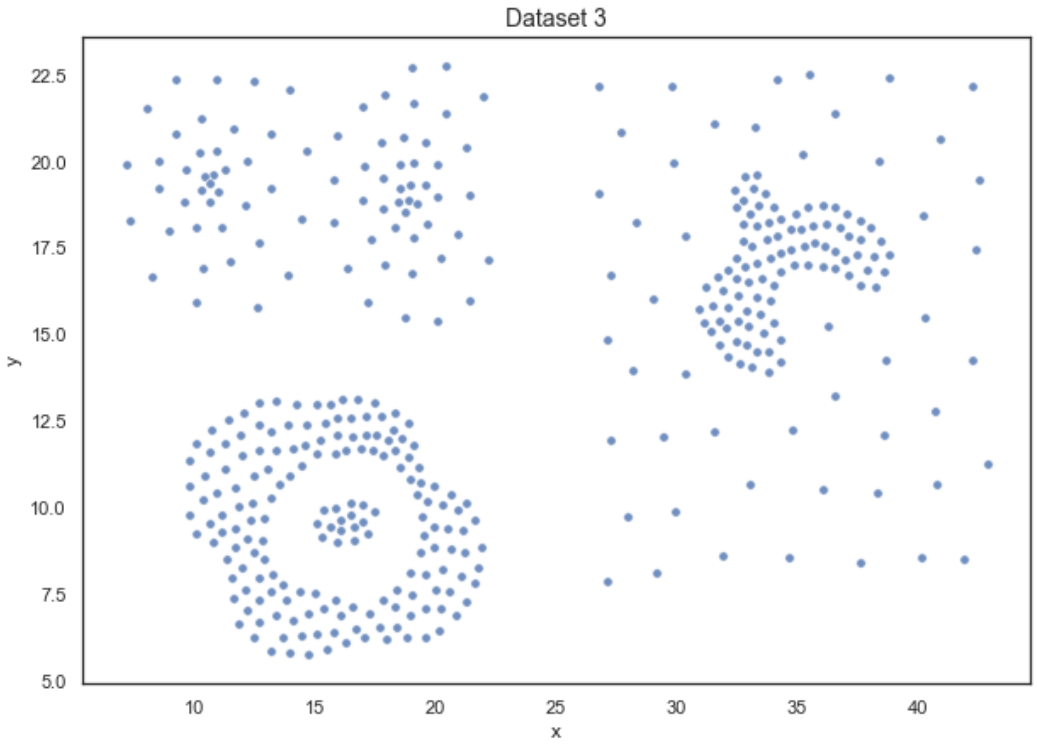


Figure 1.1 - Spread of the data in Dataset 3

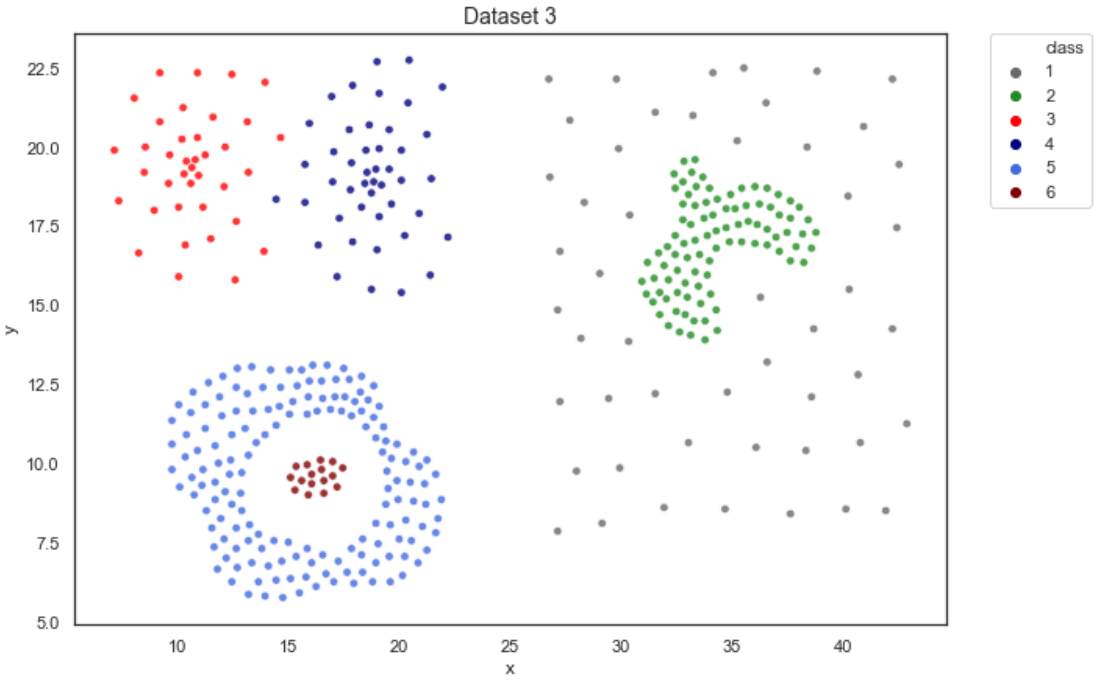


Figure 1.2 - Visualization of the 6 clusters in Dataset 3

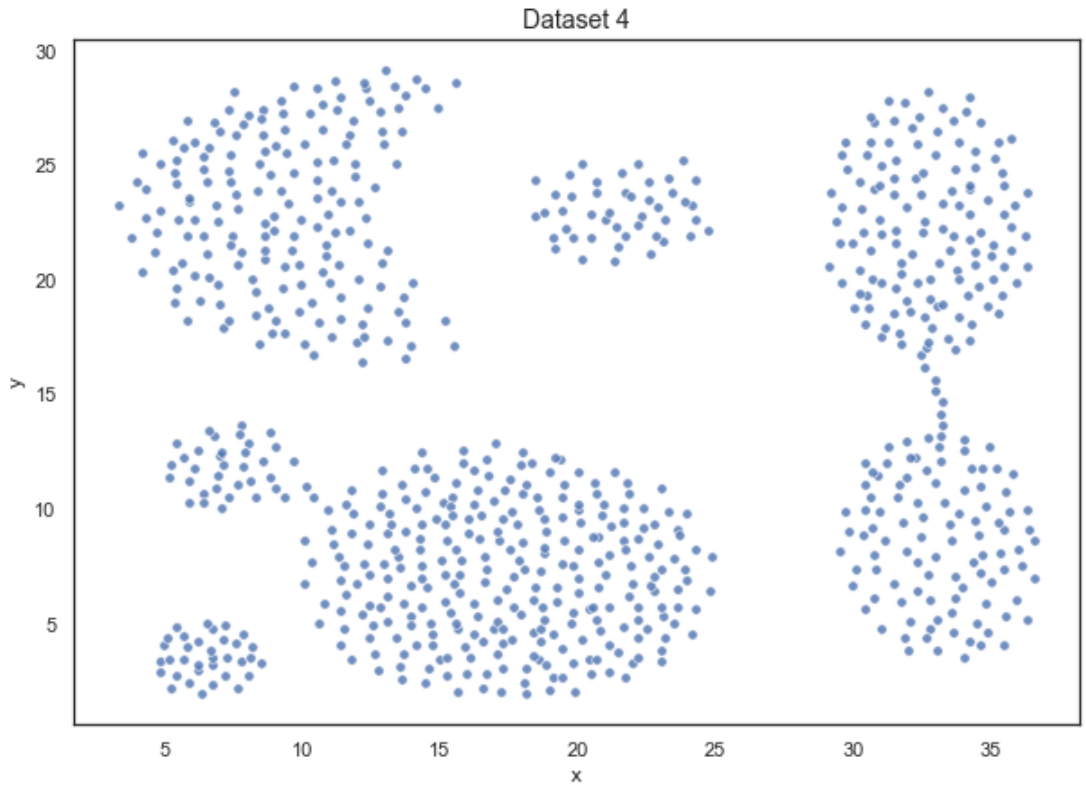


Figure 1.3 - Spread of the data in Dataset 4

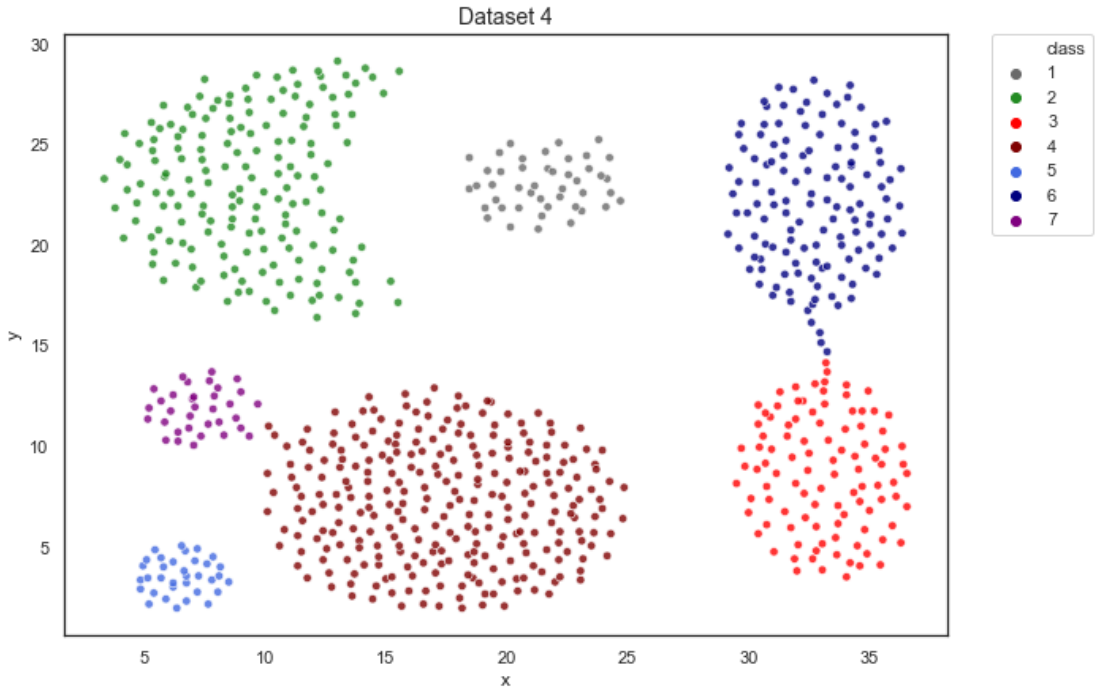


Figure 1.4 - Visualization of the 7 clusters in Dataset 4

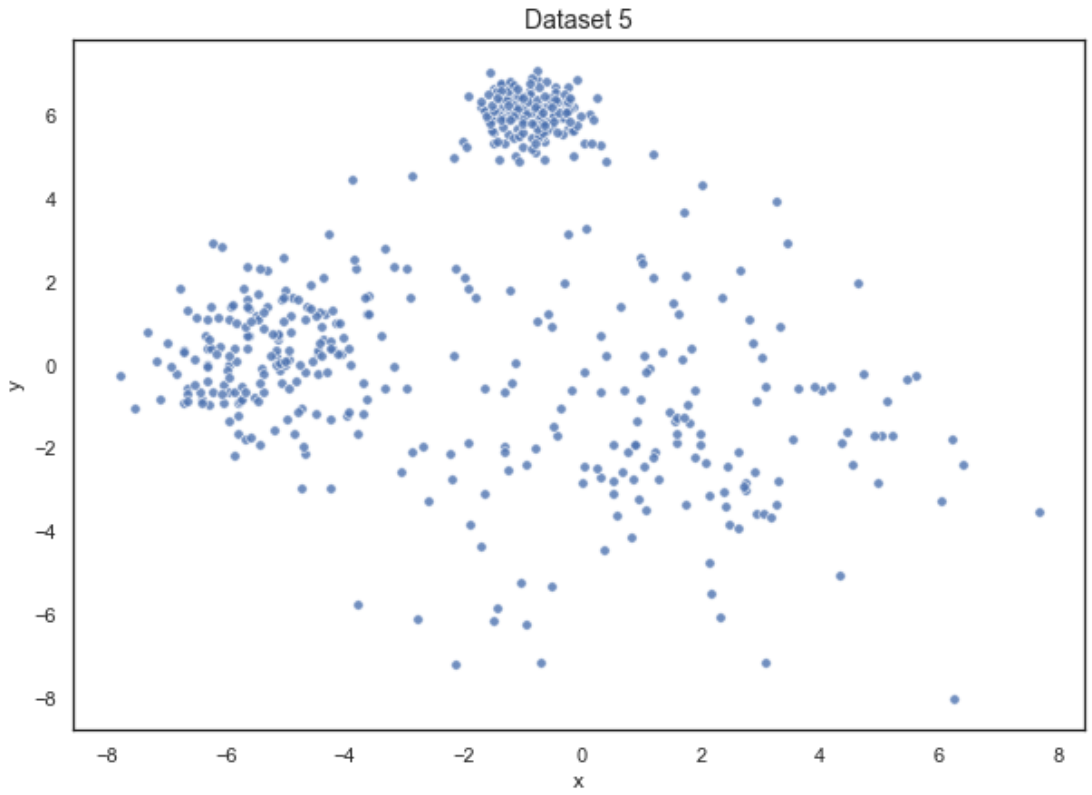


Figure 1.5 - Spread of the data in Dataset 5

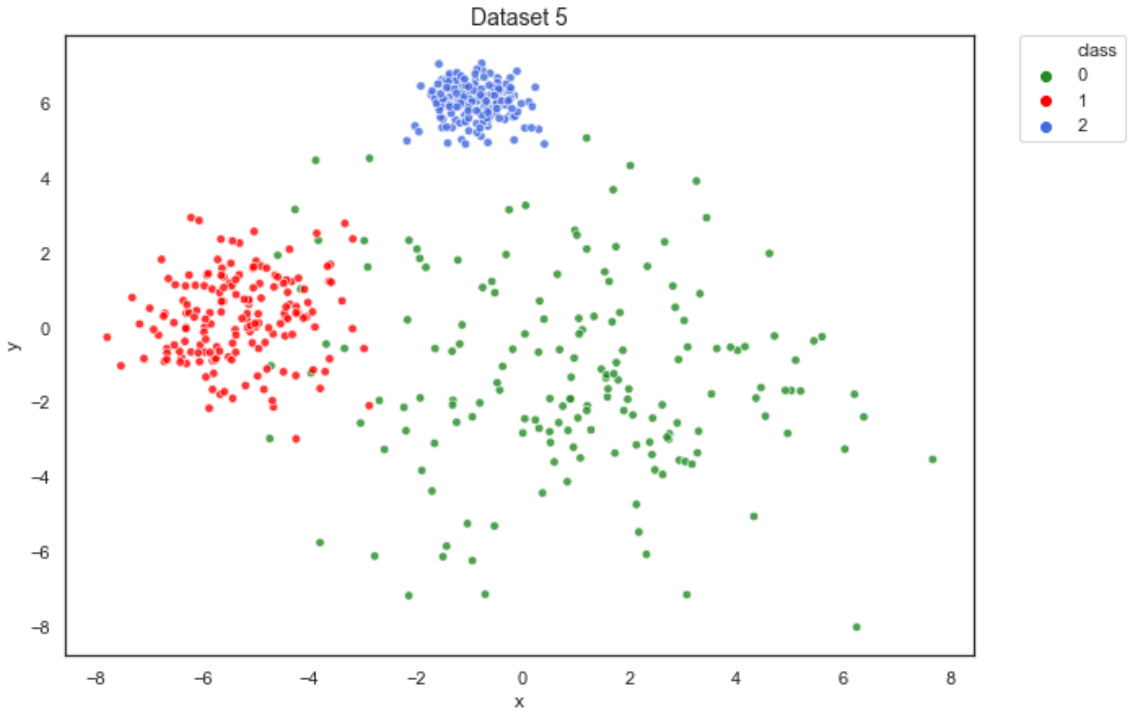


Figure 1.6 - Visualization of the 3 clusters in Dataset 5

2. Data Preprocessing

After performing the initial analysis, it was clear that the data was clean. Thus, no additional measures were taken to change the original features. No transformations were used. There was no addition of features. The main reason behind this was that from the plots, well-formed clusters were observed. Furthermore, the range of values was comparable and thus scaling was not required. The main reason behind this decision was that the arrangement of data points in the original feature space was interesting when observed by plotting them.

3. Finding the clusters

3.1 Prototype based clustering - k-Means

