CSE506: Data Mining

Assignment 2

Group 19.
Group Members: Ananya Kansal (2019458), Avishi Gupta (2019155), Jahnvi Kumari (2019469), Manvi Goel (2019472), Prachi Goyal (2019186)

Question 1

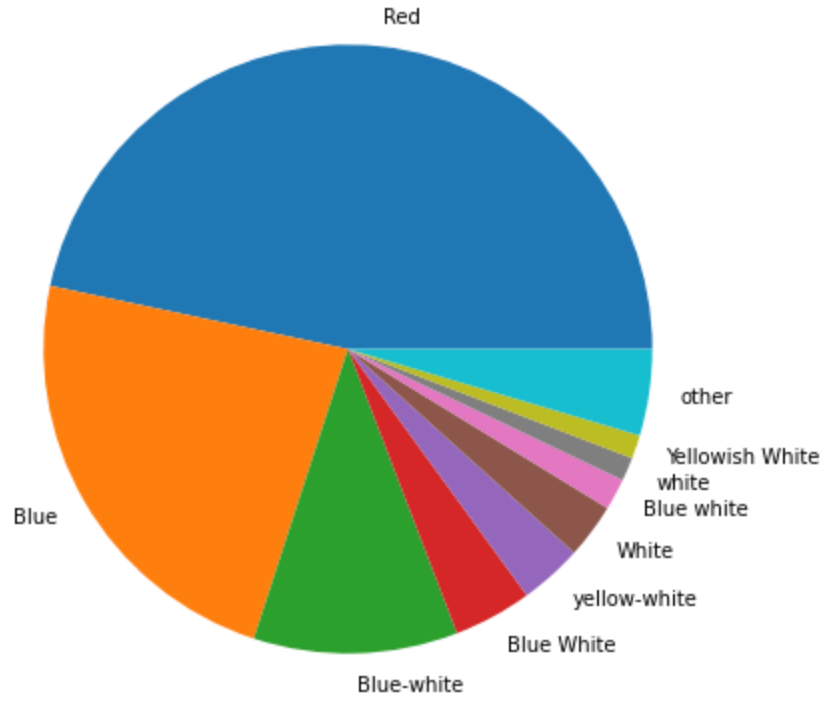
Dataset 1

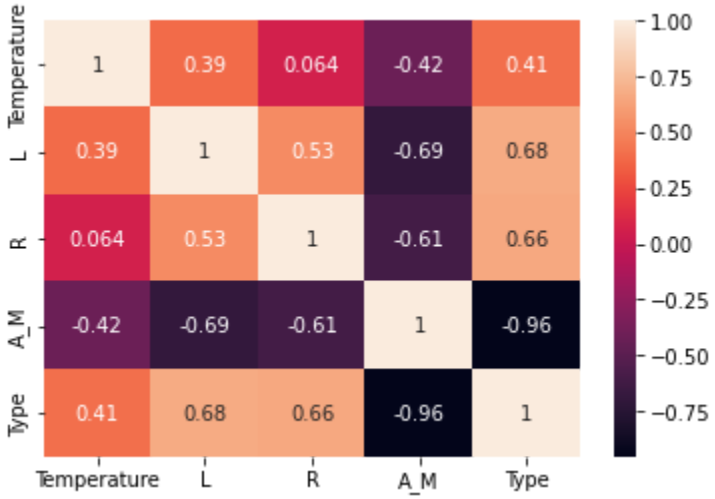
Q1 a. The dataset used is the star-type classification dataset. It was downloaded from kaggle and contains 240 data points. There are 5 target variables corresponding to the 6 types of stars - Red Dwarf, Brown Dwarf, White Dwarf, Main Sequence, Super Giants and Hyper Giants. Following is a brief description of the columns:-

1. Temperature: is the temperature of the star in Kelvin.
2. L: is the star's luminosity in units of Sun's average luminosity Lo = 3.828 x 10^26 Watts .
3. R: is the star's average radius in units of Sun's average radius Ro = 6.9551 x 10^8 m.
4. Am: Absolute magnitude
5. Color: is the general color of the spectrum
6. Spectral Class: Clustering has been applied after removing this column of target variable and extrinsic evaluation judges how correlated the clustering is to the target variable.

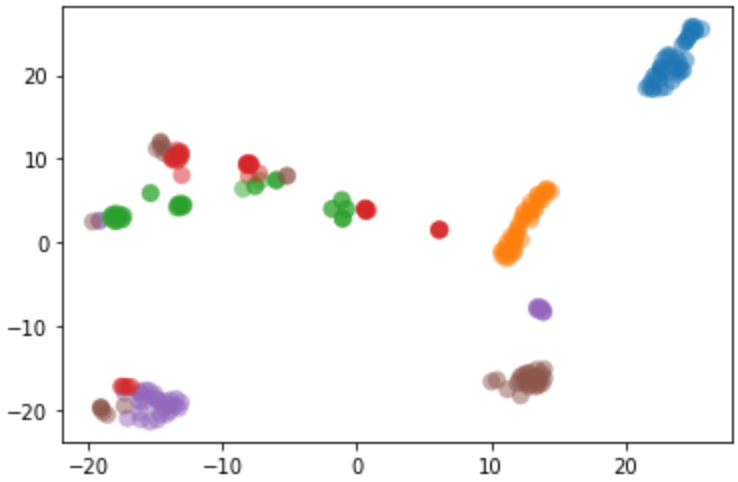
Dataset Analysis

a. **Preprocessing:** After dropping the types or the target column, the columns containing categorical data like Color and Spectral Class were one-hot encoded and the numerical attributes like temperature and a_m were normalized.
b. **EDA:**





c. **TSNE of the datapoints colored according to the type of star**



Analysis: Density based clustering is often used in stellar observations to detect the similarity between stars, nebulas, galaxies etc. (Ref https://ceur-ws.org/Vol-3036/paper17.pdf). From the EDA, we observe that red is the most prominent color of the spectrum. We also notice that there is a strong inverse correlation between the mass of the star and it's type. The TSNE shows that there are some star types may have varied behaviour. This means that the clusters are not globular and have varied shapes. Most clusters are dense proving the need for a density based clustering algorithm but we also observe a variation in the density of the clusters hence promoting the need for hierarchical density based clustering.

Learning Section and Steps to recreate
In this question, I learnt about a new algorithm HDBSCAN and its working. I also learnt about extrinsic and intrinsic methods of evaluating unsupervised learning algorithms. You can recreate the result by adding the dataset data-t4.csv into the runtime of the colab file Question2 Colab File.

Dataset 2

We choose a dataset of Customer Classification of a Telecommunication company.
The dataset has 12 attributes and 4 classes (A, B, C, D) that specify different types of customer.

Hierarchical clustering is suitable for the dataset as the data is aimed at finding similarities between different types of consumers. Agglomerative clustering allows us use a taxonomical classification for the consumers which allows us to make decisions and multiple levels.

Also, as customers can often be represented in hierarchies, a dendrogram analysis is usually the most suitable approach to go about it.

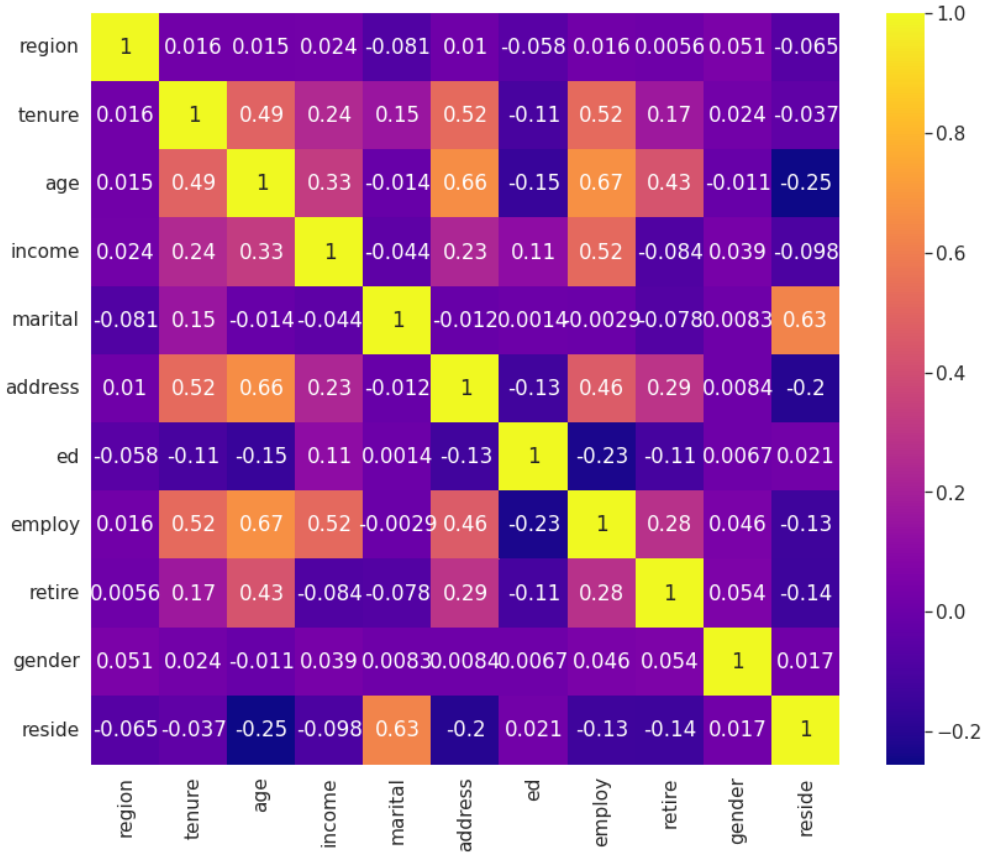
Multiple studies in the domain also suggest the same (1, 2, 3, 4).

Exploratory data analysis and Preprocessing.

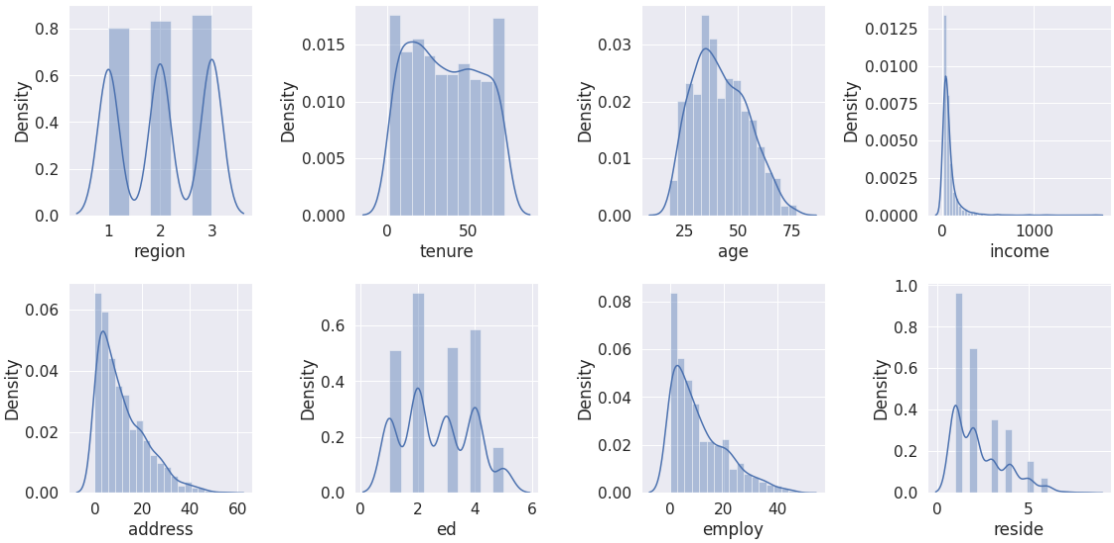
The dataset has 1000 rows and 12 columns.

It does not contain any null values.

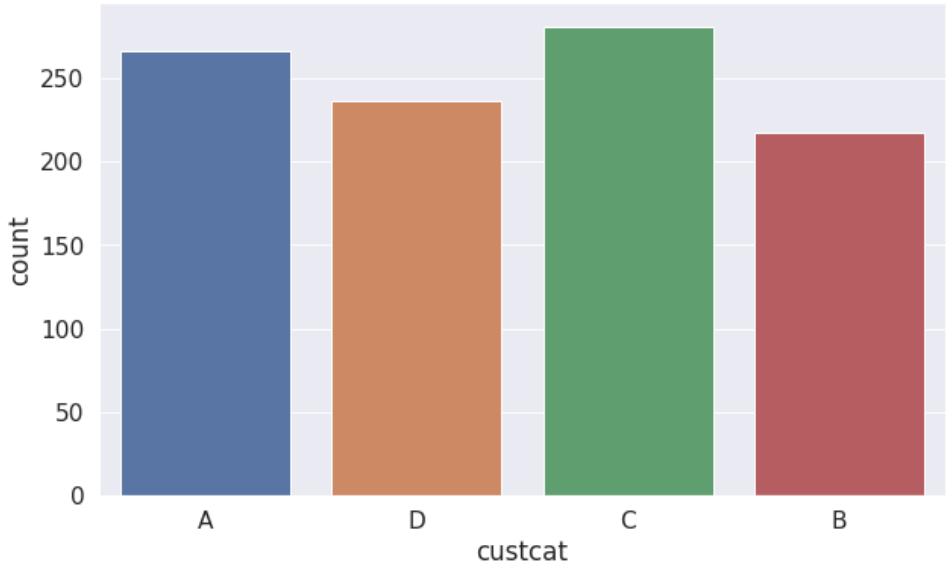
We plot a heatmap to understand the correlation between different variables



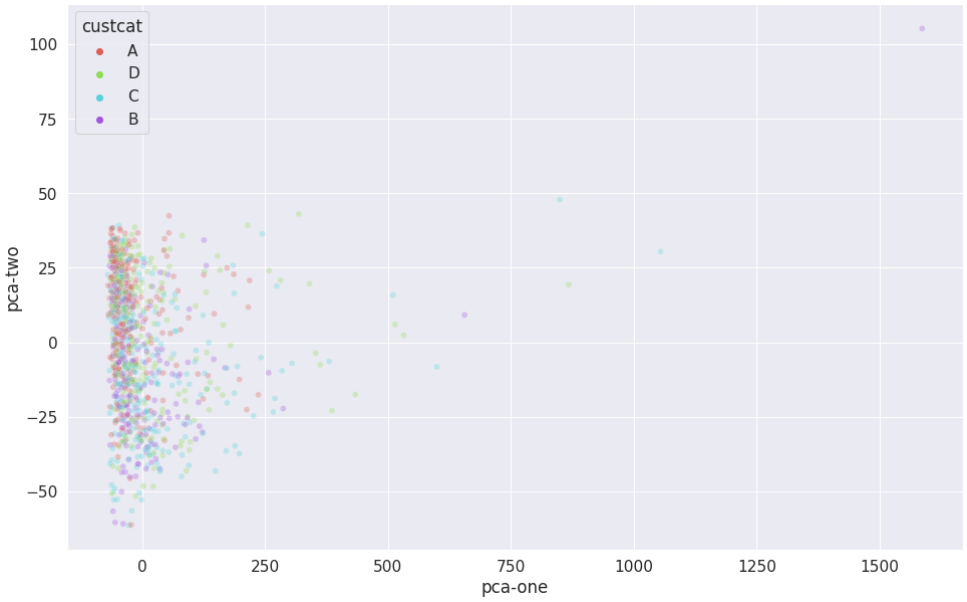
We also plot the distribution for all numerical columns.



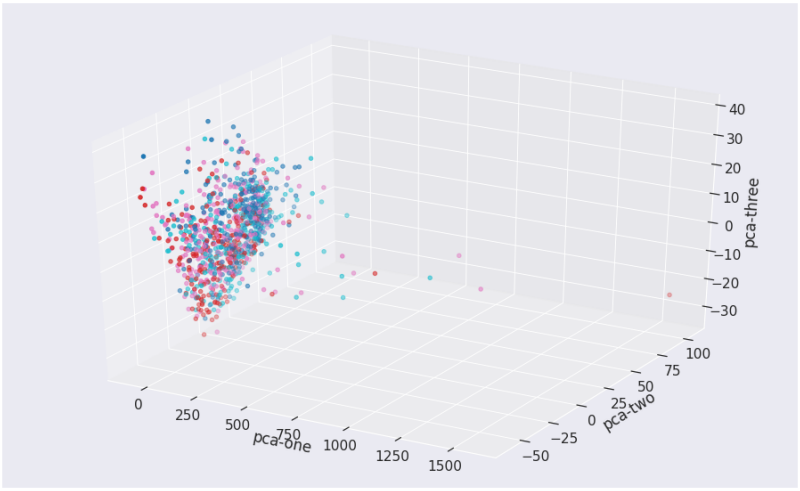
To understand the distribution of class variable, we plot a bar graph



