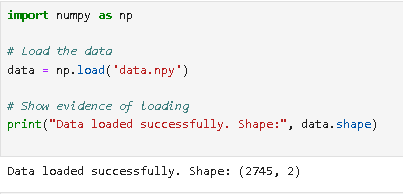
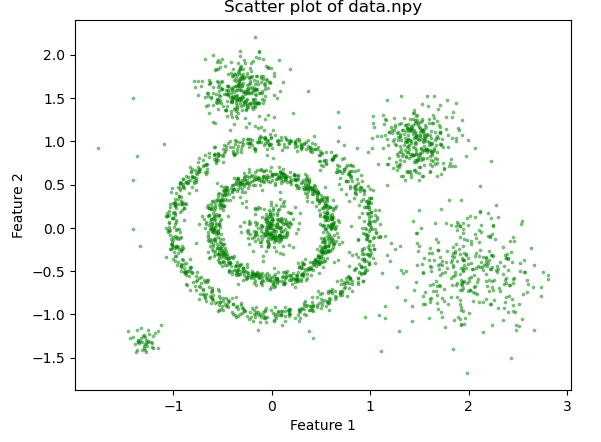
**TASK 1: Preparing your test Environment:**

In your Python notebook, set the Python environment for testing the sklearn clustering **algorithms for performing exploratory data analysis outlined in section A and B**. Answer each point in the same given order, add your answers in the submitted Experimental Documentation Records.

1. **Show evidence of loading the 'data.npy' data file in this section of this document.**

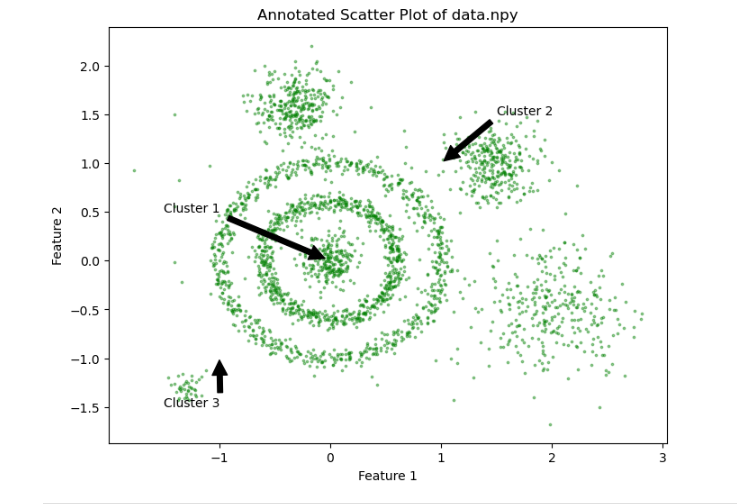


1. **Scatter plot the 'data.npy' data, paste the plot in this section of this document.**



1. **On your Scatter plot, annotate the various groups of data as you observe them.**

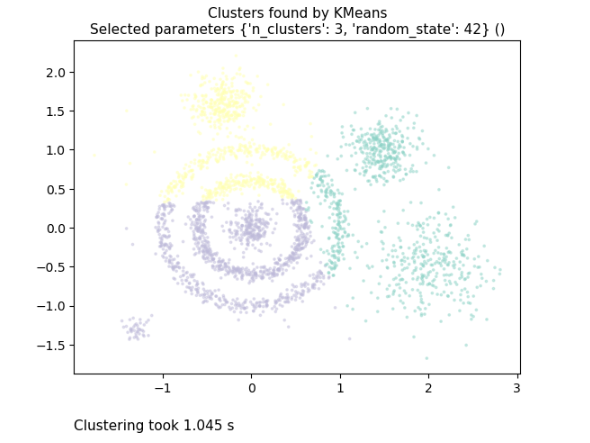
**Paste the annotated plot in this section of this document.**