Since the dataset mentions the number of clusters that exist, we use this value for k for each dataset while finding the clusters. The datasets are characterized by number of points - N, number of clusters - k, number of features - D.

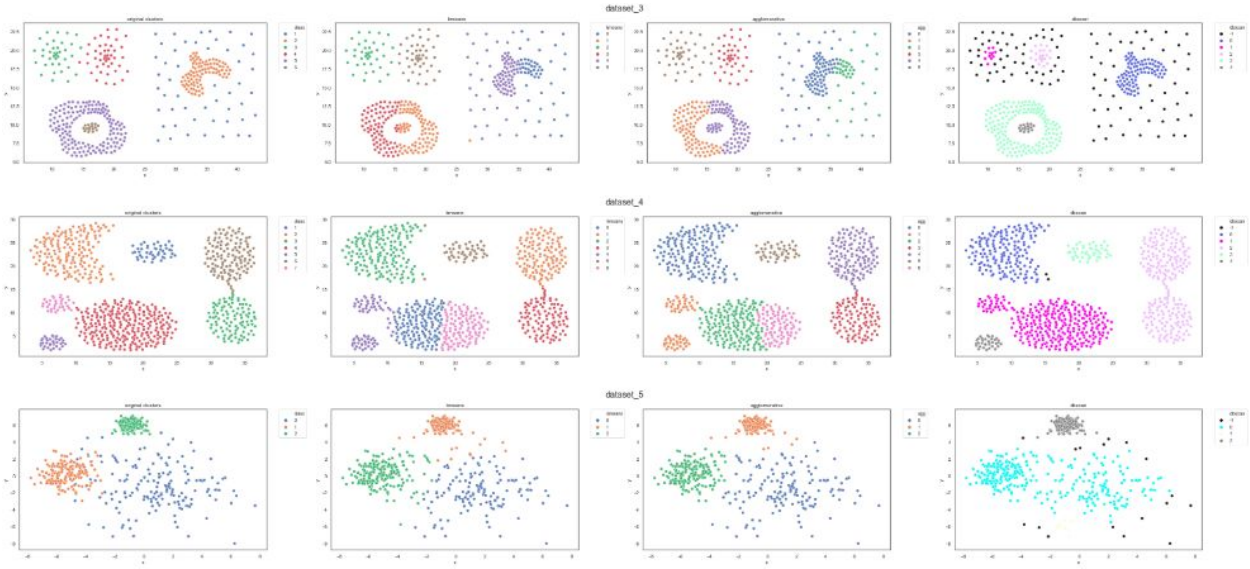


Figure 1.7.0 - Visual Representation of clusters in each dataset identified

Figure 1.7.0 gives an overview of the results obtained. The first column contains the visualization of the original datasets. The second, third and fourth column shows how the datasets' clusters were identified using kMeans, agglomerative clustering and DBscan respectively. The first row is of dataset 3, second row - dataset 4, third row - dataset 5. In the sections below, each of these graphs are shown along with the parameters and hyperparameters provided to the respective clustering algorithm.

Dataset 3 (cluster_ds3.csv): N=399, k=6, D=2

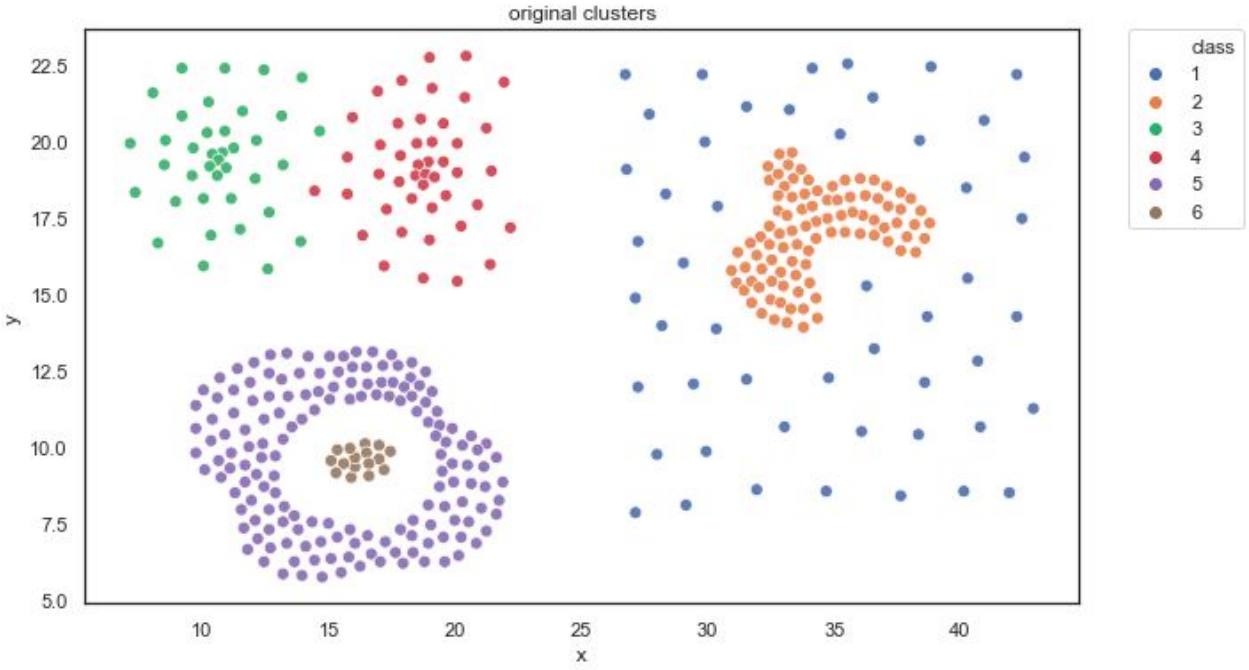


Figure 1.7 - Original clusters for reference (Dataset 3)

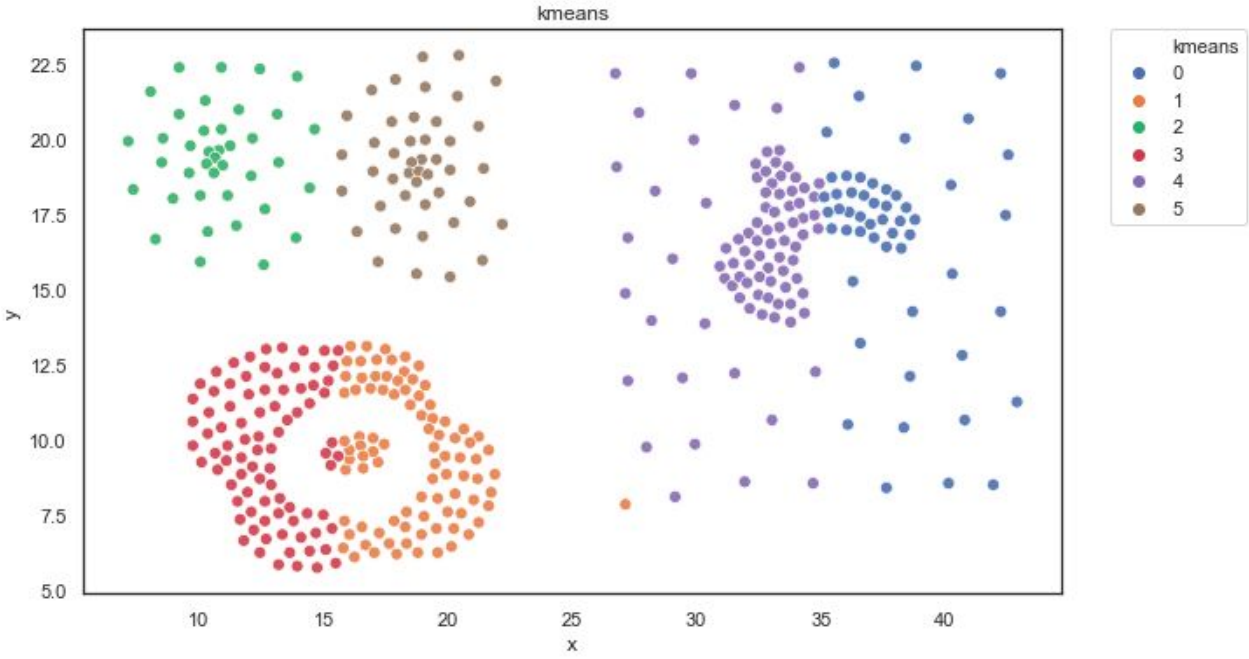


Figure 1.8 - Visualization of the 6 Clusters detected by kMeans (Dataset 3)
Parameters to the function - KMeans(n_clusters=6, n_init=50)

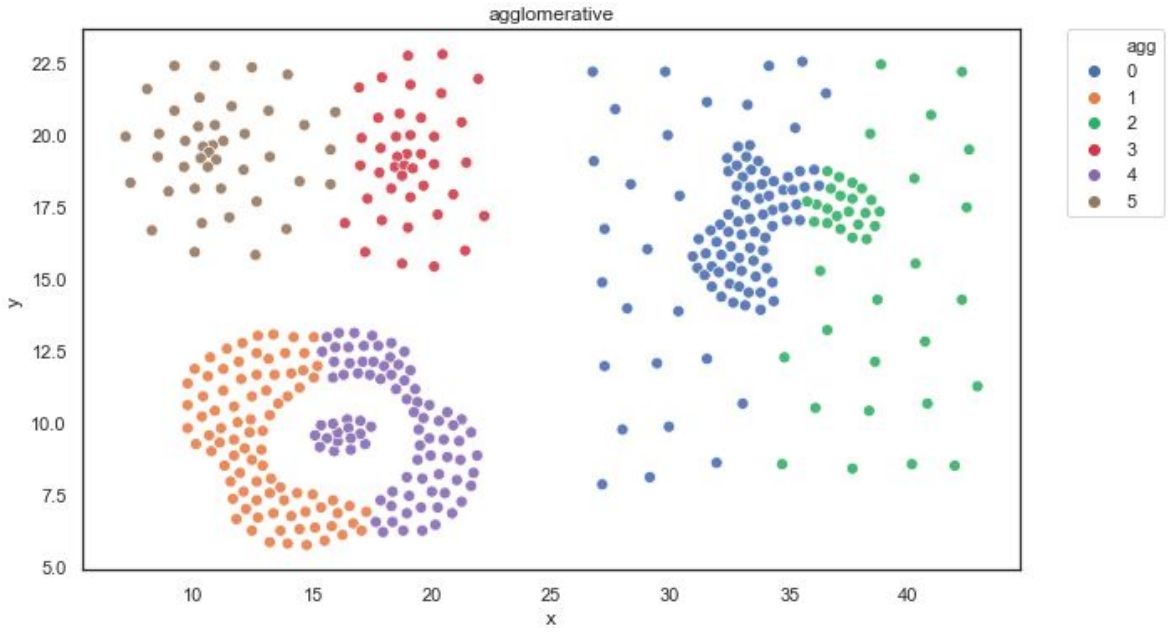


Figure 1.9 - Visualization of the 6 Clusters detected by Agglomerative Clustering (Dataset 3)
Parameters to the function - AgglomerativeClustering(n_clusters=6)