We also plot the points using PCA and TSNE
PCA - 2D

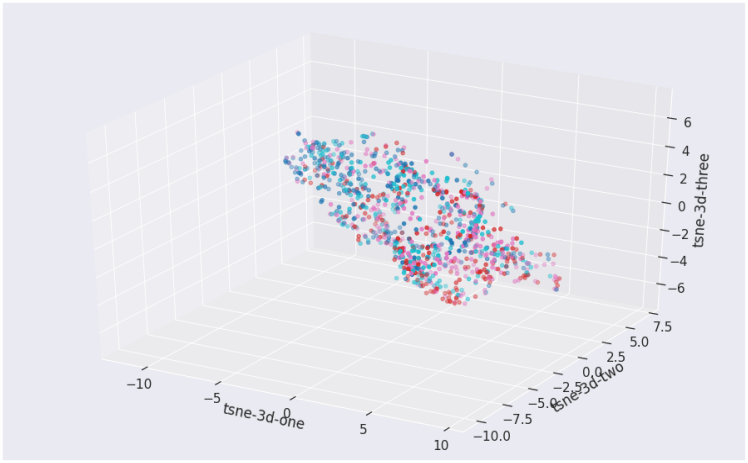


PCA 3D



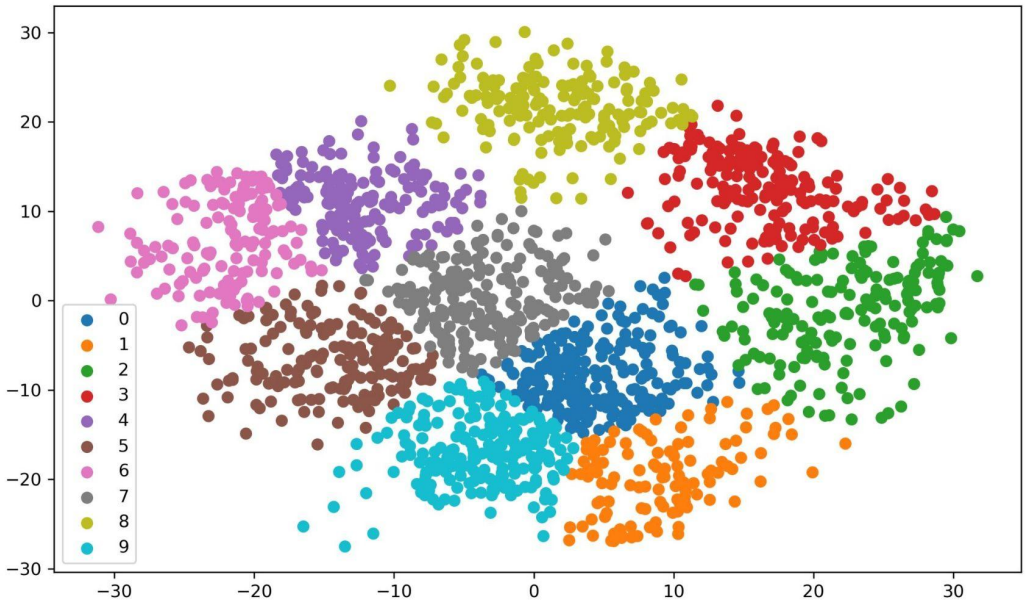
T-SNE - 2D

T-SNE - 3D



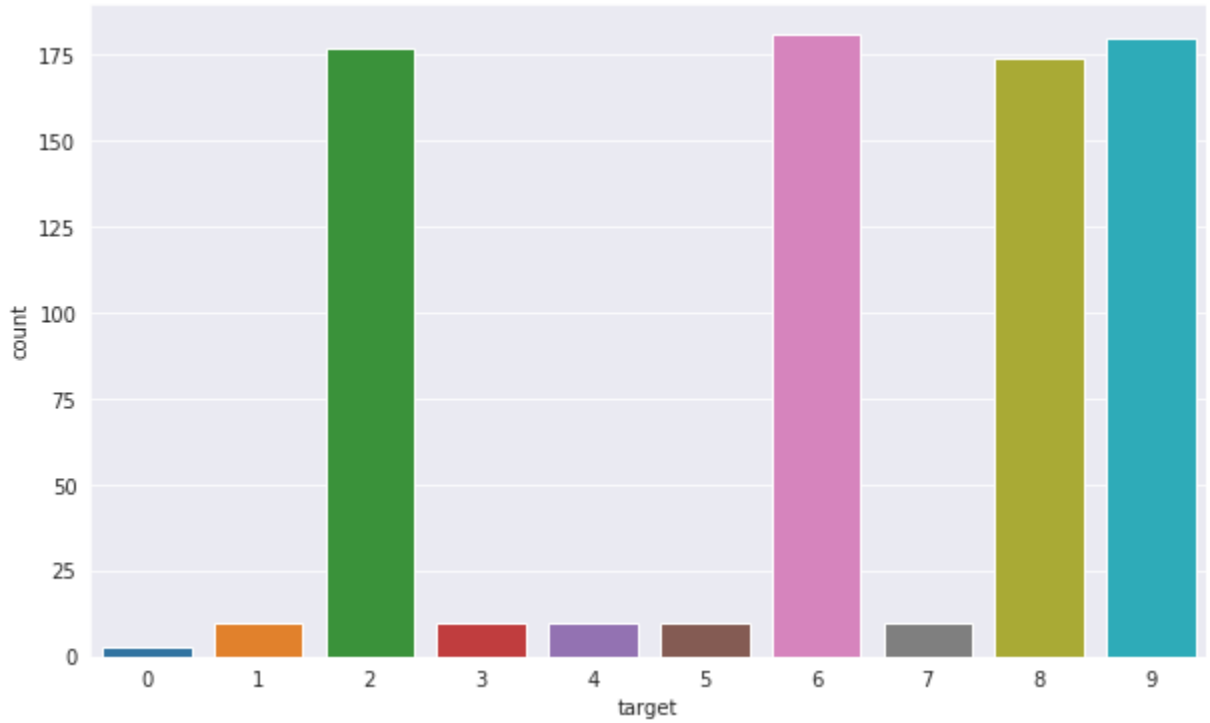
Dataset 3

We chose a digits dataset for prototype-based clustering as it is globular in nature (as shown below). Prototype-based clustering is a center based clustering that performs best with globular dataset.



For introducing outliers: We chose the digits 2, 6, 8, and 9 as target classes. Then we selected a few points from the remaining labels to introduce outliers. We chose the target classes such that the clusters are as far as possible for quality clustering.

Target count plot for the final data:



Question 2

Q2 b.
Advantages

- DBSCAN can handle clusters of arbitrary shapes and sizes. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
- By categorising points as core, non-core and outlier, it has a notion of noise which makes it robust to outliers.

Disadvantages:

- Points on the borders which can be reached from more than one cluster can be a part of either cluster. Hence their cluster is based on the order in which the data is processed. This makes DBSCAN not entirely deterministic even though this situation is relatively uncommon.
- DBSCAN can be expensive when the computation of nearest neighbours requires computing all pairwise proximities, especially for high-dimensional data. On the other hand, common distance metrics especially Minkowski metrics can make it difficult to find an appropriate ε for sparse data due to the curse of dimensionality.

Q2 c.
HDBSCAN is a density based clustering algorithm which extends DBSCAN and converts it into a hierarchical clustering algorithm. DBSCAN locally approximates the density ie. f there are more than min_points data points in the epsilon bal(user parameter)l, then the region is declared dense. HDBSCAN **estimates** the density. It answers the question - after drawing circles of infinite number of radii, what radius would i take before dbscan declares it dense ie. which radius will enclose min_points and maps them on the points. The connected components of level sets form a tree. Under the hood, HDBSCAN closely resembles computing single linkages or minimum spanning trees. An efficient algorithm to make a MST for dense graphs (our use case) is Burovka's algorithm which uses spatial index trees for nearest neighbour queries. Thus the runtime is practically reduced to O(nlogn). It uses a notion of mutual reachability distance which wants the points not only to be close in the euclidean space but also be dense. A flat cut through the generated dendogram is equivalent to running dbscan for a given epsilon parameter. Finally, it chooses the most persistent clusters ie. clusters that live for long and have many points

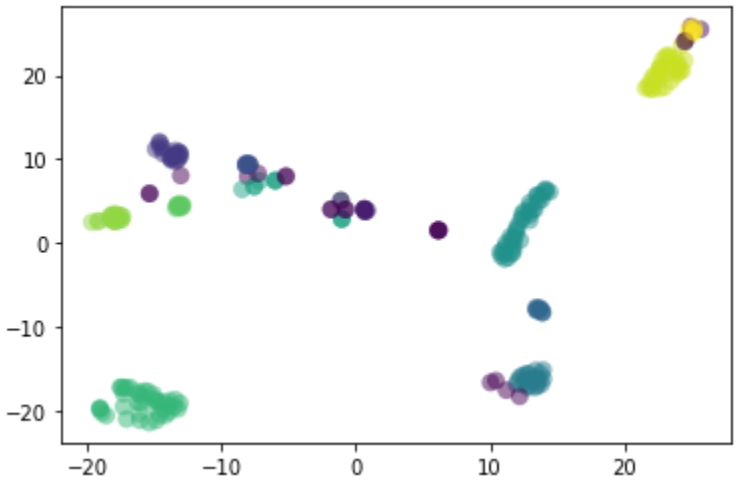
Two of its advantages over DBSCAN are:-

- Being a hierarchical clustering algorithm, it creates a natural hierarchy over the clusters doing away with the resolution parameter epsilon in DBSCAN. This means HDBSCAN doesn't assume any prior knowledge of the data.
- Theoretically, both DBSCAN and HDBSCAN are $O(n^2)$ algorithms and can be sped up to O(nlogn), the time taken by DBSCAN is highly dependent on epsilon; for smaller epsilon, DBSCAN would classify more points as noise and would require less distance

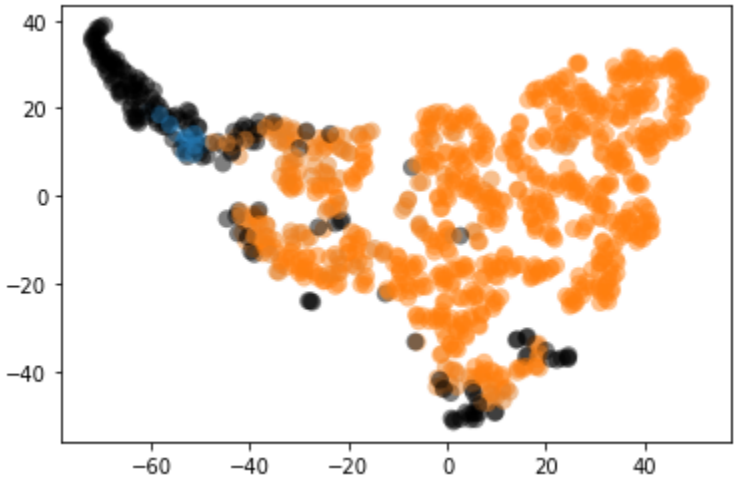
computations than a higher epsilon. Though a single run of DBSCAN is slightly faster than a single run of HDBSCAN, one run of HDBSCAN is equivalent to running DBSCAN multiple times for all possible values of epsilon. Hence the most time and effort saving method is to run HDBSCAN and find the flat cut (epsilon value) suitable for the usecase.

Q2 d.
The following diagram shows the color coded clusters for the three datasets:-

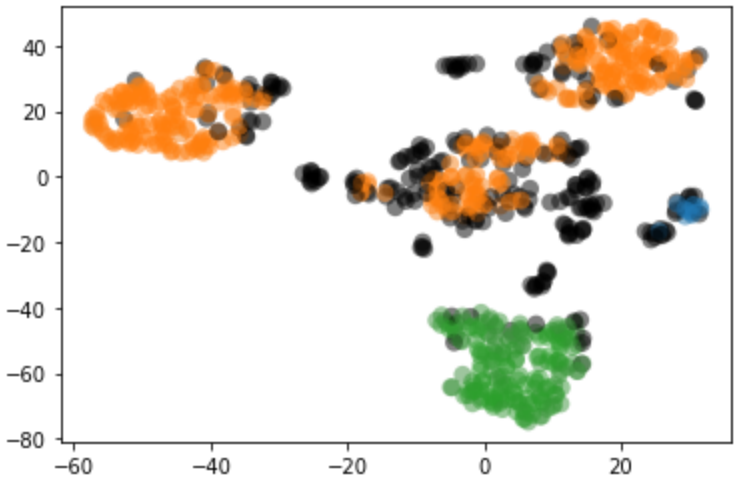
Dataset 1:



Dataset 2



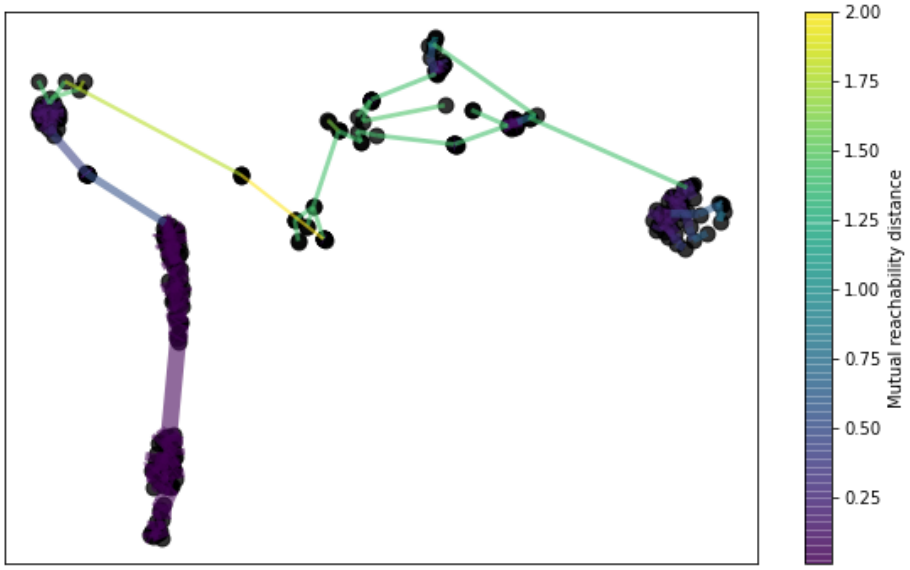
Dataset 3



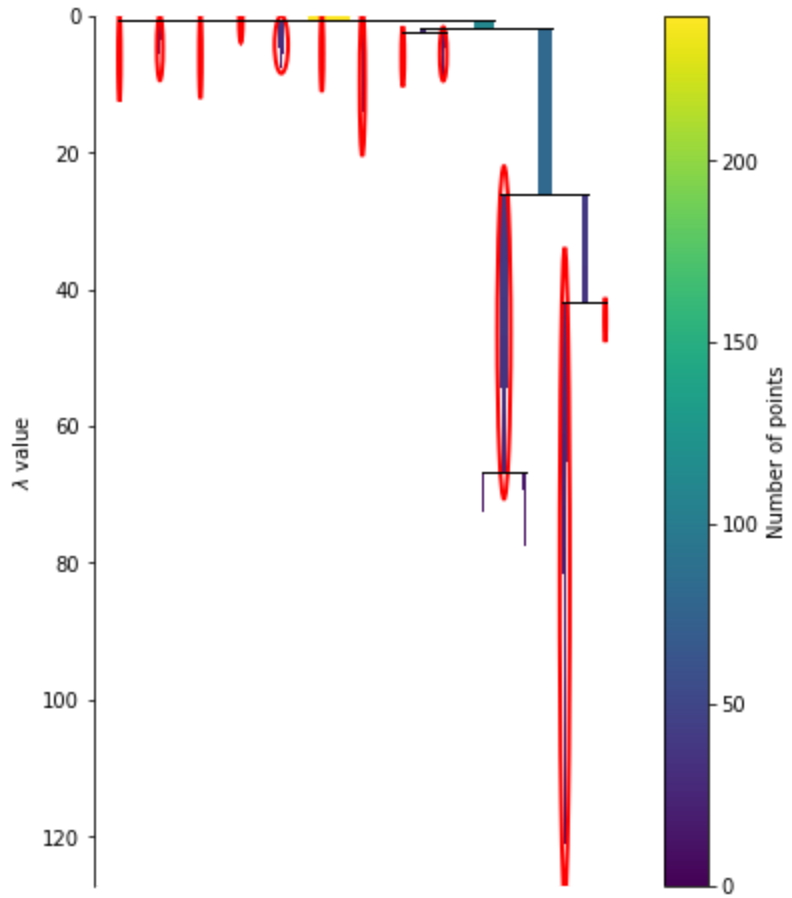
Q2 e. The following parameters were changed:-

Control experiment

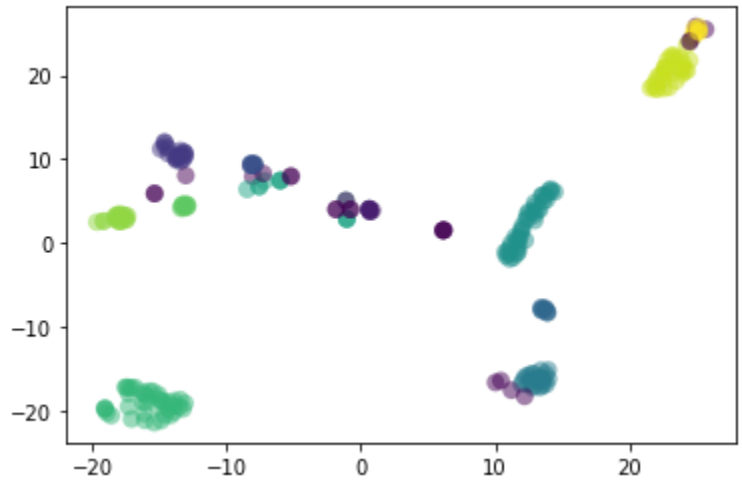
Minimum Spanning Tree:



Condensed density Cluster Tree

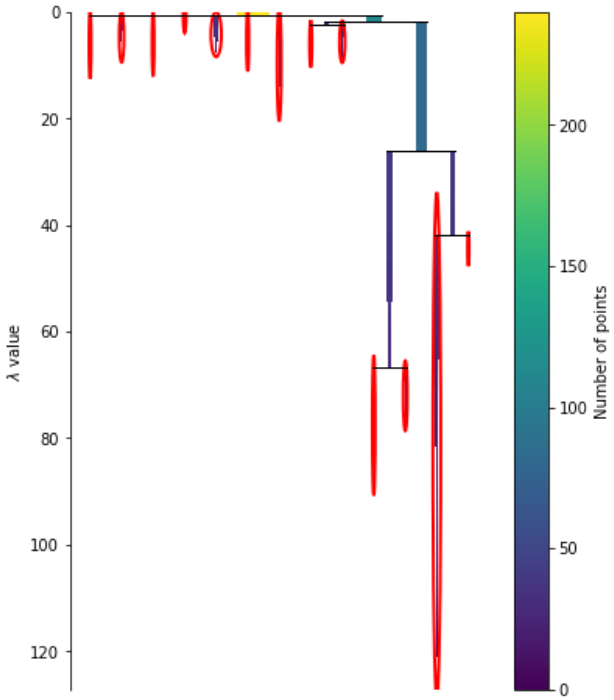


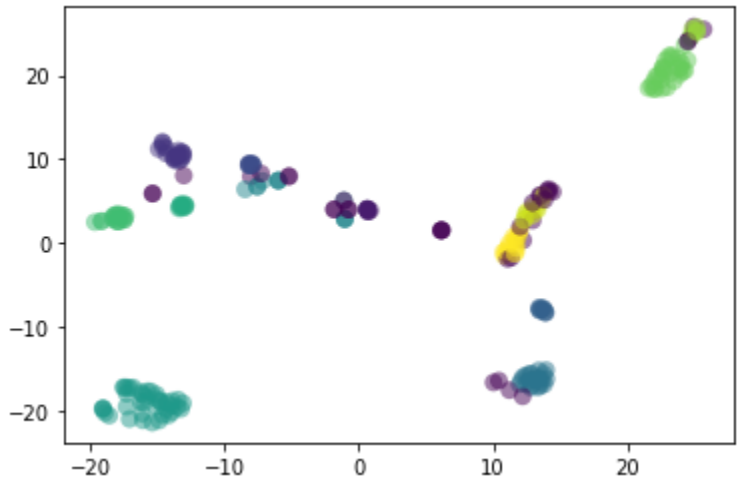
TSNE plot with color coded clusters



1. **Cluster selection method changed from eom to leaf.** The standard approach for HDBSCAN is to use an Excess of Mass algorithm to find the most persistent clusters. Alternatively we can select the clusters at the leaves of the tree – this provides the most fine grained and homogeneous clusters.

