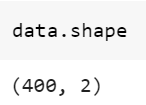
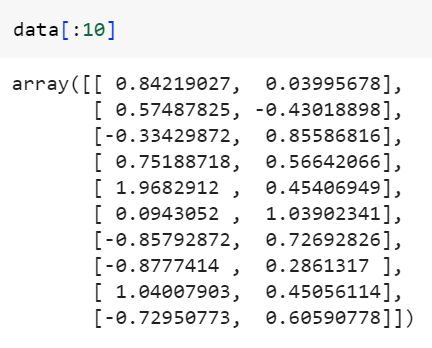
Pragya Bhatia

[W1968251@westminster.ac.uk](mailto:W1968251@westminster.ac.uk)

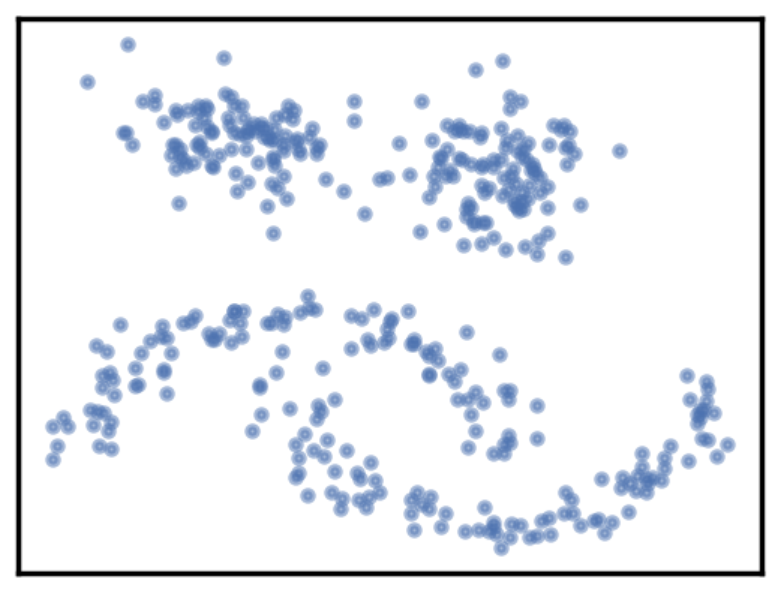
TASK 1: Preparing your test Environment:

1. Show evidence of Loading the 'data.npy' data file.

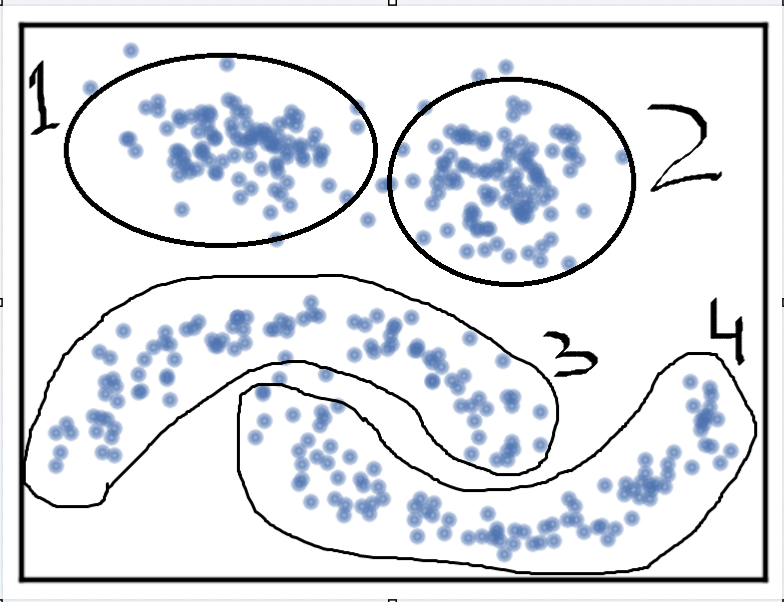




1. Scatter plot the 'data.npy' data.



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



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: K-Means, DBSCAN, Birch, OPTICS, Affinity Propagation, HDBSCAN, Agglomerative Clustering, Spectral Clustering

|  |  |
| --- | --- |
| Algorithm Name | Syntax |
| K-Means | plot\_clusters (data, KMeans, (), {‘n\_clusters’:value, ‘init’:value, ‘max\_iter’:value, ‘algorithm’:value}) |
| Affinity Propagation | plot\_clusters (data, AffinityPropagation, (), {‘preference’:value, ‘damping’:value, ‘convergence\_iter’:value, ‘max\_iter’:value, ‘affinity’=value'}) |
| DBSCAN | plot\_clusters (data, DBSCAN, (), {‘eps’:value, ‘min\_samples’:value, ‘metric’:value}) |
| Birch | plot\_clusters (data, Birch, (), {‘n\_clusters’:value, ‘branching\_factor’:value, ‘threshold’:value }) |
| OPTICS | plot\_clusters (data, OPTICS, (), {‘min\_samples’:value, ‘metric’:value, ‘max\_eps’:value }) |
| HDBSCAN | plot\_clusters (data, HDBSCAN, (), {‘min\_cluster\_size’:value, ‘metric’:value }) |
| Agglomerative Clustering | plot\_clusters (data, AgglomerativeClustering, (), {‘n\_clusters’:value, ‘metric’:value, ‘linkage’:value}) |
| Spectral Clustering | plot\_clusters (data, SpectralClustering, (), {‘n\_clusters’:value, ‘affinity’:value }) |

