DMG - Assignment 2

Group 12

Parth Chhabra	Rishi Singhal	Samyak Jain	Sarthak Johari	Kishan Sinha
2019069	2019194	2019098	2019099	2019428

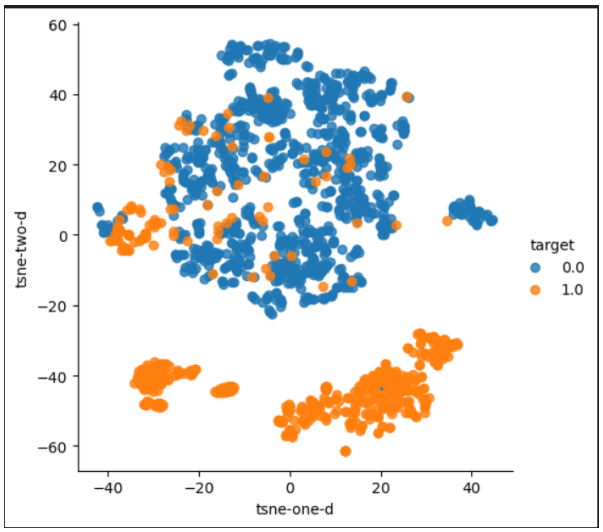
Q1

DATASET 1-

For Density-Based Clustering, we have used the credit card fraud detection dataset available in the GitHub Repo of DMG Assignment 3. The dataset link is as follows-

https://github.com/ayan-iiitd/ADRepository-Anomaly-detection-datasets/tree/main/numerical%20data/DevNet%20datasets

It is also available on Kaggle and the link is mentioned below:-
https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud



We have used the credit card fraud dataset for density-based clustering because of the following reasons -

1. The first reason is that the t-sne plot of this dataset is densely packed in one area corresponding to label 0. Also there seems some outliers in that region in

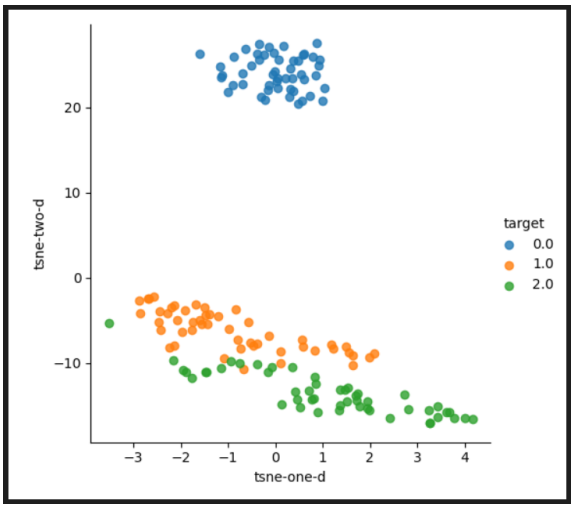
the form of label 1. Also this cluster region is well seperated from the other 2 regions.

2. For label 1 there are 2 dense orange regions which are quite far from each other. This might be problematic because the 2 regions belong to the same label but they are far from each other. However, we hypothesize that in this case as well these 2 regions would be clustered into 2 different clusters because they are highly densely packed.

We didn't use digits dataset for density based clustering even though it was densely packed. This was due to its high dimensionality of 64x1 vector size of each sample. As we know that due to high dimensionality distance metrics like euclidean won't work. Also, since, algorithm like HDBSCAN rely on such metrics without doing any PCA. Thereby, we believe that the results for the digits dataset using HDBSCAN won't be good due to the curse of dimensionality.
Furthermore, we also didn't use iris because it was not densely packed.

DATASET 2-

For hierarchical clustering we have utilized the iris dataset which is available in sklearn.datasets.



We have made use of iris dataset for hierarchical clustering because of the following reasons-

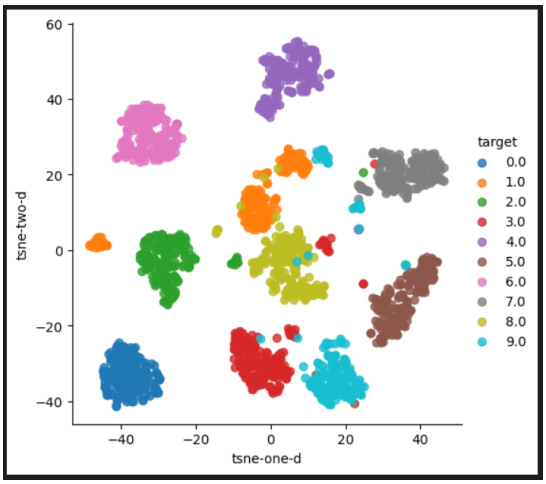
1. This is because there are only two distinct clusters in the iris dataset because Iris Virginica and Iris Versicolor share many characteristics.

In contrast, we can split a single cluster into numerous sub-clusters using hierarchical clustering. This can be understood to mean: Iris Versicolor and Iris Virginica are both members of the same "superclass." Agglomerative clustering performs best on the Iris dataset because hierarchical clustering does a great job of capturing superclasses.

2. On the other hand, the digits dataset lacks such a "superclass," making agglomerative clustering inferior to other clustering techniques.
3. The credit card fraud detection dataset is the same in that there is no superclass that is further subdivided into subclasses. Thus, we are hypothesizing to make use of iris dataset out of all 3 datasets for hierarchical clustering.

DATASET 3-

For prototyping-based clustering, we have used the digits dataset from sklearn.datasets library.



It is because of the following reasons-

1. This is because of the fact that the digits dataset is quite high in dimensionality and is well separated and has globular-shaped clusters which is visible through it's t-sne plot before clustering.
2. Even though digits dataset has well separated clusters as seen in the t-sne plot. We still didn't use density-based clustering because of its

high dimensionality of 64x1 for each data sample. Thus, we hypothesize that techniques like k-means, k-medoids and spectral clustering would work better than other density-based and hierarchical clustering methods.

Question 2

PART B

Advantages of DBSCAN:-

1.) DBSCAN doesn’t require any pre-input of the number of clusters like in K-Means. Furthermore, the clusters formed can be of arbitrary shapes, unlike in K-Means, where spherical clusters are generally formed.
2.) DBSCAN is able to detect outliers, i.e., it is robust to outliers.

Disadvantages of DBSCAN:-

1.) While DBSCAN performs well in distinguishing between high-density and low-density clusters, it has trouble with clusters of similar densities.
2.) DBSCAN's other drawback is that it is much more susceptible to noise, which could result in incorrect clustering.

PART C

We chose HDBSCAN over DBSCAN because of the following reasons:-

1.) One of the major disadvantages of DBSCAN is it being more susceptible to noise. Thus, HDBSCAN reduces this noise clustering problem as it focuses on high-density clustering.

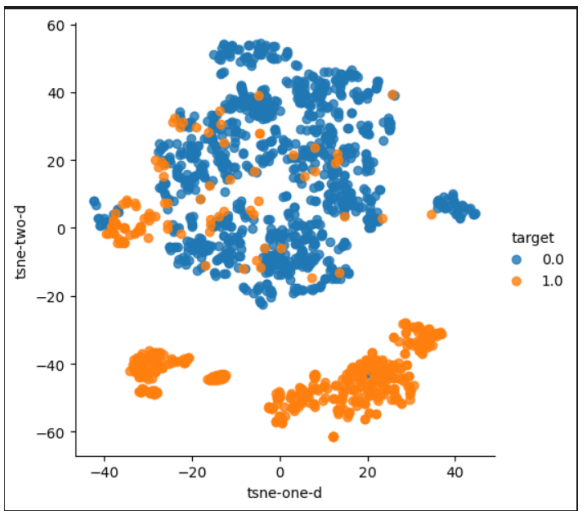
2.) DBSCAN struggles to group data of different densities. On the other hand, HDBSCAN can cluster data of varying densities.

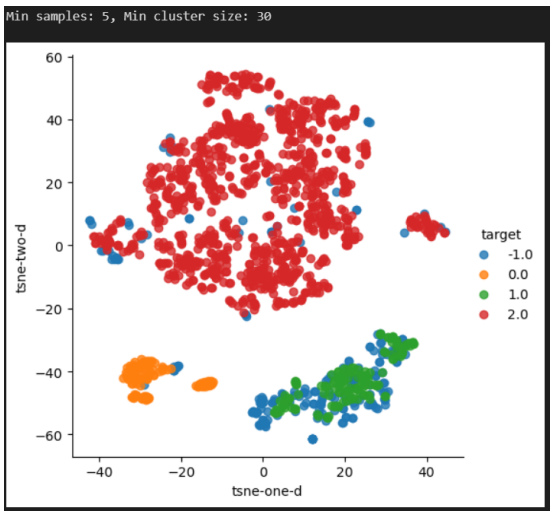
PART D

We used HDBSCAN to check the algorithm's performance on the 3 datasets. We did hyperparamterization of 2 input variables of the HDBSCAN algorithm - “min_cluster_size” and “min_samples.” To check the performance across different input configurations, 2 extrinsic() and 2 intrinsic() measures have been used. Alongwith this, we are also visually comparing the t-sne plots before and after clustering.