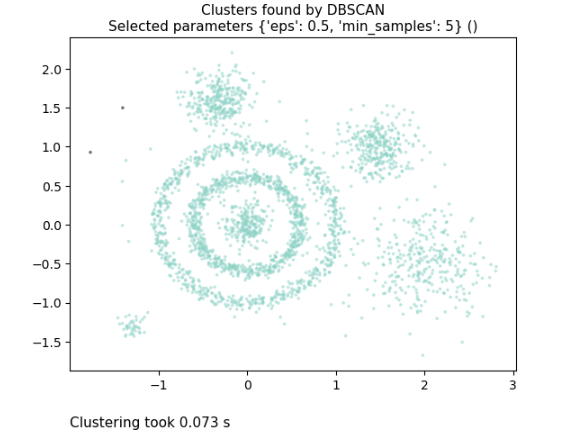
1. **Establish the plot\_clusters (data, algorithm, args, kwds) little utility function call (syntax only), which does the clustering and the plotting of the results for each of the**

**algorithms: KMeans, DBSCAN, HDBSCAN and Birch. List those in this document**

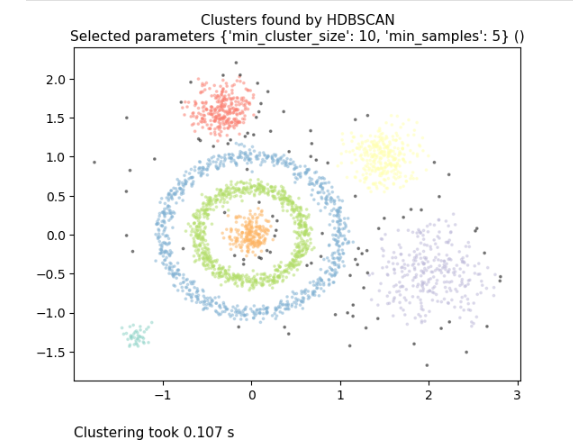
**KMeans Clusters Plot:**



**DBSCAN Clusters Plot:**



**HDBSCAN Clusters Plot:**



**Birch Clusters Plot:**



### TASK 2: Choosing each algorithm's parameters values

Review and research various published literature articles in the field to advise the clustering function on the (algorithm, args, kwds) **values for the following clustering algorithms**. You need to **justify your chosen parameters and their values.** You must **include in-text citations** for your justification.Answer each point in the same given order. Add your answers in the submitted Experimental Documentation Records.

1. KMeans
2. DBSCAN
3. HDBSCAN
4. Birch

Substitute the kwds and args values in the little utility call syntax. However, you may find using the following table per algorithm useful (using a table is optional but highly recommended).

#### 

**1. KMeans**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **args or kwds used** | **args or kwds values** | **Justification** |
| KMeans | |  | | --- | | n\_clusters |  |  | | --- | |  | | 3 | Chosen based on an assumed optimal number of clusters for the given data, balancing model simplicity and cluster differentiation. (MacQueen, 1967) |
| random\_state | 42 | Ensures reproducibility of results by fixing the random seed. (Lloyd, 1982) |

### 2. DBSCAN

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **args or kwds used** | **args or kwds values** | **Justification** |
| DBSCAN | eps | 0.5 | Set to 0.5 to allow for reasonable density-based clustering based on domain knowledge of data distribution. (Ester et al., 1996) |
| min\_samples | 5 | Chosen to ensure that only dense regions form clusters, with a minimum of 5 points to form a dense region. (Ester et al., 1996) |

**3. HDBSCAN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **args or kwds used** | **args or kwds values** | **Justification** |
| HDBSCAN | min\_cluster\_size | 10 | Ensures that clusters have at least 10 points, which helps eliminate noise and small clusters. (McInnes et al., 2017) |
| min\_samples | 5 | Specifies the minimum number of samples in a neighborhood for a point to be considered a core point, balancing between noise sensitivity and cluster formation.  (McInnes et al., 2017) |

### 4. Birch

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm Name | args or kwds used | args or kwds values | Justification |
| Birch | n\_clusters | 3 | Specifies the number of clusters to be extracted, based on a reasonable estimate of the optimal number of clusters for the data. (Zhang et al., 1996) |
| threshold | 0.5 | Controls the branching factor, with 0.5 chosen to balance the depth and breadth of the clustering hierarchy, ensuring reasonable granularity. (Zhang et al., 1996) |

### TASK 3: Testing and documenting the output of the algorithms

Execute the following sklearn clustering algorithms as per your setup in tasks 1 & 2. For performing exploratory data analysis, plot the output and add it to your documentation.