Condensed cluster tree and color coded TSNE plot

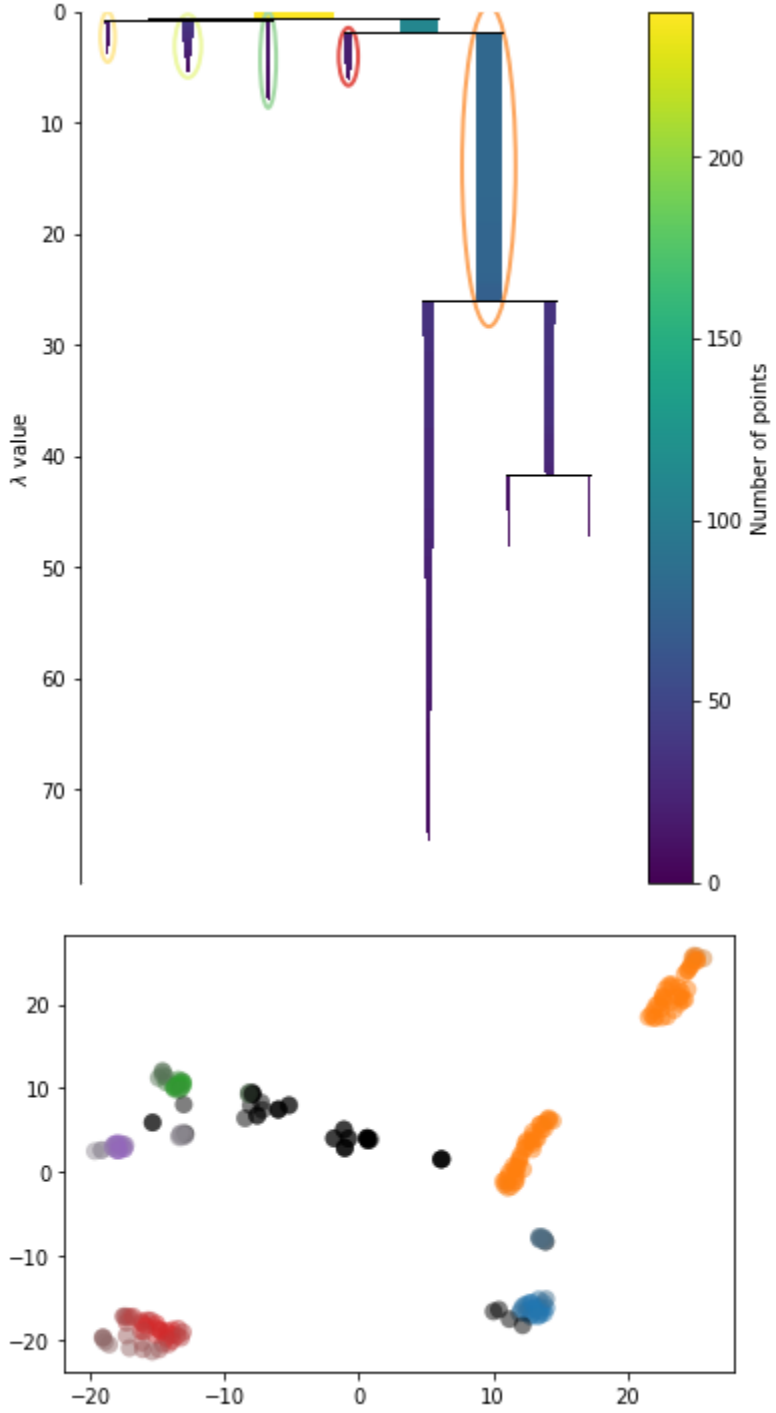




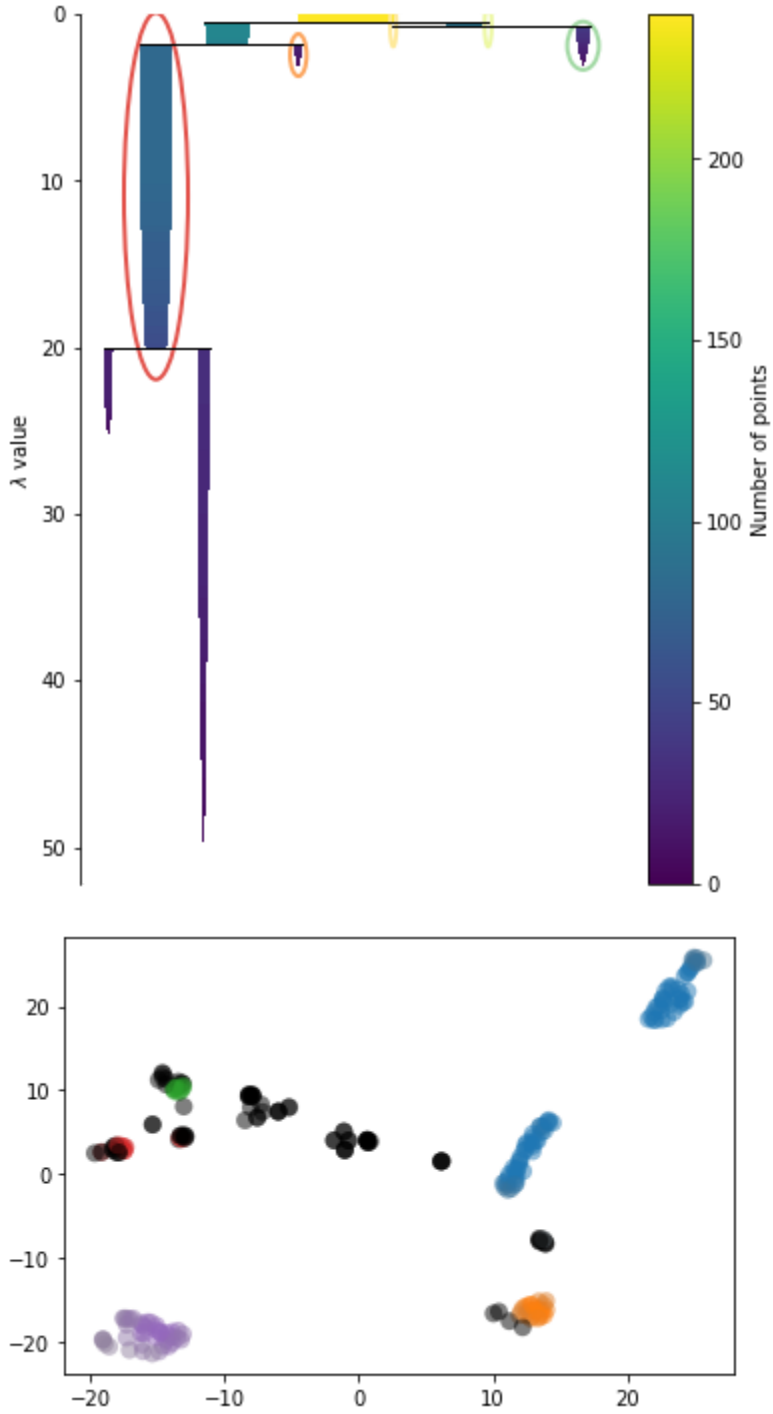
Analysis: We notice that the algorithm now forms the clusters from the leaves in the tree hierarchy rather than selecting the most persistent clusters. As a result, some clusters are further broken down into smaller clusters (see the new yellow cluster).

2. **Changed the min sample size** (number of samples in a neighbourhood for a point to be considered a core point)

a. **min sample size = 10**
Color coded condensed cluster tree and TSNE plot



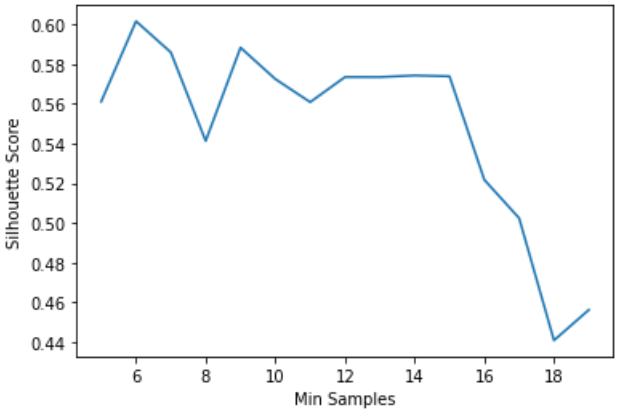
b. **min sample size = 20**

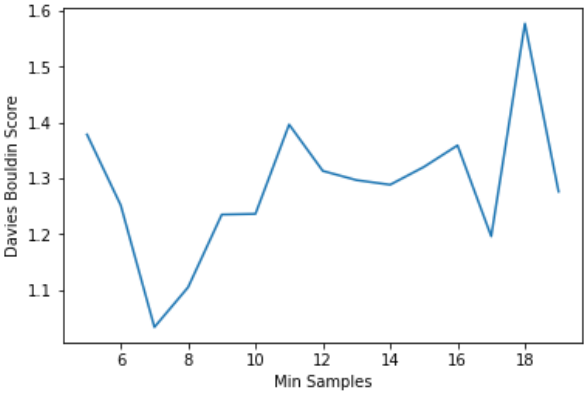


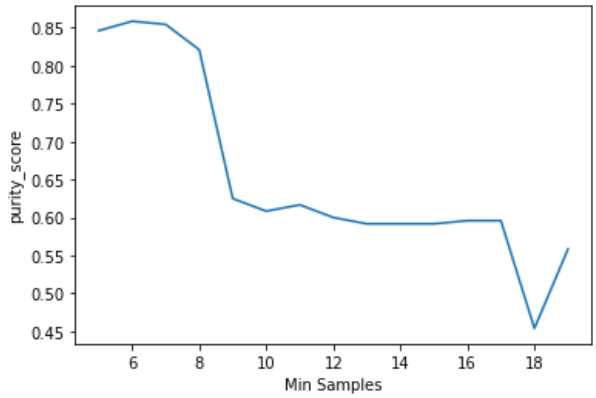
Analysis: We observe that some points which were earlier included in the clusters due to them or their neighbours being a core point are now classified as outliers. This is because the criteria for being a core point changed from having 20 neighbours to having 40 neighbours. As min sample size increases, we will observe more outliers. **The minimum sample size = 10 gives**

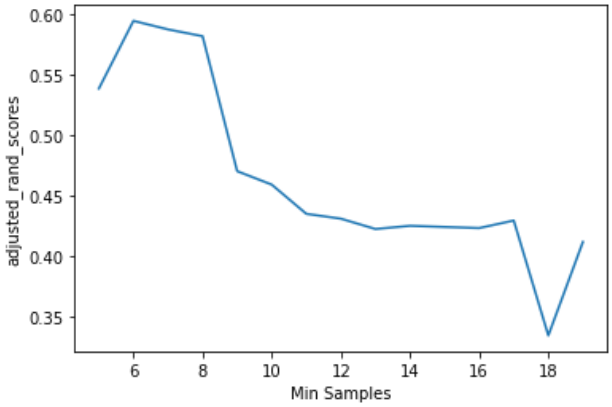
the best evaluation scores for our experiment but we observe that the number of clusters has increased by 1 and there is an increase in the number of points deemed outliers**.**

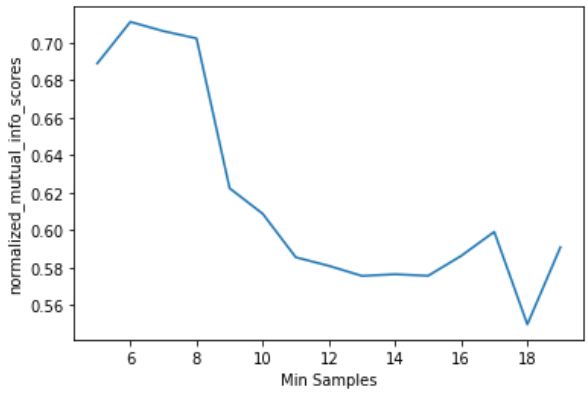
The general trend for the minimum sample size on the parameters is as follows:-





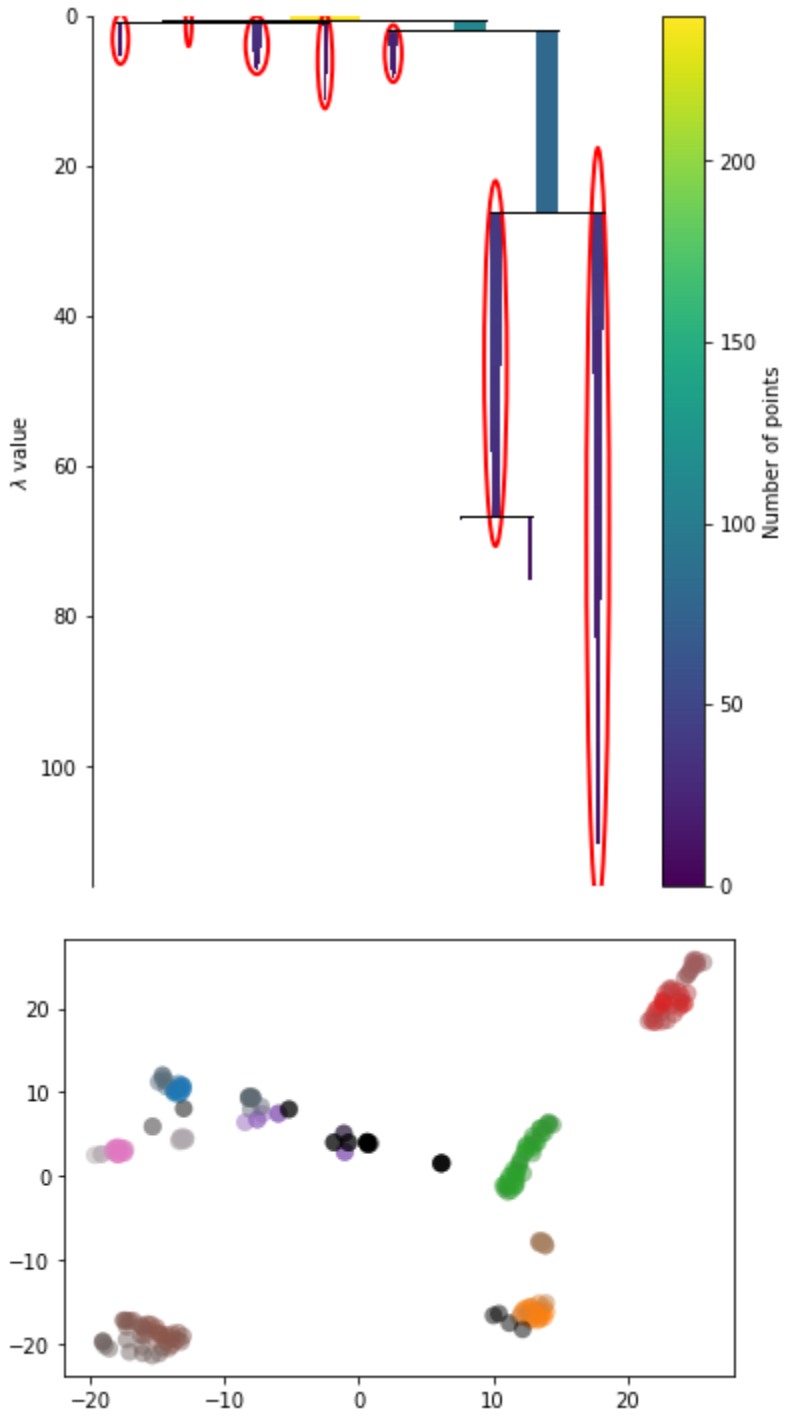




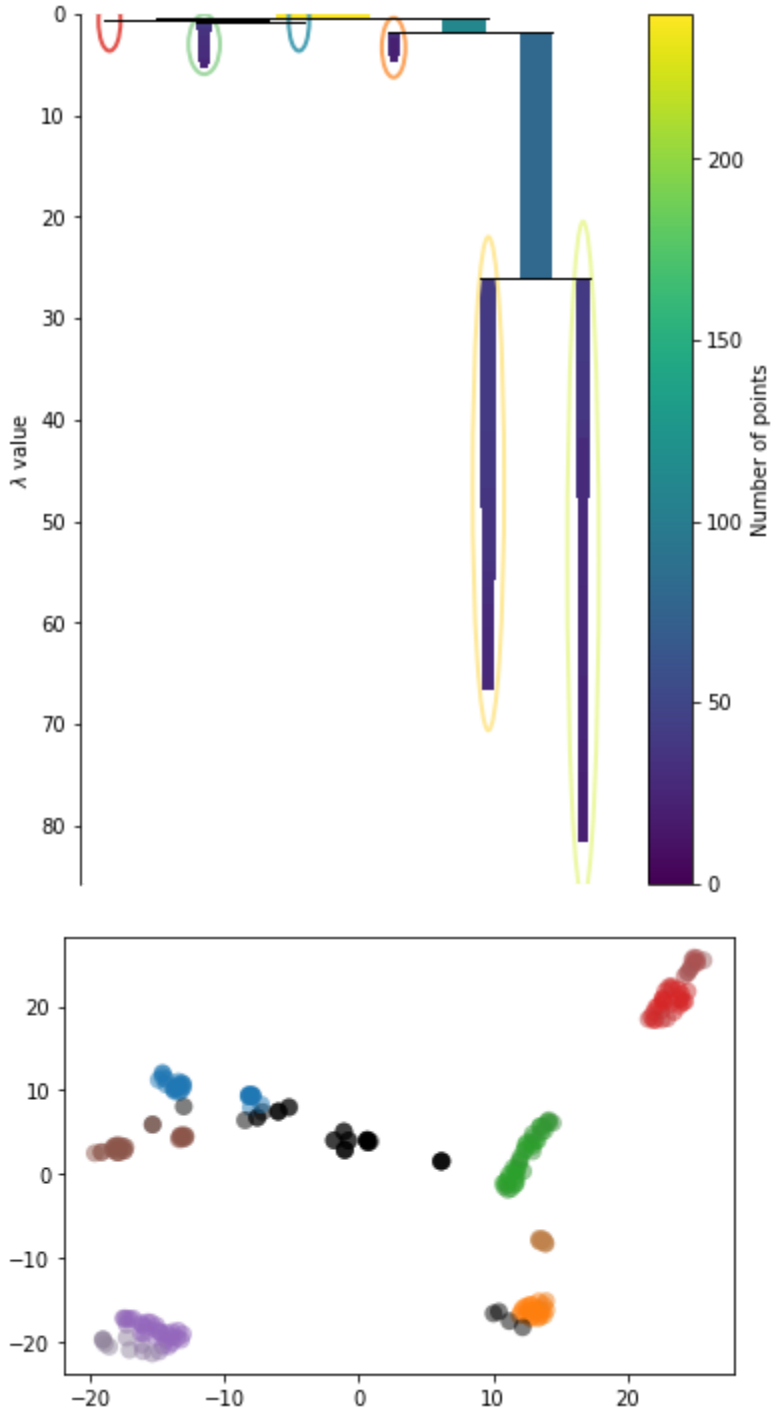


3. **Changed the minimum cluster size**: Single linkage splits that contain fewer points than this will be considered points “falling out” of a cluster rather than a cluster splitting into two new clusters.