The best results for the 3 datasets are as follows:-

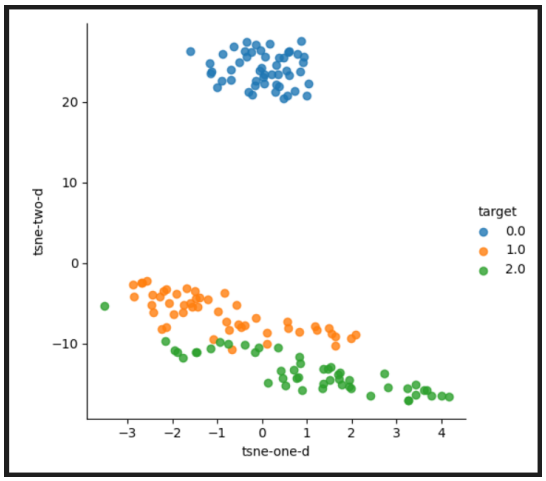
DATASET 1:- (Credit Card Fraud Detection Dataset) - min_samples = 5, min_cluster_size = 30

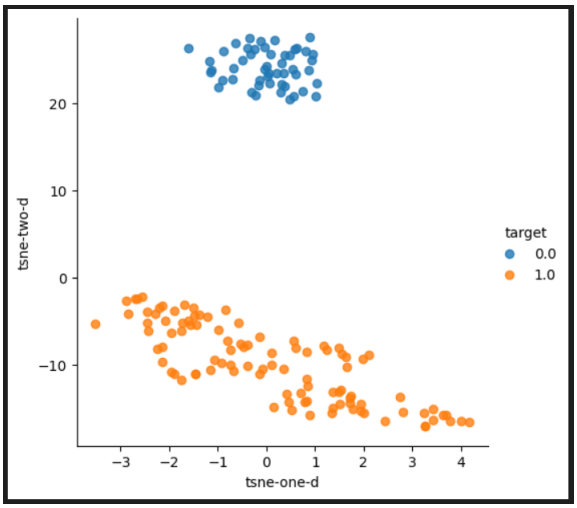





```
Estimated number of clusters:  4
Intrinsic Measures:  {'silhouette score': 0.3799638776750195, 'calinski_harabasz_index': 704.6588038590676, 'DB index': 5.283591206720166}
Extrensic Measures:  {'adjusted_rand_index': 0.6550096998164158, 'mutual_information': 0.5095963588100617}
```

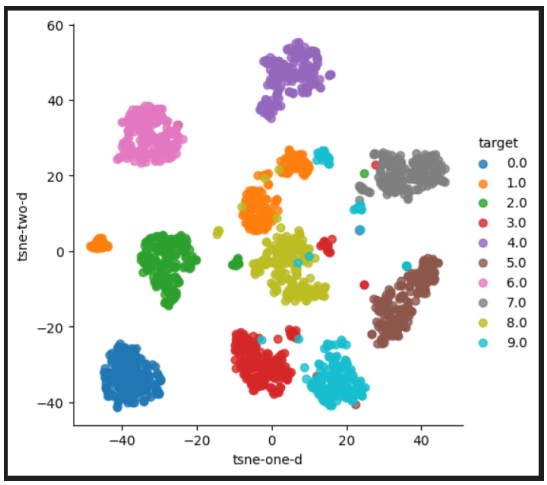
DATASET 2:- (Iris Dataset) - min_samples = 2, min_cluster_size = 20

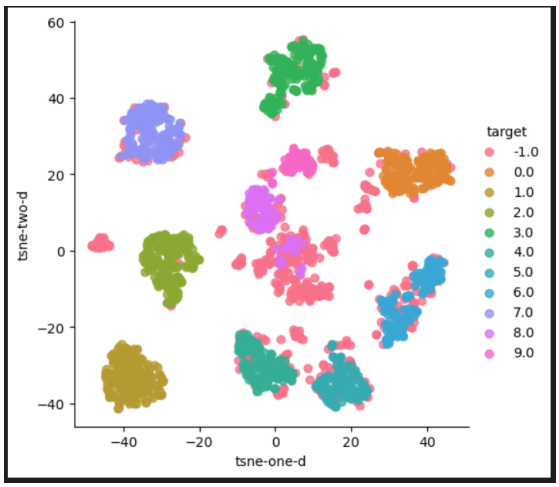





```
Estimated number of clusters:  2
Intrinsic Measures:  {'silhouette score': 0.6867350732769777, 'calinski_harabasz_index': 502.82156350235897, 'DB index': 0.38275284210068705}
Extrensic Measures:  {'adjusted_rand_index': 0.5681159420289855, 'mutual_information': 0.7315847607219573}
```

DATASET 3:- (Digits Dataset) - min_samples = 2, min_cluster_size = 30



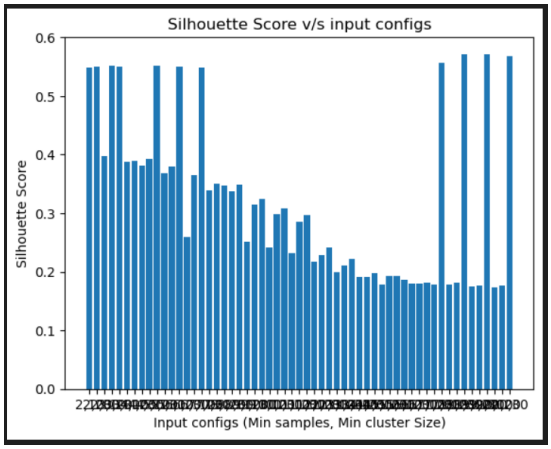


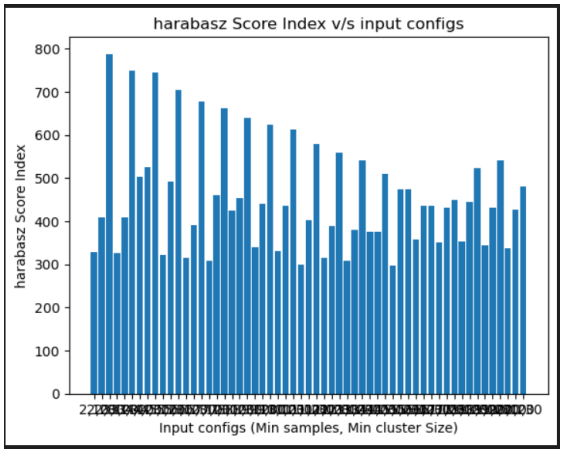

```
Estimated Number of Clusters:  11
Intrinsic Measures:  {'silhouette score': 0.13040046927804297, 'calinski_harabasz_index': 127.3924728905354, 'DB index': 2.1992396422653964}
Extrinsic Measures:  {'adjusted_rand_index': 0.5903209567287874, 'mutual_information': 0.7838342404846185}
```

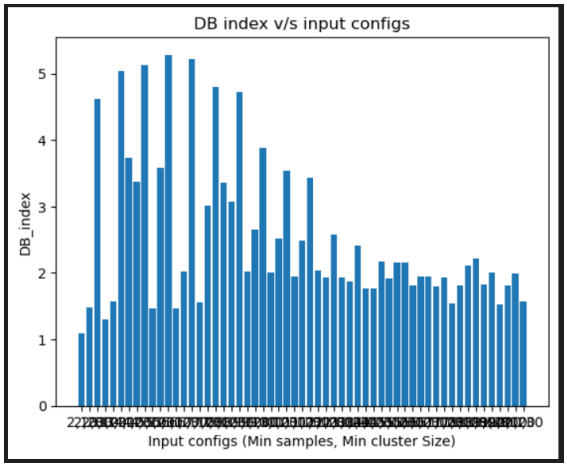
PART E

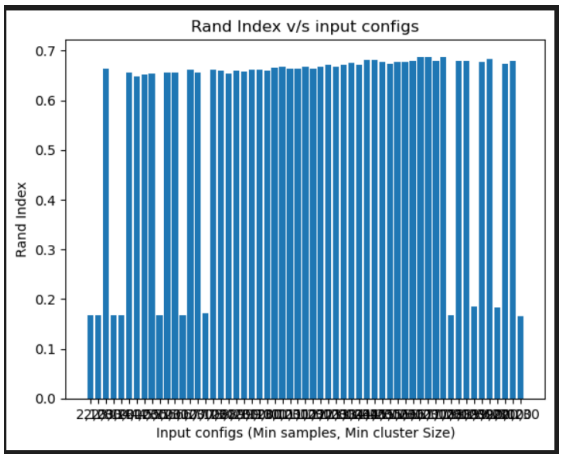
For DATASET 1:-

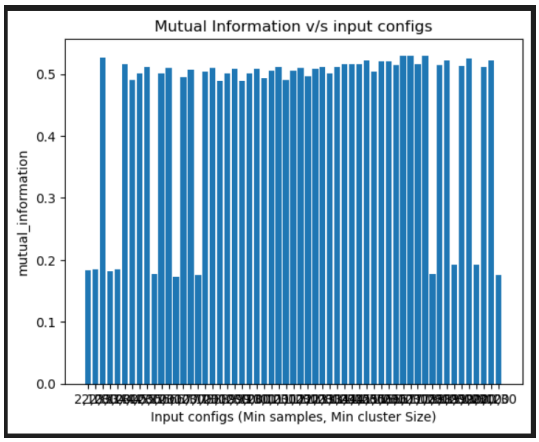
We used different configurations of the HDBSCAN algorithm by varying min_samples(range 2-20 both included) and min_cluster_size (10,20,30). After doing this we achieved the following metrics values shown in the form of bar plots.