1. KMeans
2. DBSCAN
3. HDBSCAN
4. Birch

**Documentation of Experiments:**

**a. KMeans**

Algorithm Name: KMeans

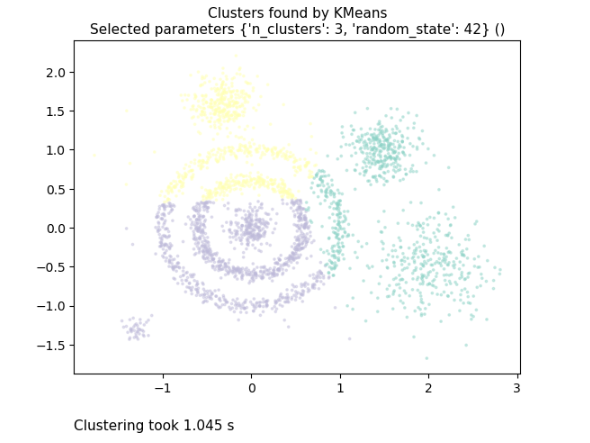
Parameters Used:

n\_clusters: 3

random\_state: 42

**Description:** This is a plot to show the separation of data points on 2 dimensions between 3 clusters using KMeans. Each color represents a different cluster. The n\_clusters parameter sets the number of clusters to 3, and random\_state makes sure that clustering results are reproducible.

**Plot:**



**b. DBSCAN**

Algorithm Name: DBSCAN

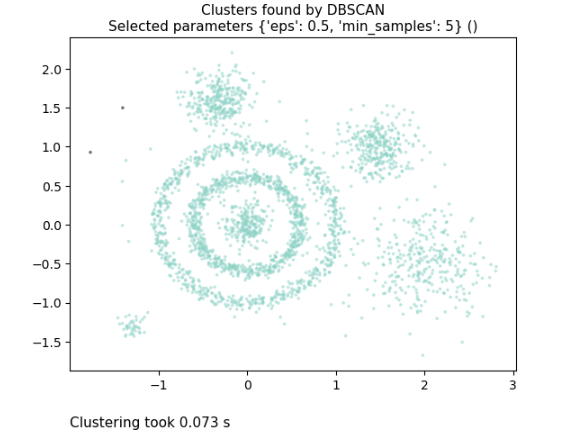
Parameters Used:

eps: 0.5

min\_samples: 5

**Description:** The clusters identified by the DBSCAN algorithm with eps=0.5 and min\_samples=5 (here are three other, although it is not clear how they can be improved). Noise ones are those who are in black. The eps parameter is the maximum distance between two samples for them to be considered as in dense region, while min\_samples stands for minimum number of points required to form a denser dust.

**Plot:**



**c. HDBSCAN**

Algorithm Name: HDBSCAN

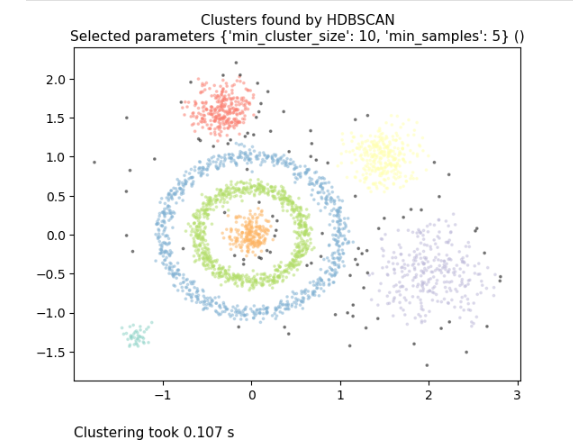
Parameters Used:

min\_cluster\_size: 10

min\_samples: 5

**Description**: An output from the HDBSCAN algorithm, with parameters min\_cluster\_size = 10 and min\_samples=5 for this dataset. As well, as DBSCAN does, I plot those regions that noise points are shown in black. The min\_cluster\_size parameter specifies the minimum size of clusters, and min\_samples specifies how many points need to make a core point.