TASK 2: Choosing each algorithm's parameter values

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| K-Means | n\_clusters | 4 | We can observe 4 distinct clusters  The best numbers of clusters obtained from elbow method [1] |
| init | k-means++ | It is said to be a better and standard choice as it is based on probability proportional to the squared distance away from a given point's nearest existing centroid. Also, as my data set is small so random is faster than k-means++ but it is unstable as it randomly selects data points that can be highly volatile. So, for large datset, k-means++ is preferable [2] |
| max\_iter | 100 | It minimizes the sum of squared distances within clusters (inertia). Although it is showing the same result for our small dataset if we use 100 or 300 (default) but it is crucial parameter if we have large dataset.  “the difference between with-in cluster similarity become negligible with increasing number of iterations” [3] |
| algorithm | Elkan | We are getting same result with Elkan and Lloyds, so we are going with Elkan. Elkan is faster in processing than Lloyd as it uses a lower & upper-bound approach. Lloyd spends a lot of processing time computing the distances between each of the k cluster centres and the n data points. Since points usually stay in the same clusters after a few iterations, much of this work is unnecessary, making the naive implementation very inefficient. However, The Elkan algorithm tends to be more efficient when the number of clusters (k) is small, and the dimensionality of the data is not too high. If you have a large dataset with a relatively small number of clusters, Elkan might offer faster convergence. [4] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| Affinity Propagation | preference | Value -> (- Number of Pref/10) = 4 (No of cluster)  No.of Pref = -40  Value = -40 | The preference parameter represents the diagonal of the similarity matrix and indicates how likely a data point is to be chosen as an exemplar (cluster center).  For a common preference of −600, affinity propagation found 62 clusters = abs(-600/10) = 60 clusters. [5] |
| damping | Default Value – 0.5 | Each iteration in the affinity propagation involved (i) revising all responsibilities based on availabilities, (ii) adjusting all availabilities based on responsibilities, and (iii) merging availabilities and responsibilities to monitor exemplar decisions and conclude the algorithm [5]. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| DBSCAN | eps | Mean Nearest Neighbour Distance  Value - 0.15 | First, we found the epsilon range using Euclidean with nearest neighbours.[6] |
| min\_samples | Value - 4 | These are core points of a cluster. As we have only 4 clusters, DBSCAN will find denser clusters, whereas if it is set to a lower value, the found clusters will be more sparse. [7] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| Birch | n\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| threshold | Value – 0.5 | The radius of the subcluster obtained by merging a new sample and the closest subcluster should be lesser than the threshold. Otherwise, a new subcluster is started. Setting this value to be very low promotes splitting and vice-versa.[8] |
|  | Branching\_factor | No. Samples/2  Value - 200 | Maximum number of CF subclusters in each node. If a new sample enters such that the number of subclusters exceed the branching factor, then that node is split into two nodes with the subclusters redistributed in each. The parent subcluster of that node is removed and two new subclusters are added as parents of the 2 split nodes. [8] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| OPTICS | Min\_samples | abs (Fraction no. of samples)  value = 400/10000 = 0.04 | The number of samples in a neighbourhood for a point to be considered as a core point. Also, up, and down steep regions can’t have more than min samples consecutive non-steep points. Expressed as an absolute number or a fraction of the number of samples (rounded to be at least 2). [9] |
| xi | Default – 0.05 | This is the minimum steepness on the reachability plot that forms a cluster boundary. is an extension of the "reachability" concept in DBSCAN. The ratio from one point to its successor being at most 1-xi [9] |
|  | Min\_cluster\_size | abs (Fraction no. of samples)  value = 400/10000 = 0.04 | Minimum number of samples required for a cluster.  Fraction of the number of samples. [9] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| HDBSCAN | Min\_cluster\_size | 5 | This was the size which was clustered as an expected group. Lower value than this was creating noise.[10] |
| cluster\_selection\_epsilon | 0.5 | If our data set also contains partitions with high concentrations of objects, this parameter setting can result in a large number of micro-clusters. Selecting a value for cluster\_selection\_epsilon helps us to merge clusters in these regions. Or in other words, it ensures that clusters below the given threshold are not split up any further.[10] |
|  | Min\_samples | 10 | Conservative Cluster we can obtain. The larger the value of min\_samples you provide, the more conservative the clustering – more points will be declared as noise, and clusters will be restricted to progressively more dense areas. The min\_cluster\_size at 60, but reducing min\_samples to 1.[10] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| Agglomerative Clustering | N\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| linkage | ward | Ward is better as it causes minimum increase in information loss. Moreover, complete merges the close groups of clusters (as it determines the most significant distance between any two objects in the different clusters). [11] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Args or kwd used** | **Args or kwd values** | **Justification** |
| Spectral Clustering | N\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| affinity | rbf | Nearest\_neighbor is computed based on the K-nearest neighbors of each data point. It connects each point to its K nearest neighbors in the dataset.  RBF measures the similarity between data points using the exponential of the negative squared Euclidean distance. We calculated the silhouette score for them and got the RBF as best. [12][13] |
| gamma | 10 | We can use the Gamma that controls the width of the Gaussian kernel in the RBF affinity matrix.  As our clusters are shaped in smile structure, we can smooth them using gamma.  Default value is 1.0 A higher gamma results in a narrower Gaussian distribution, making the kernel more sensitive to the distance between data points. |

**References:**

Helpful site URL: https://scikit-learn.org/stable/modules/clustering.html#clustering

[1] Marutho, D., Handaka, S. H., Wijaya, E., & Muljono. (2018) 'The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News,' in 2018 International Seminar on Application for Technology of Information and Communication, Semarang, Indonesia, pp. 533-538. doi: 10.1109/ISEMANTIC.2018.8549751.

[2] Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2004) 'Selection of K in K-means clustering,' Manufacturing Engineering Centre, Cardiff University, Cardiff, UK. Manuscript received on 26 May 2004, accepted after revision for publication on 27 September 2004. DOI: 10.1243/095440605X8298.

[3] Zahra, S., Ghazanfar, M. A., Khalid, A., & Prugel-Bennett, A. (2015) 'Novel centroid selection approaches for KMeans-clustering based recommender systems,' Information Sciences, 320, DOI: 10.1016/j.ins.2015.03.062.

[4] Yu, Q., Chen, K.-H., & Chen, J.-J. (2020) 'Using a Set of Triangle Inequalities to Accelerate K-means Clustering,' in Similarity Search and Applications: 13th International Conference, SISAP 2020, Copenhagen, Denmark, September 30 – October 2, 2020, Proceedings, pp. 297–311. doi: 10.1007/978-3-030-60936-8\_23.

[5] Frey, B. J., & Dueck, D. (2007) 'Clustering by Passing Messages Between Data Points,' Science, 315(5814), 972-976. DOI: 10.1126/science.1136800.

[6] Rahmah, N., & Sitanggang, I. S. (2016) 'Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra,' IOP Conference Series Earth and Environmental Science, 31(1), 012012. DOI: 10.1088/1755-1315/31/1/012012. License: CC BY 3.0.

[7] Merk, A., Cal, P., & Wozniak, M. (2017) 'Distributed DBSCAN Algorithm – Concept and Experimental Evaluation,' DOI: 10.1007/978-3-319-59162-9\_49.