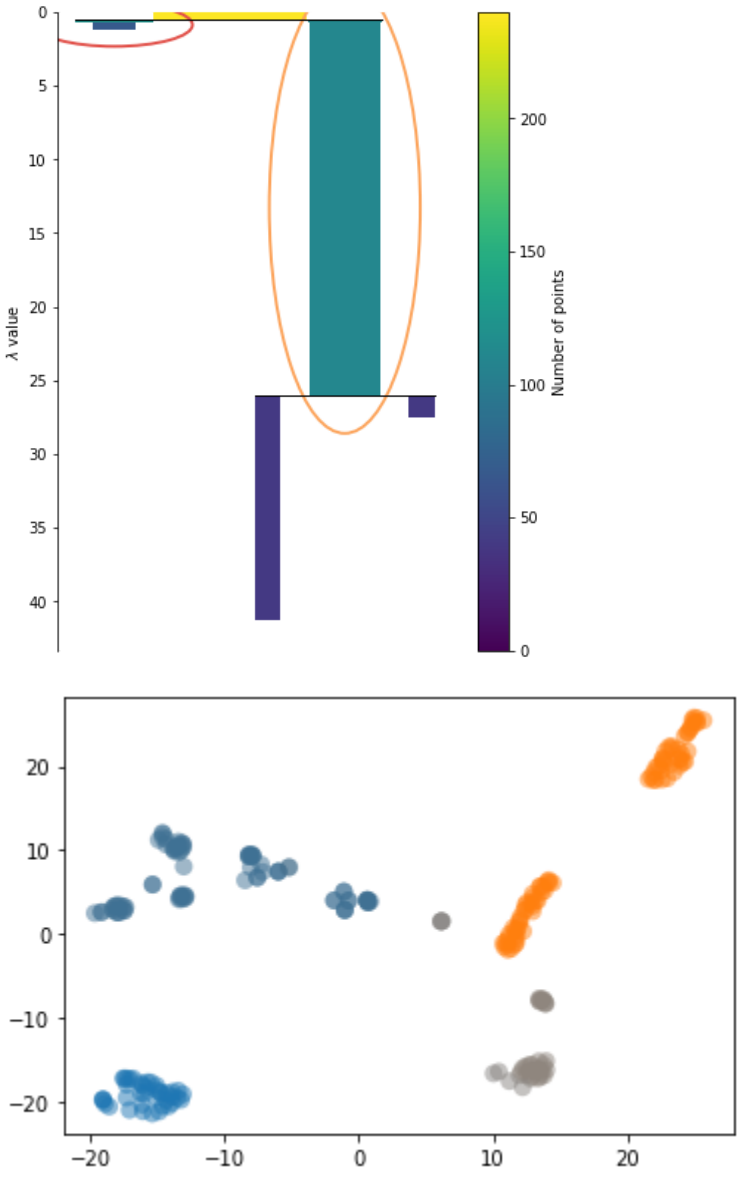
    a. **Min cluster size =** 10



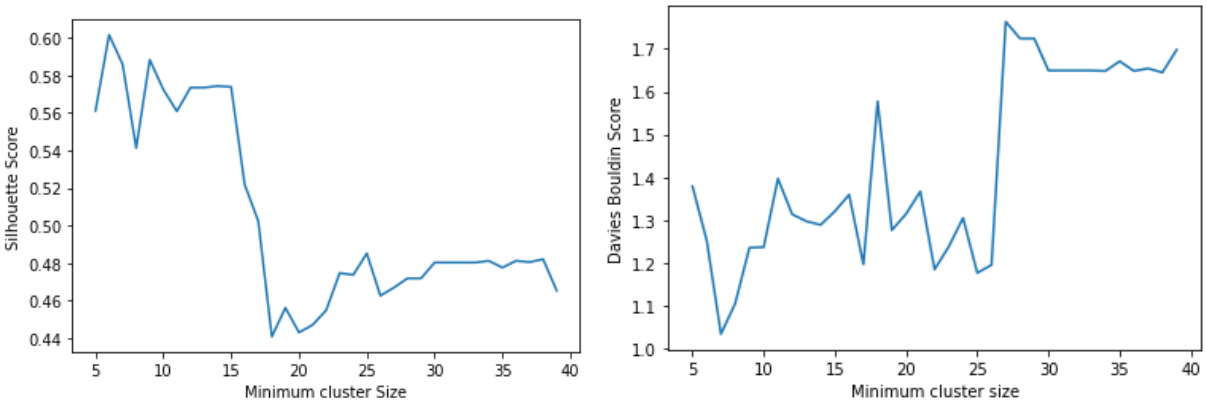
b. **Min cluster size =** 20



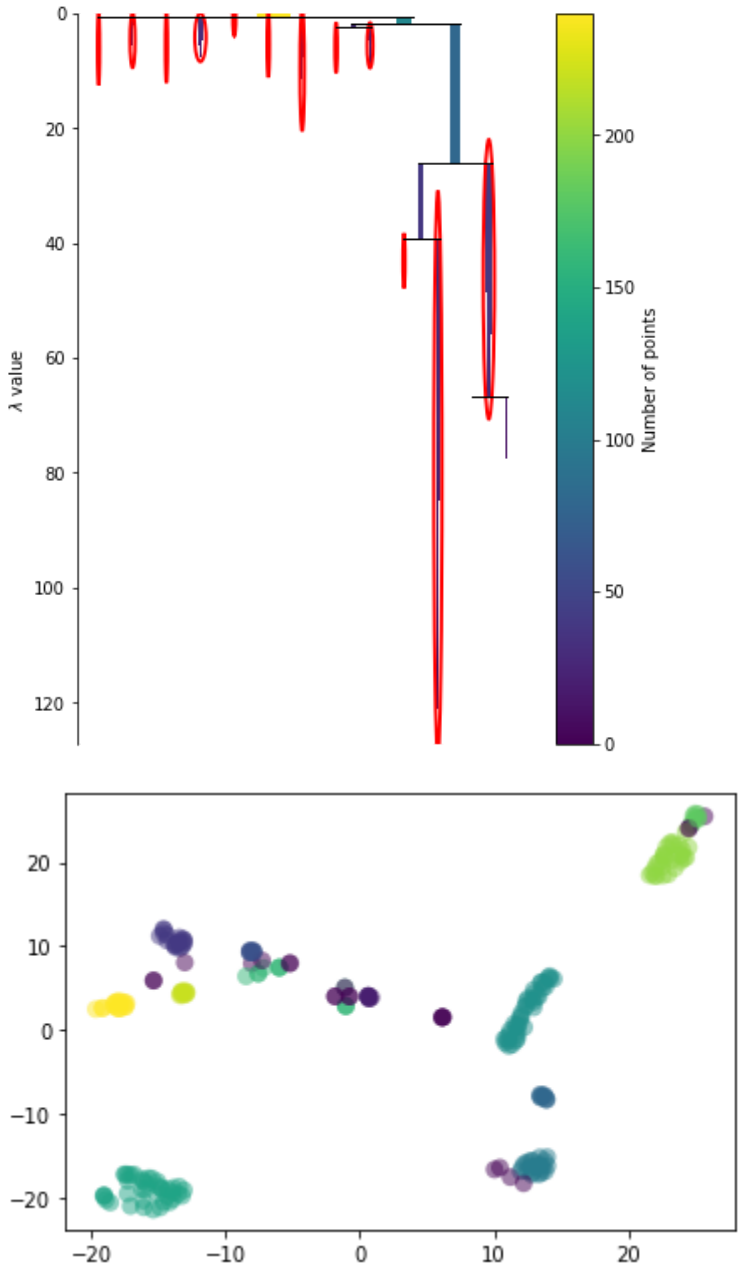
c. **Min cluster size =** 40



Analysis: We observe that as expected, on increasing the minimum cluster size the threshold number of points in a single linkage for a cluster to divide increases. As a result, the most persistent clusters chosen by HDBSCAN contain a large number of points (see the width of the dendrogram tree). As a result the number of clusters reduces. Also, since points are less likely to fall out of clusters, the number of outliers also decreases (25 in min cluster size=10 to 0 in min cluster size=40).



4. **Changed algorithm:** The implementation by default selects the best algorithm looking at the nature of the data. But, we can manually set the algorithm.
    a. Algorithm: We showed only the Boruvka tree algorithm because it was giving a better Silhouette score than ‘best’.



Analysis: Both the silhouette score and the davis bouldin score improved to 0.57 and 1.373 respectively when using boruvka_tree algorithm with a leaf size of 10. On experimenting, we found that the implementation chose prims_balltree as the algorithm most suitable for the dataset. This means that ‘best’ actually led to a suboptimal choice which means that the implementation optimizes some criteria different from the intrinsic parameters used by us.

Q2 d **HDBSCAN on all three datasets** (with algorithm=best, min_sample_size=10 and min_cluster_size=5)

Measure Type	Measure	Dataset 1	Dataset 2	Dataset 3
Intrinsic	Silhouette Score	0.5725	0.48	0.11
	Davies Bouldin Score	1.23	1.30	3.046
Extrinsic	Purity score	0.8458	0.302	0.529
	Adjusted Rand Score	0.538	0.0013	0.307
	Normalized Mutual Information Score	0.689	0.0125	0.429

Q2 f. The above table shows the performance of HDBSCAN on the three datasets. As observed from this and other such tables, all the scores are better for HDBSCAN ie. performs the best on this dataset which can be confirmed visually. The only anomaly we observe is a higher silhouette score in hierarchical clustering (by 0.04). As studied in class, this may be due to an intrinsic bias in the formula of silhouette score against density based clustering algorithms like DBSCAN.

Q2 g.To check if the results of our clustering techniques are significantly different from random data, we assign random labels to the datapoints and calculate their evaluation metrics. We further repeat the process 20 times and calculate the confidence interval with 99% confidence.

*The **confidence level** represents the long-run proportion of corresponding CIs that contain the true value of the parameter. For example, out of all intervals computed at the 95% level, 95% of them should contain the parameter's true value. (Source)*

The min-max values of confidence intervals are as follows:-

Confidence Interval	Silhouette Score	Davies Bouldin Score	Purity score	Adjusted Rand Score	Normalized Mutual Information Score
Random	(-0.145,-0.12)	(63.7, 135.2)	(0.233, 0.251)	(-0.006, 0.0016)	(0.0254, 0.0382)

Analysis: With 99% confidence, the scores are vary far and much worse than the algorithm's score (for example, silhouette score is even negative). Hence, we can say that the clustering done is good and that there is <1% probability that a random distribution of labels would produce a better clustering.

Question 3

a) There are two types of hierarchical clustering: divisive (top-down) and agglomerative (bottom-up)
From the book,
Divisive: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.
Agglomerative: Start with the points as individual clusters and, at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity.

Agglomerative clustering belongs to the latter category. Agglomerative clustering is also more common than divisive due to the latter's complex algorithm. In the case of divisive clustering, we need a flat clustering method as a "subroutine" to split each cluster until we have each data in its own singleton cluster.

The process is slow as for agglomerative clustering we start by computing distances among the N objects. There are $N(N-1)/2$ calculations, but each is very fast. Each step "up" requires fewer calculations, and each is very fast. On the contrary, with divisive, we start with 2^N comparisons and each is time-consuming. And the costs stay high because, while each cluster gets smaller and the number increases.

b) We use four linkage methods namely - single, complete, average and ward.

c) We run the four linkage methods for each dataset and represent them visually using - PCA (2D and 3D) and T-SNE (2D and 3D) and dendrogram diagram.

A dendrogram diagram shows the hierarchical relationships between objects.

Steps to recreate.
Upload the three datasets and specify which dataset is used. Then run the entire colab file for parts.

Evaluation

		Dataset 1	Dataset 2	Dataset 3
Single	Davies Bouldin Index	0.429010302269248	0.040064011830938225	0.9050195769353596
	Silhouette Coefficient	0.1478760034514985	0.9387170490111375	-0.10379472454375818
	Purity Score	0.7916666666666666	0.282	0.8366013071895425

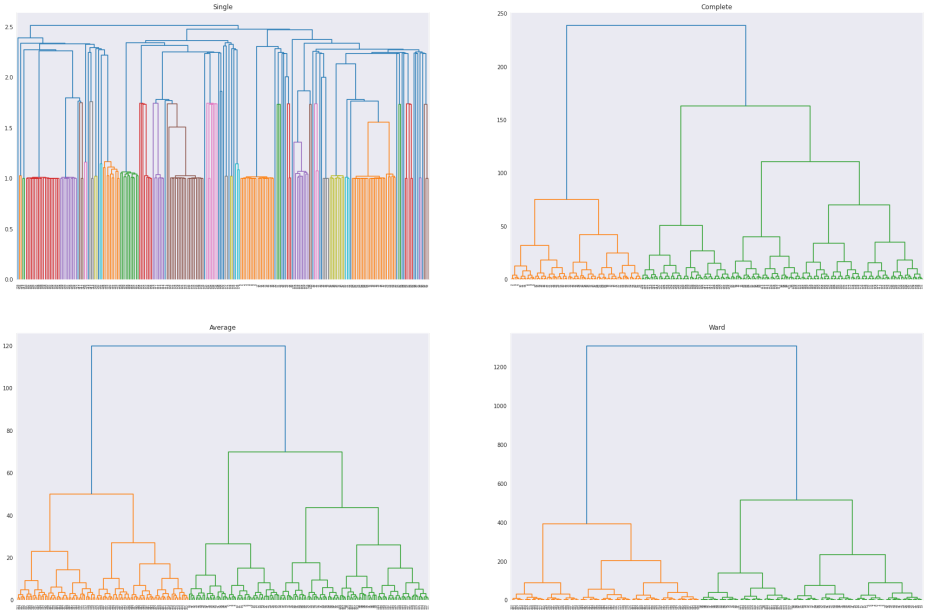
	Adjusted Rand Score	0.1785166754 4407782	0.0001902302 1714247208	0.6417422864 596949
	Normalized Mutual Info	0.5527210505 351642	0.0022022680 393947824	0.7115769318996483
Complete	Davis Bouldin Index	0.5013592480 263714	0.2851608215 0240885	2.3113854421 68663
	Silhouette Coefficient	0.5486849220 69238	0.9161607478 34404	0.1051785297 967763
	Purity Score	0.2083333333 3333334	0.281	0.9006535947 712418
	Adjusted Rand Score	0.0063926274 55691369	-3.2348114656 751916e-05	0.4480750235 197549
	Normalized Mutual Info	0.0180026918 8162553	0.0019259728 211161568	0.6409673980 888057
Average	Davis Bouldin Index	0.5010139168 026061	0.2851608215 0240885	1.4240550853 543172
	Silhouette Coefficient	0.6063285570 779985	0.9161607478 34404	0.1199581136 6347287
	Purity Score	0.2083333333 3333334	0.281	0.9803921568 627451
	Adjusted Rand Score	0.0116025272 8317059	-3.2348114656 751916e-05	0.6219297140 406735
	Normalized Mutual Info	0.0239989191 12731585	0.0019259728 211161568	0.7451779258 463783
Ward	Davies Bouldin Index	0.5009543118 201337	0.5619231876 648753	1.8403334010 594579
	Silhouette Coefficient	0.6186401174 274316	0.6786133299 222863	0.1820438748 9779805
	Purity Score	0.2083333333 3333334	0.293	0.6627450980 392157
	Adjusted Rand Score	0.0046596965 37263011	0.0019003657 284070437	0.4841047698 492194

	Normalized Mutual Info	0.014770314577515339	0.013055768287073296	0.6400743236052747