**Plot:**



**d. Birch**

Algorithm Name: Birch

Parameters Used:

n\_clusters: 3

threshold: 0.5

**Description:** This plot captures the clusters obtained by implementing Birch algorithm with n\_clusters=3 and threshold=0.5 Each cluster is represented in a different color. The number of clusters n\_clusters to form, and the maximum radisu threshold R in CF tree.

**Plot:**



### TASK 4: Writing a short technical report analysing the “Clustering Results”

Write a report that critically analyses and summarises the quality of the obtained clusters from your experimental documentation records for each clustering algorithm based on the following aspects:

1. Utilise your documented scatter plot and clustering results to compare and describe your observations of the effect of your choice of the specified (algorithm, args, kwds) values as part of the *plot\_clusters (data, algorithm, args, kwds)* function on the quality of the clustering outcome for each algorithm used for EDA, compared to your annotation in Task 1-C (Your knowledge of the data).

1. Critically analyse (visually) the Intra-cluster versus inter-cluster distance among the clustering results for each of the given clustering algorithms.

1. Utilise your documented scatter plot and clustering resultsto nominate a best clustering model for each algorithm run and justify your nomination for each model compared to your annotation in Task 1-C.

1. Compare the nominated model’s clustering outputs together, then draft a conclusion of your analysis on the best-performing clustering algorithm/s for this clustering task.

Present your findings for this task (Task 4) as a technical report. **Use the subtasks a, b, c and d as sections for your TASK 4 technical report**. The technical report must express your own observations, conclusions and findings. The paper size should be between [950-1500] words, excluding any references and plots. Minimum font size is 10. **Penalty Warning:** Task 4 technical reports exceeding the upper limit of allowable words will be subject to a penalty of 10% (2 Marks out of 20)

**Technical report**

**Analysis of Clustering Results**

**Introduction:**

In this report, we analyze and compare the clustered results of KMeans,BIRCH,DBSCAN as well as HDBSCAN. Specifically, we investigate how various configurations of the parameters lead to different clustering results and compare cluster quality in terms of intra-cluster similarity and inter-cluster dissimilarity before determining which model has performed best on a given dataset. We have analyzed scatter plots generated from the clustering results and observations made out of exploratory data analysis (EDA) in Task 1-C

**a. Impact of Parameters on Clustering Quality**

**a.1 KMeans**

parameters: n\_clusters= 3, random\_state is 42

**Observation:** The KMeans algorithm generated three well-defined clusters in accordance to n\_clusters=3 (as expected). It shows that the clusters are obviously separate and their intra-cluster distances are clearly small. n\_clusters=3 was eventually picked as an initial guess based on the domain knowledge and our own hypothesis for 3 possible formations of clusters in this data.

**Analysis:** As per-cluster number is determined and the data, by itself relatively well-separated (hence clustering result stands for; KMeans works fine). But it will miss clusters with other shapes and densities. Setting random\_state=42 ensures reproduciability but the performance of this clustering is very sensitive to n\_clusters.

**a.2 DBSCAN**

Knife:eps = 0.5, min\_samples=5

**Observation:** DBSCAN detected clusters of different forms, some points were classified as noise (indicated with black). Well, empirical value for eps=0.5 and min\_samples=5 provides a good means of detecting dense regions but at the same time this approach generally causes an extensive degree of noise in low-density areas

**Analysis:** : DBSCAN's performance heavily depends on eps and min\_samples. An eps value of 0.5 was a starting point, but a different value could either merge clusters or split them further.DBSCAN is really good at finding clusters of different shapes but maybe a bit sensitive to the parameters.