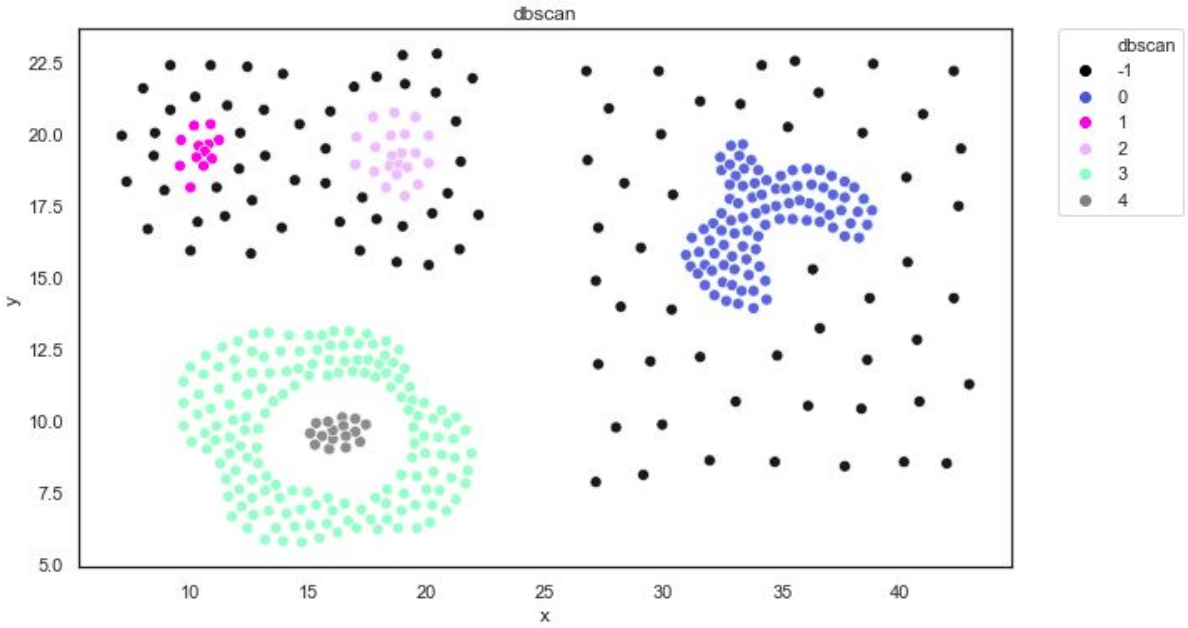


Figure 1.10 - Visualization of the 5 Clusters detected by DBScan (Dataset 3)
Parameters to the function - AgglomerativeClustering(n_clusters=6)

- N_clusters was taken as 6 for both k-Means and Agglomerative clustering. Other hyperparameters were left as default.
- For DBSCAN, min_eps was taken as 0.912 and min_samples as 4

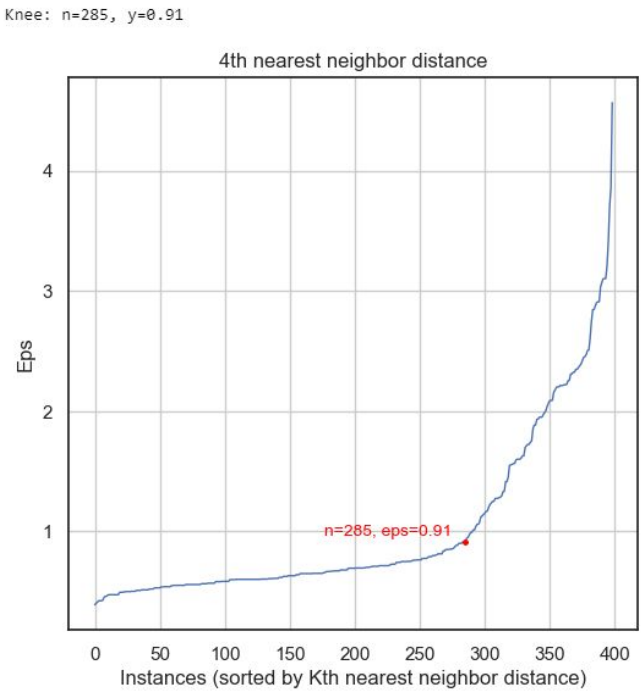


Figure 1.10.1 - Computing the minimum epsilon value for DBscan with kth Nearest Neighbors

Dataset 4 (cluster_ds4.csv): N=788, k=7, D=2

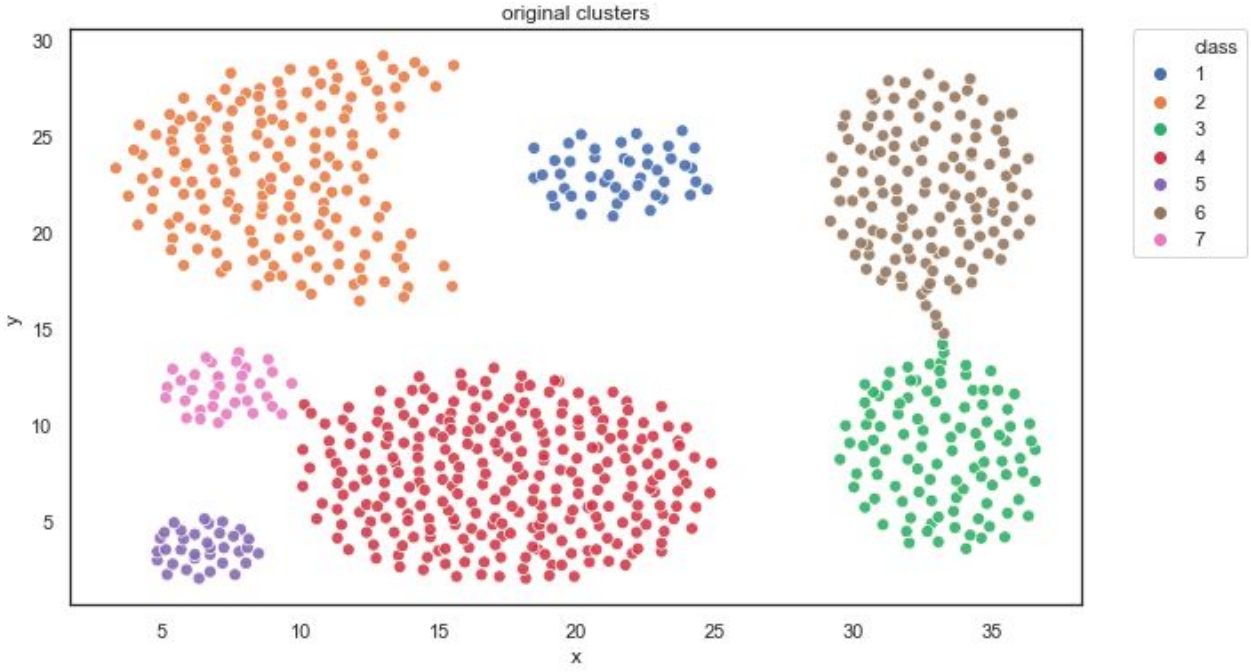


Figure 1.11 - Visualization of the original clusters (Dataset 4)

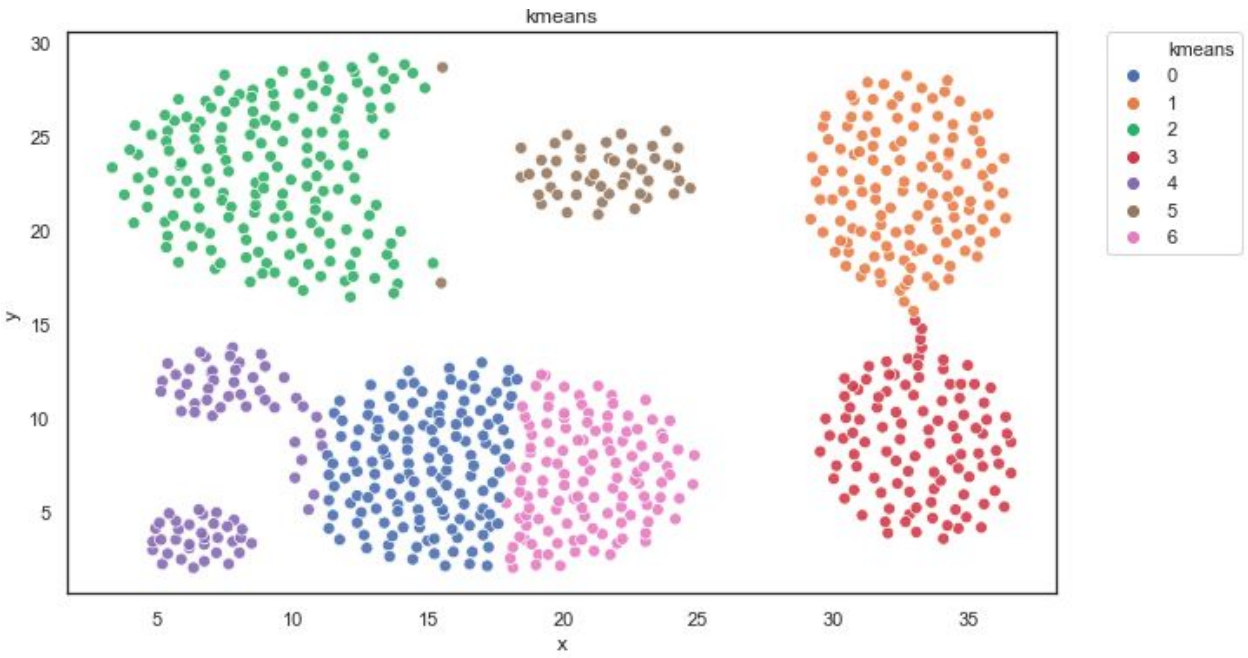


Figure 1.12 - Visualization of the 6 Clusters detected by kMeans (Dataset 4)

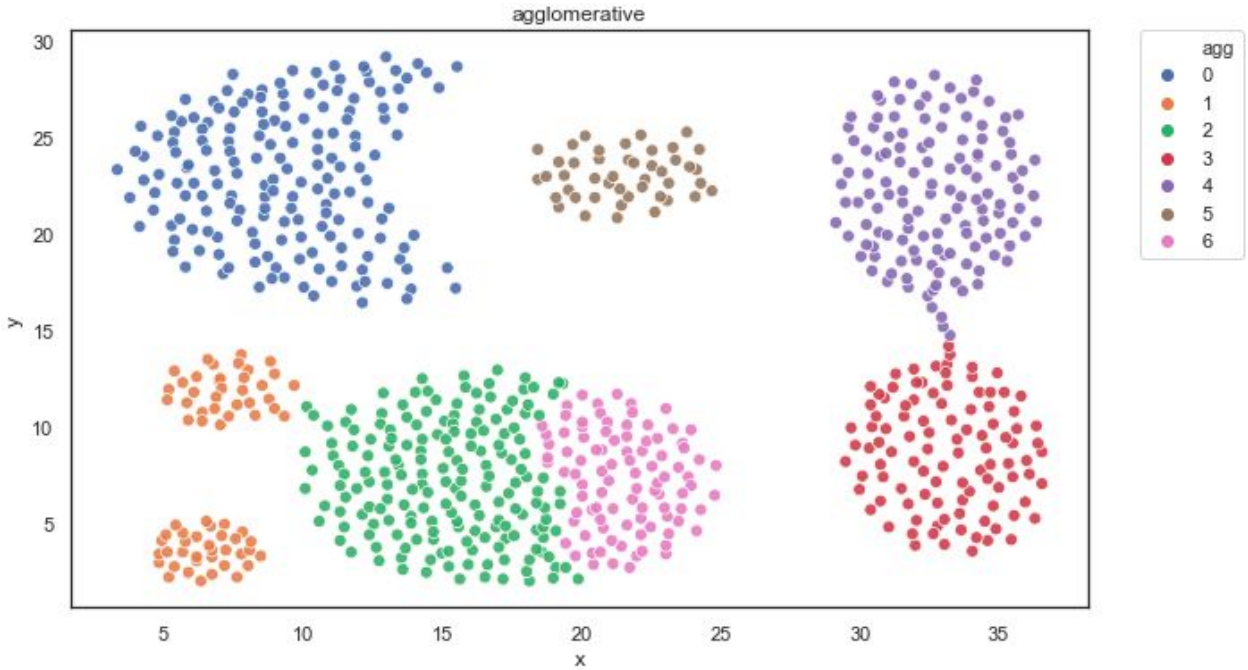


Figure 1.13 - Visualization of the 7 Clusters detected by Agglomerative Clustering (Dataset 4)

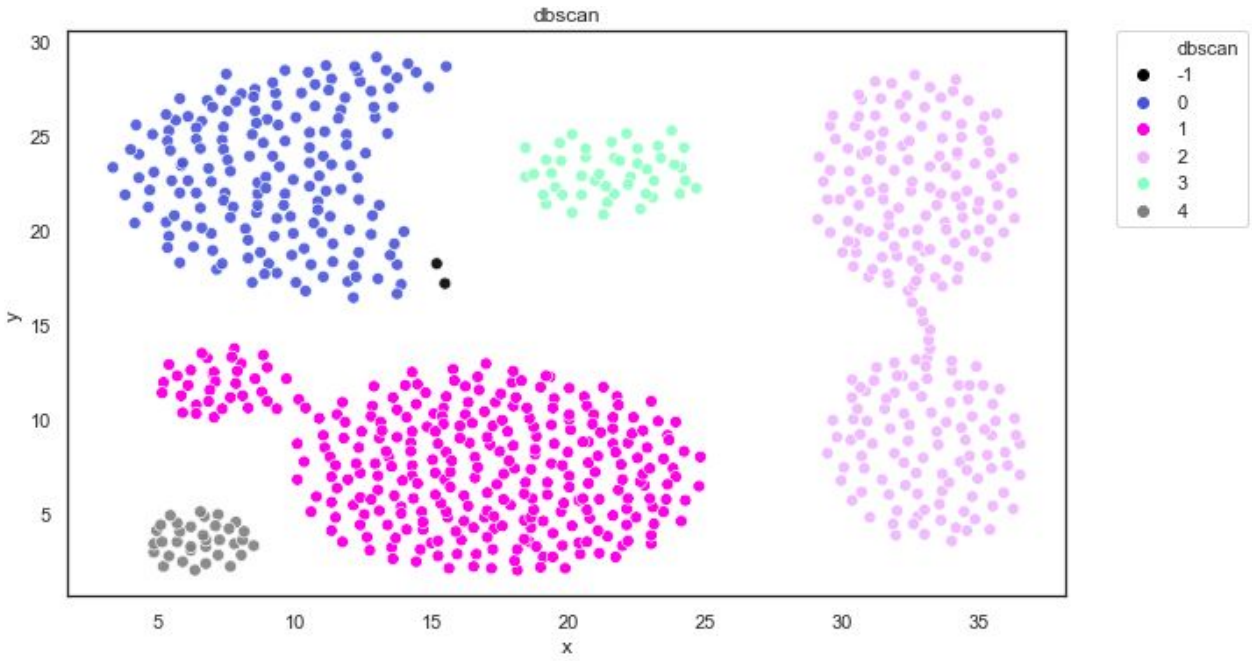


Figure 1.14 - Visualization of the 5 Clusters detected by DBScan (Dataset 4)