Visual Evaluations

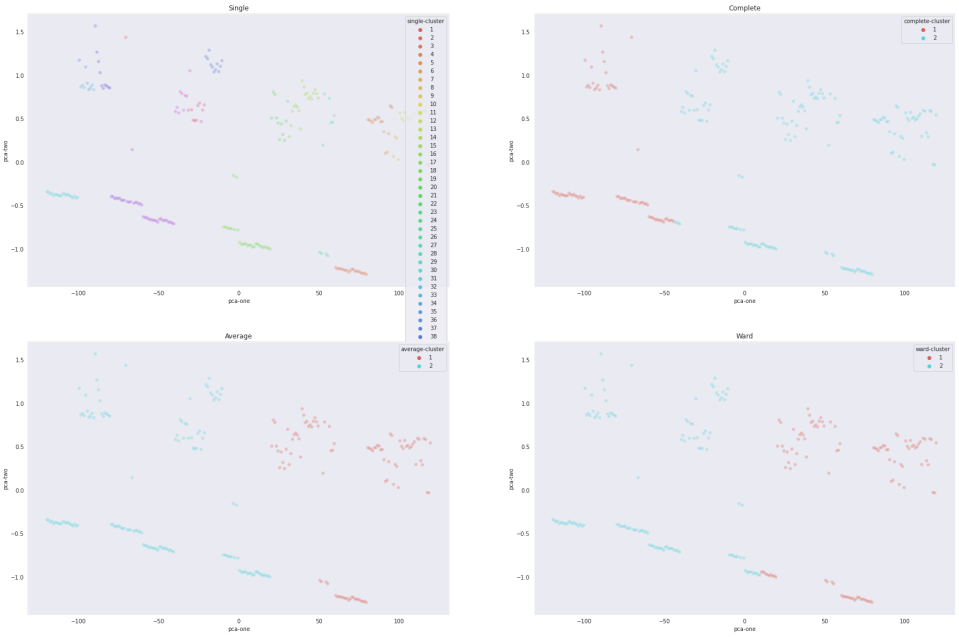
Dataset 1

Dendogram Diagram

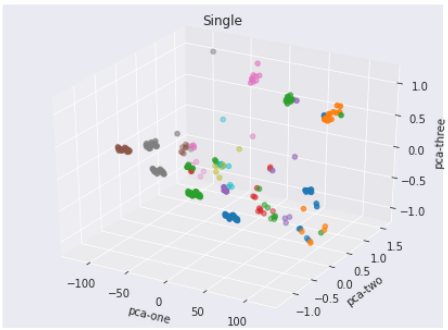


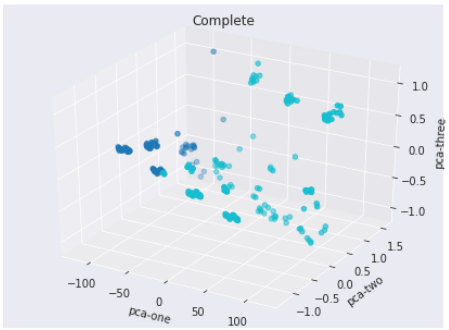
PCA plots

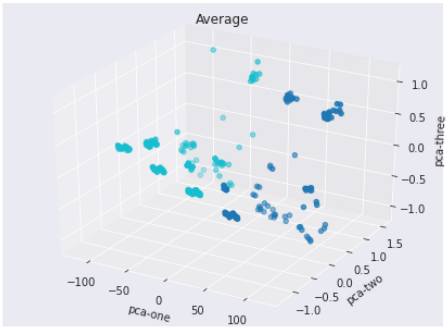
2D

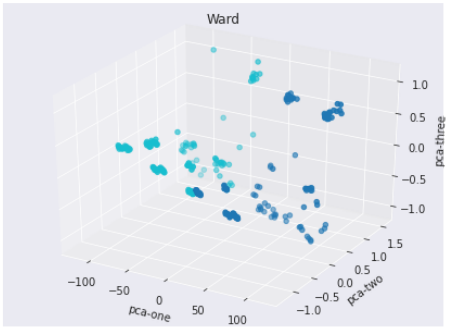


3D



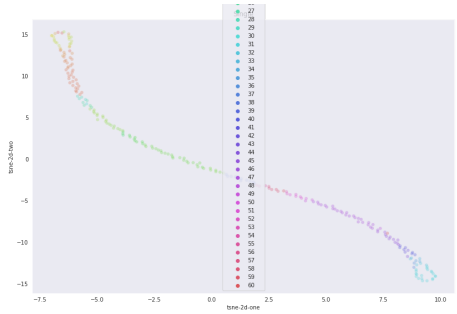


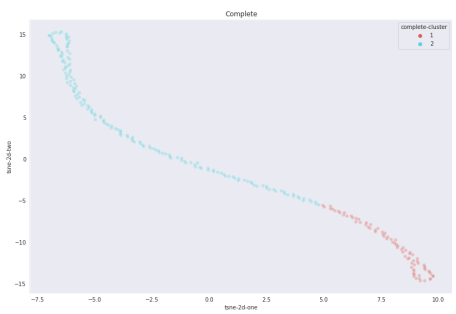


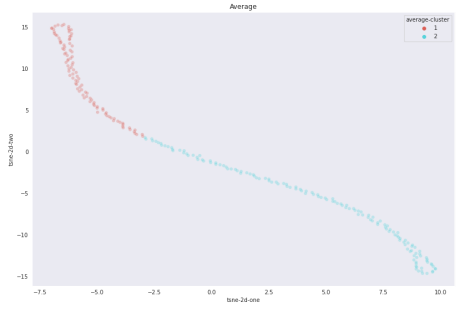


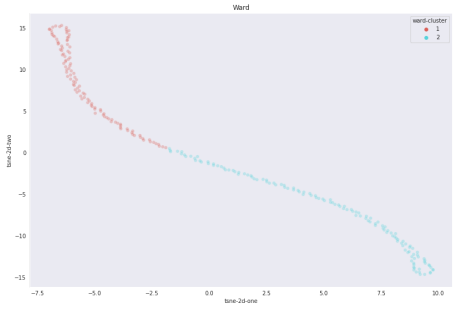
T SNE plots

2D

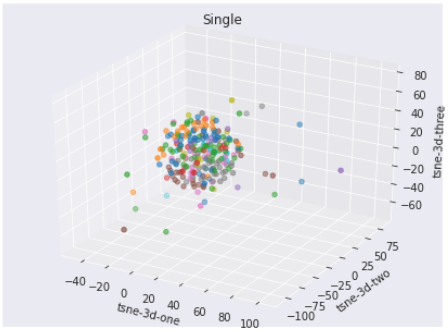


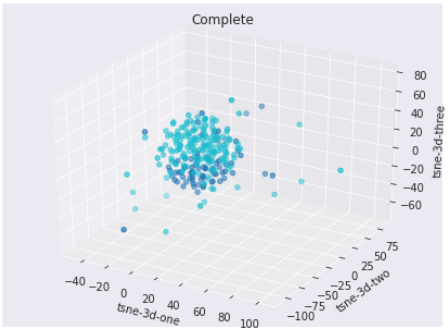


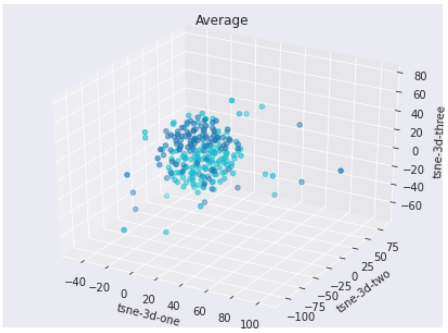


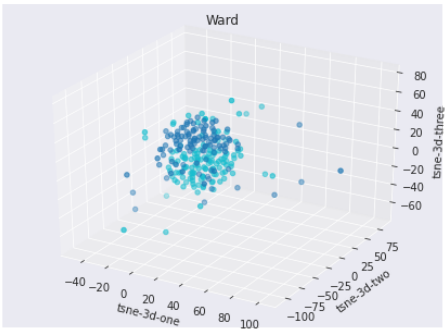


3D



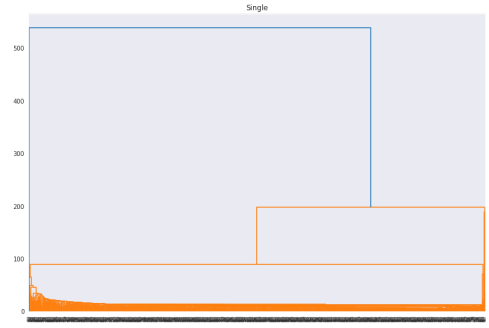


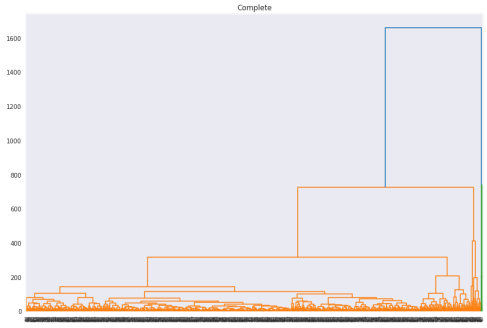


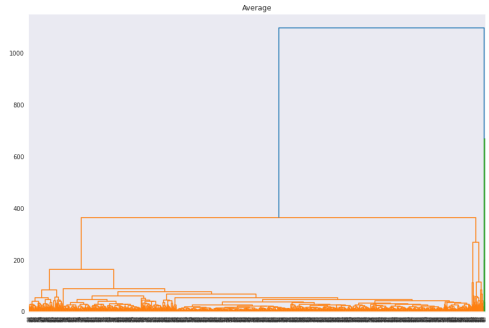


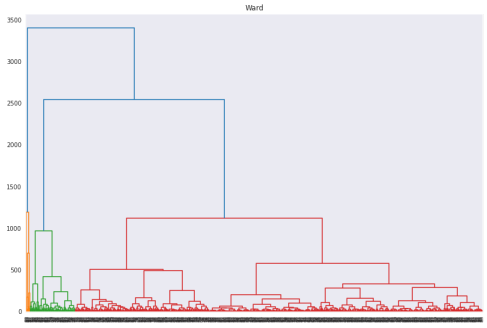
Dataset 2

Dendrogram Diagram



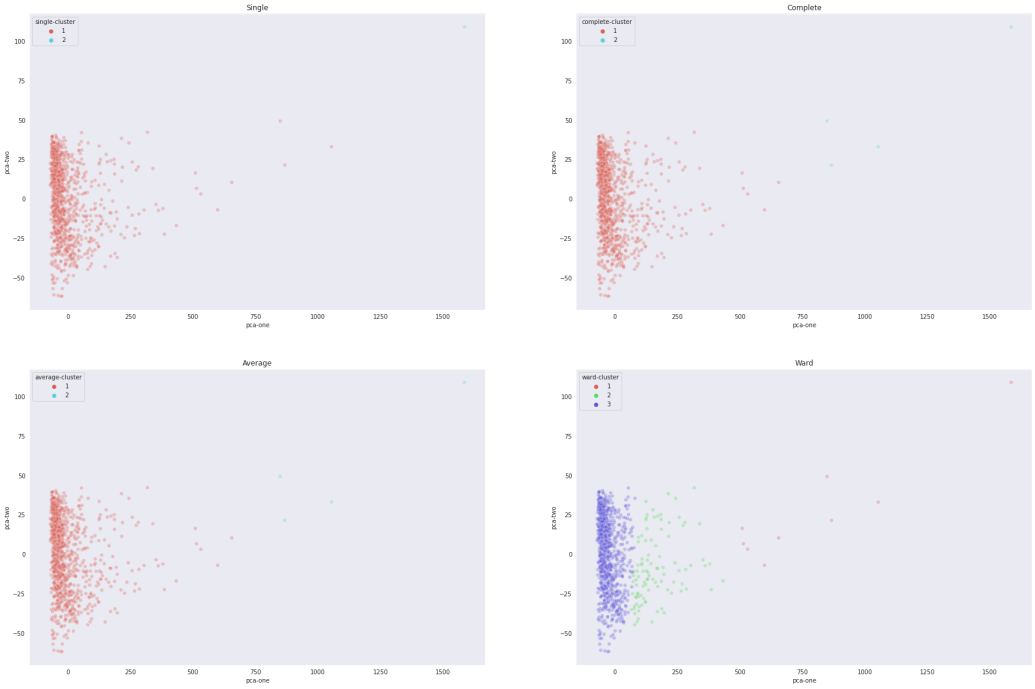




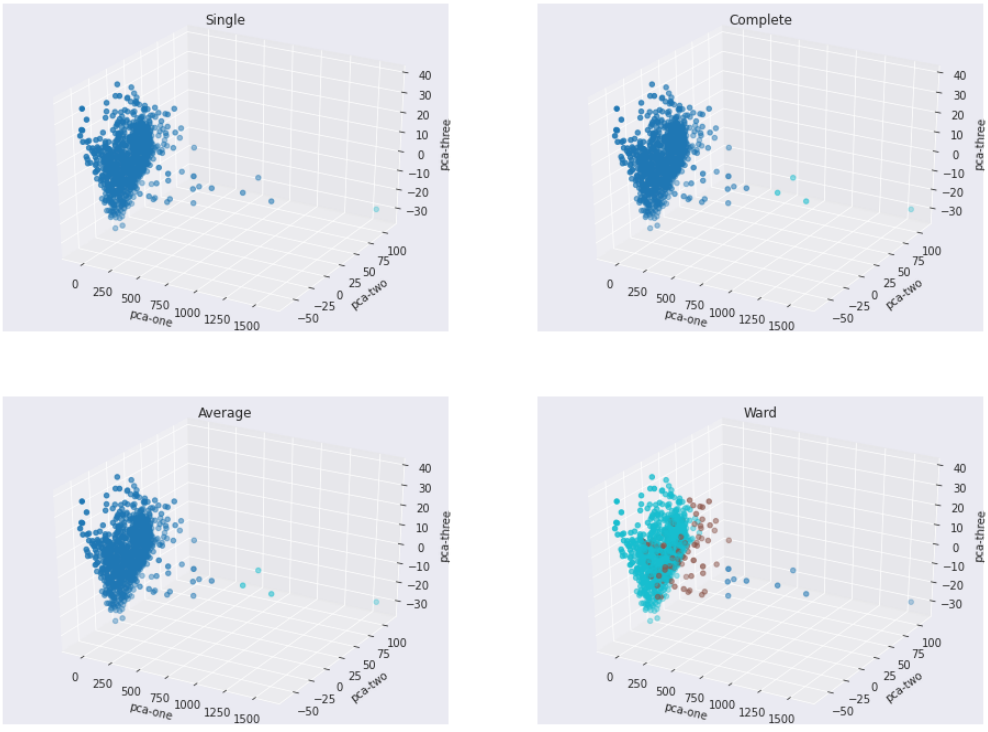


PCA plots

2D

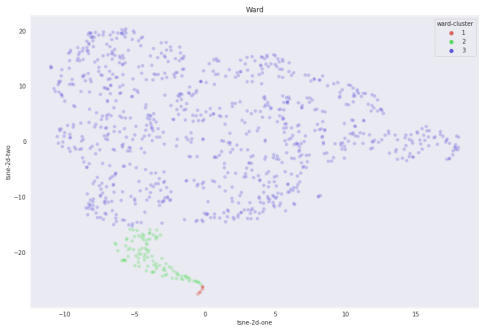


3D

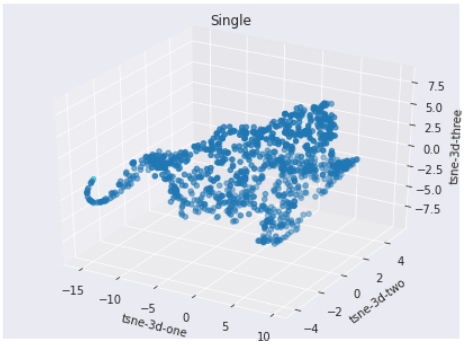


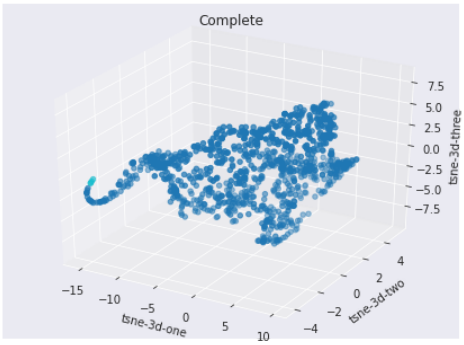
T SNE plots

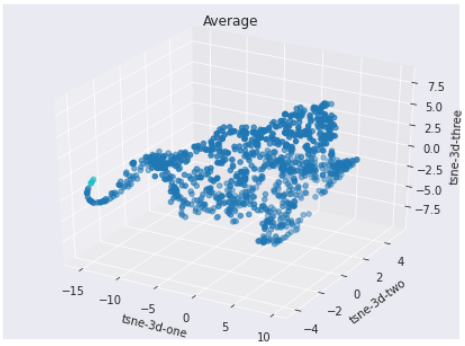
2D

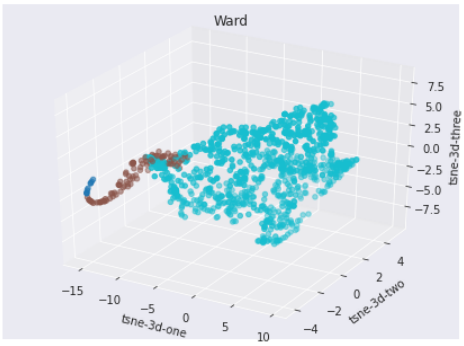


3D



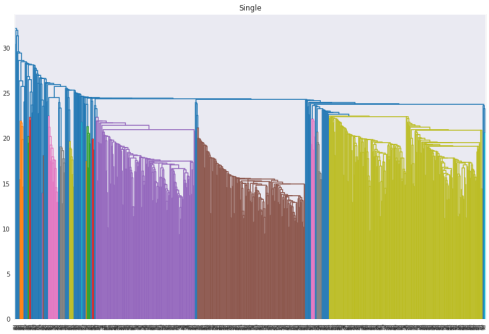


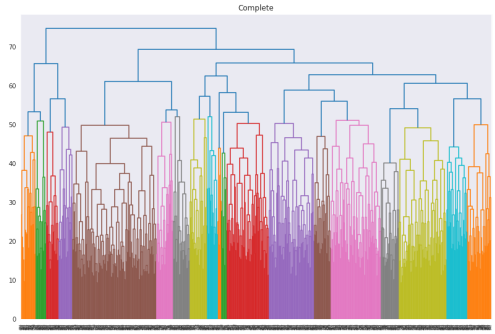


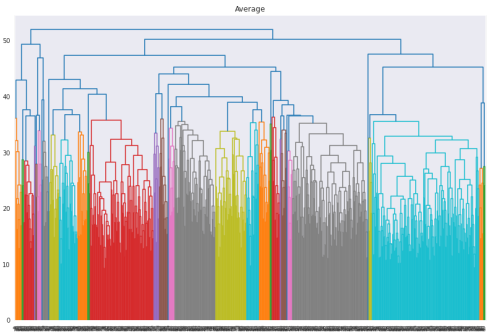


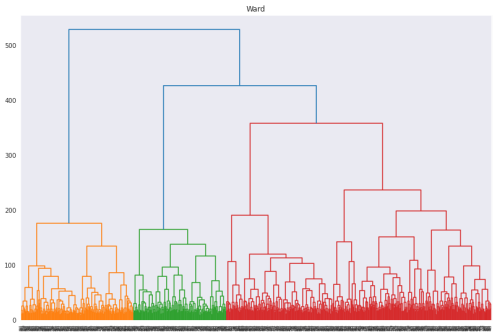
<u>Dataset 3</u>

Dendrogram Diagram



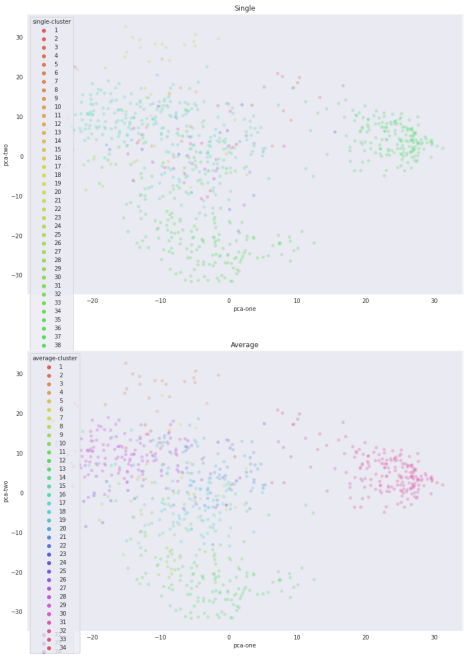


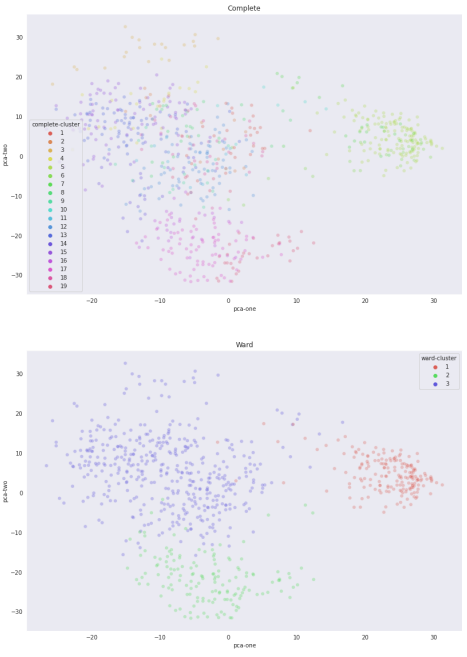




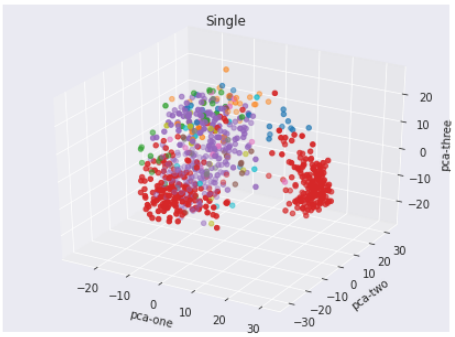
PCA plots

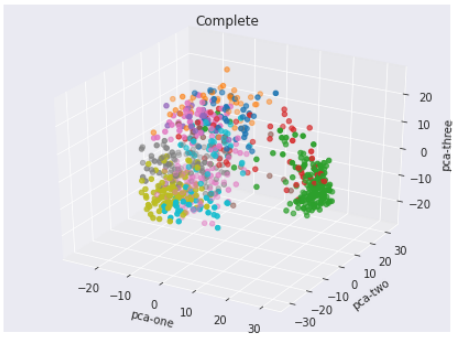
2D

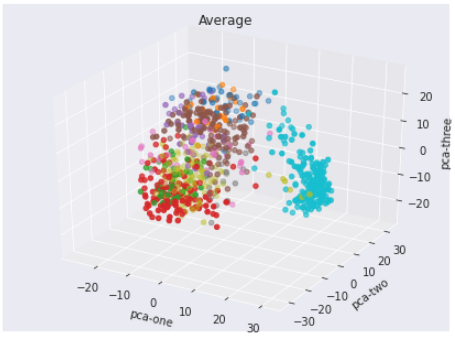


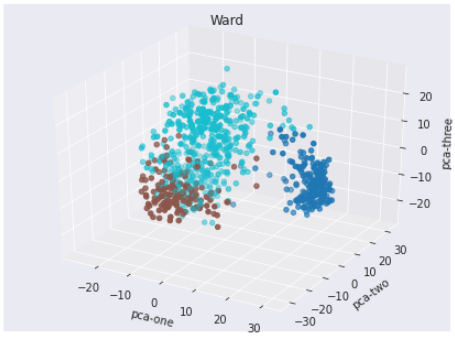


3D



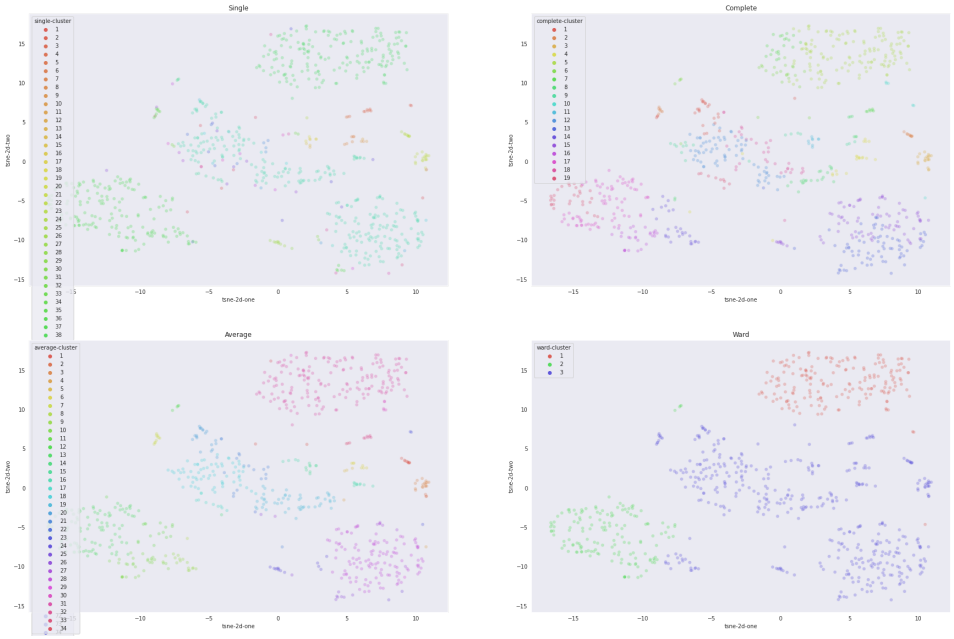




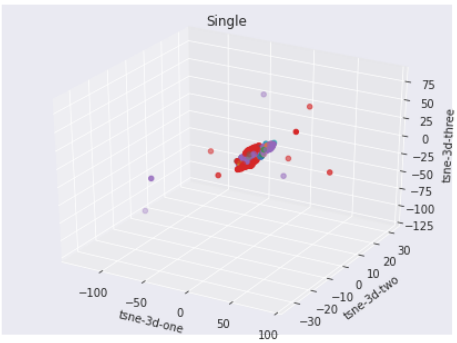


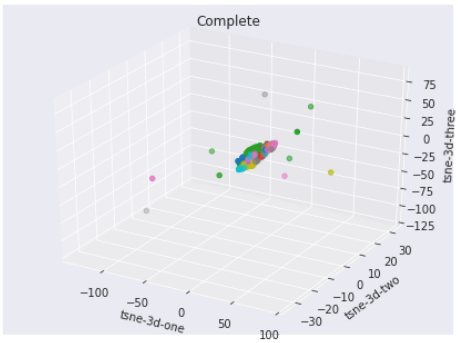
T SNE plots

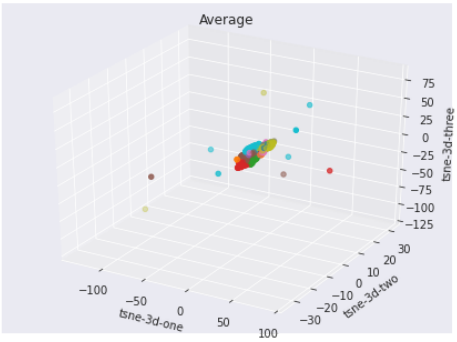
2D

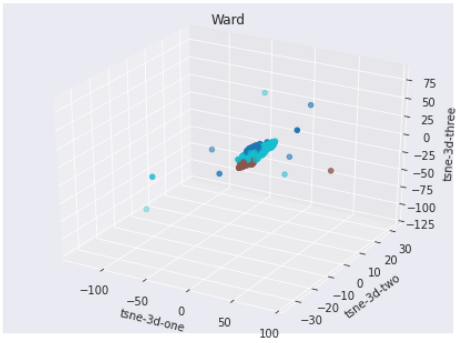


3D









d) According to our hypothesis, agglomerative clustering should outperform all other algorithms for dataset 2. We use the evaluation table to compare the clustering performance for each algorithm.
We find that the silhouette coefficient for this dataset is the highest for agglomerative clustering, almost reaching 0.92 for each linkage except ward whereas for other algorithms the highest is 0.67 (k-means clustering). Similarly, davies bouldin index, compared to our 0.04, every algorithm has much higher score. We can see the same for extrinsic measure where, normalized mutual information is almost 0 (0.0002) for single linkage compared to 0.0125 for the other clustering algorithm. We can extend this to each evaluation metric to see that our clustering method works the best for our dataset which confirms our hypothesis.
We can also check from the visualizations that our agglomerative clustering suits the dataset.

e) To ensure our assignments of cluster is significantly different from randomly assigned cluster labels, we will compare the results received from our clustering algorithm. We run the random assignment 20 times and calculate confidence interval for each metric with 99% confidence.
We have for dataset 2.

Confidence Interval	Silhouette Score	Davies Bouldin Score	Purity score	Adjusted Rand Score	Normalized Mutual Information Score

Random	(-0.09,-0.06)	(31.36, 52.85)	(0.290, 0.300)	(-0.0010, 0.0011)	(0.0025, 0.0053)

We can see the minimum and maximum values that the random clustering will assign for each metrics. We can notice the values for each metrics are awful and cannot compare to the results of the algorithm. For example, the minimum value of davis bouldin index is 31.36 compared to the 0.04 (best) achieved by our dataset and 0.58 (worst) is much larger. We can similarly compare all metrics to see that our algorithm significantly outperforms the random clustering. Thus, our results are valid.

Learnings

From this question we learnt and understood about the working of agglomerative clustering. We also learned the use of multiple linkages used in clustering. We also learned about dendogram and analysing the same.

Question 4

(a) Limitations

The 2 limitations of K-means algorithm are -