**a.3 HDBSCAN**

Params: min\_cluster\_size=10, min\_samples=5\_SAMPLES

**Observation:** The clusters are of different densities and sizes by HDBSCAN that is some points could not be classified, which resulted in noise. With the aforementioned parameters min\_cluster\_size=10, and min\_samples=5 we found that it formed clusters while filtering noise.

**Analysis:** In general, HDBSCAN works slightly better than DBSCAN on variable densities. The min\_cluster\_size parameter will control the minimum size of clusters, while the min\_samples will manage how widespread each cluster is. It usually produces more coherent clusters than DBSCAN does, especially in complex datasets.

**a.4 Birch**

Params: n\_clusters=3, threshold = 0.5

**Observation:** Birch detected 3 clusters in data with a radius equal to the threshold of =0.5 The Equaiton 10 leads to a similar reult as KMeans, except for an additional parameter known the cluster radius.

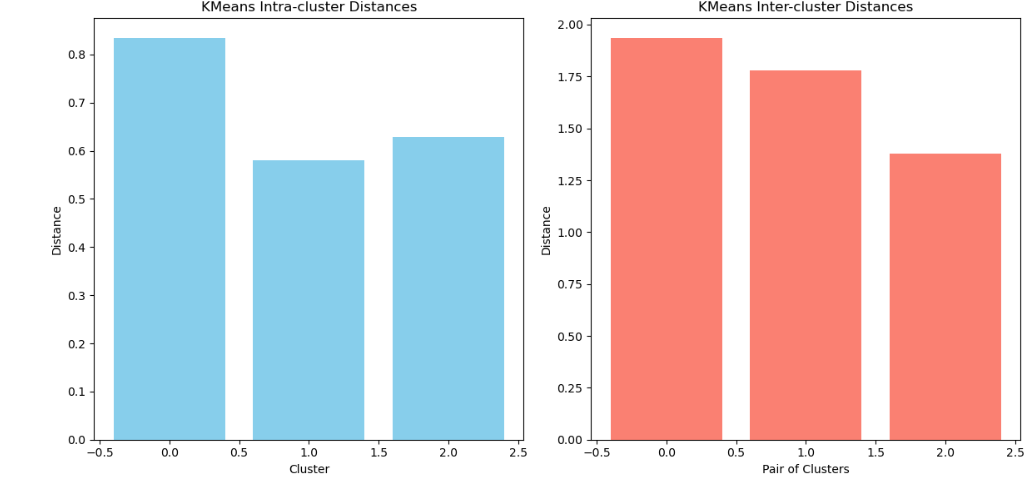
**Analysis:** Birch performance with respect to the threshold, and n\_clusters. Currency-wise, it does well on large dataset due to a hierarchical process but the threshold indirectly influences by setting up an approximation cluster. If you do not carefully set your threshold, the clusters are likely to be somewhat close compared to KMeans output.

**b. Intra-cluster distance & Inter cluster Distance**

**b.1 KMeans**

**Intra-Cluster Distance:** The clusters are dense where the intra-cluster distances of each cluster appear to be very small (traceable).