[8] Ramadhani, F., Zarlis, M., & Suwilo, S. (2020) 'Improve BIRCH algorithm for big data clustering,' IOP Conference Series Materials Science and Engineering, 725(1), 012090. doi: 10.1088/1757-899X/725/1/012090

[9] Ankerst, M., Breunig, M. M., Kröger, P., & Sander, J. (1999) 'OPTICS: Ordering Points to Identify the Clustering Structure,' ACM SIGMOD Record, 28(2), 49-60. DOI: 10.1145/304182.304187

[10] McInnes, L., Healy, J., & Astels, S. (2017). "HDBSCAN: Hierarchical density based clustering." The Journal of Open Source Software, 2(11), 205. DOI: 10.21105/joss.00205.

[11] Fernandes, A. A. R., & Solimun, S. (2022) 'Comparison of the Use of Linkage in Cluster Integration With Path Analysis Approach,' Frontiers in Applied Mathematics and Statistics, 8. DOI: 10.3389/fams.2022.790010

[12] Langone, R., Mall, R., Alzate, C., & Suykens, J. A. K. (2016) 'Kernel Spectral Clustering and Applications,' DOI: 10.1007/978-3-319-24211-8\_6

## [13] Gandhi,Vipul.(2020)’ Spectral Clustering - Detailed Explanation’, Available at: https://www.kaggle.com/code/vipulgandhi/spectral-clustering-detailed-explanation

Gamma value

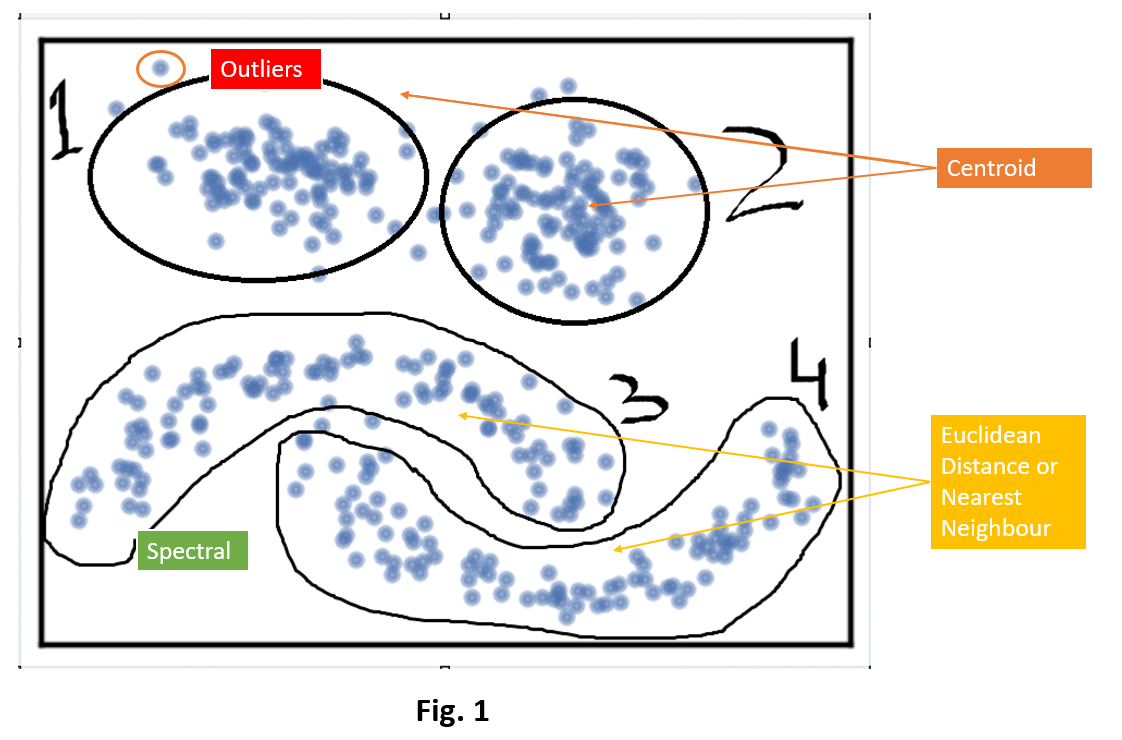
**TASK 3:** Testing and documenting the output of the algorithms. Run as it is setup in Task 1 & 2 For performing exploratory data analysis plot the output.

|  |  |
| --- | --- |
| 'n\_clusters':4,'init':'k-means++','algorithm':'elkan','max\_iter':100 |  |
| 'preference': -40,'damping':0.5 |  |
| 'eps':0.15, 'min\_samples':4 |  |
| 'branching\_factor': 200,'threshold':0.5,'n\_clusters':4 |  |
| 'min\_samples':0.04, 'xi':0.05, 'min\_cluster\_size':0.04 |  |
| 'min\_cluster\_size': 5,'cluster\_selection\_epsilon':0.5,'min\_samples':10 |  |
| 'n\_clusters': 4, 'linkage':'ward' |  |
| 'n\_clusters': 4,'affinity':'rbf','gamma':10 |  |

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

1. **The domain:**
2. We get below observations from the given data file data.npy:

* We have small dataset with 2 features with 400 observations.
* When we plot that using scatter plot, we get 2-D visualisation. By just looking at the visual image, we can see there are 4 clusters. It has 2 circular density shapes and 2 non globular shapes; includes some noise as well which makes it difficult to find the exact boundaries of these shapes.