1. *Scaling with number of dimensions*
   When dimensionality increases, data becomes largely sparse as the objects may be in different dimensional subspaces and as a result these objects can be considered equidistant from one another. Due to this our distance metric is rendered useless which was earlier used to group similar objects together with a lower number of dimensions. Also, since the number of features increases, the noise increases which can mask the real clusters. It becomes difficult to find the relevant attributes for clustering.
   The curse of dimensionality results in increased memory usage and high processing power due to large databases.
2. *Clustering Outliers*
   Outliers can greatly affect clusters by dislocating their centroids. This means that the centroid is not a true representation of a closely packed globular cluster.
   Sometimes groups of outliers get their own centroid.

(b) Algorithms

The 2 existing algorithms to mitigate these limitations are -

- *For clustering outliers:* **Kmedoids clustering**. The medoid of a cluster is a data point whose average dissimilarity to all points in the cluster is minimal. In K-Medoids clustering, the cluster centers are the medoids of each cluster respectively, instead of being the mean average of all cluster points. This reduces its susceptibility for cluster

centers to be shifted towards groups of outliers and the cluster center can end up in a spatially low-density location, which is not an ideal representative point for the cluster.

- *For scaling with number of dimensions:* **Spectral clustering**
  It is a type of subspace clustering and helps to project data points into a lower dimensional subspace like Principal Component analysis as a preprocessing step to traditional clustering algorithms. It uses graph distance like nearest neighbors which makes more sense since distance metrics like euclidean become meaningless in higher dimensions.
  To view and run the code for spectral clustering and hyper parameter tuning for the datasets, run the colab file - “DMG_A2_Q4_dimensions.ipynb”.

(c) Results

Comparing evaluation measures (for Dataset 1)

Evaluation metric	**K-Means**	**K-Medoids (Manhattan)**	**K-Medoids (Euclidean)**	**Spectral clustering**
Sum of Squared Errors (Intrinsic)	184.013130989071 84	448.994853644106 2	225.658638027500 05	-
Davies Boudin score (Intrinsic)				0.502175309518047 3
Silhouette Coefficient (Intrinsic)	0.5181769500958 624	0.45423830228864 454	0.37893676621566 79	0.579310719051497 4
Purity score (Extrinsic)	0.4541666666666 6666	0.29583333333333 334	0.29583333333333 334	0.2375
Rand_index (Extrinsic)	0.3259273581787 576	0.09221376823108 729	0.10387833003488 621	0.016016922748716 904

Comparing evaluation measures (for Dataset 2)

Evaluation metric	**K-Means**	**K-Mediods (Manhattan)**	**K-Mediods (Euclidean)**	**Spectral clustering**

Sum of Squared Errors (Intrinsic)	3258175.040614787	60310.0	34925.055257360 21	-
Davies Boudin score (Intrinsic)				0.913826522544300 9
Silhouette Coefficient (Intrinsic)	0.6704673250239 235	0.2594732695593435	0.2594732695593435	-0.020282638844525 11
Purity score (Extrinsic)	0.295	0.348	0.338	0.2392156862745098 1
Rand_index (Extrinsic)	0.0014302779705 772583	0.0244088103954083372	0.0196317110659796 1	-0.0002320938808 1300308

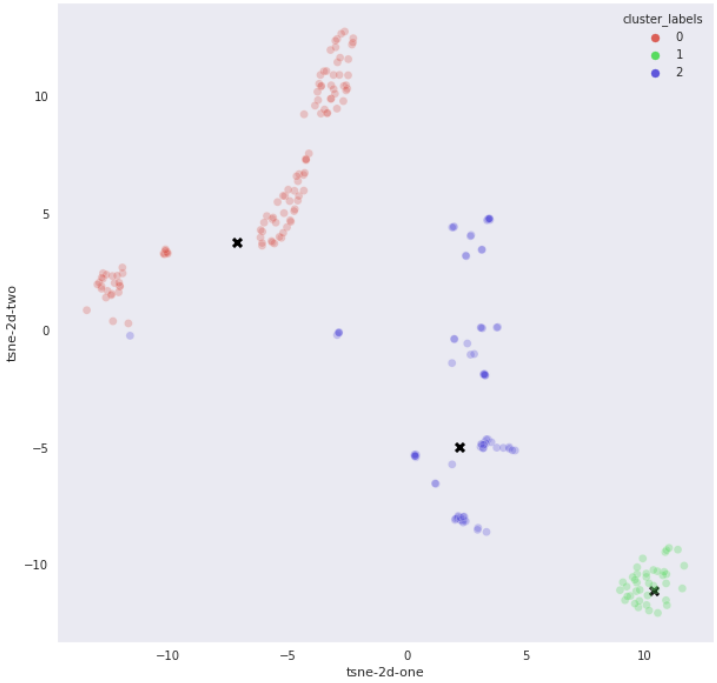
Comparing evaluation measures (for Dataset 3)

Evaluation metric	**K-Means**	**K-Mediods (Manhattan)**	**K-Mediods (Euclidean)**	**Spectral Clustering**
Sum of Squared Errors (Intrinsic)	542923.9391450146	104601.0	22394.959196007 39	-
Silhouette Coefficient (Intrinsic)	0.2039819250456 0156	0.2281963923655 1778	0.1787321811512 3857	0.3903065291957 5565
Davies Boudin score (Intrinsic)				0.6528303080739 942
Purity score (Extrinsic)	0.8431372549019 608	0.8130718954248 366	0.8274509803921 568	0.4535947712418 3006
Rand_index (Extrinsic)	0.6890491811446 796	0.6413318241612 036	0.6433240737990 74	0.1626394974319 495

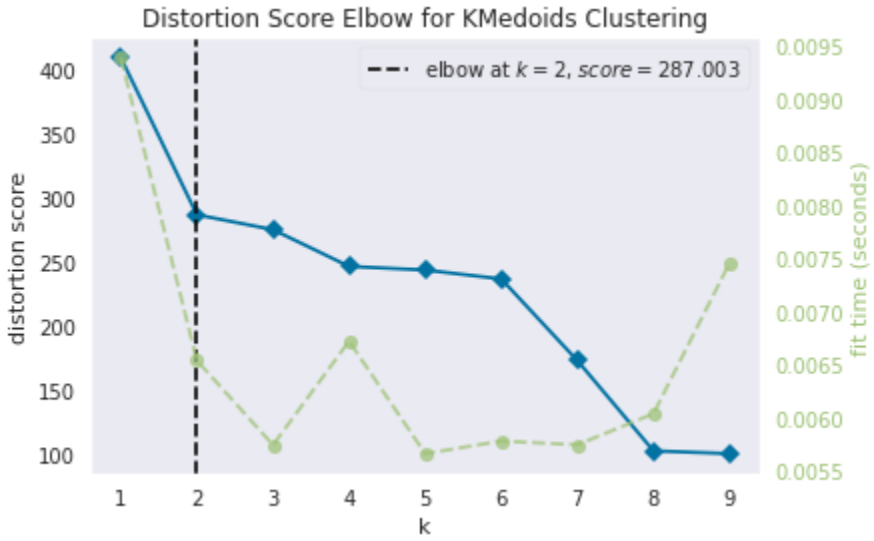
(d) Hyperparameter Tuning

We used the *Elbow method* to determine the optimal K value for the clustering algorithms. The elbow shows the optimal number of clusters.