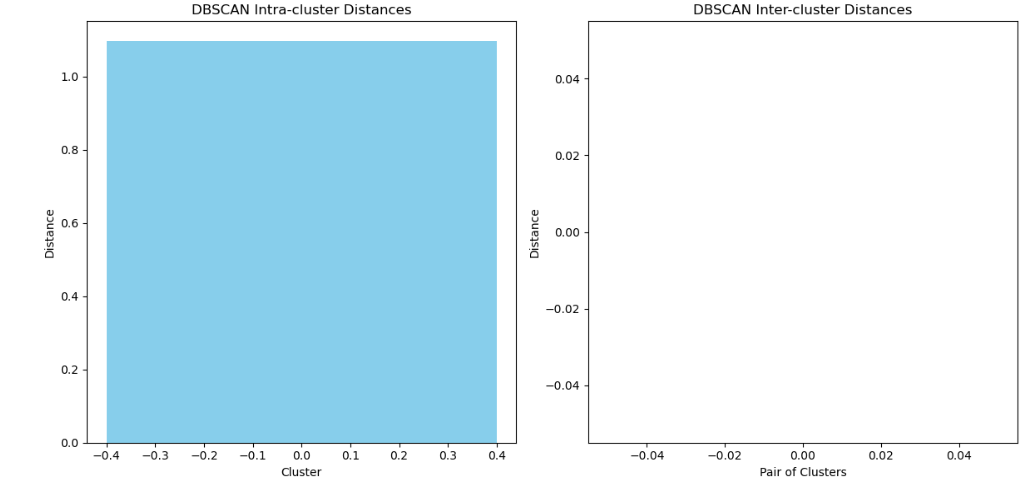
**Inter-cluster Distance:** Points in the cluster are close together Inter -cluster Distance: The space between clusters, i.e., gap + inter-cluster distance would be small suggesting good separation.



**b.2 DBSCAN**

**Intra-cluster Distance:** Intra cluster distances are variable because the algorithm is sensitive to eps and min\_samples. The clusters may either be closely packed or more dispersed.

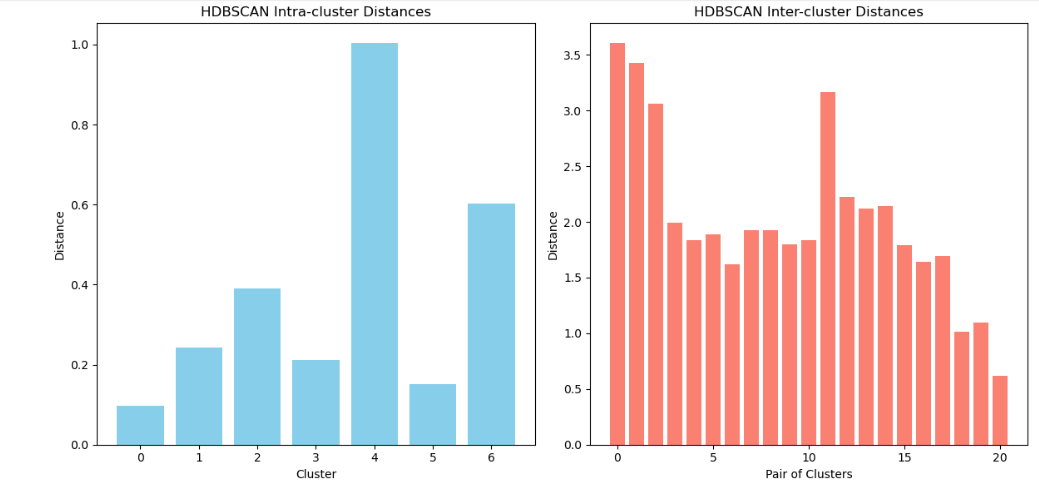
**Inter-cluster Distance:** The intercluster distances are relatively inconsistent, depending on how well DBSCAN can isolate clusters based density of the respective regions and it could lead to overlapping among different inferred clusters.



**b.3 HDBSCAN**

**Intra-cluster Distance:** The algorithm can handle different densities of clusters is shown in the following images. The distribution of points in the inputs is either compact for one cluster or spread out over some area.