Furthermore, we also compared different t-sne plots after clustering with before clustering t-sne plots, to see if the clustering results resemble the original labels distribution. We observed that for (min_samples = 5 and min_cluster_size = 10) the best silhouette score of 0.55 is observed, however the rest of the metrics are of less value. Also the clustering results too are not that good because two major clusters of distinct labels are clustered into the same sample. This is also quite visible in various other

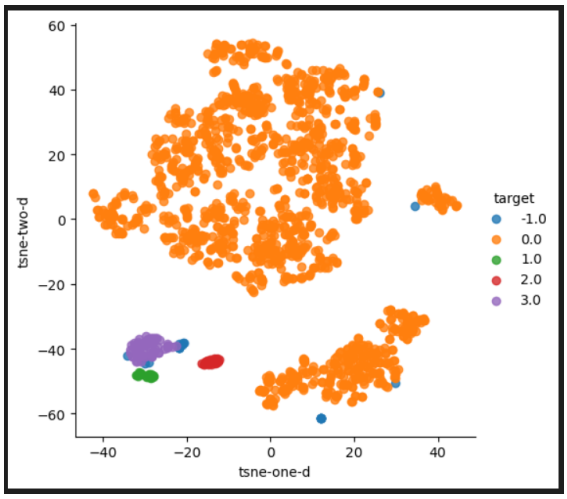
configurations like (min_samples = 2, min_cluster_size = 10) & (min_samples = 2, min_cluster_size = 20).

Thus, it is also needed to see if the clustered t-sne plots are resembling the original t-sne plots.

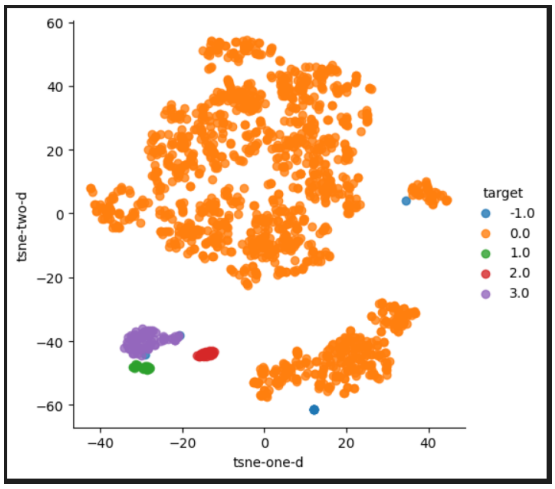
This was seen in many configurations, however the best performance measures was observed in the case where min_samples = 3, min_cluster_size = 30. We didn't pick the configuration where the silhouette score was the best(discussed above) because its clustering results were not that good. The results for these configurations alongwith their clustered t-sne plots are shown in the table below.

S.No.	Min_Samples	Min_cluster_size	Silhouette Score	Harabasz Score Index	DB Index	Rand Index	Mutual Information
1.	5	10	0.55	320.628	1.47	0.16	0.176
2.	2	10	0.548	327.574	1.08	0.167	0.182
3.	**3**	**30**	**0.387**	**747.714**	**5.033**	**0.656**	**0.516**
4.	11	30	0.296	578.982	3.435	0.664	0.509

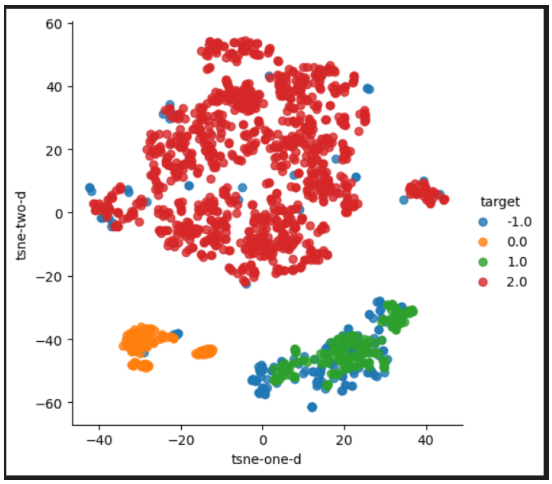
Table of Hyper-params Results (Bold one is the best case)



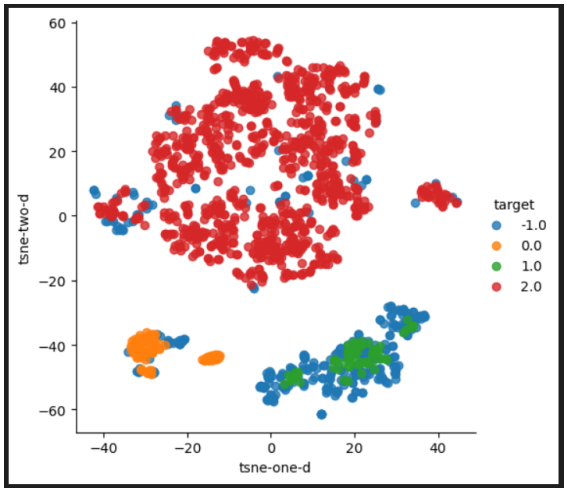
Min_Samples = 5, Min_Cluster_Size = 10



Min_Samples = 2, Min_Cluster_Size = 10



Min_Samples = 3, Min_Cluster_Size = 30



Min_Samples = 11, Min_Cluster_Size = 30

PART F

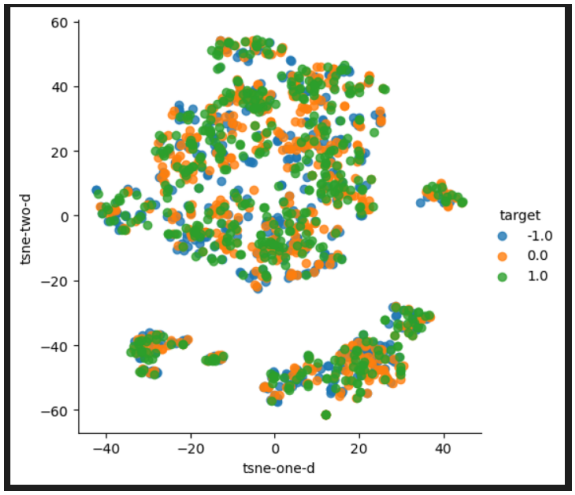
Our initial hypothesis was that one of the clusters before clustering was dense enough and well separated from the other label cluster which was not that densely separated. Thus, we initially thought that density-based clustering would give good results.
From the experiments as well, we saw that one of the larger clusters was well-labelled and resembled the original t-sne plot cluster. Also, there were some orange labels in the blue label region before clustering. These data samples were classified as outliers by HDBSCAN which is quite intuitive because those orange labels were in quite low quantity in the large blue cluster. Furthermore, after clustering 2 more clusters are formed because label 1 cluster in the original t-sne was seperated into 2 portions both dense, these regions were later labelles into different clusters due to the large inter-region distance, showing that HDBSCAN was able to work in dense clusters. Thus, by keeping a tradeoff of min_samples and min_cluster_size we were able to achieve sufficiently good results on the basis of the 4 performance metrics.

PART G

We have done the following technique to confirm whether HDBSCAN clustering results are random or not.

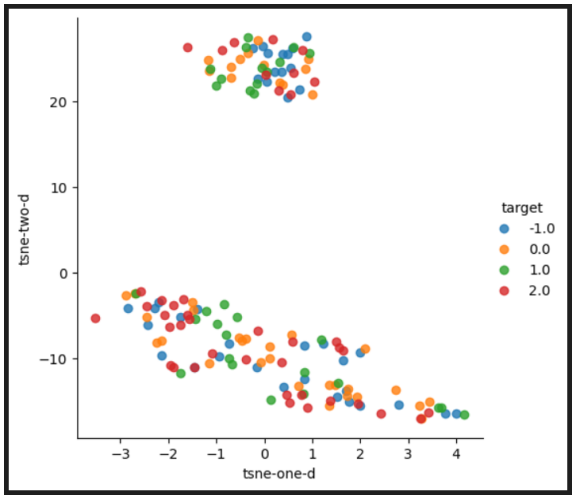
As we know the total number of labels in the given dataset, thus we assigned random cluster labels to all the data samples in the range [-1 to n-1] where -1 represents outliers (since HDBSCAN can detect outliers) and n is the total number of labels in the dataset.
We then plot the T-SNE plot using these random cluster labels and then found the extrinsic and intrinsic measure values. Later, we compared these results with the best results that we achieved on the given dataset using HDBSCAN.