- N_clusters was taken as 7 for both k-Means and Agglomerative clustering. Other hyperparameters were left as default.
- For DBSCAN, min_eps was taken as 1.15 and min_samples as 4

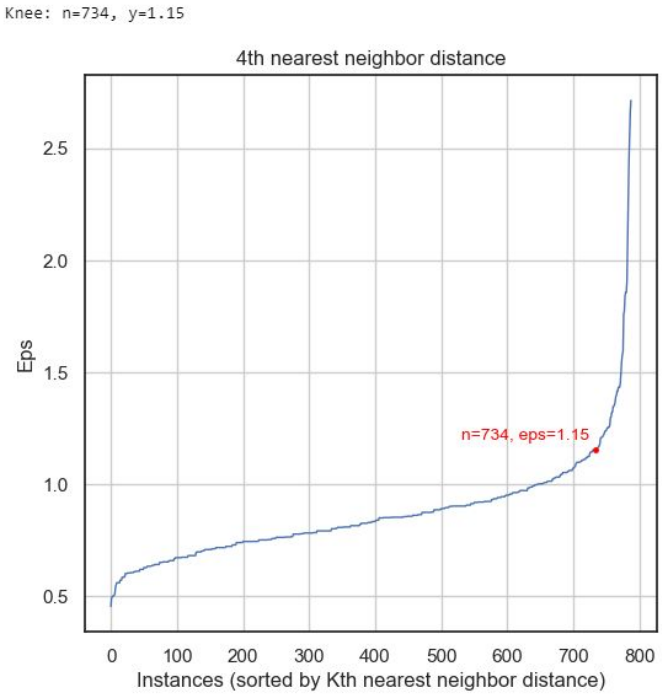


Figure 1.14.1 - Computing the minimum epsilon value for DBscan with kth Nearest Neighbors

Dataset 5 (cluster_ds5.csv): N=500, k=3, D=2

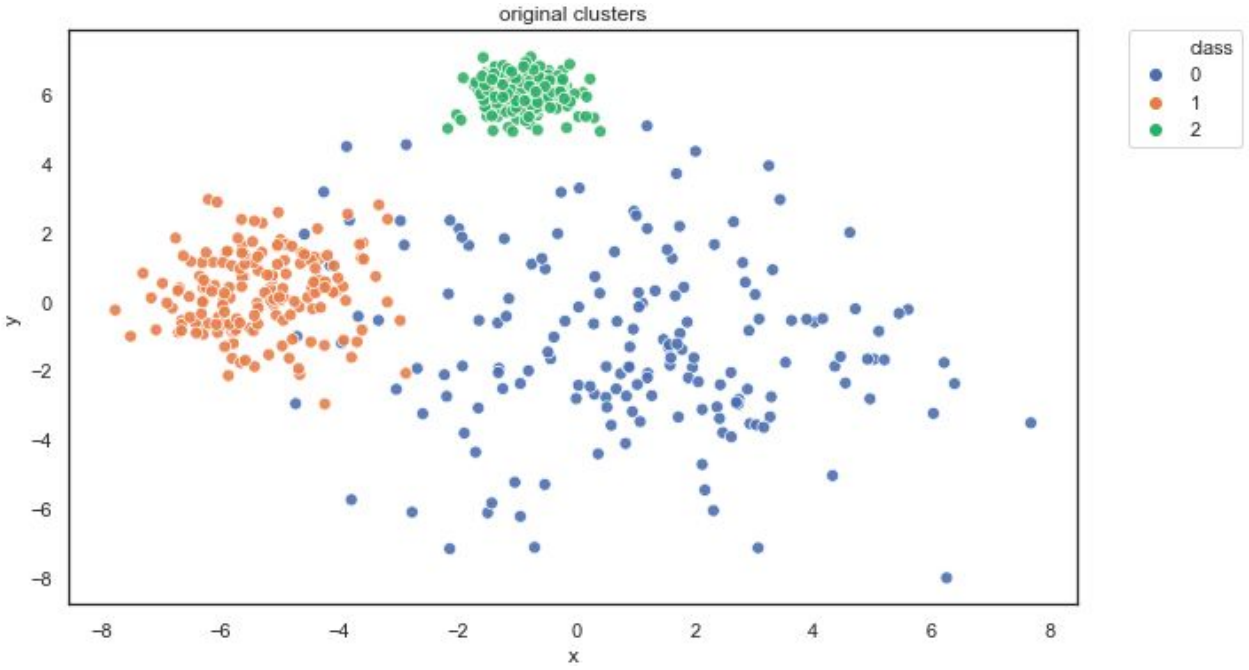


Figure 1.15 - Visualization of the original clusters (Dataset 5)

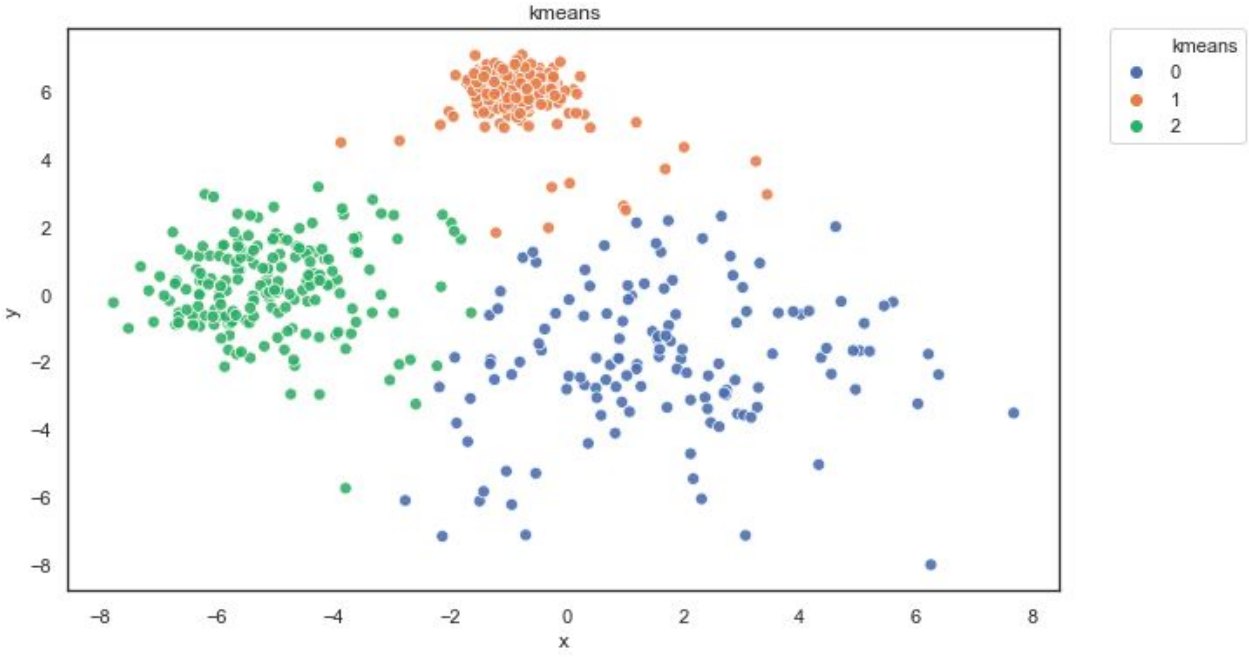


Figure 1.16 - Visualization of the 3 Clusters detected by kMeans (Dataset 4)

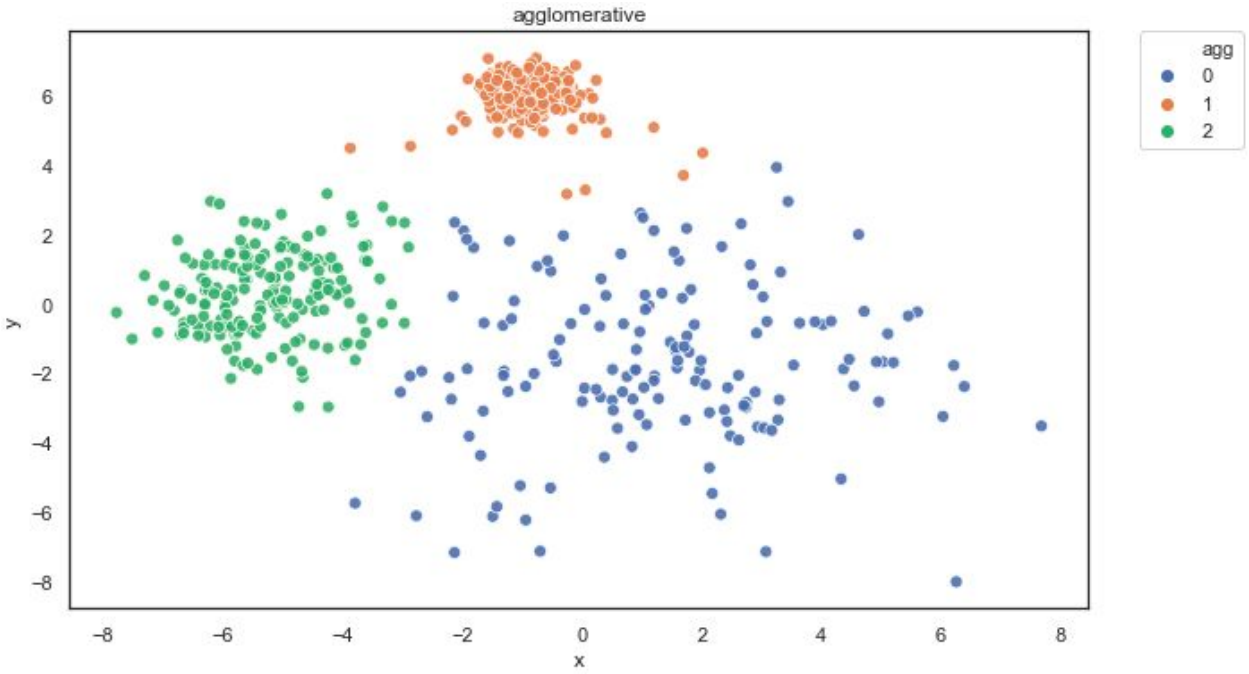


Figure 1.17 - Visualization of the 3 Clusters detected by Agglomerative Clustering (Dataset 5)

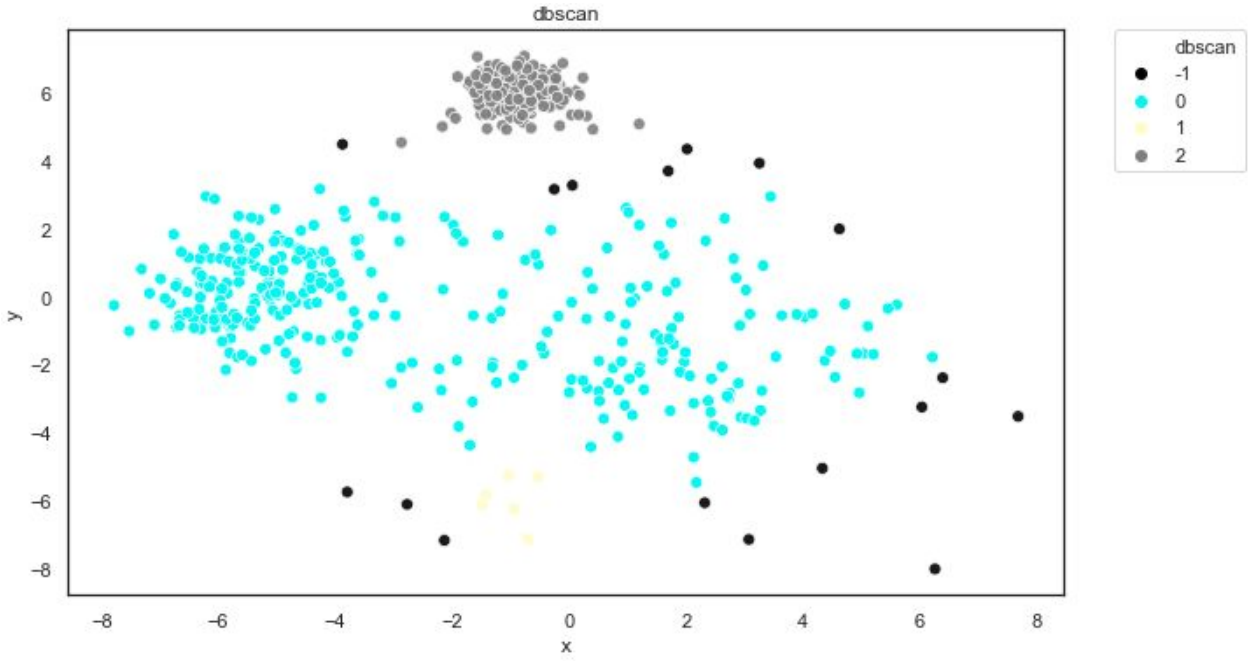


Figure 1.18 - Visualization of the 3 Clusters detected by DBScan (Dataset 5)