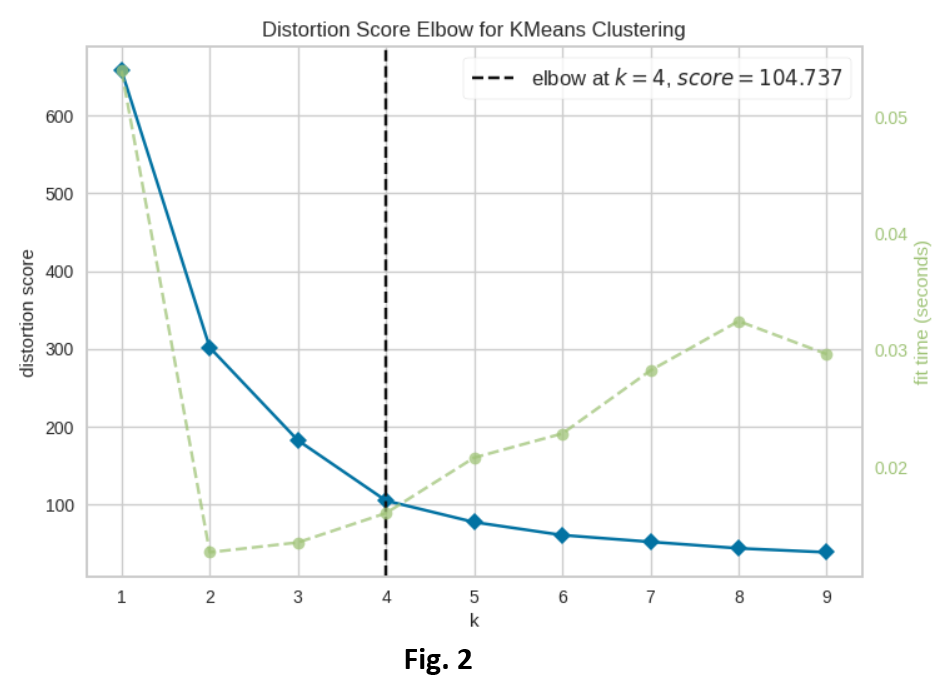


1. **Intuitive Parameters** from Fig 1 –

* Number Of Cluster – 4 We can see 4 different observable patterns. With Elbow method we can verify the same
* For Cluster 1 & 2 Centroid is stable
* For Cluster 3 & 4 Mean centroid will not work we need to use Euclidean or Nearest neighbour Distance

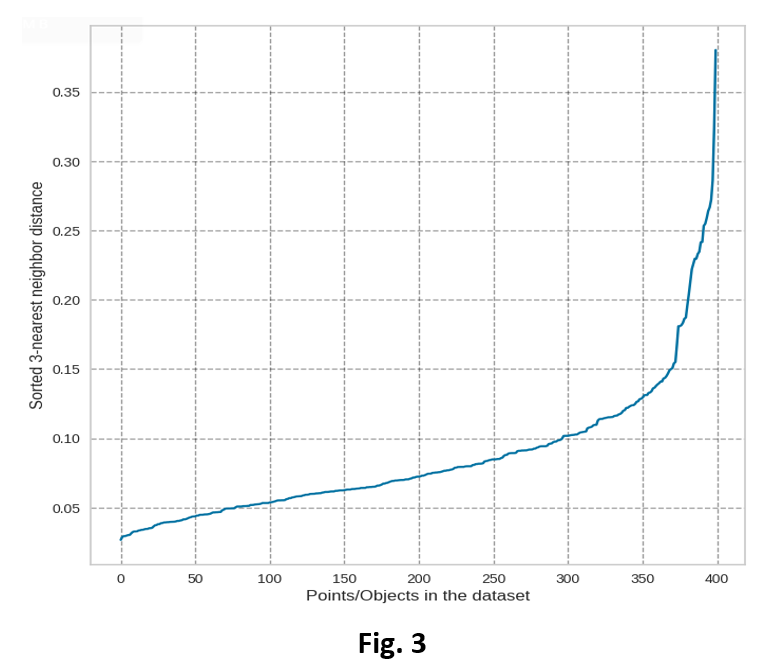
1. **Stable Cluster** – Cluster 1 & 2 can be stable and with changing multiple parameters won’t affect that much compared to the remaining clusters.
2. **Performance** – For high computational speed K-Means or DBSCAN is commonly used due to processing of big data.
3. The specified (algorithm, args, kwds) values as part of the plot\_clusters (data, algorithm, args, kwds) employed function

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| K-Means | n\_clusters | 4 | We can observe 4 distinct clusters.  The best numbers of clusters obtained from elbow method [1]. The Fig. 2 Shows the Distortion score with k value. |
| init | k-means++ | It is said to be a better and standard choice as it is based on probability proportional to the squared distance away from a given points nearest existing centroid. Also, as my data set is small so random is faster than k-means++ but it is unstable as it randomly selects data points that can be highly volatile. So, for large dataset, k-means++ is preferable [2] |
| max\_iter | 100 | It minimizes the sum of squared distances within clusters (inertia). Although it is showing the same result for our small dataset if we use 100 or 300 (default) but it is crucial parameter if we have large dataset.  The difference between with-in cluster similarity become negligible with increasing number of iterations” [3] |
| algorithm | Elkan | We are getting same result with Elkan and Lloyds, so we are going with Elkan. Elkan is faster in processing than Lloyd as it uses a lower & upper-bound approach. Lloyd spends a lot of processing time computing the distances between each of the k cluster centres and the n data points. Since points usually stay in the same clusters after a few iterations, much of this work is unnecessary, making the naive implementation very inefficient. However, The Elkan algorithm tends to be more efficient when the number of clusters (k) is small, and the dimensionality of the data is not too high. If you have a large dataset with a relatively small number of clusters, Elkan might offer faster convergence. [4] |



|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| Affinity Propagation | preference | -50 | The preference parameter represents the diagonal of the similarity matrix and indicates how likely a data point is to be chosen as an exemplar (cluster center).  For a common preference of −600, affinity propagation found 62 clusters.  Using this we can derived value = Number of cluster = abs (No. Of Preference/10).  According to this value we obtained the value -40.  Adjusting the damping value was giving a better number of clusters. So, We adjusted the value lower median similarities to find our number of 4 clusters by using best model silhouette score and model.label\_ for range **[-50, -40,-30,-20, -10, 0, 10]** |
| damping | 0.9 | Each iteration in the affinity propagation involved (i) revising all responsibilities based on availabilities, (ii) adjusting all availabilities based on responsibilities, and (iii) merging availabilities and responsibilities to monitor exemplar decisions and conclude the algorithm [5].  Despite attempting 10 iterations no alterations in the outcome were observed. Consequently, we undertook an iterative process on range of damping values **[0.5,0.6,0.7,0.8,0.9],** and found silhouette score to determine a suitable value to address oscillations, aiming to enhance the stability of the algorithm. |