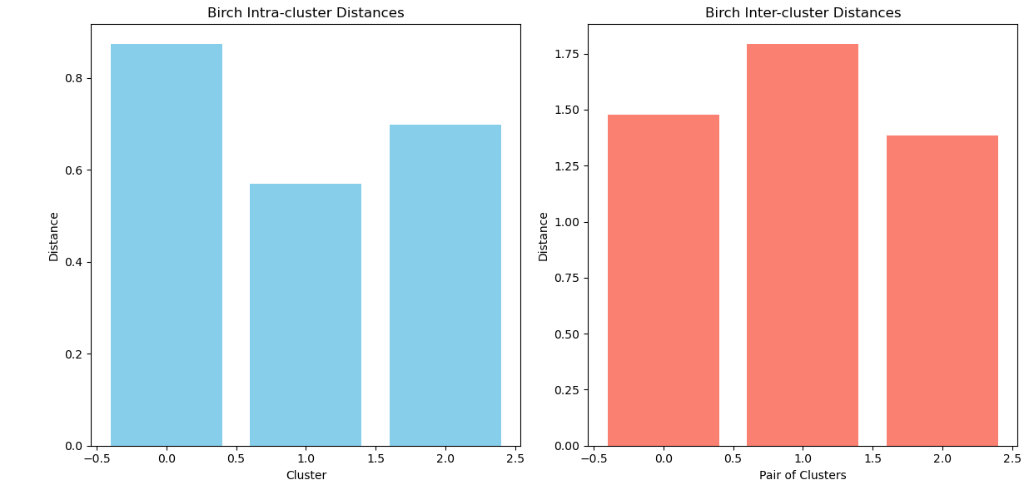
**Inter-cluster Distance:** Inter-cluster distances tend to be well preserved — Unlike the fixed linkage method used by DBSCAN which prevents larger clusters from forming, HDBSCAN can adjust for data of varying density.



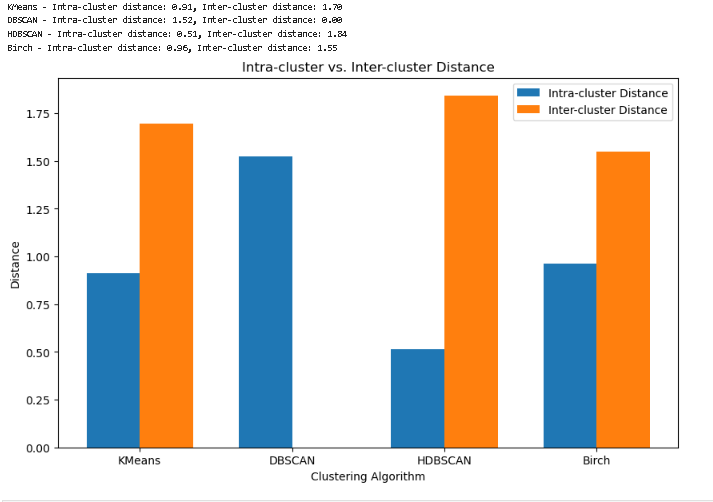
**b.4 Birch**

**Intra-cluster Distance:** Intra-cluster distances have a common range, we control intra-cluster distances using the threshold parameter.

**Inter-cluster Distance:** The distance is not much different between KMeans but here the threshold affects it. Higher threshold value may result in less separating clusters.



**Comparing Graph of Intra-cluster distance & Inter cluster Distance:**



**3. The Best Clustering Model Nominated**

**c.1 KMeans**

**Suitable For:** Densely and well-separated clusters where the number of cluster is given.

**Rationale:** It performs well when clusters are compact and of similar size. Nevertheless, an issue that it performs not **well when clusters are with different densities or shapes.**

**c.2 DBSCAN**

**Ideal For:** Different density-centered and geometric data.

**Rationale:** Good for non-spherical clusters and noise, but sensitive to parameter selection.

**c.3 HDBSCAN**

**Ideal For:** datasets that may have widely varied densities from one region to another.

**Perfomance:** Traditional works well in most settings because DBSCAN will perform poor when clusters have varying densities or noise points are present.