- N_clusters was taken as 3 for both k-Means and Agglomerative clustering. Other hyperparameters were left as default.
- For DBSCAN, min_eps was computed as 1.077 and min_samples as 4

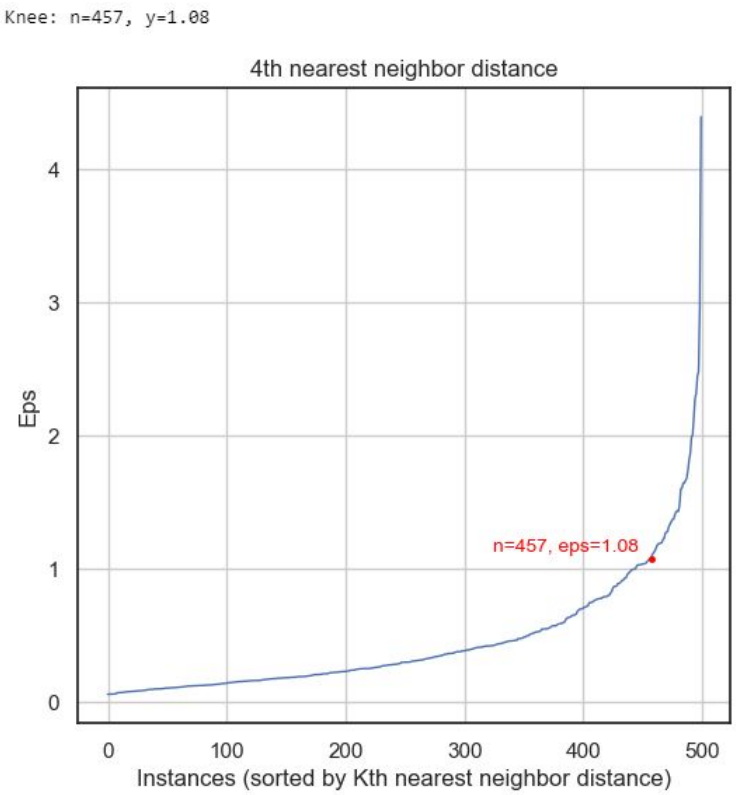


Figure 1.18.1 - Computing the minimum epsilon value for DBscan with kth Nearest Neighbors

In this section, default hyperparameters were used to find the clusters in order to compare the baseline performance of each algorithm on each of the datasets and evaluate the performance based on these.

4. Evaluating the performance of the three clustering algorithms

In this section, each of the three clustering algorithm's performance is evaluated both visually and quantitatively. The quantitative performance evaluation measures used are silhouette plots and accuracy.

4.1 Visual analysis

Dataset 3

From Figures 1.8 and 1.9, we see that the performance of k-Means and Agglomerative clustering was very similar and identified clusters of similar patterns. Since k-Means and Agglomerative clustering are well-suited for clusters with a circular shape, both algorithms identify the single irregularly shaped cluster as two different clusters, each with its own centroid. In Figure 1.9, it is observed that Agglomerative clustering is sensitive to outliers and includes these outliers in clusters 0 and 2. From Figure 1.10, we observe that DBSCAN identified dense clusters very well but the points far from each other were misidentified as noise points.

Dataset 4

Since all the clusters appear to have similar densities, DBSCAN works comparatively better with this dataset than dataset 3, as illustrated in Figure 1.12. For certain clusters in this dataset, k-Means and Agglomerative outperformed DBSCAN because of what appears to be a bridge of points of similar density between the two clusters. These bridges made DBSCAN identify 2 clusters connected by the bridge as one and the same. Overall performance of k-Means and Agglomerative was again very similar based on visual inspection. DBSCAN with the chosen parameters fails to identify the right number of clusters and requires further optimization.

Dataset 5

Since all clusters had varying densities in this dataset, DBSCAN performs poorly as seen from Fig. 1.18 for its default parameters except for epsilon and minimum samples. Performance of k-Means and Agglomerative, as seen in Fig. 1.16 and 1.17, were much better and quite similar to each other. This is because although the points are spread out they share a common centroid in kMeans as shown in Figure 1.19 below:

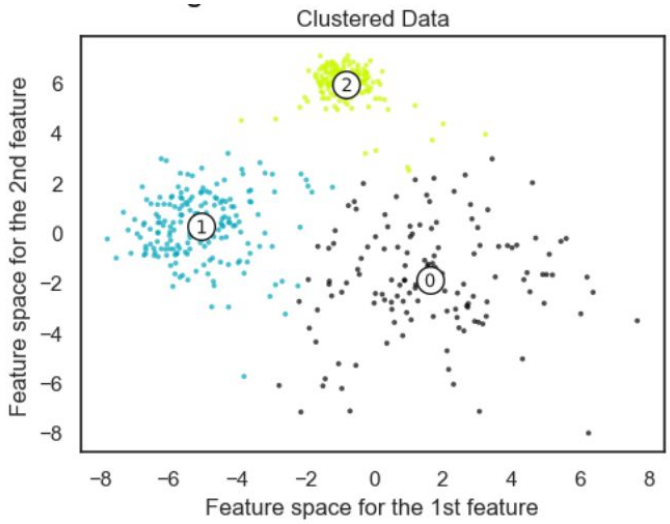


Figure 1.19 - Centroids for each of the clusters in Dataset 5

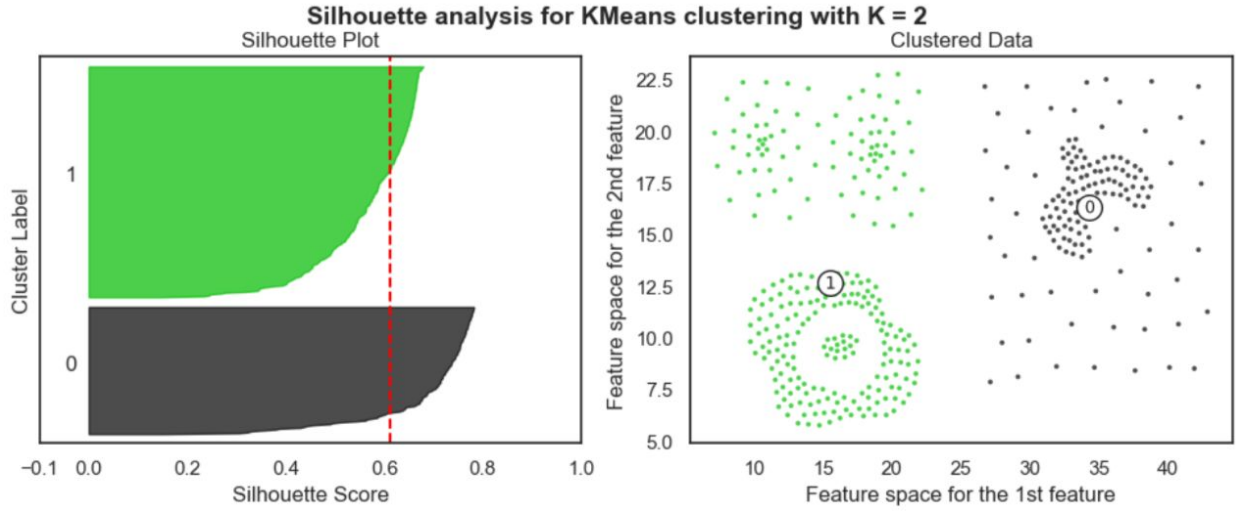
4.2 Silhouette Analysis

Silhouette analysis was employed to determine the number of clusters in k-means analysis. Silhouette analysis can also be used to study the separation distance between the resulting clusters.

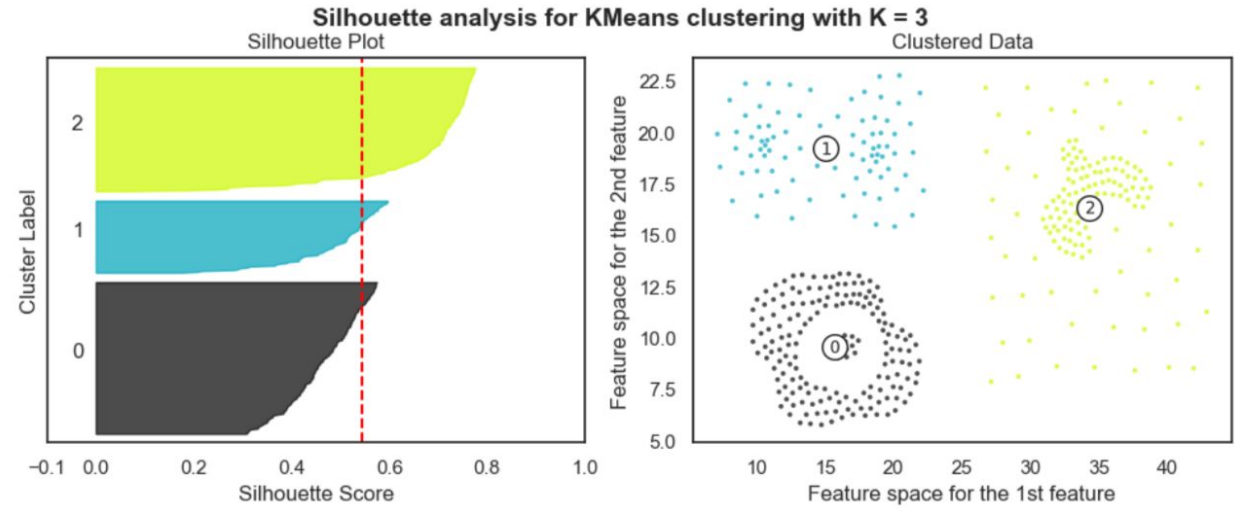
Dataset 3

For dataset 3, the outputs from the silhouette analysis are ambivalent because of the irregular shape of clusters. kMeans inability to distinguish nested clusters, eg. ring like clusters and those with varying densities, is visible from the silhouette analysis. This is because though silhouette plots with K=4 or 5 appear to indicate optimal values of k, visual analysis shows that kMeans cannot distinguish the clusters correctly regardless of the value of K.

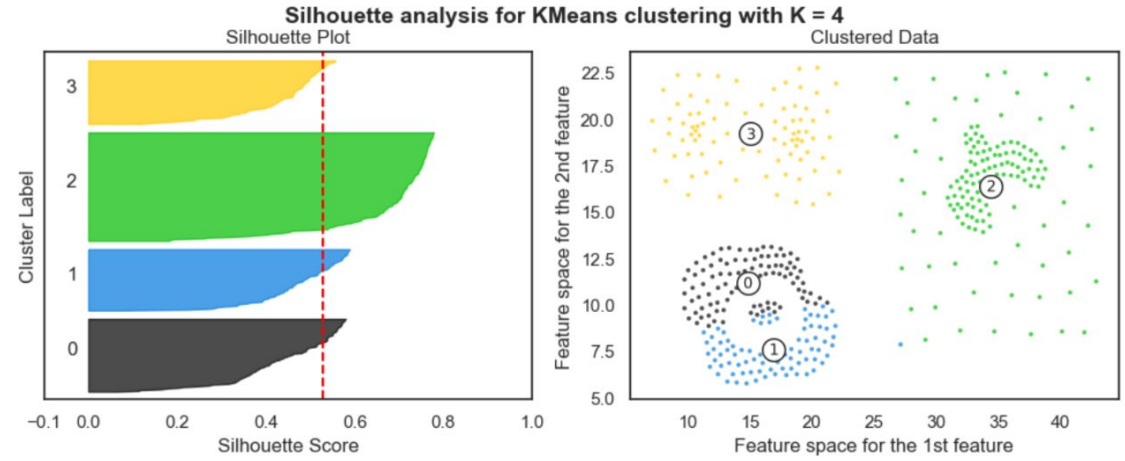
```
For K = 2, the average silhouette score is 0.611
```



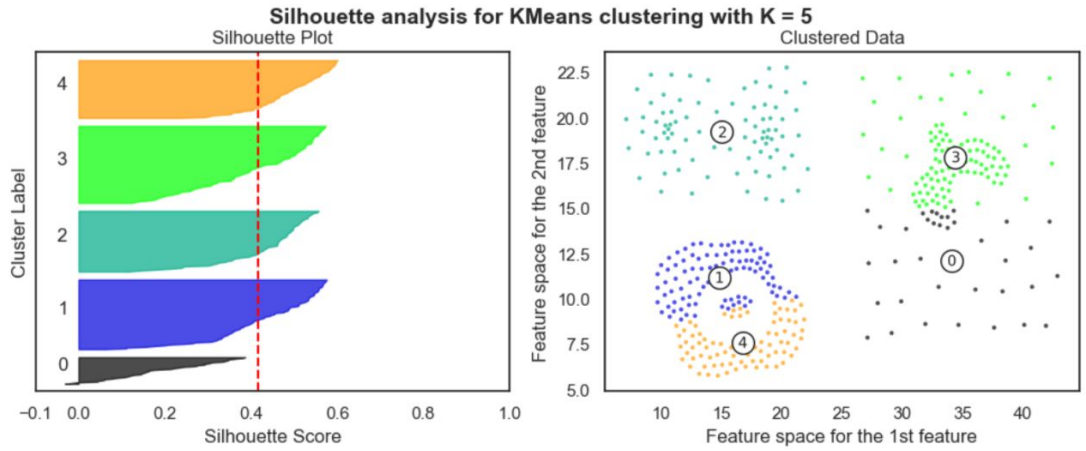

```
For K = 3, the average silhouette score is 0.543
```