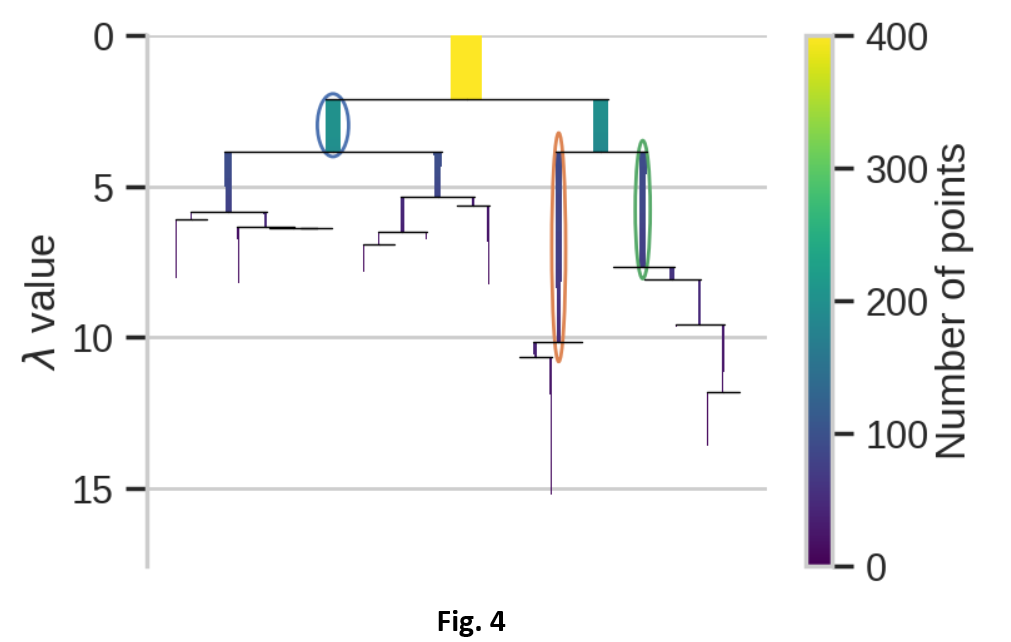
|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| DBSCAN | eps | 0.24 | First, we found the epsilon range using Euclidean with nearest neighbours. We got range as [0.10 to 0.25]. We can see in Fig. 3. Then we found the silhouette score and best epsilon value within range 0.23. We Also use number of clusters = 4 to check as DBSCAN will automatically generate clusters.[6] |
| min\_samples | 10 | These are core points of a cluster. DBSCAN will find denser clusters, whereas if it is set to a lower value, the found clusters will be sparser. As we have only 4 clusters, so we took our minimum range from -2 to 20.  Again, we used silhouette score and model labels to find the best min samples (as per our 4 clusters) [7] |



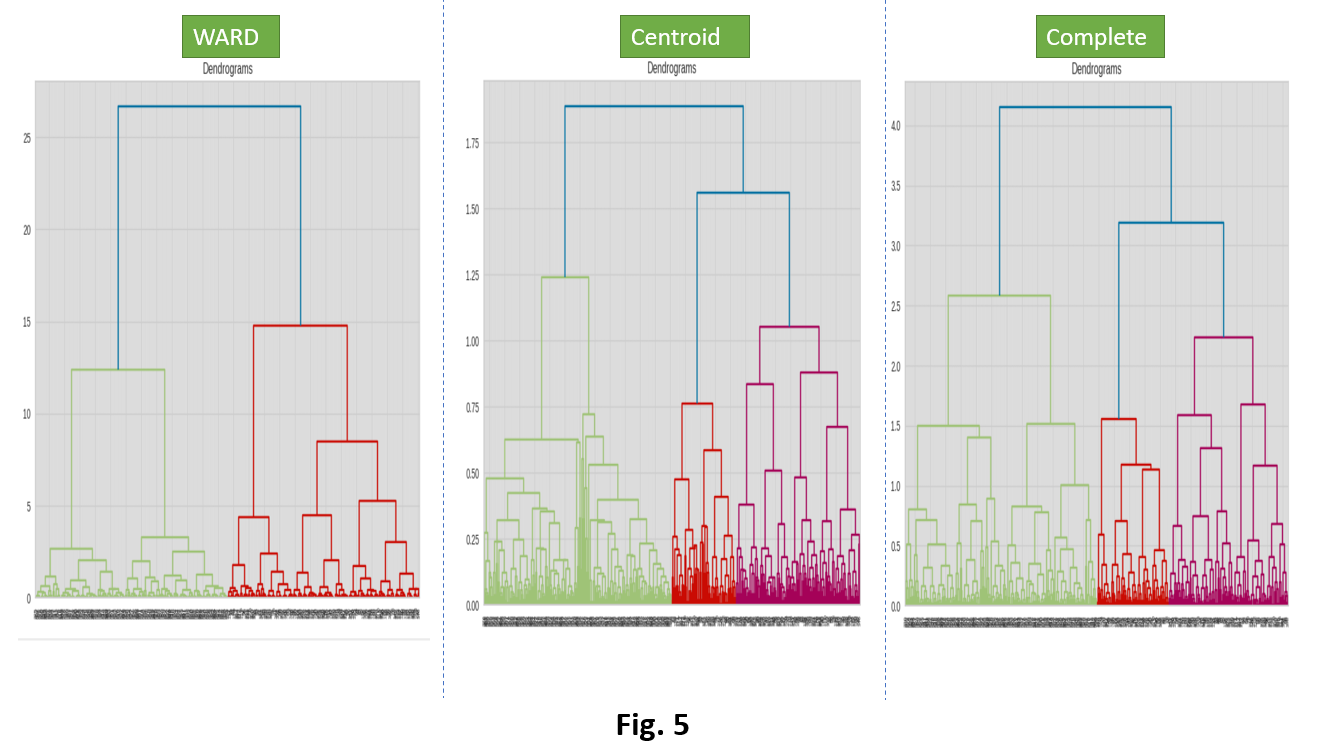
|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| Birch | n\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| threshold | 0.9 | The radius of the subcluster obtained by merging a new sample and the closest subcluster should be lesser than the threshold. Otherwise, a new subcluster is started. Setting this value to be very low promotes splitting and vice-versa.  Default value is 0.5, but that is not giving us expected clear 4 clusters rather its promoting splitting within a cluster, so we took the estimated range from 0.1 to 1 and found the best threshold using silhouette score.[8] |
|  | Branching\_factor | 350 | Maximum number of CF subclusters in each node. If a new sample enters such that the number of subclusters exceed the branching factor, then that node is split into two nodes with the subclusters redistributed in each. The parent subcluster of that node is removed and two new subclusters are added as parents of the 2 split nodes.  Adding default branching factor, is giving no different result but when using the range from 50 to 400, to find the best value. It is giving better result. Used the silhouette score metric to find best value. It explains that our dataset exhibit patterns which are better with large branching factor. [8] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| OPTICS | Min\_samples | 0.1 | The number of samples in a neighbourhood for a point to be considered as a core point. Also, up, and down steep regions can’t have more than min samples consecutive non-steep points. Expressed as an absolute number or a fraction of the number of samples (rounded to be at least 2).  Took a default range of float from 0 to 1. And found best value. [9] |
| xi | 0.009 | This is the minimum steepness on the reachability plot that forms a cluster boundary. is an extension of the "reachability" concept in DBSCAN. The ratio from one point to its successor being at most 1-xi. Took the default range of float from 0 to 1 and found best value using silhouette metric. [9] |
|  | Min\_cluster\_size | 0.009 | Minimum number of samples required for a cluster. Found the best value through silhouette score from a range of default values of float 0 to 1 [9] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| HDBSCAN | Min\_cluster\_size | 5 | This was the size which was clustered as an expected group. Lower value than this was creating noise.[10] We can see in Fig. 4 the lambda value can clearly segregate the hierarchical cluster. |
| cluster\_selection\_epsilon | 0.2 | Our data set also contains partitions with high concentrations of objects, this parameter setting can result in many micro-clusters. Selecting a value.  For cluster\_selection\_epsilon helps us to merge clusters in these regions. Or in other words, it ensures that clusters below the given threshold are not split up any further.[10]  We incremented the values from 0.5 to 0.1 and significant change observed in the noise whereas high values were reducing the same.  Obtained 0.2 with best changes in noise. |
|  | Min\_samples | 8 | Conservative Cluster we can obtain. The larger the value of min\_samples you provide, the more conservative the clustering – more points will be declared as noise, and clusters will be restricted to progressively more dense areas. The min\_cluster\_size at 60 but reducing min\_samples to 1.  With samples 8 was the best creating 4 cluster whereas samples 9 was creating 3. 8 is the threshold value obtained. |