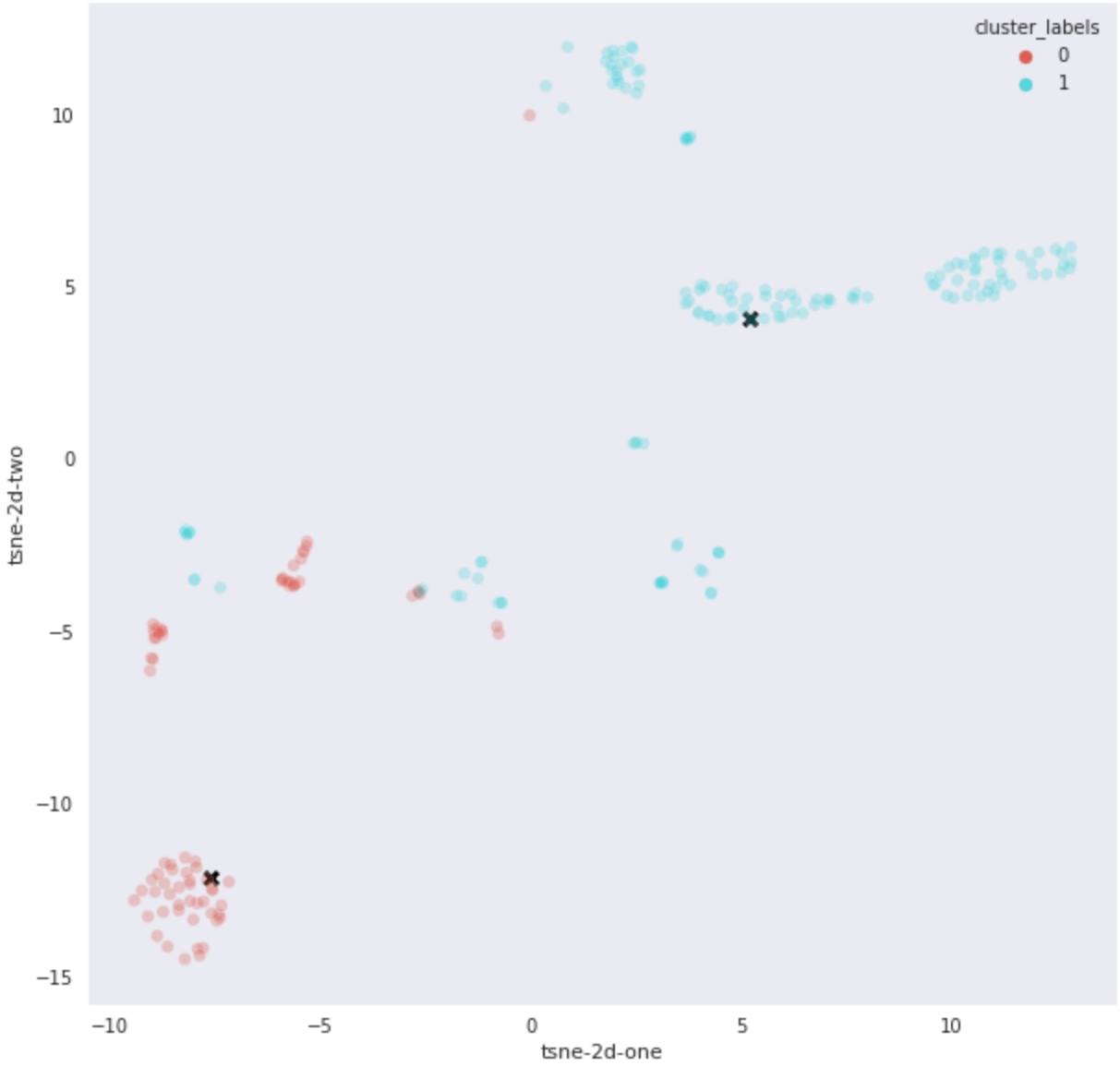
Dataset 1 - KMeans



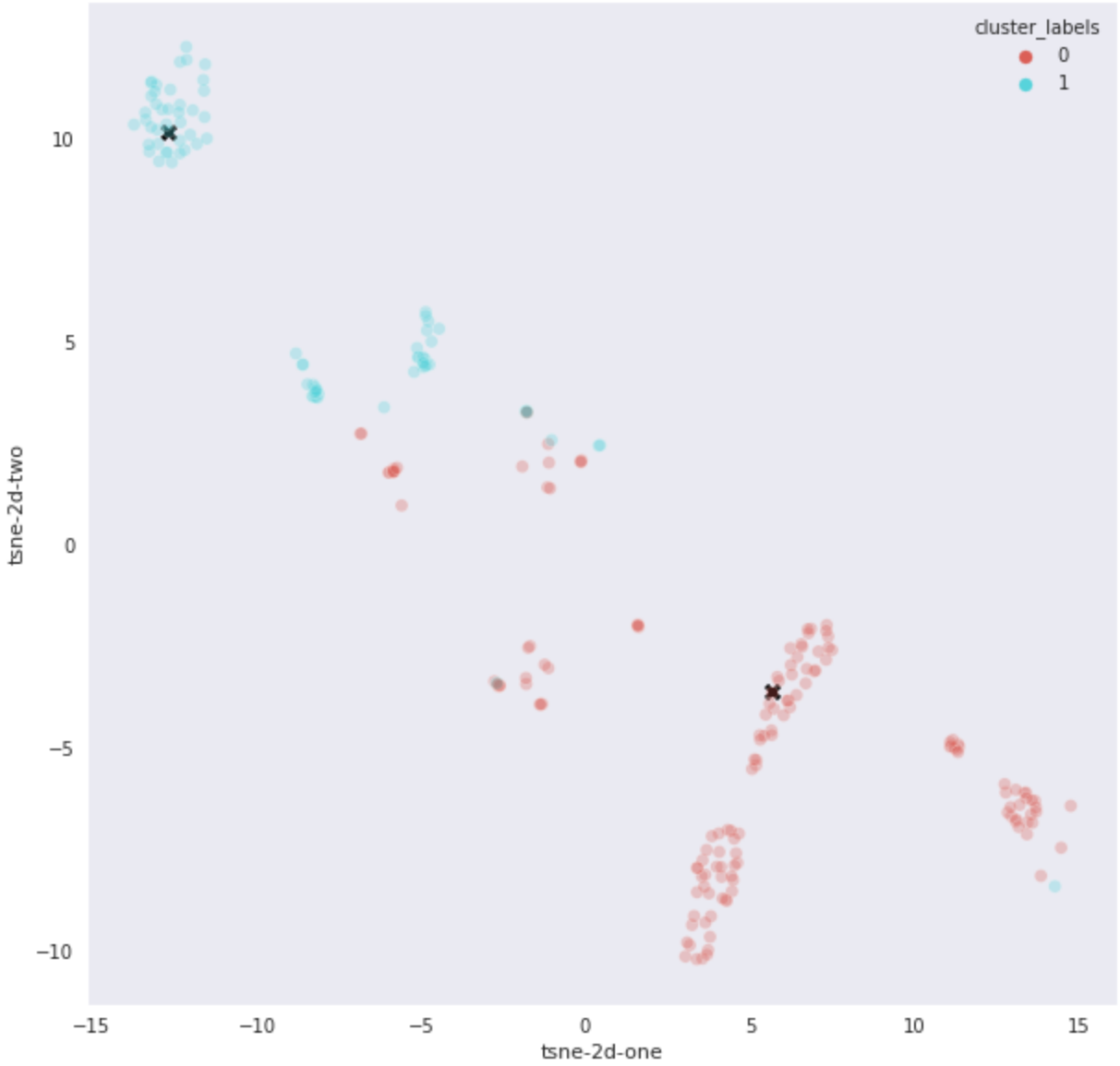
Dataset 1 - KMediods



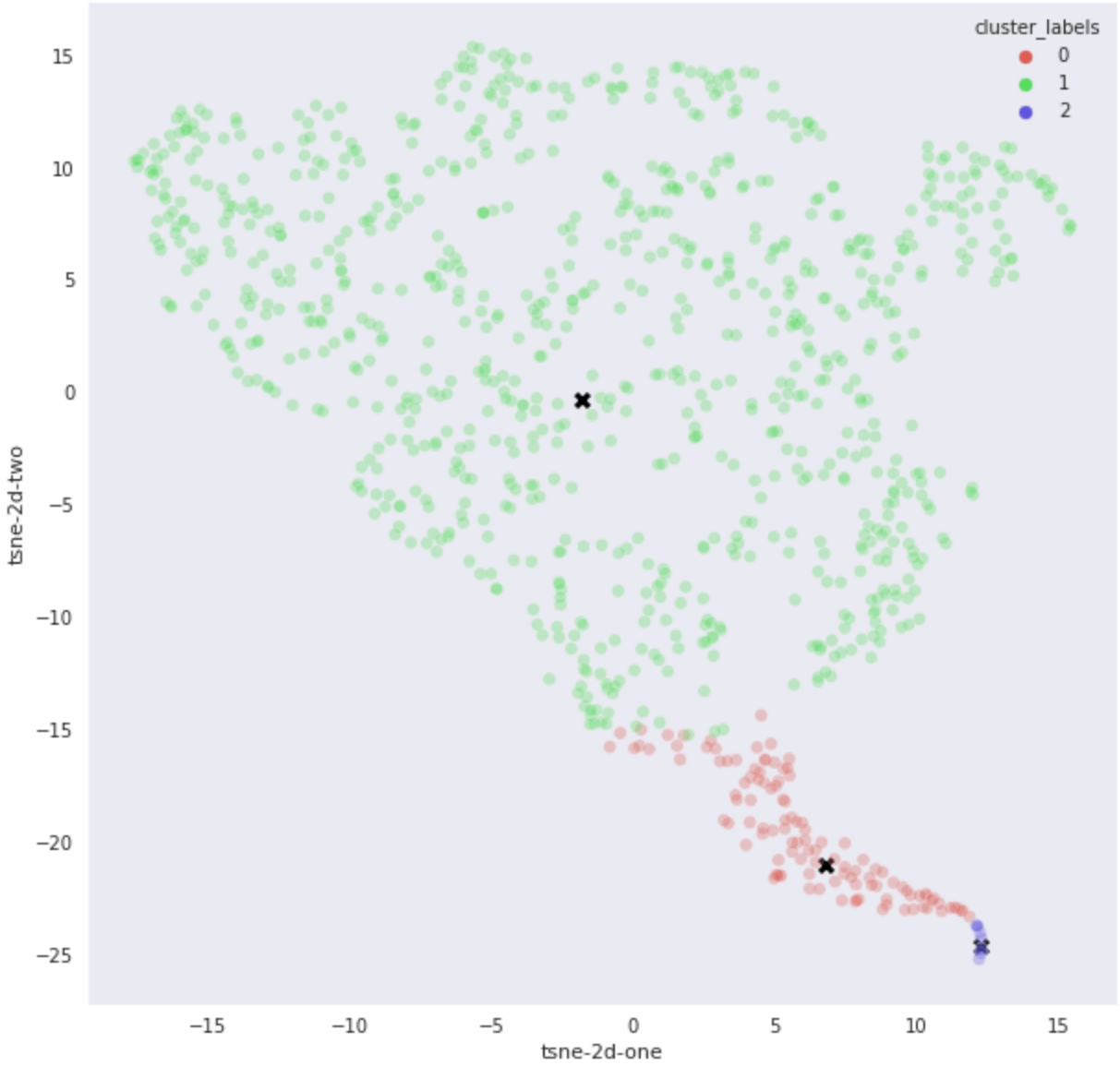
KMediods Manhattan:



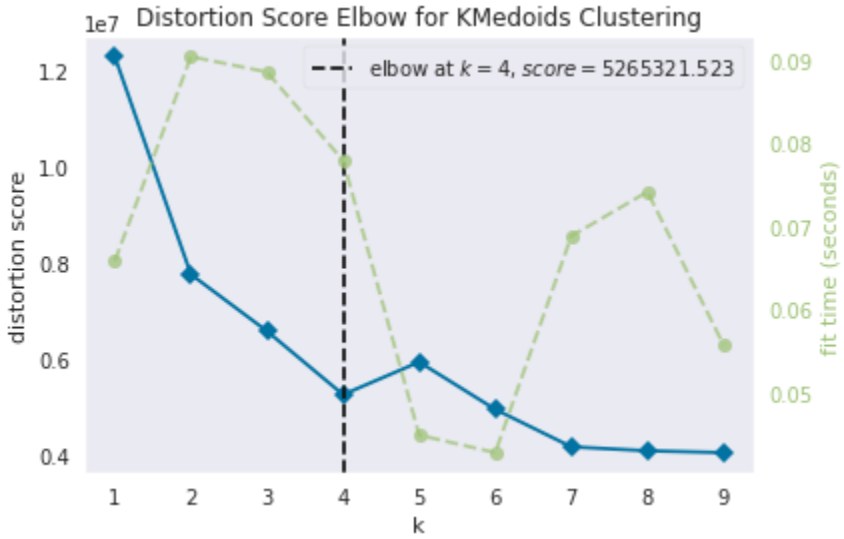
KMediods Euclidean:


cluster_labels
0
1
10
5
0
−5
−10
tsne-2d-two
−15
−10
−5
0
5
10
15
tsne-2d-one

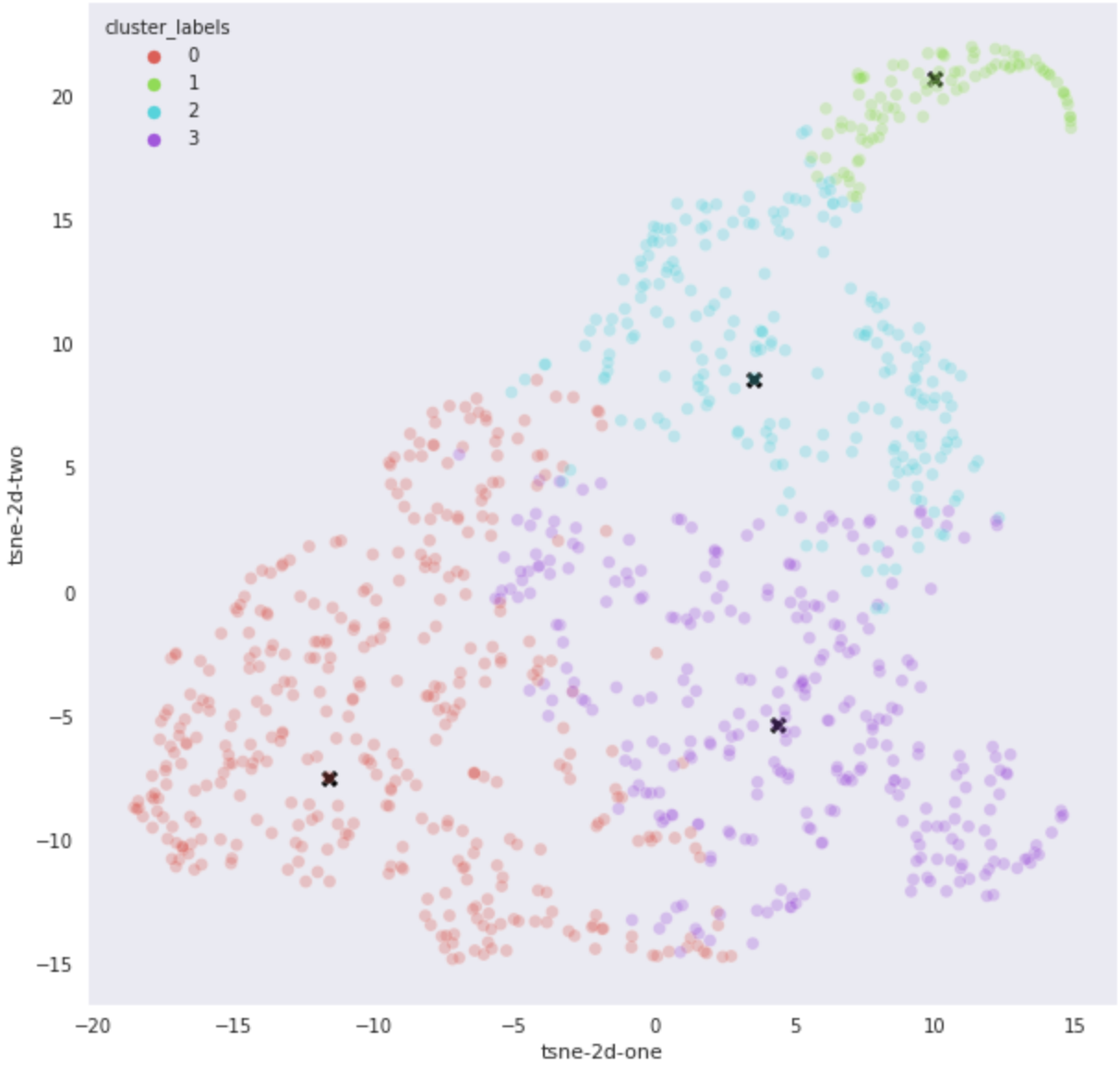
Dataset 2 - KMeans


cluster_labels
0
1
2
15
10
5
0
−5
−10
−15
−20
−25
tsne-2d-two
−15
−10
−5
0
5
10
15
tsne-2d-one