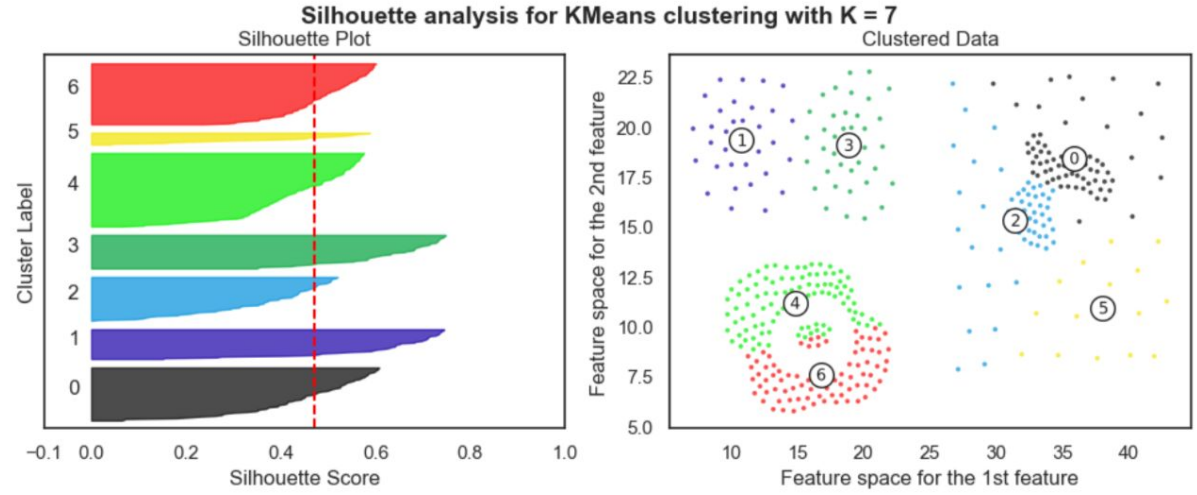

```
For K = 4, the average silhouette score is 0.527
```



```
For K = 5, the average silhouette score is 0.416
```



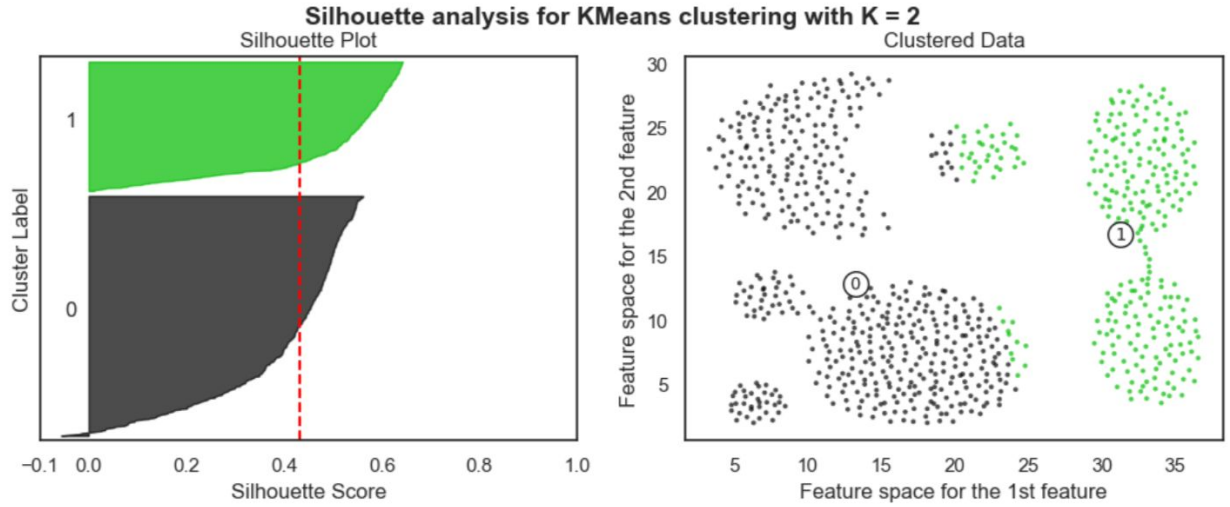

```
For K = 7, the average silhouette score is 0.470
```



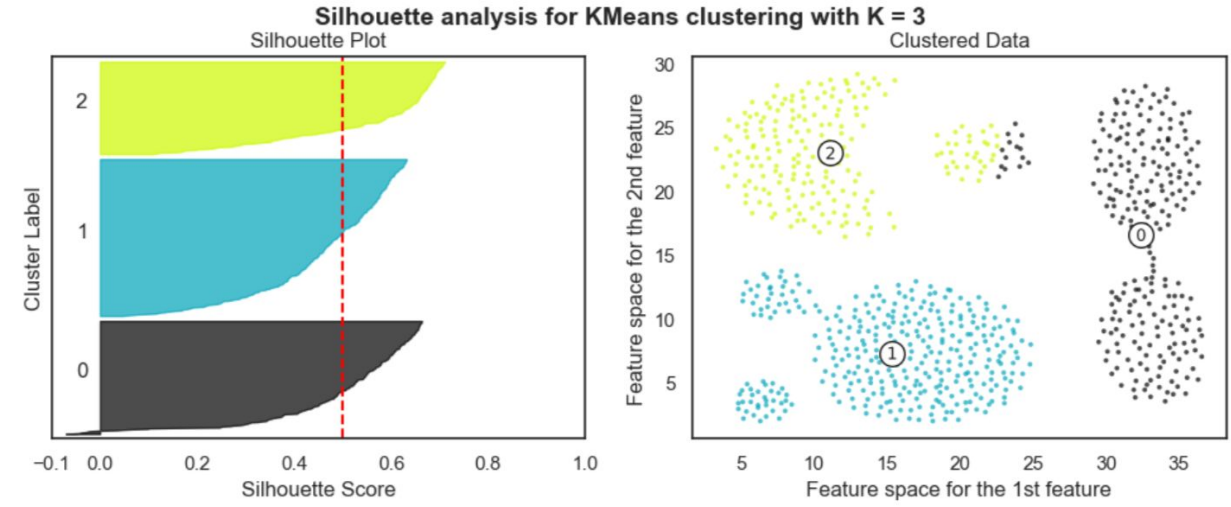
Dataset 4

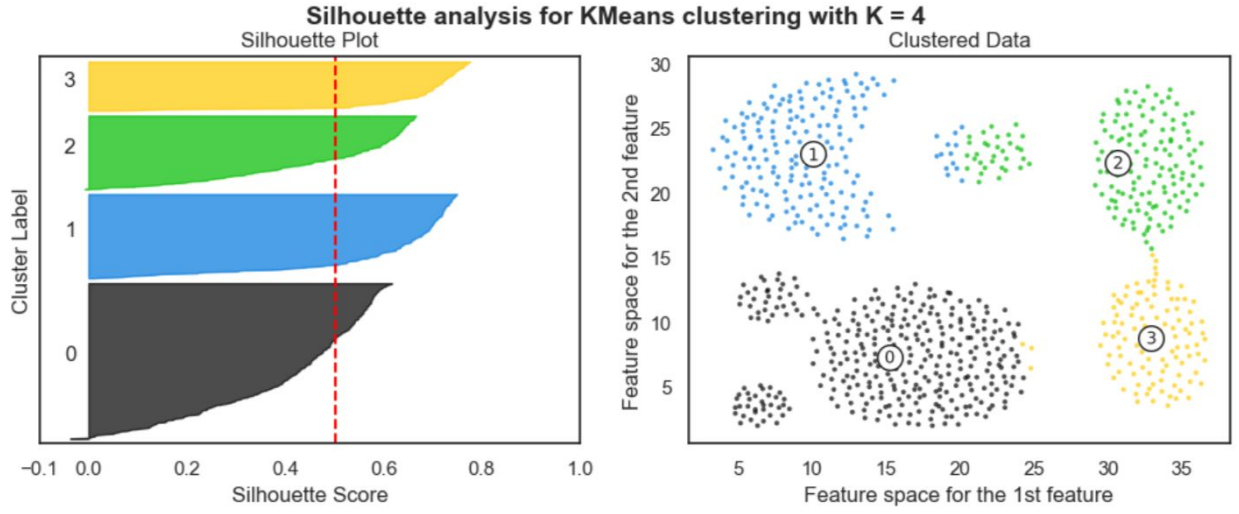
The average silhouette score is about 0.5 for all values of K. From the silhouette plots, K values of 4, 5 and 6 can be considered. Since the silhouette plots for K=5 and 6 have samples more or less equally distributed in width, they are preferred choices of K over 4. When K=7, from the plots, we observe that class 3 has a value below the average silhouette score, indicating that these points were likely misclassified.

```
For K = 2, the average silhouette score is 0.432
```

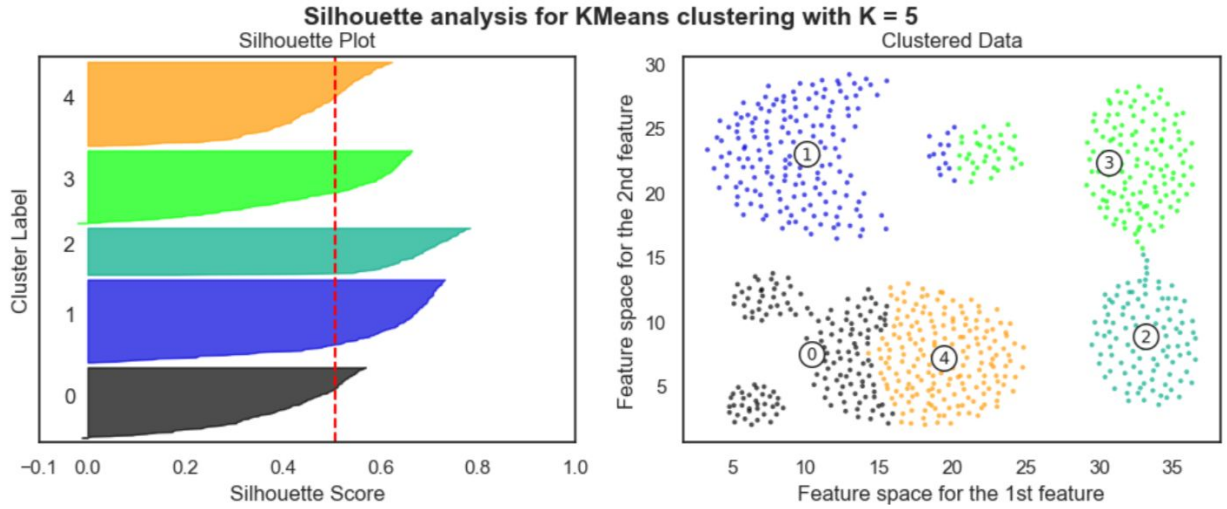



```
For K = 3, the average silhouette score is 0.500
```

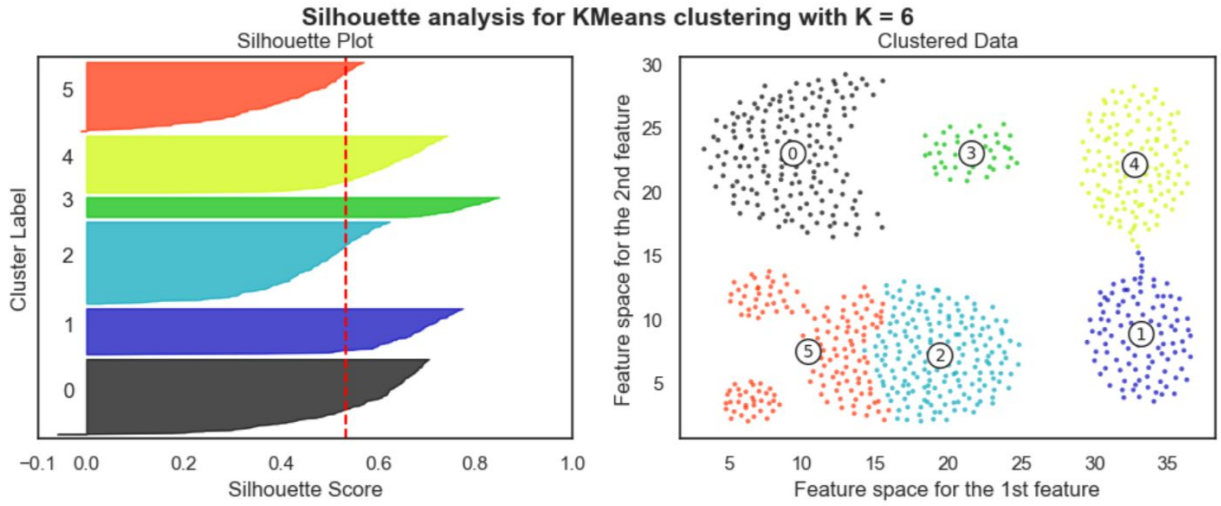




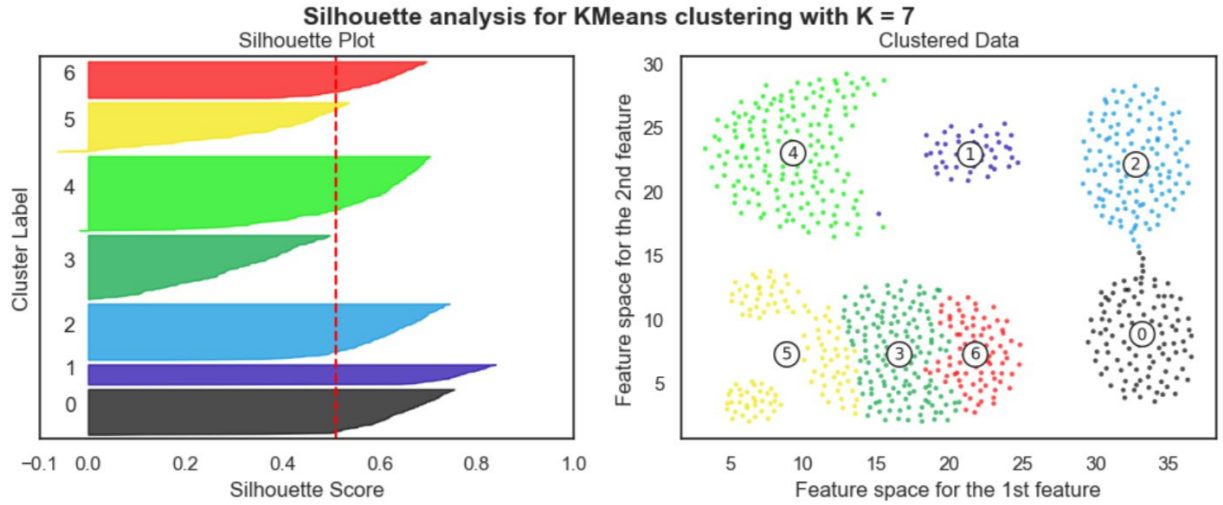
```
For K = 5, the average silhouette score is 0.506
```




```
For K = 6, the average silhouette score is 0.533
```