Dataset 1:-



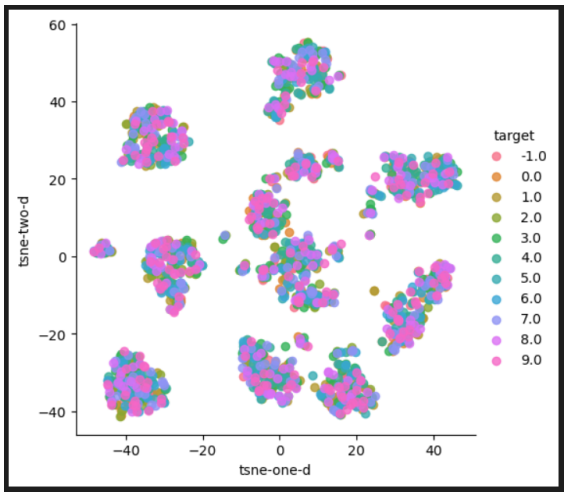

```
Estimated number of clusters:  3
Extrinsic Measures:  {'adjusted_rand_index': 0.001656935419097549, 'mutual_information': 0.0013023804124130314}
Intrinsic Measures:  {'silhouette score': -0.019203137780032235, 'calinski_harabasz_index': 1.2499327973234644, 'DB index': 45.11743602798947}
```

Dataset 2:-




```
Estimated Number of Clusters:  4
Extrinsic Measures:  {'adjusted_rand_index': -0.008342734517809982, 'mutual_information': -0.008569133494379221}
Intrinsic Measures:  {'silhouette score': -0.07198698008525652, 'calinski_harabasz_index': 0.4022626881356364, 'DB index': 21.91105083213749}
```

Dataset 3:-




```
Estimated Number of Clusters:  11
Extrinsic Measures:  {'adjusted_rand_index': -0.0006932201966487469, 'mutual_information': -0.0012975237684628759}
Intrinsic Measures:  {'silhouette score': -0.01825689730508569, 'calinski_harabasz_index': 0.9593657759762586, 'DB index': 23.048146285540945}
```

We can clearly see that for every dataset, the T-SNE plot after clustering is very random and doesn't significantly to the original T-SNE plot before clustering. On the other hand, without doing random clustering (PART D), we can see a significant resemblance between the T-SNE plots before and after clustering. Also, the performance measures (extrinsic and intrinsic) are quite bad in comparison to the results achieved in PART D.

Question 3

PART A

There are 2 main categories of hierarchical clustering and they are as follows:-

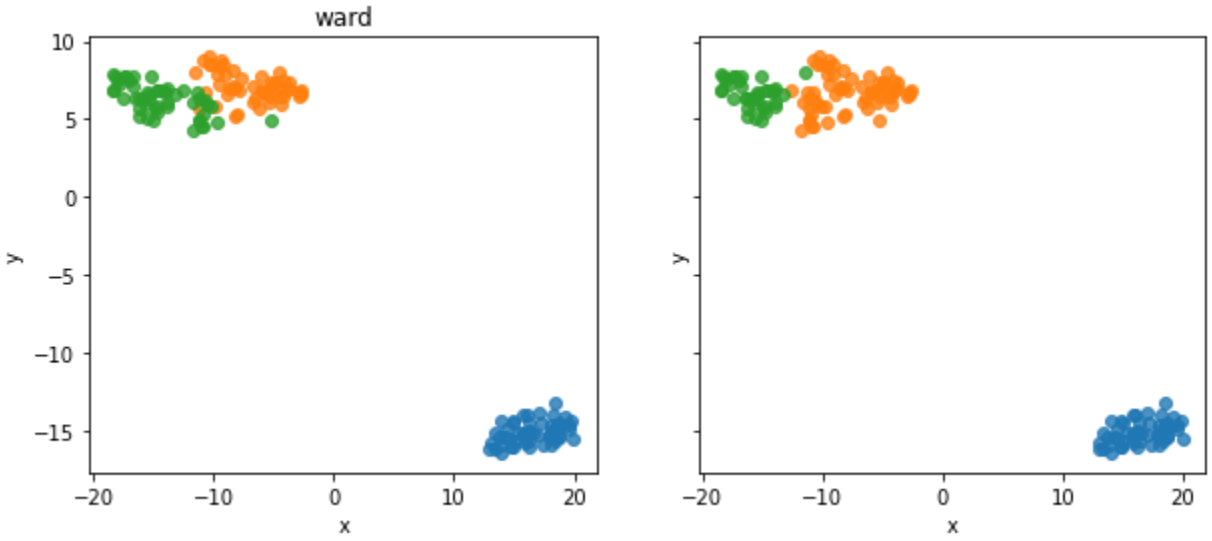
1.) Agglomerative:-. This type of clustering has a bottom-up approach
2.) Divisive:- It is a kind of top-down approach
Both these approaches define a notion of the proximity of clusters where we start the points as individual clusters and then start merging the closest pairs into clusters.
Among the 2 techniques, agglomerative clustering is more frequently used. Additionally, the divisive approach is of exponential complexity than the

agglomerative, which has a complexity of O(n^3) in its most fundamental form and even drops to O(n^2) in their optimized versions.
Because it is less complicated than the divisive approach, agglomerative clustering is used more frequently than the other method. Furthermore, Agglomerative clustering can distinguish between clusters with sufficient accuracy in a short amount of time, despite not providing the best solution.

PART C

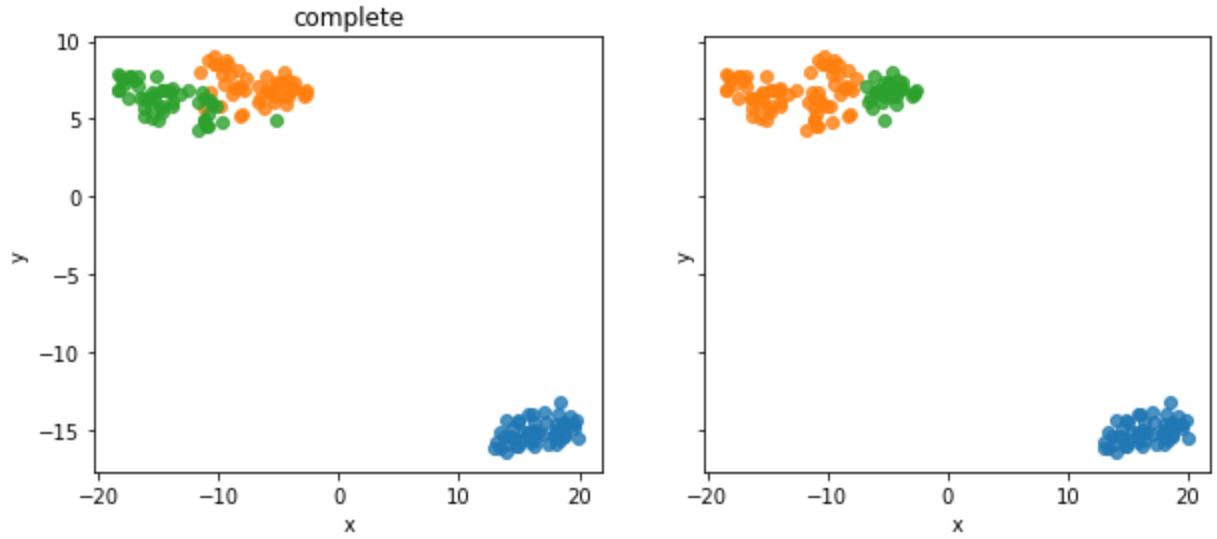
Iris Dataset

Linkage: ward



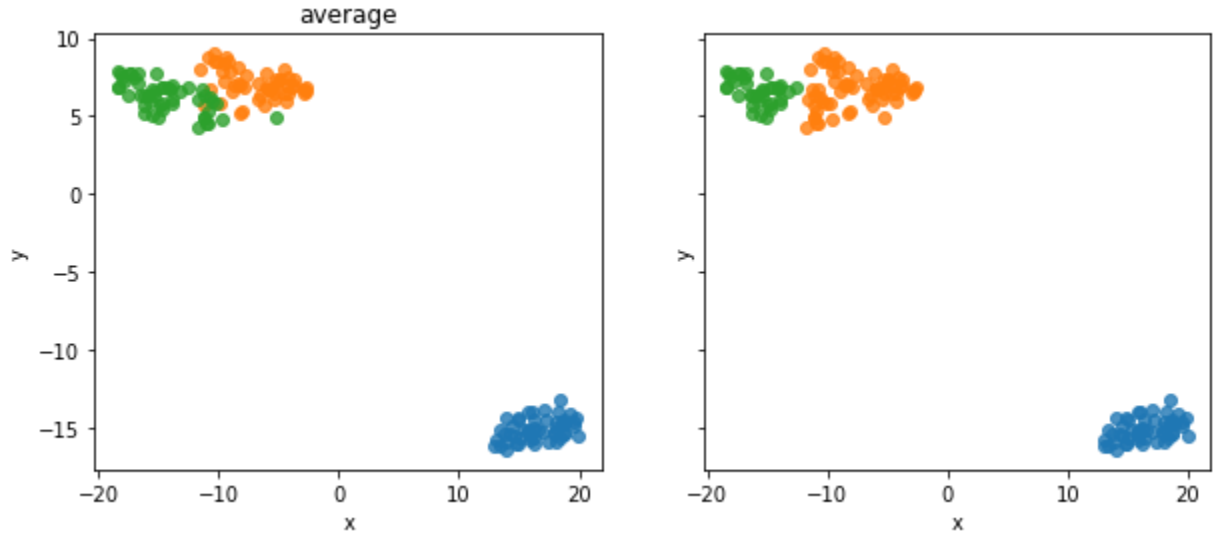
{'adjusted_rand_index': 0.7311985567707746, 'adjusted_mutual_information': 0.7671669615713113}
{'silhoutte_coefficient': 0.5543236611296419, 'calinski_harabasz_index': 558.0580408128307, 'davies_bouldin_index': 0.6562564540642021}