|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| Agglomerative Clustering | N\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| linkage | ward | Ward is better as it causes minimum increase in information loss. Moreover, complete merges the close groups of clusters (as it determines the most significant distance between any two objects in the different clusters). [11]  We have plotted Dendrograms to find the WARD method was giving separated clusters comparing to others methods like centroid and complete shown in Fig. 5. |

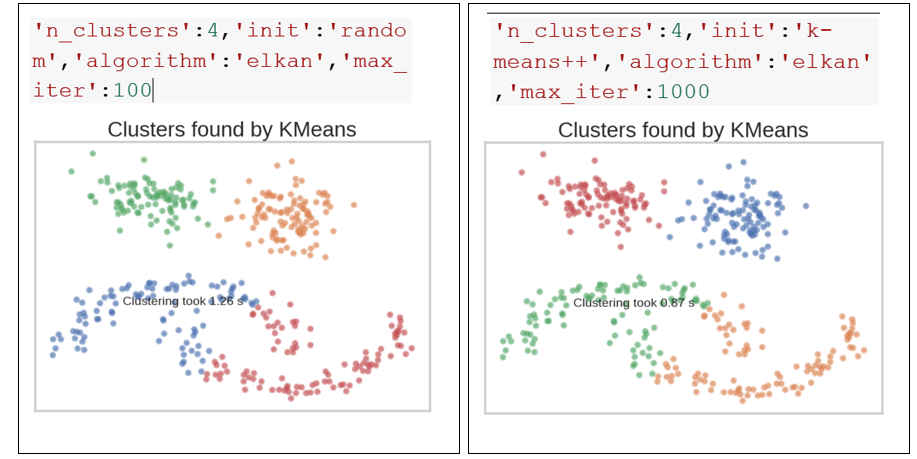


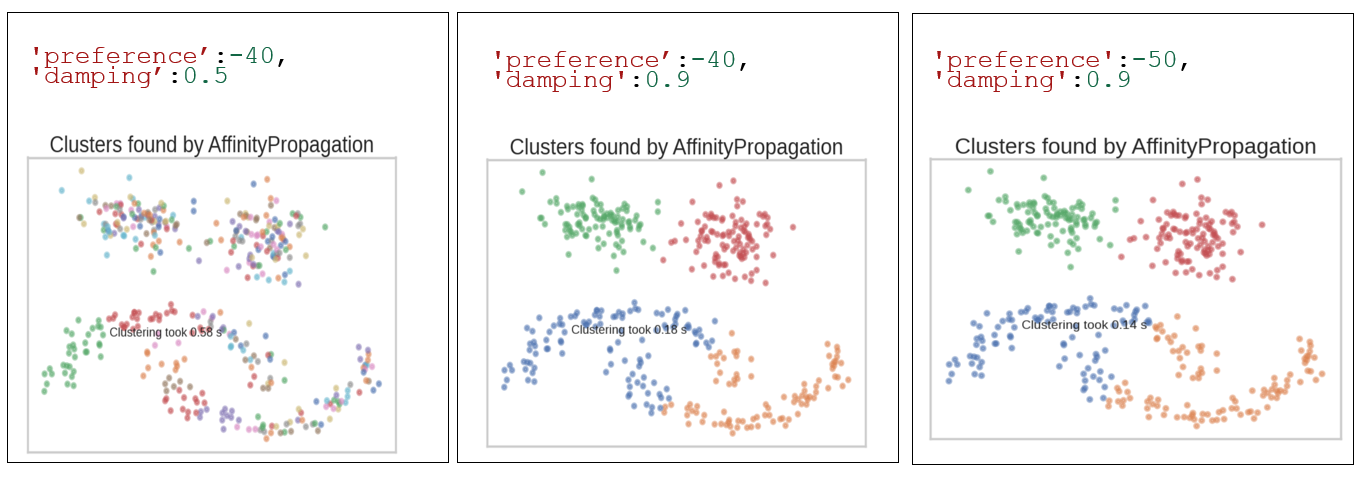
|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm Name** | **Parameter** | **Best values** | **Justification** |
| Spectral Clustering | N\_clusters | 4 | As explained above: We can observe 4 distinct clusters. The best numbers of clusters obtained from elbow method [1] |
| affinity | rbf | Nearest\_neighbor is computed based on the K-nearest neighbors of each data point. It connects each point to its K nearest neighbors in the dataset.  RBF measures the similarity between data points using the exponential of the negative squared Euclidean distance. We calculated the silhouette score for them and got the RBF as best. [12][13] |
| gamma | 60 | We can use the Gamma that controls the width of the Gaussian kernel in the RBF affinity matrix.  As our clusters are shaped in smile structure, we can smooth them using gamma.  Default value is 1.0 A higher gamma results in a narrower Gaussian distribution, making the kernel more sensitive to the distance between data points. |