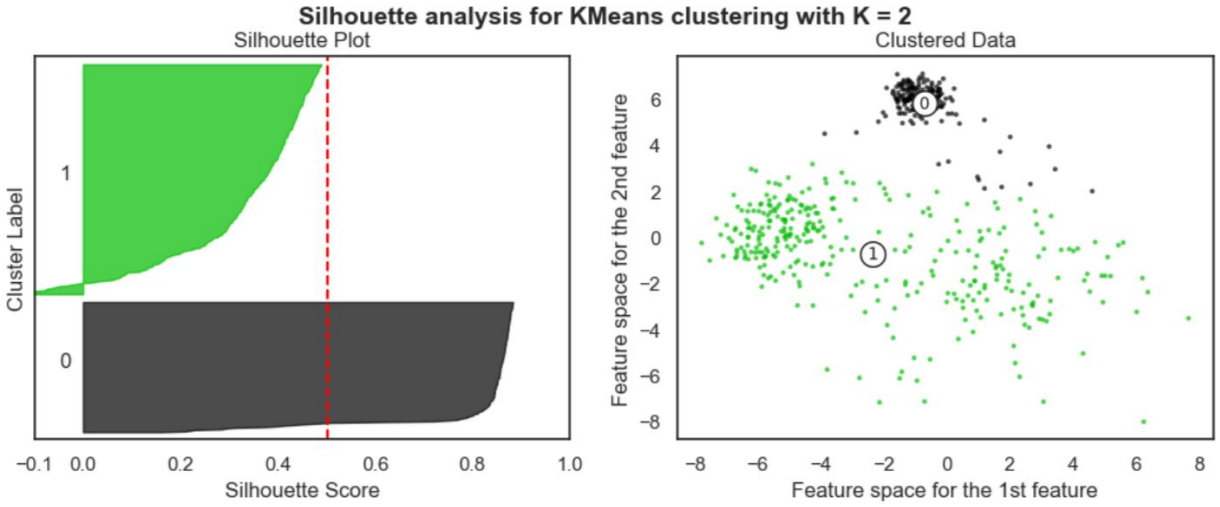
```
For K = 7, the average silhouette score is 0.511
```



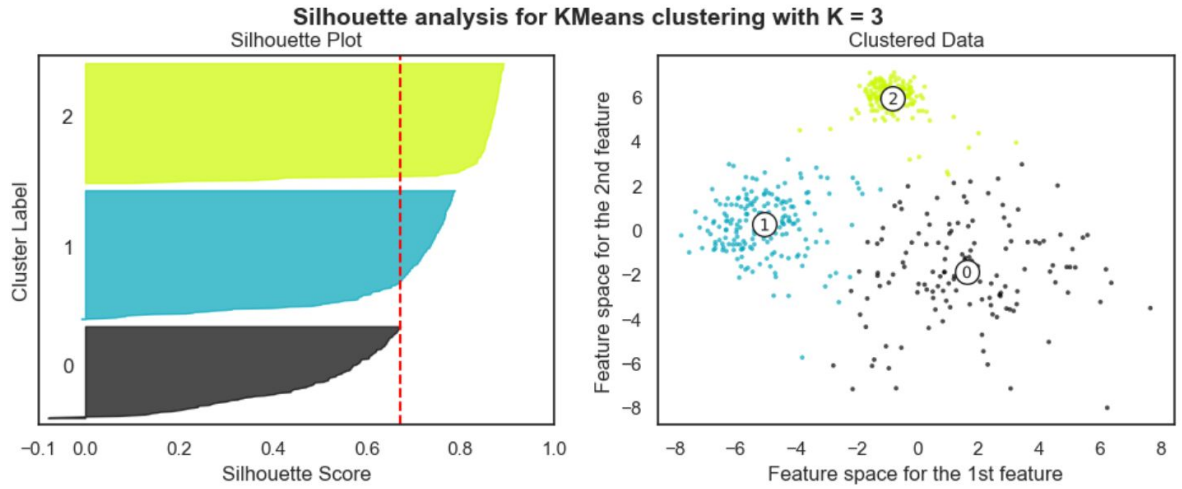
Dataset 5

From the silhouette plots of dataset 5, ideal value of K is 3 because the average silhouette score is high and for each class, most of the silhouette score is above the average silhouette value.

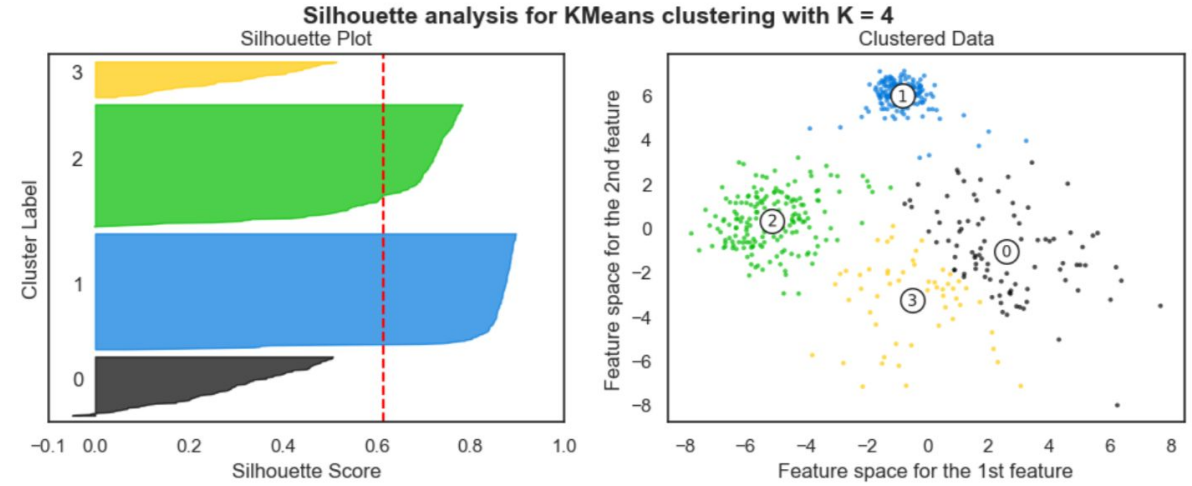
```
For K = 2, the average silhouette score is 0.503
```



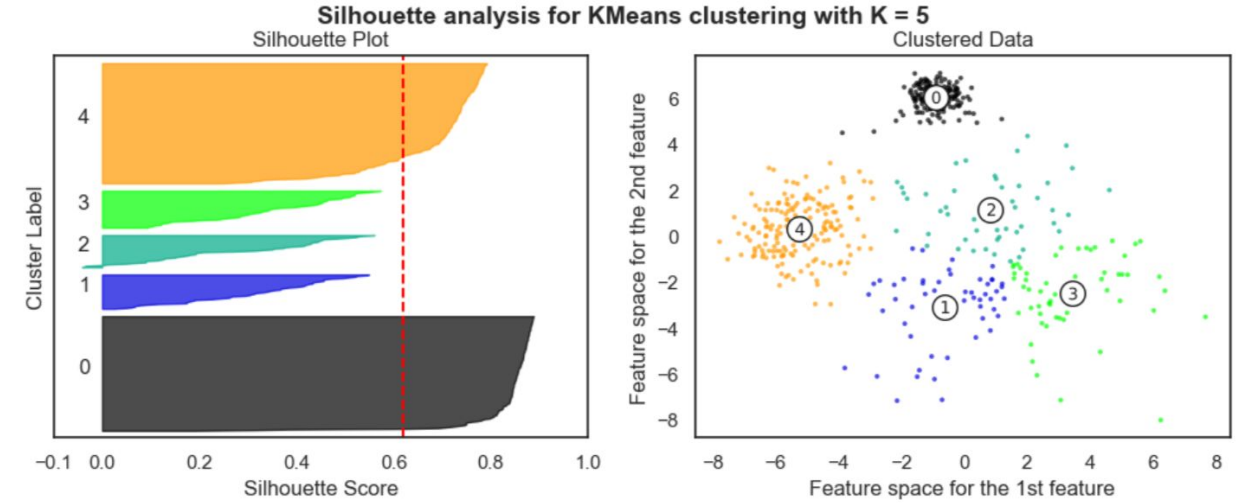
```
For K = 3, the average silhouette score is 0.672
```




```
For K = 4, the average silhouette score is 0.615
```




```
For K = 5, the average silhouette score is 0.620
```

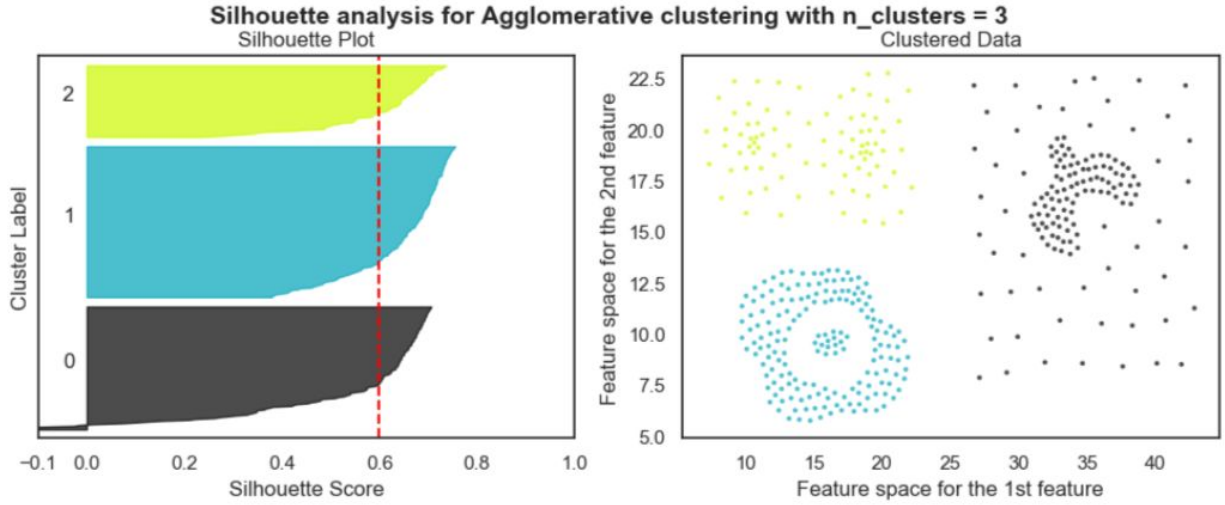


Agglomerative Clustering

Only the best silhouette plots are shown for each of the datasets which indicate the values for hyperparameter n_clusters

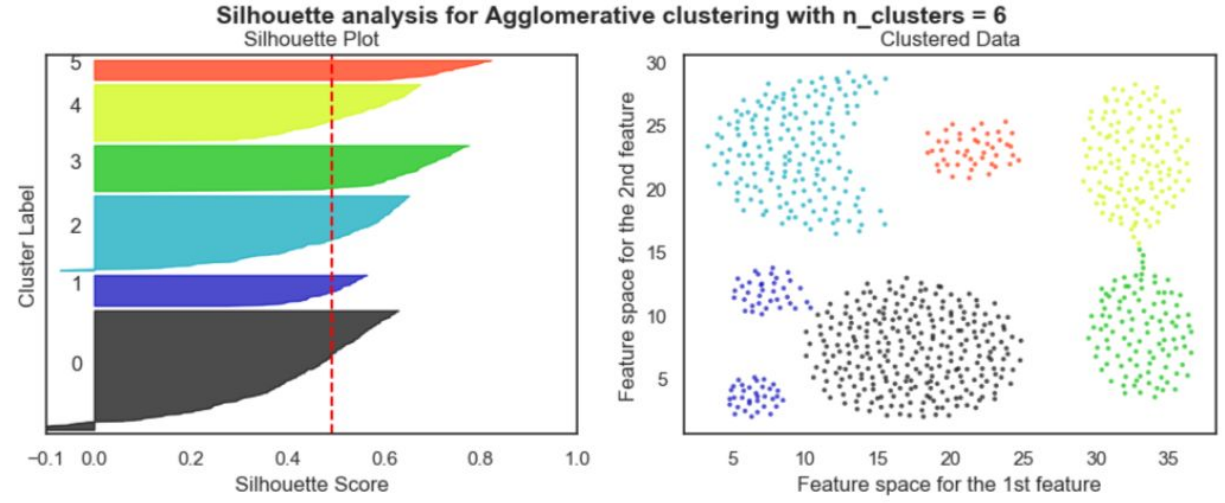
Dataset 3

```
For K = 3, the average silhouette score is 0.600
```