Linkage: complete



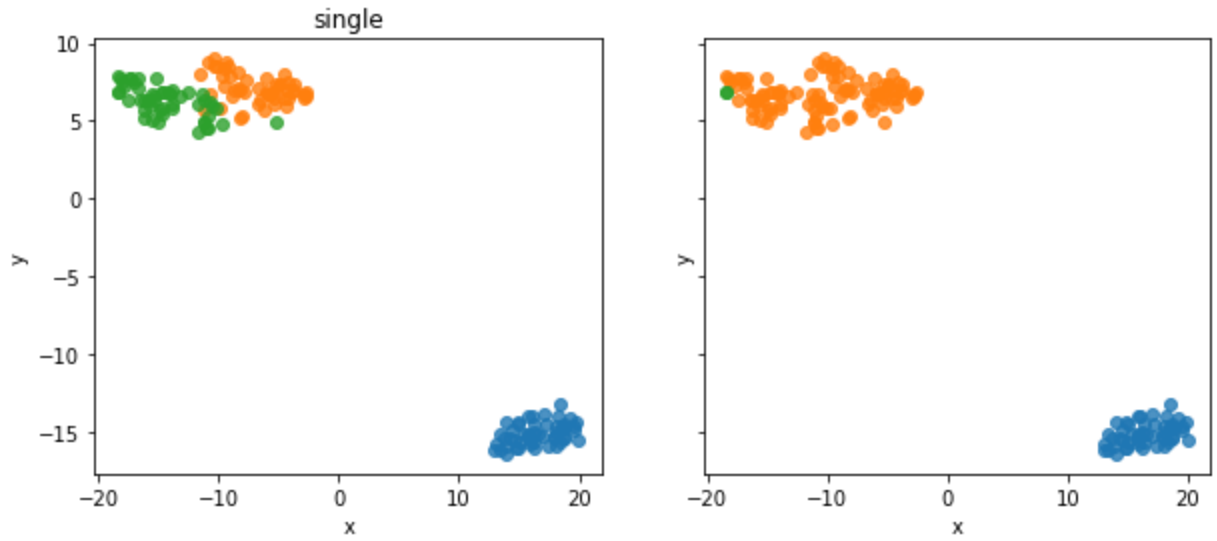
{'adjusted_rand_index': 0.6422512518362898, 'adjusted_mutual_information': 0.7184641371994783}
{'silhoutte_coefficient': 0.5135953221192214, 'calinski_harabasz_index': 485.9050227341817, 'davies_bouldin_index': 0.6333339304359707}

Linkage: average



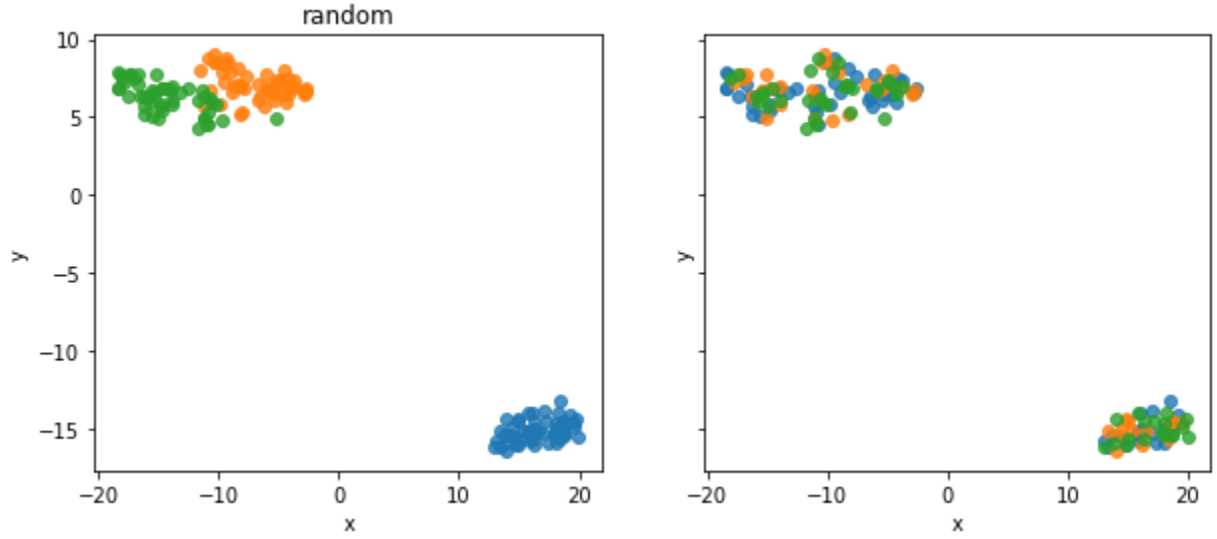
{'adjusted_rand_index': 0.7591987071071522, 'adjusted_mutual_information': 0.8032287370935434}
{'silhoutte_coefficient': 0.5541608580282851, 'calinski_harabasz_index': 556.8795419179529, 'davies_bouldin_index': 0.658444278322429}

Linkage: single



{'adjusted_rand_index': 0.5637510205230709, 'adjusted_mutual_information': 0.7125764811325078}
{'silhoutte_coefficient': 0.5121107753649307, 'calinski_harabasz_index': 277.9946762646194, 'davies_bouldin_index': 0.4471537628542408}

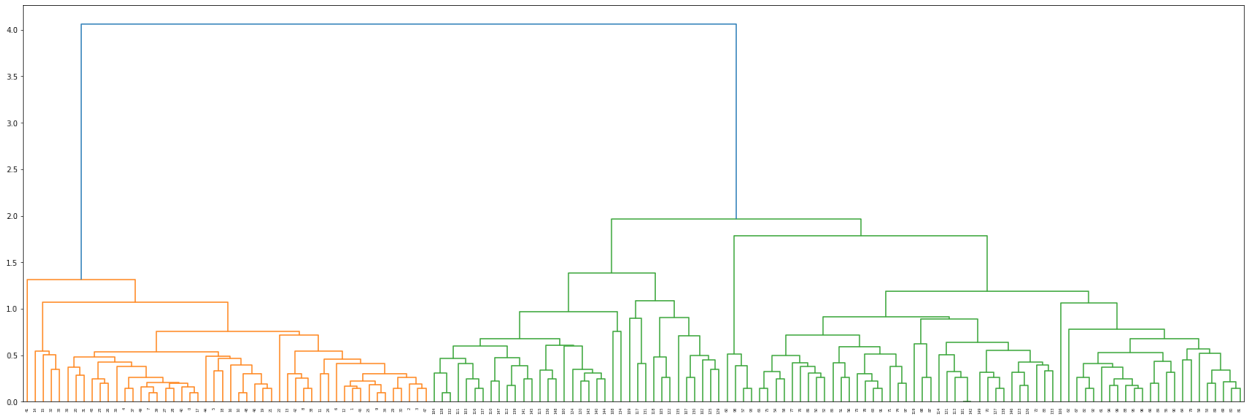
Random clustering



{'adjusted_rand_index': -0.011522586950724426, 'adjusted_mutual_information': -0.010882017797028453}
{'silhoutte_coefficient': -0.030443408695375224, 'calinski_harabasz_index': 0.24690327044645627, 'davies_bouldin_index': 20.995734132300644}

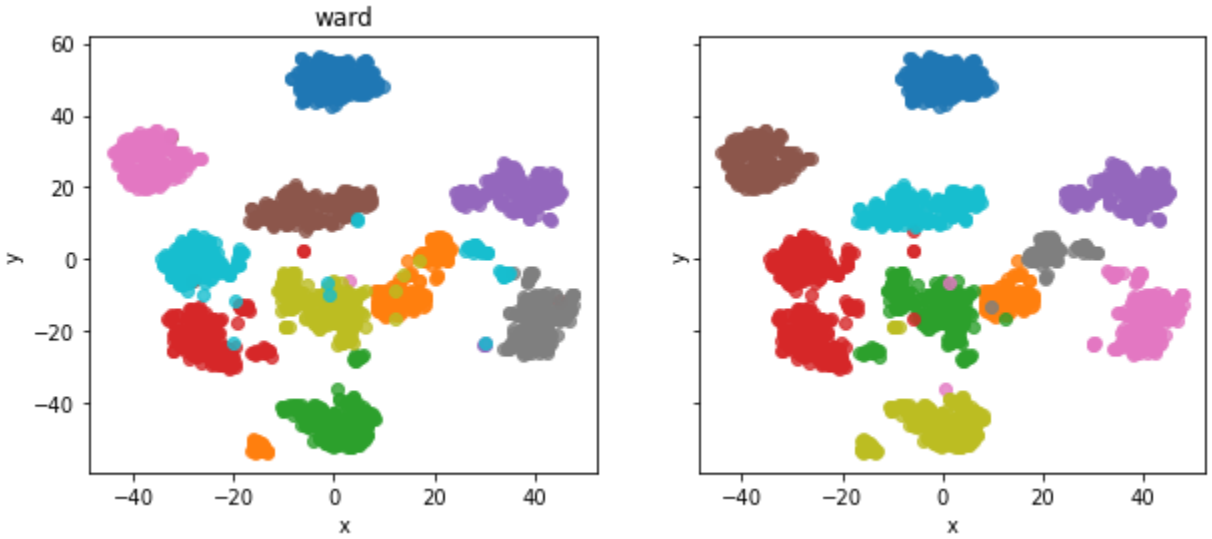
For the iris dataset, the best clustering results are obtained using **average linkage**. It has the highest adjusted rand index and adjusted mutual information scores. The intrinsic measures are behind ward (minimize variance) linkage but only by a very small margin. From the TSNE also, it is clear that this is the best clustering since actual labels and clusters are quite similar.