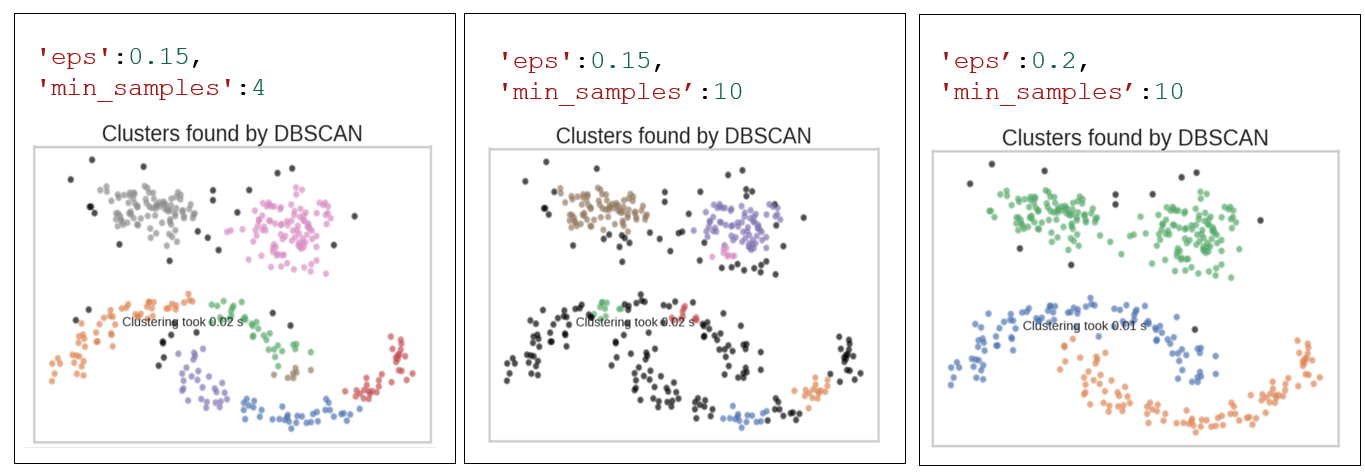
1. Intra-Cluster versus inter-cluster distance

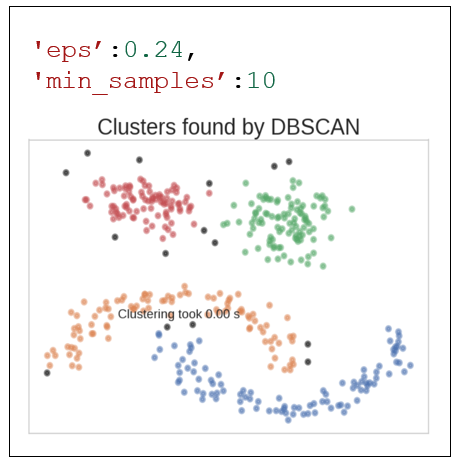
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **No. Of Clusters** | **Noise/Outlier Detection** | **Silhouette Score** | **Compulational Speed in Sec** | **Visual Pattern Match** | **Intra Cluster** | **Infer Cluster** | **Noise Proportion** |
| K-Means | 4 | No | 0.55 | 0.22 | No | Yes | No |  |
| Affinity Propagation | 4 | No | 0.55 | 0.14 | No | Yes | No |  |
| DBSCAN | 4 | Yes | 0.45 | 0 | Yes | Yes | Yes | [-1 0 1 2 3] [ 15 98 97 100 90] |
| BIRCH | 4 | No | 0.54 | 0.04 | No | Yes | No |  |
| OPTICS | 4 | Yes | 0.46 | 0.29 | No | Yes | No |  |
| HDBSCAN | 4 | Yes | 0.42 | 0.02 | Yes | Yes | Yes | [-1 0 1 2 3] [31 94 97 86 92] |
| Agglomerative | 4 | No | 0.53 | 0.01 | No | Yes | No |  |
| Spectral | 4 | No | 0.54 | 0.69 | Yes | Yes | Yes |  |

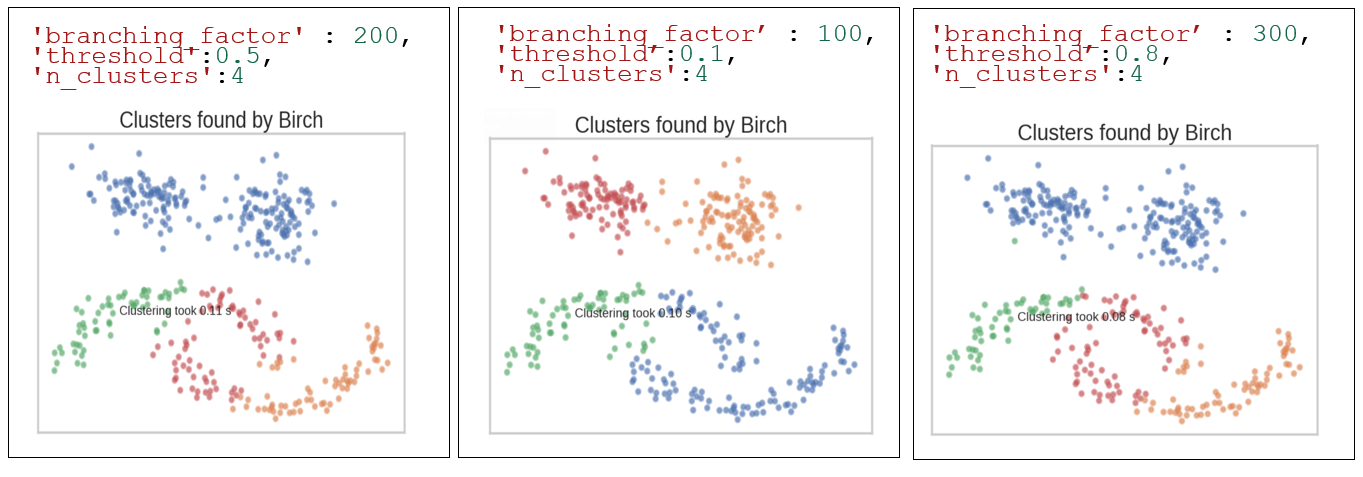
1. The documented scatter plots.