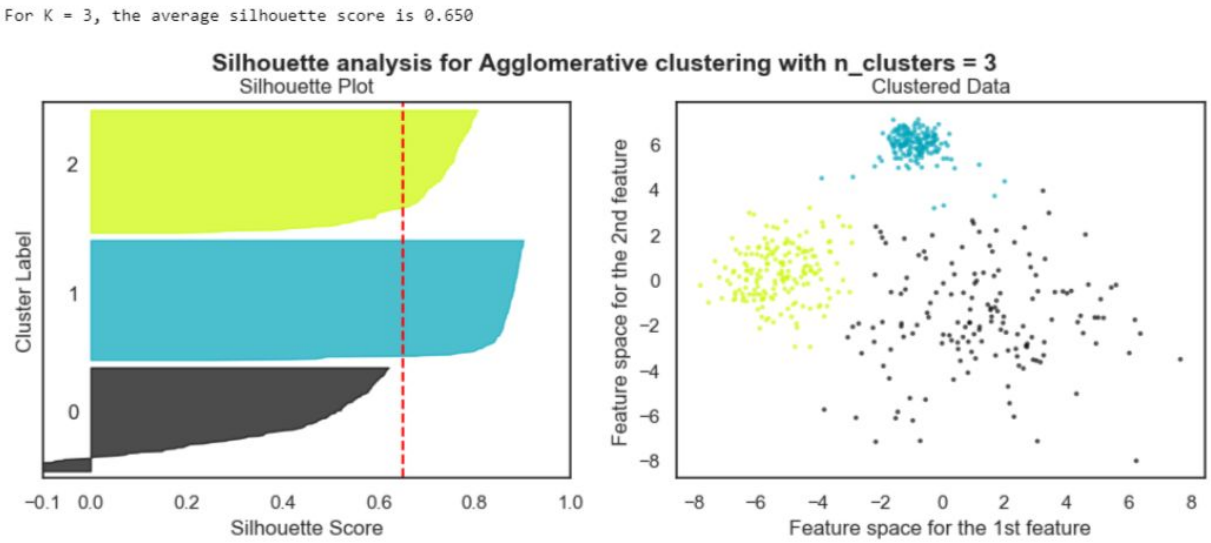


Dataset 4

```
For K = 6, the average silhouette score is 0.491
```



Dataset 5



4.2 Accuracy measure

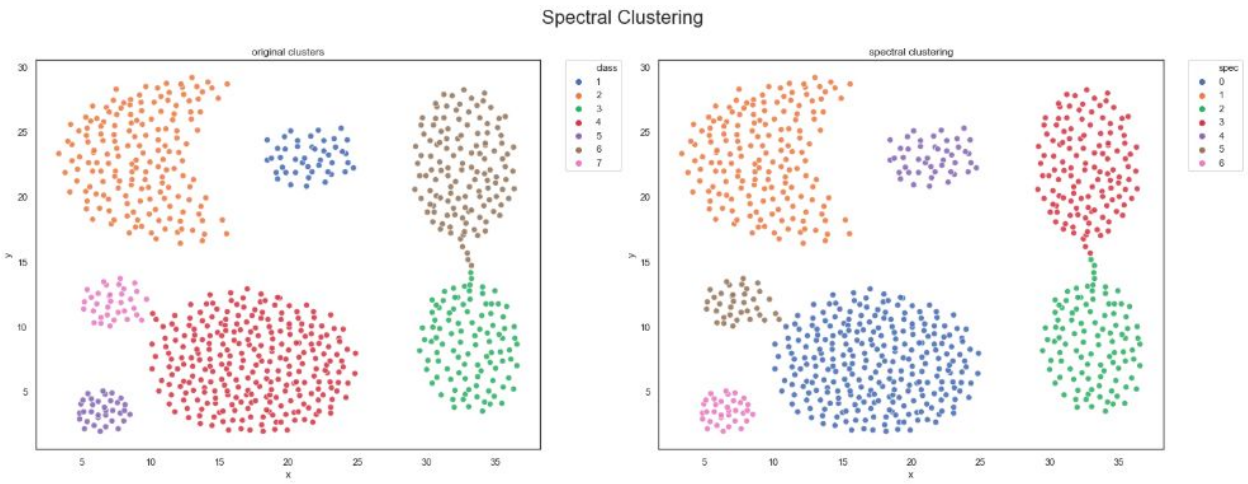
In this case, the true clusters are provided so in order to evaluate the individual algorithm's performance on each dataset, accuracy can be used as a measure. However, the output of the clustering algorithms is random, which means that it can allocate a random ID to any of the clusters. Therefore, in order to measure the accuracy, first the algorithm outputs were mapped back to the original dataset. Then using these new cluster IDs, the accuracy score was computed.

Algorithm	**k-Means**	**Agglomerative Clustering**	**DBSCAN**
Dataset 3	65.7 %	68.9 %	**74.9 %**
Dataset 4	78.5 %	**83.7 %**	82.4 %
Dataset 5	93 %	**96.2 %**	67.8 %

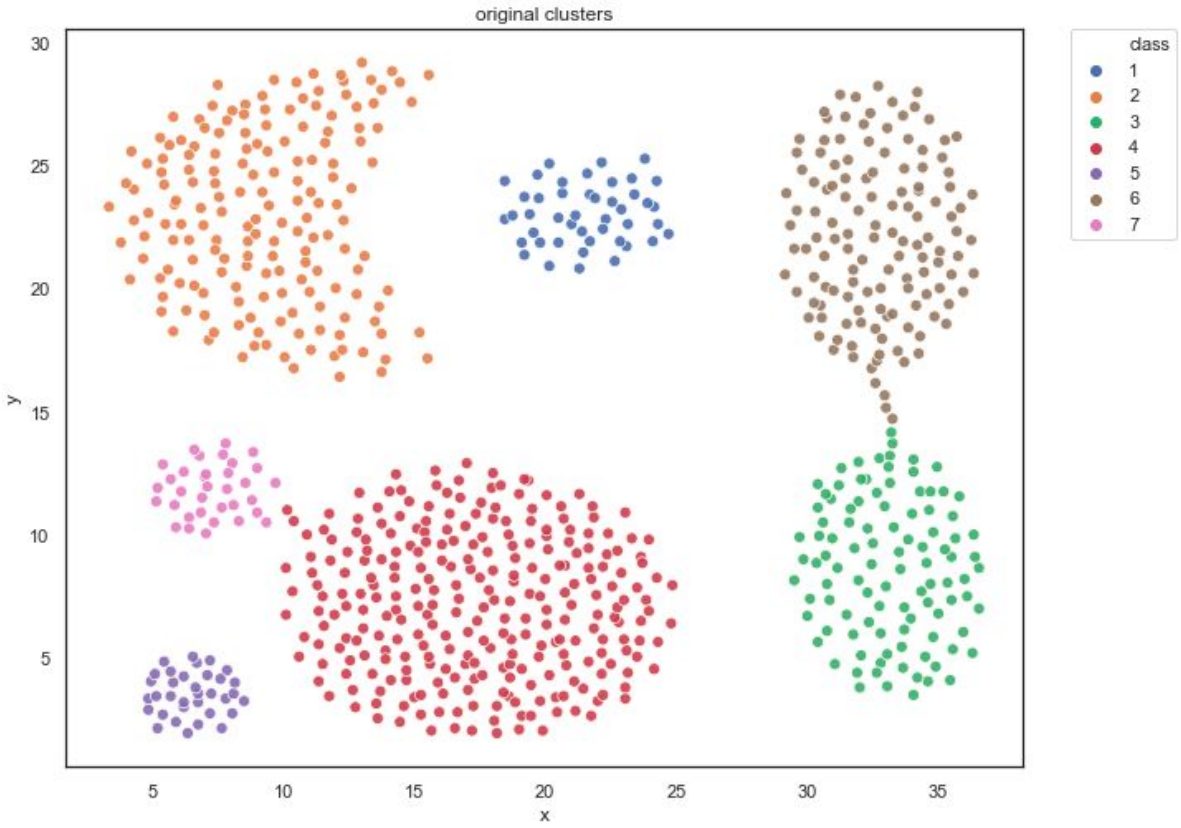
Best scores have been put in **bold** and the worst have been underlined.

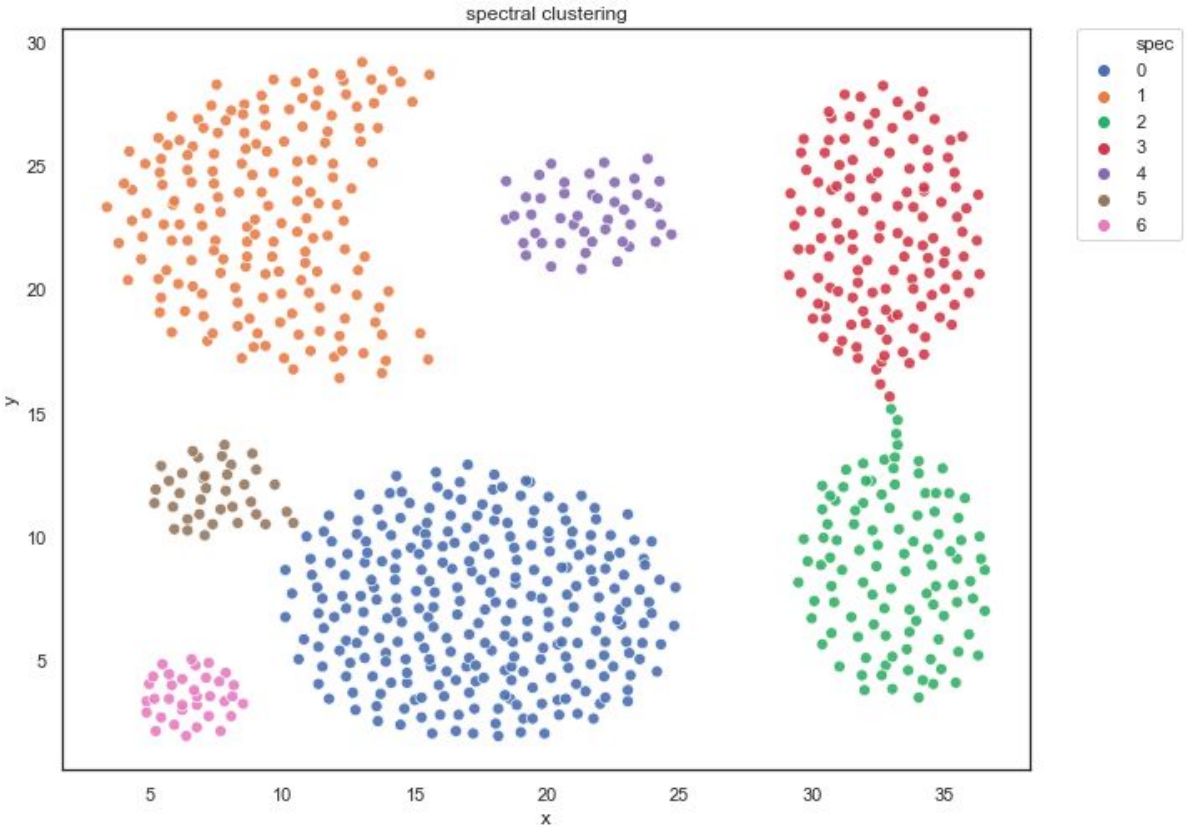
5. Improving performance

From the original plots it was observed that dataset 4 had well formed clusters but none of the basic clustering algorithms were able to perform very well. Thus, in this section a different clustering algorithm was used to improve the performance on this dataset. The best accuracy from basic algorithms was of DBSCAN, which was **82.5%**. In order to improve this measure, Spectral Clustering was used.



- From the above plots of the clusters, as identified by Spectral Clustering algorithm, it is obvious that the algorithm performs extremely well and much better than the previous ones.
- This is also verified by the accuracy score of the algorithm, which is **99.5%**.





6. Conclusion

Below are the main observation made in this lab:

- Dataset was clean, with no missing values, no null values, and no transformations were needed.
- No additional features were used because the original features represented the data points very well.
- No outliers were present in the dataset as the data was generated synthetically.
- No test set was required for this lab.
- The algorithms and their respective accuracies on the 3 datasets were:
    - k-Means
        - Dataset 3 - 65.7 %
        - Dataset 4 - 78.5 %
        - Dataset 5 - 93 %
    - Agglomerative Clustering
        - Dataset 3 - 68.9 %
        - Dataset 4 - 83.7 %
        - Dataset 5 - 96.2 %
    - DBSCAN
        - Dataset 3 - 74.9 %
        - Dataset 4 - 82.4 %
        - Dataset 5 - 67.8 %

- Spectral clustering outperformed all the algorithms for dataset 4 by identifying 99.5 % of the data points in correct cluster.
- In a real world situation, cluster analysis can be more tedious because the number of clusters is not always available.
- This can be resolved by using domain-specific knowledge which would give some idea about how many clusters can there be. Other ways to approximate the correct number of clusters have been used, such as using Silhouette plots for different values of number of clusters in the algorithm.
- It is best to observe the kind of patterns the original data points make in the feature space before deciding a clustering algorithm as different algorithms perform better on different arrangements of clusters and data points inside those clusters.
- Through the analysis, very good results were obtained for datasets 4 and 5.
- For future work, the performance can be improved for dataset 3. This can be done by optimizing the DBSCAN algorithm further as other algorithms fail to identify the clusters with ring shape.
- The assertion is based on the fact that the algorithm performed competitively, however, hyperparameter tuning can be time consuming so an advanced algorithm, in this case Spectral clustering was used instead.
- Overall, the task of cluster analysis becomes much simpler if the number of clusters is known prior to performing any test on the data. Two ways of determining this number without performing any test are either through visualization of the original dataset, or through domain knowledge. Other tests include silhouette plots, elbow plots among other tests.

8. References

[1] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[2] https://en.wikipedia.org/wiki/Limited-memory_BFGS
[3] https://scikit-learn.org/stable/modules/naive_bayes.html
[4] https://scikit-learn.org/stable/modules/tree.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[6] https://scikit-learn.org/stable/modules/svm.html