Dendrogram of average clustering:



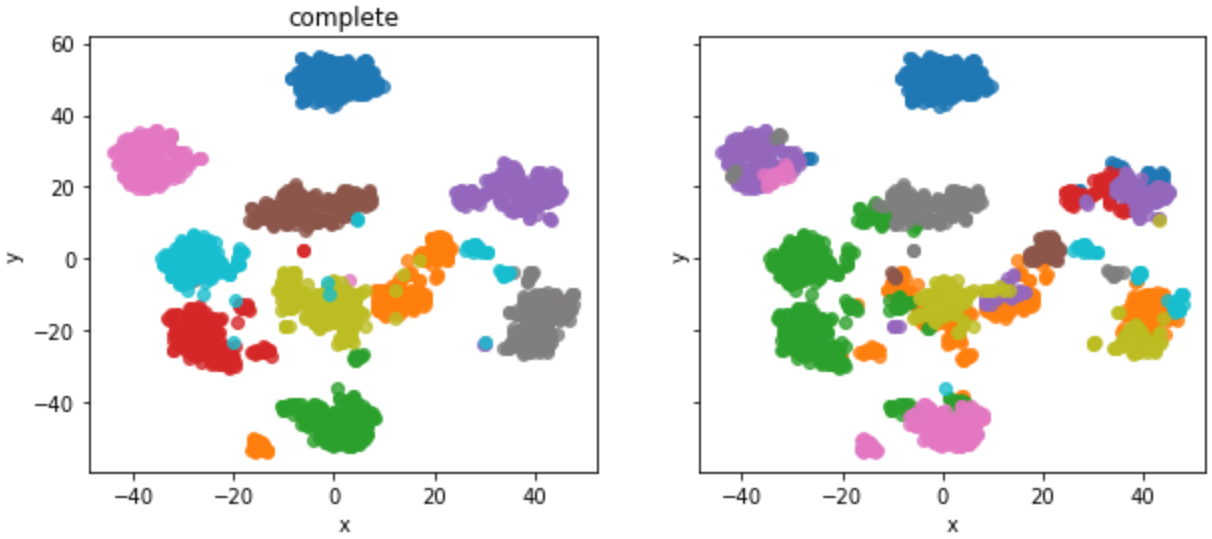
Digits dataset

Linkage: ward



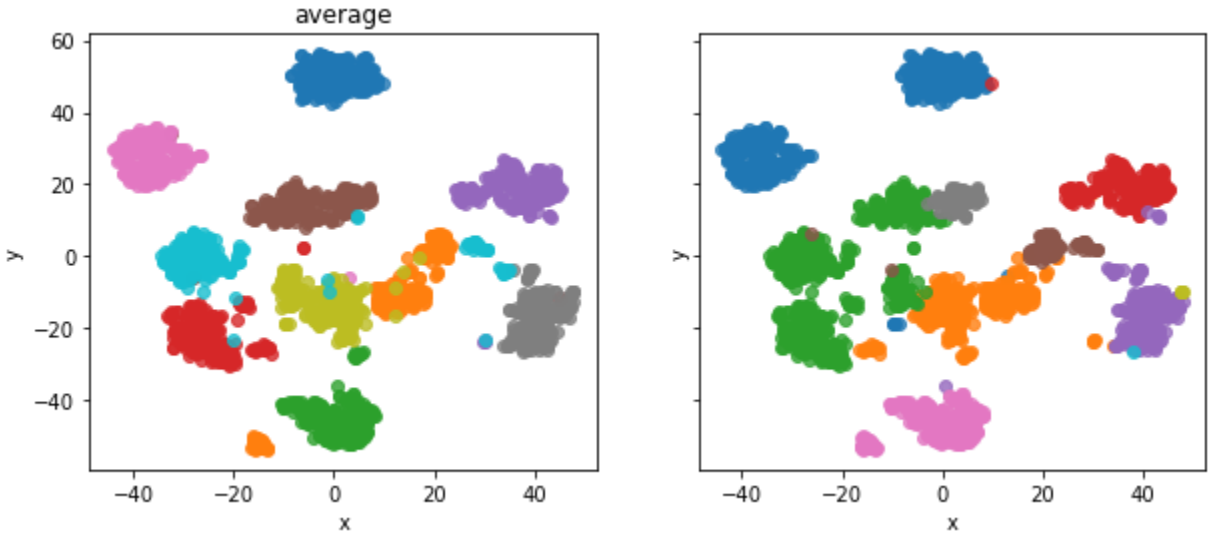
{'adjusted_rand_index': 0.7940031835568753, 'adjusted_mutual_information': 0.8668321489750319}
{'silhoutte_coefficient': 0.17849659940596496, 'calinski_harabasz_index': 161.20475281721804, 'davies_bouldin_index': 1.8889885974945109}

Linkage: complete



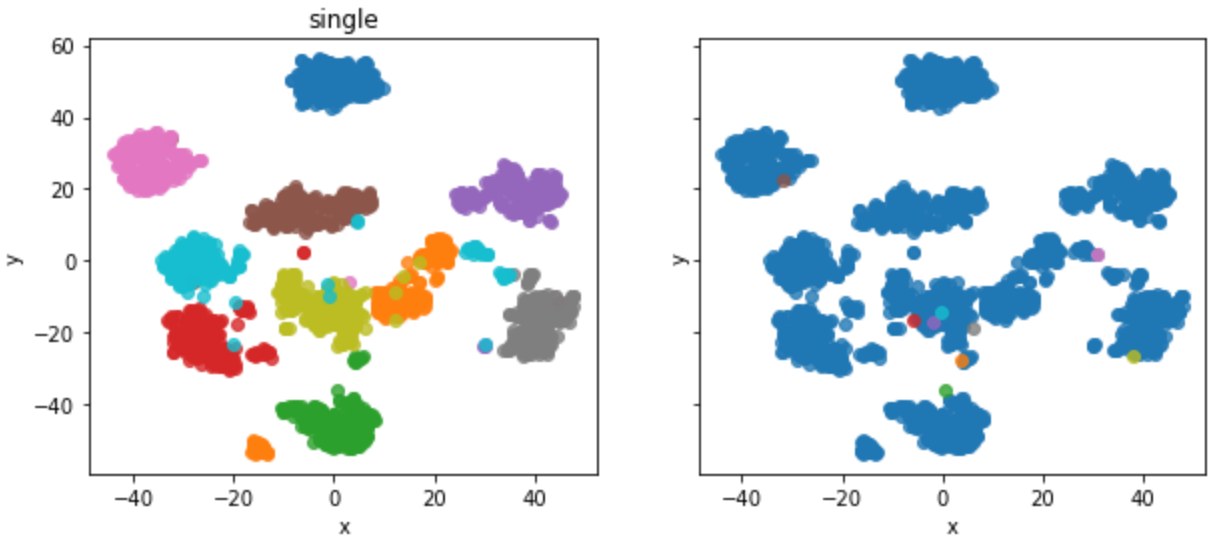
{'adjusted_rand_index': 0.4286888142937744, 'adjusted_mutual_information': 0.6091974206985212}
{'silhoutte_coefficient': 0.1192708999859292, 'calinski_harabasz_index': 117.41350520458832, 'davies_bouldin_index': 2.303120200595817}