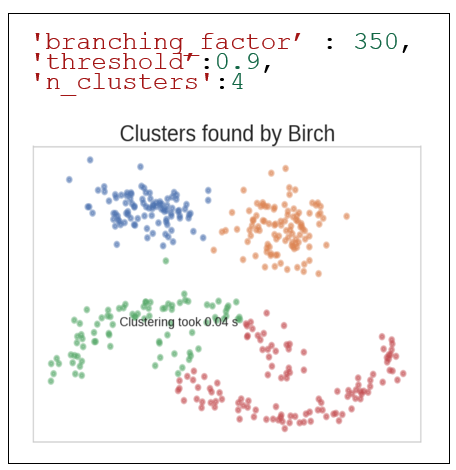


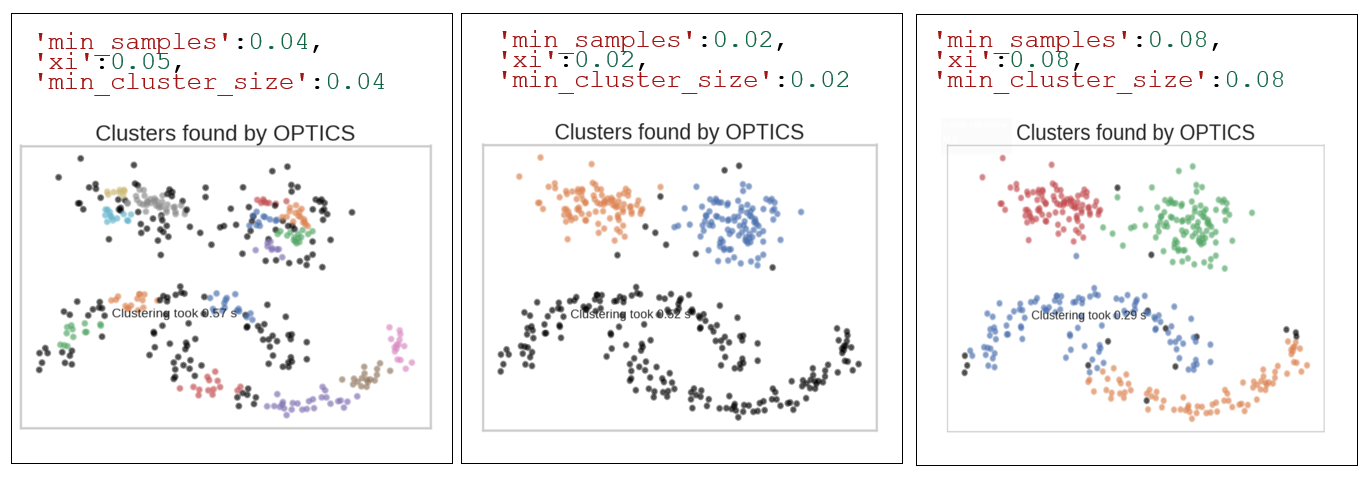


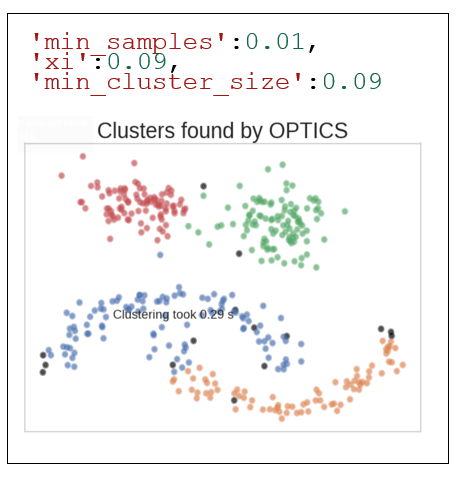


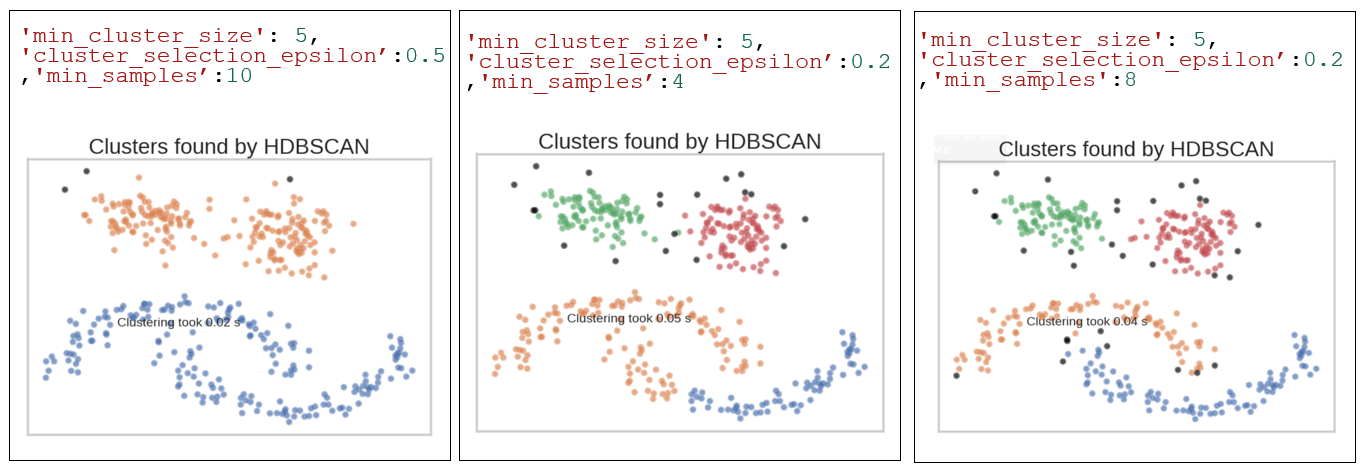


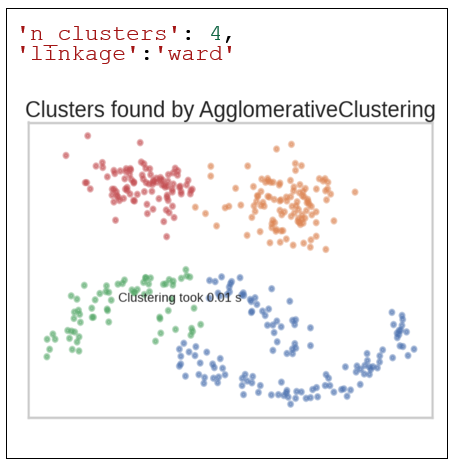














1. **Conclusion** – After doing EDA the DBSCAN performs out well compared to other algorithms.

According to our requirement DBSCAN out grows in following parameter.

* Pattern Matching – Non-Globular => DBSCAN identifies clusters based on the density of data points. It is effective in discovering clusters of arbitrary shapes and sizes, making it suitable for datasets with irregular cluster structures.
* Computation speed is best and can work with the Big Data=> DBSCAN does not require storing the entire dataset in memory during the clustering process. It processes data points sequentially, making it memory-efficient and scalable to large datasets.
* Able to capture the Noise or Outliers as well – In reality there will be more outliers in the data. The Pattern is different to identify the same Outliers works exception well. => DBSCAN is robust to outliers and noise in the data. It can identify and label outliers as noise, preventing them from being assigned to any cluster. This makes DBSCAN suitable for datasets with varying levels of noise.
* Silhouette Score is good Greater than 0