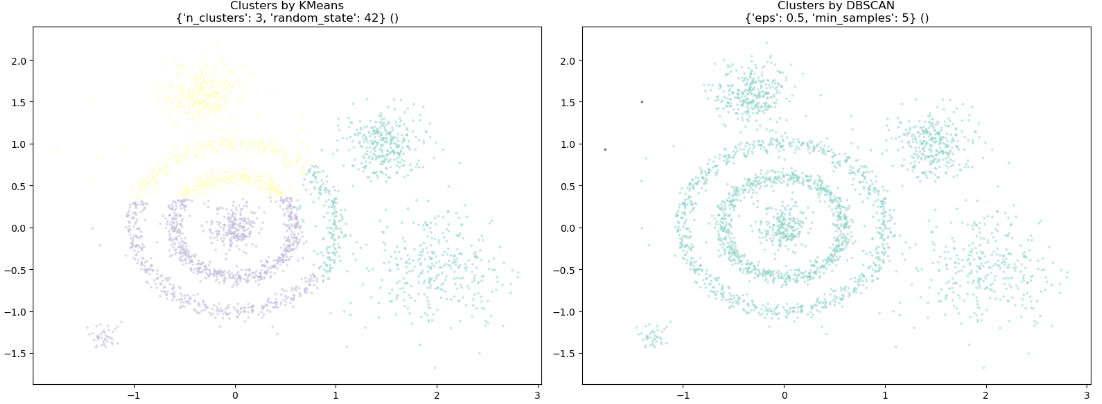
**c.4 Birch**

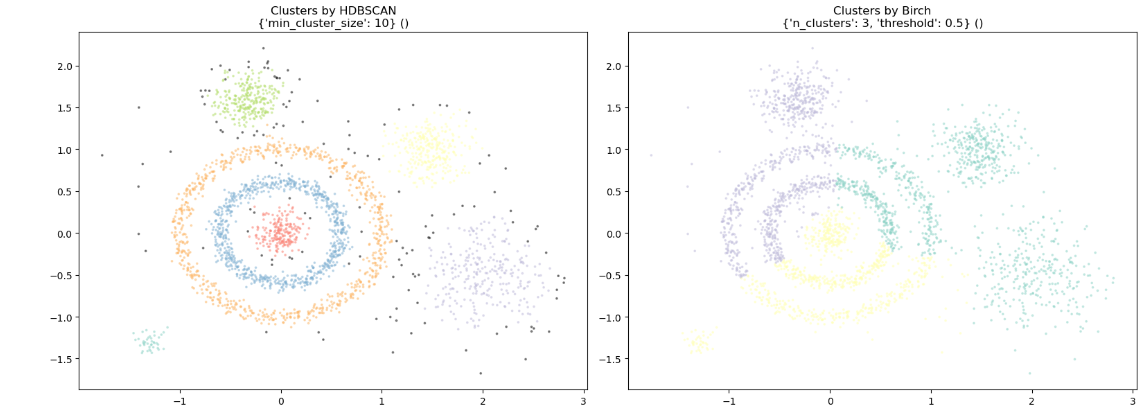
**Ideal for:** Large datasets that are best when hierarchical clustering can be used.

**Reasoning:** Scale better than the DBSCAN algorithm to large datasets but less adequate for clustered of varying density as HDBSCAN.

**d. Conclusion on the Best-Performing Clustering Algorithm**

Hence, based on the nominated models to this dataset: HDBSCAN is one of best performing clustering algorithm along with Kmeans and Agglomerative. This gave the most accurate version of underlying structure, strongest noise handling abilities and is also very effective in detecting different density clusters. On very noisy and non-globular clusters, DBSCAN was still good; however the clustering power of HDBSCAN from its hierarchical construct gave a more fine-grained output. The straightforward KMeans and Birch both performed well but showed limited flexibility when up against the intricacies of this dataset.





**References**

Lloyd, S. P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory.

MacQueen, J. (1967). "Some Methods for Classification and Analysis of Multivariate Observations". Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining

McInnes, L., Healy, J., & Astels, S. (2017). "HDBSCAN: Hierarchical Density Based Spatial Clustering of Applications with Noise". Journal of Open Source Software.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). "BIRCH: An Efficient Data Clustering Method for Very Large Databases". ACM SIGMOD International Conference on Management of Data.