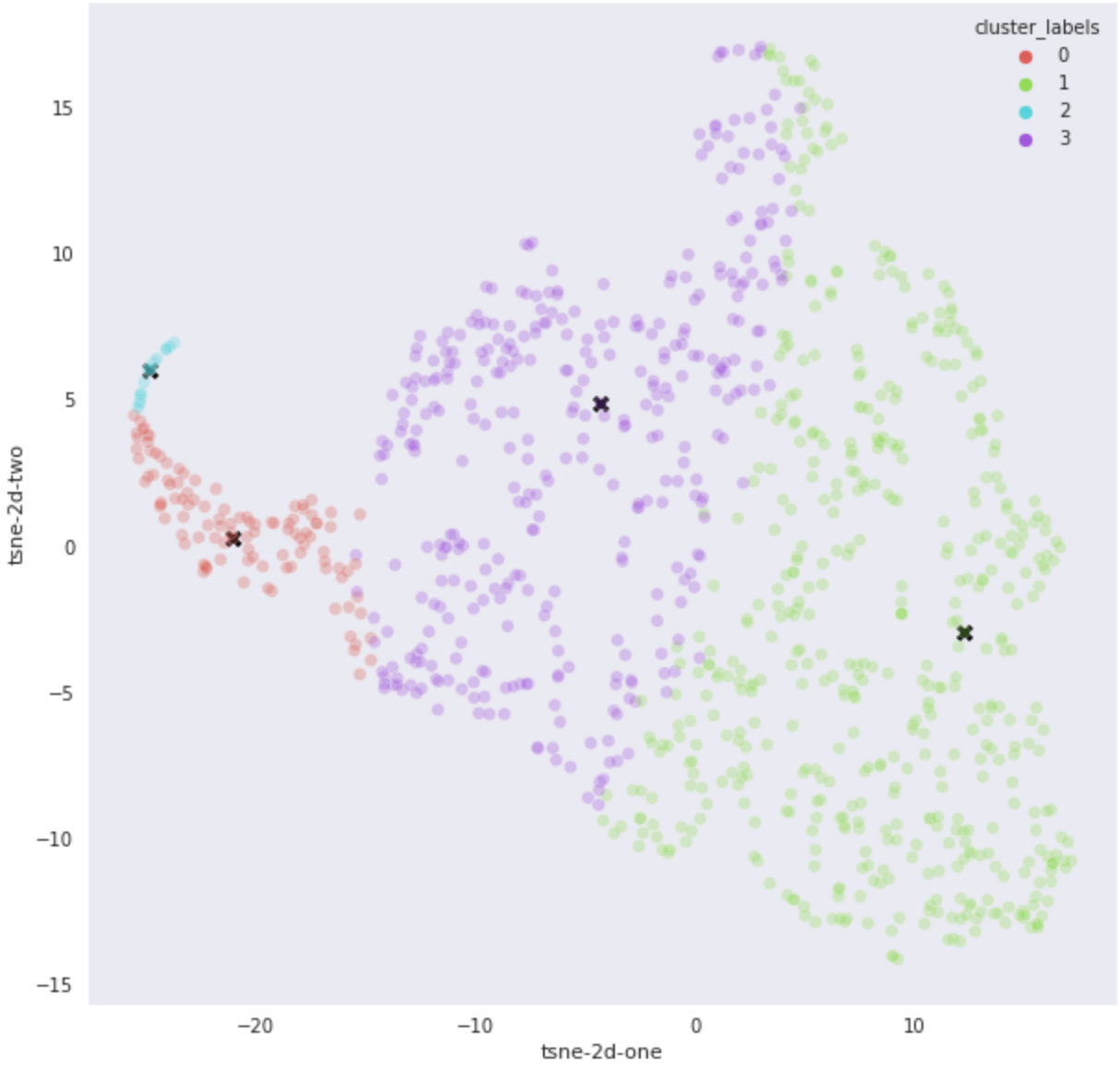
Dataset 2 - KMediods



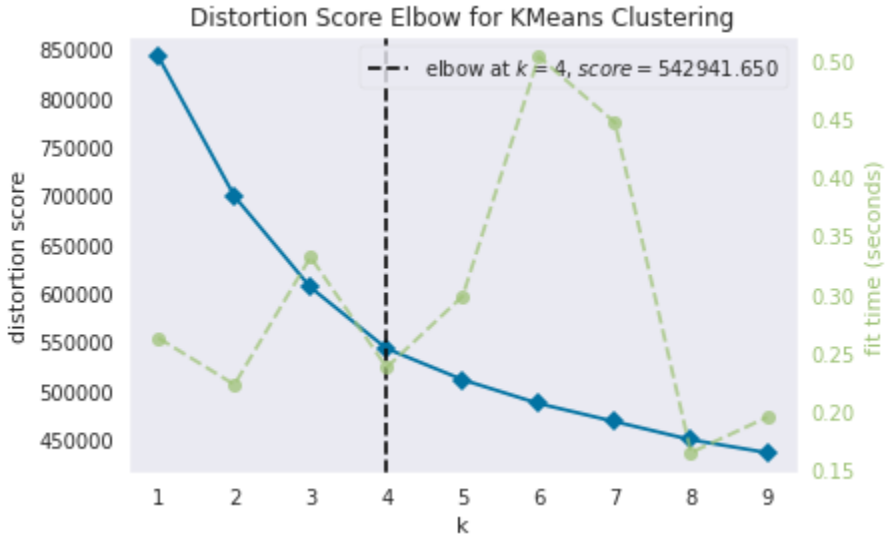
Kmediods Manhattan:

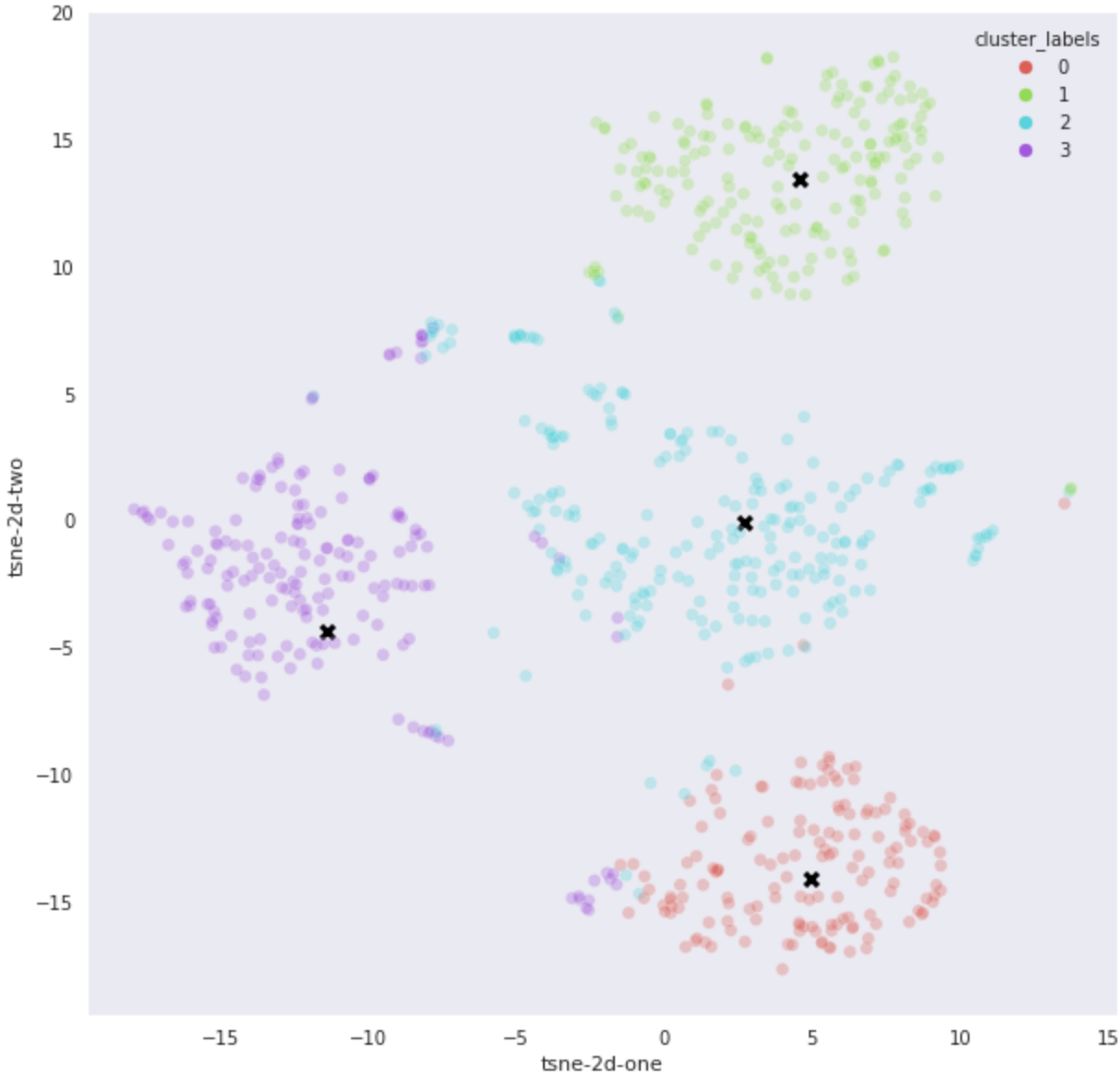


Kmediods Euclidean:

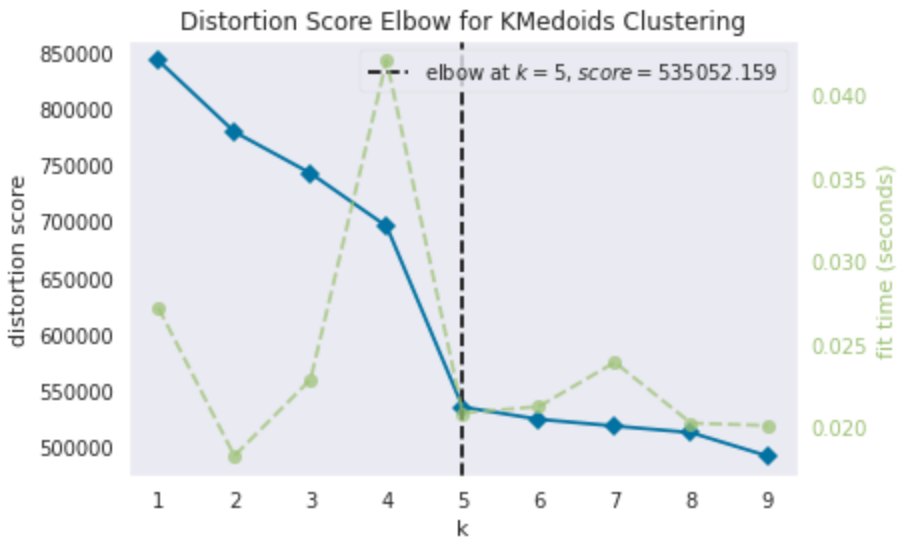

cluster_labels
0
1
2
3
15
10
5
0
−5
−10
−15
tsne-2d-two
−20
−10
0
10
tsne-2d-one

Dataset 3 - KMeans

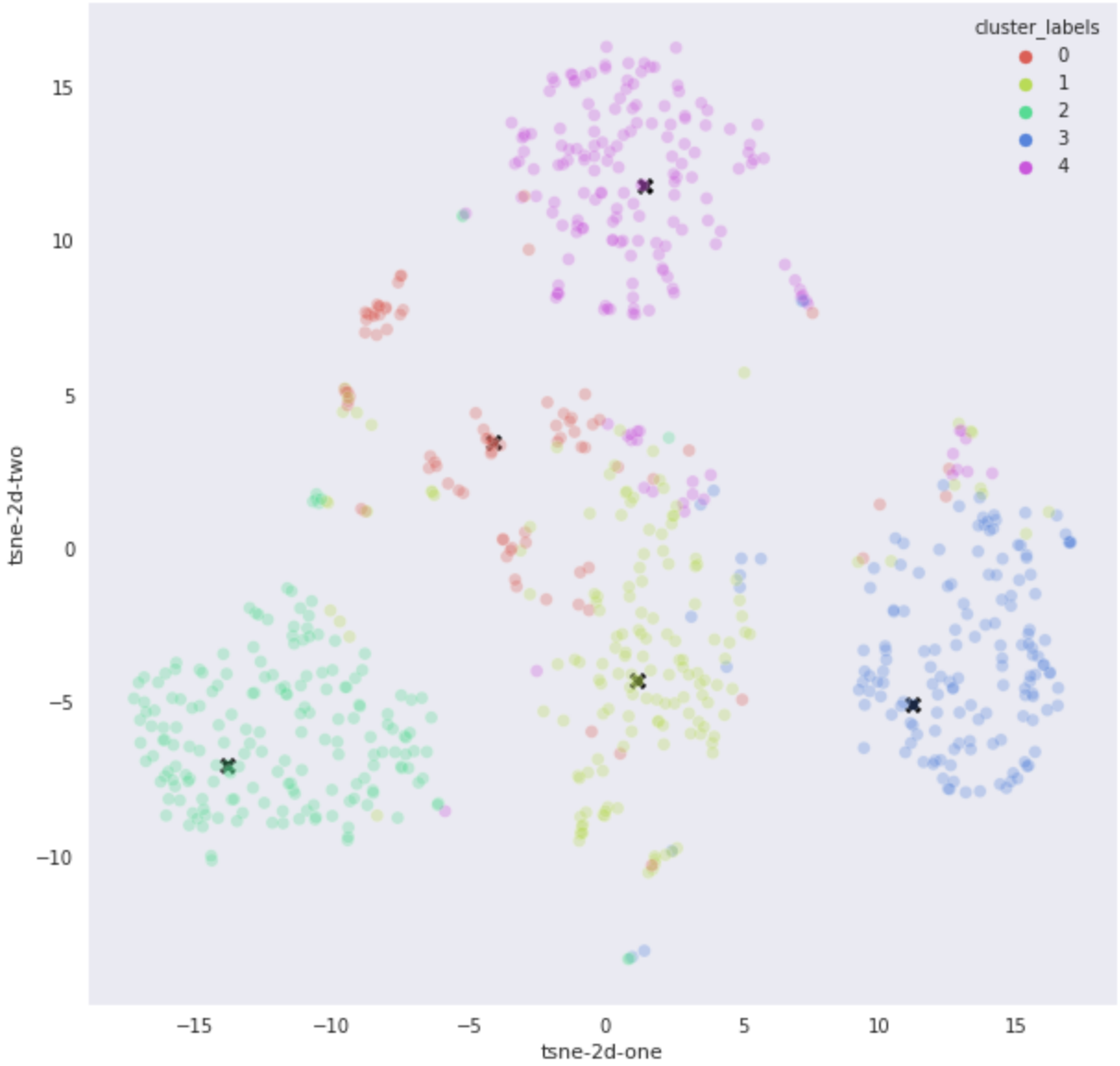




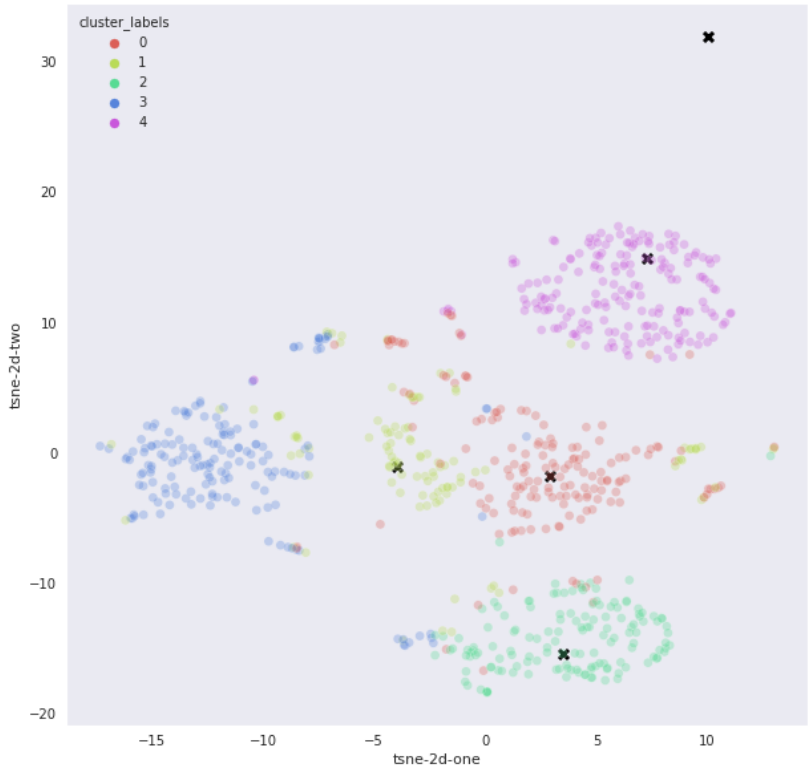
Dataset 3 - KMediods



KMediods Manhattan:



KMediods Euclidean:



(e) Selection Hypothesis

Looking at the tables we can infer that our initial hypothesis was correct and that our algorithm works better for our chosen dataset, especially spectral clustering. But between k-means, k-medoids and spectral we can't observe much difference because the no. of the features in the dataset was small. Hence we were not actually able to observe the difference in performance to a great extent.

(f) Result Confirmation

Confidence Interval	Silhouette Score	Davies Bouldin Score	Purity score	Adjusted Rand Score	Normalized Mutual Information Score
Random	(-0.034, -0.029)	(15.20, 16.73)	(0.282, 0.291)	(-0.001, 0.0003)	(0.028, 0.033)

We observe that the score of spectral clustering and kmediods clustering are much better than the scores shown in the table above. Hence a random assignment of labels does far worse clustering than our model. Thus, our clustering gives significant results.

Question 5

Definitions of the Extrinsic and Intrinsic parameters used

Silhouette score: It is a method of evaluating clusters which combines both cohesion and separation. The following steps explain how to compute the silhouette coefficient for an individual point:-

1. For the i th object, calculate its average distance to all other objects in its cluster. Call this value a_i.
2. For the i th object and any cluster not containing the object, calculate the object's average distance to all the objects in the given cluster. Find the minimum such value with respect to all clusters; call this value b_i.
3. For the i th object, the silhouette coefficient is $s_i = (b_i - a_i)/ \max(a_i, b_i)$.

The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A value near 0 indicates overlapping clusters.

Davies bouldin Score: The score is the avg. similarity measure between each cluster and its most similar cluster where this similarity ratio is calculated as ratio of within cluster distances to between cluster distances. Hence lower the score better the clustering.Lies between 0-1.

Purity score : We assign a label to each cluster based on the most frequent class in it. Then the purity becomes the number of correctly matched class and cluster labels divided by the number of total data points. Higher the purity better the clustering. Lies between 0 and 1.

Adjusted Rand Score: The Rand Index computes a similarity measure between two clusterings by calculating the following value -

RI = (number of agreeing pairs) / (number of pairs)

Lies between 0 and 1. Higher the score, better the clustering.

References

1. https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/#:~:text=Divisive%20clustering%20is%20more%20complex,having%20its%20own%20singleton%20cluster.
2. https://stats.stackexchange.com/questions/417382/hierarchical-clustering-why-always-agglomerative
3. https://www.emerald.com/insight/content/doi/10.1108/IMDS-01-2015-0027/full/html
4. HDBSCAN demo video
5. API Reference — hdbscan 0.8.1 documentation
6. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html#sklearn.metrics
7.