Linkage: average



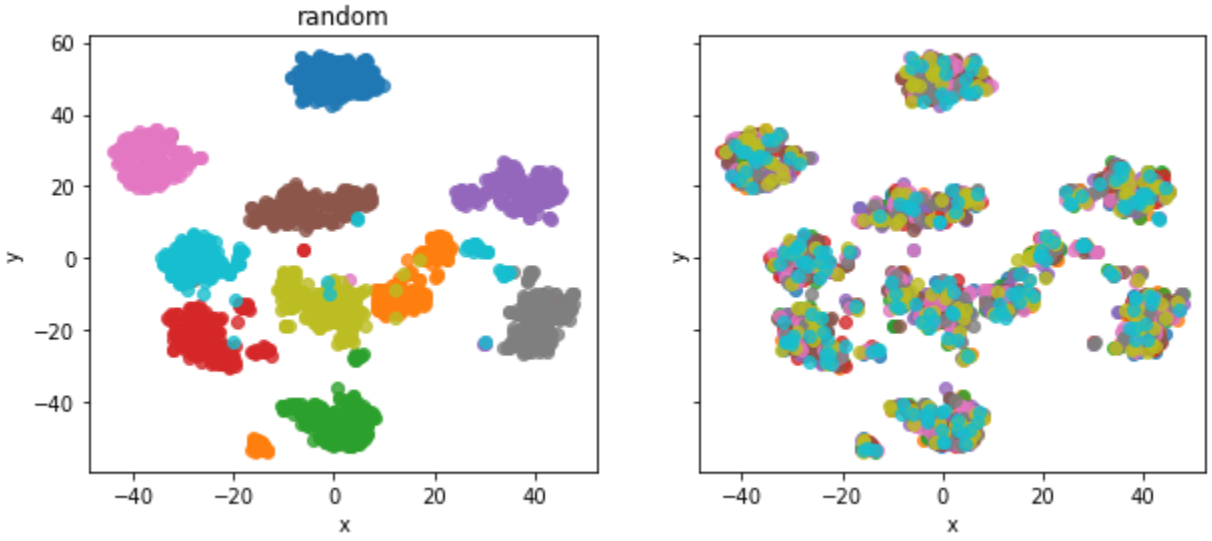
{'adjusted_rand_index': 0.5142255948681158, 'adjusted_mutual_information': 0.7103101155580299}
{'silhoutte_coefficient': 0.14662473962932557, 'calinski_harabasz_index': 122.1106003051252, 'davies_bouldin_index': 1.6868910302128672}

Linkage: single



{'adjusted_rand_index': 4.313084144605482e-05, 'adjusted_mutual_information': 8.438956846943478e-05}
{'silhoutte_coefficient': -0.1336211433529001, 'calinski_harabasz_index': 1.2457366312330127, 'davies_bouldin_index': 0.9239238655143689}

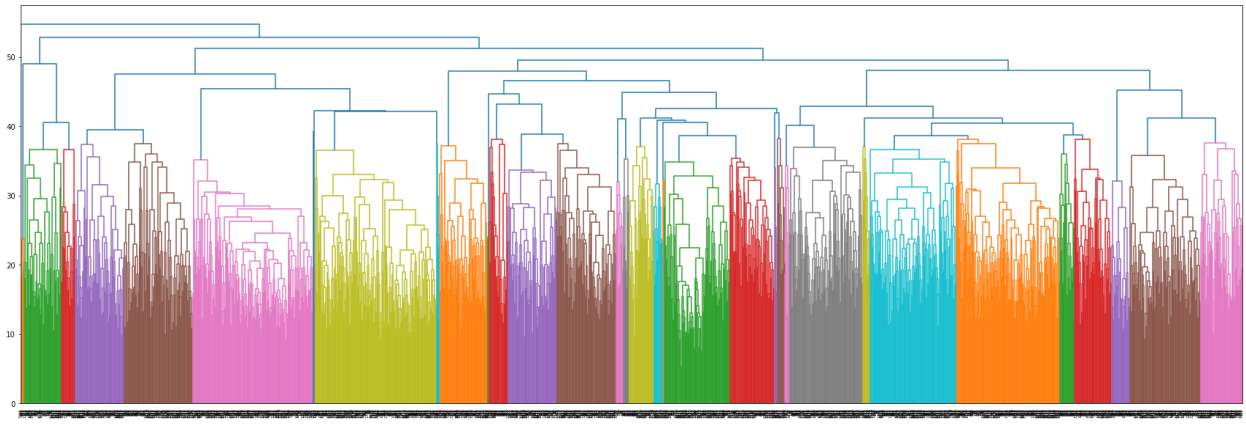
Random clustering



{'adjusted_rand_index': 0.0007665017919979328, 'adjusted_mutual_information': 0.001891693228082355}
{'silhoutte_coefficient': -0.014596159360139938, 'calinski_harabasz_index': 0.9269625875770197, 'davies_bouldin_index': 25.479874841558097}

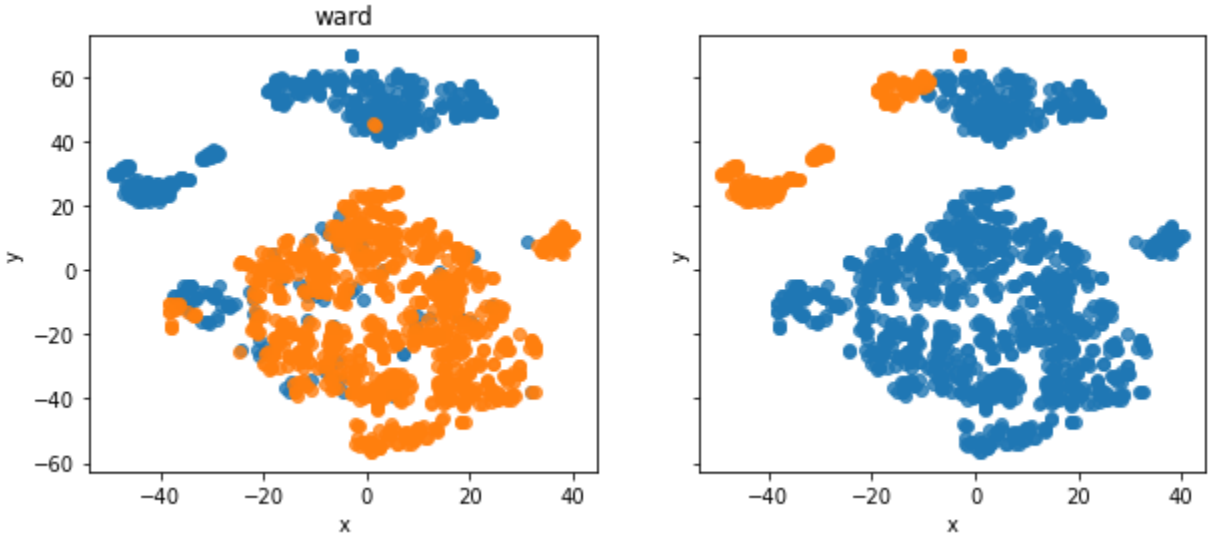
For digits dataset, the best results are obtained using **ward linkage**. It has better values of both intrinsic and extrinsic measures than other linkages. It is also evident from the TSNE that ward linkage produces clusters very similar to the true labels.

Dendrogram of average clustering:



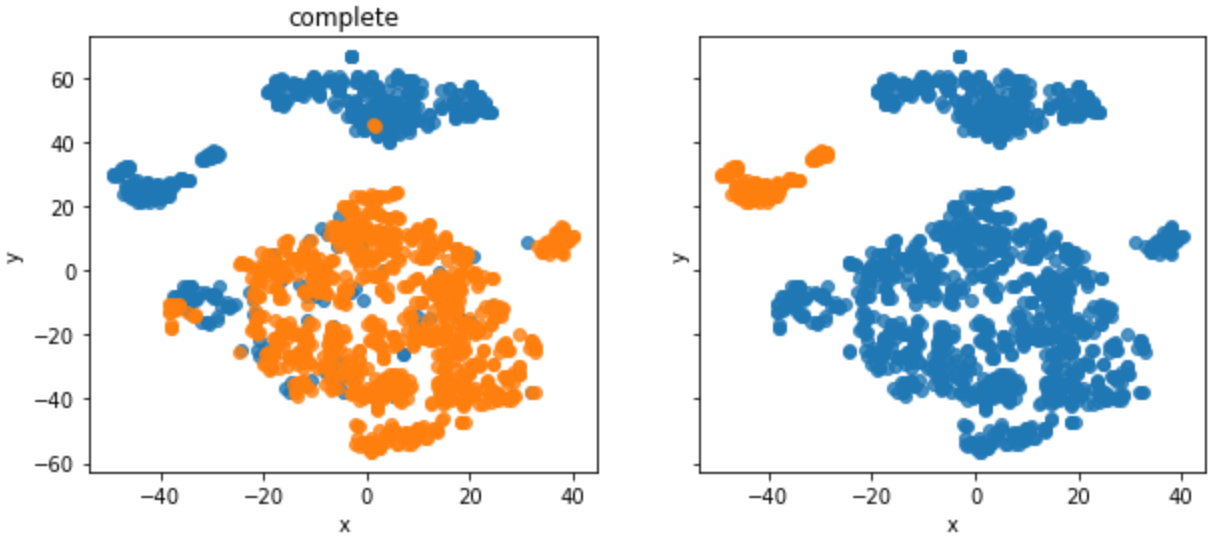
Credit Card Fraud Dataset

Linkage: ward



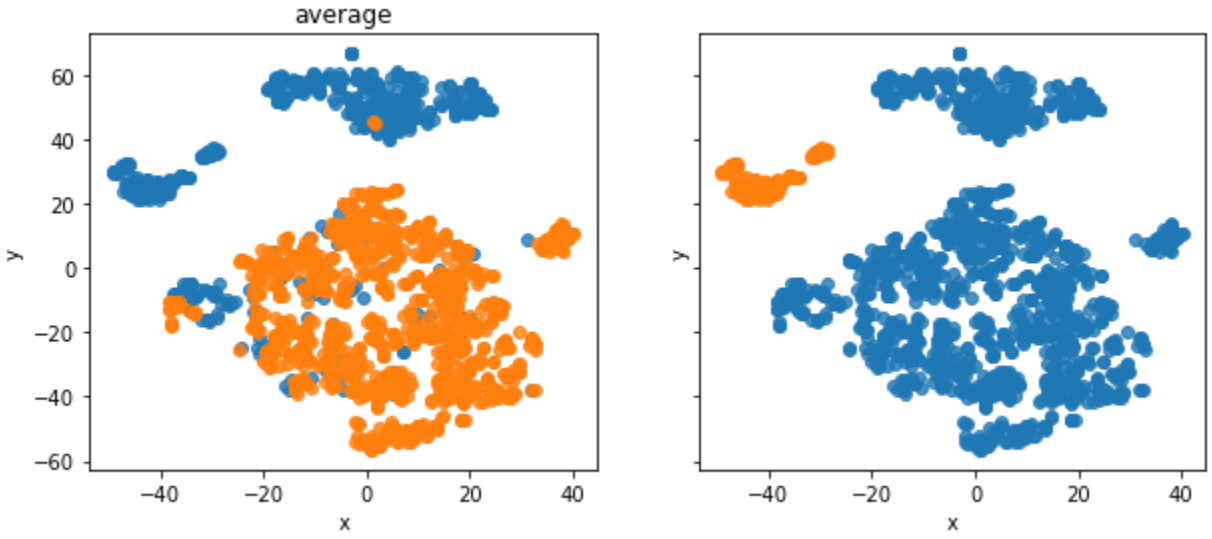
{'adjusted_rand_index': 0.27076842211221824, 'adjusted_mutual_information': 0.2854562141596948}
{'silhoutte_coefficient': 0.616058771099793, 'calinski_harabasz_index': 1465.1122546374725, 'davies_bouldin_index': 0.7303820743456186}

Linkage: complete



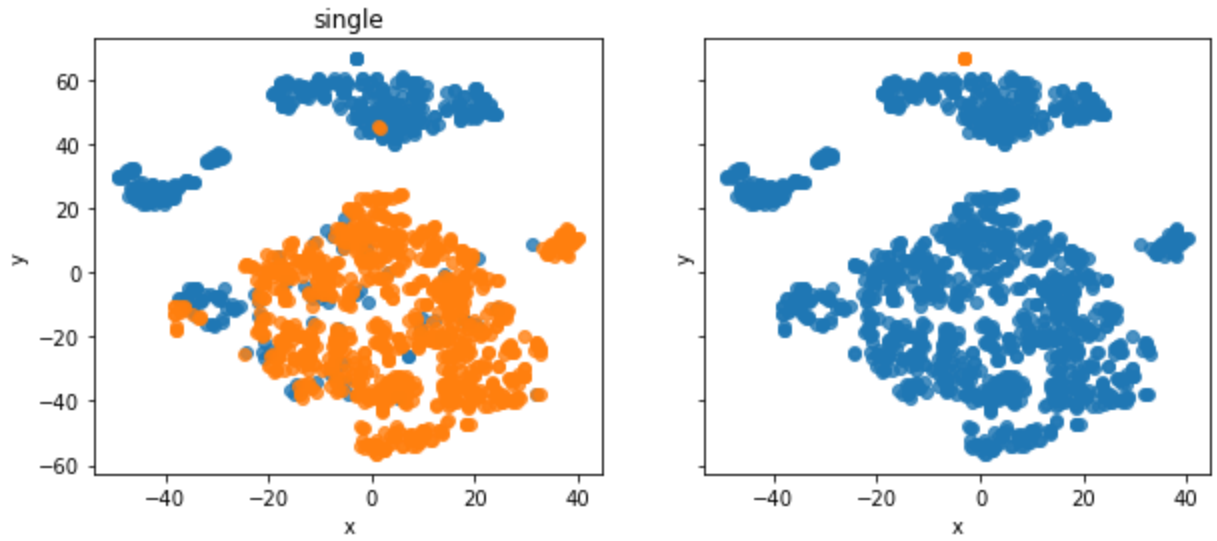
{'adjusted_rand_index': 0.16283326784844426, 'adjusted_mutual_information': 0.190766023260643}
{'silhoutte_coefficient': 0.6332235895986923, 'calinski_harabasz_index': 1115.3231428051263, 'davies_bouldin_index': 0.6402175854341382}

Linkage: average



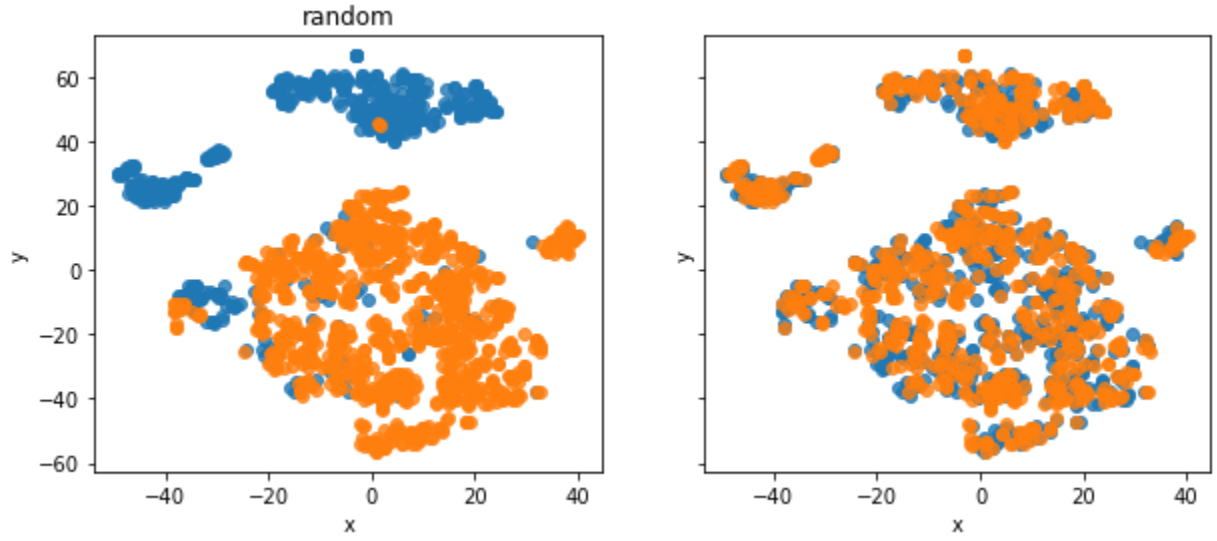
{'adjusted_rand_index': 0.16283326784844426, 'adjusted_mutual_information': 0.190766023260643}
{'silhoutte_coefficient': 0.6332235895986923, 'calinski_harabasz_index': 1115.3231428051263, 'davies_bouldin_index': 0.6402175854341382}

Linkage: single



{'adjusted_rand_index': 0.008340962505214019, 'adjusted_mutual_information': 0.012404664954984173}
{'silhoutte_coefficient': 0.48271640000632, 'calinski_harabasz_index': 25.04426451658953, 'davies_bouldin_index': 0.4101553764083515}

Random clustering

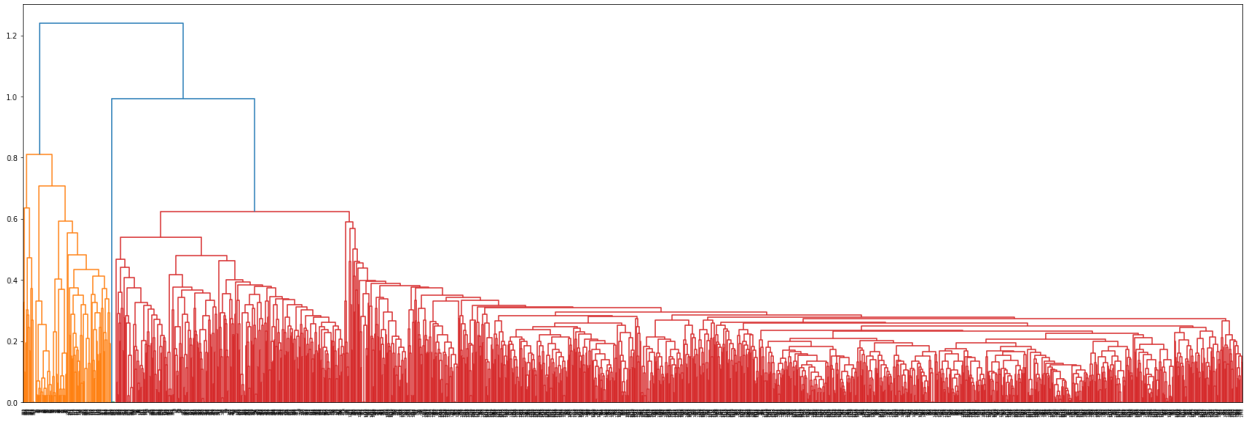


{'adjusted_rand_index': -0.0004168138125957458 3, 'adjusted_mutual_information': -0.00030804572088447793}
{'silhoutte_coefficient': -0.0002415660842069361, 'calinski_harabasz_index': 1.046248933009503, 'davies_bouldin_index': 31.671008429850033}

Ward linkage performs best for credit card fraud detection dataset. It outperforms all other linkages in all intrinsic and extrinsic measures. TSNE also shows that its clusters resemble actual labels the most. However, in general the clustering is not very good. Since both classes

have a lot of overlap in the data, agglomerative clustering does not work very well with this dataset.

Dendrogram of average clustering:



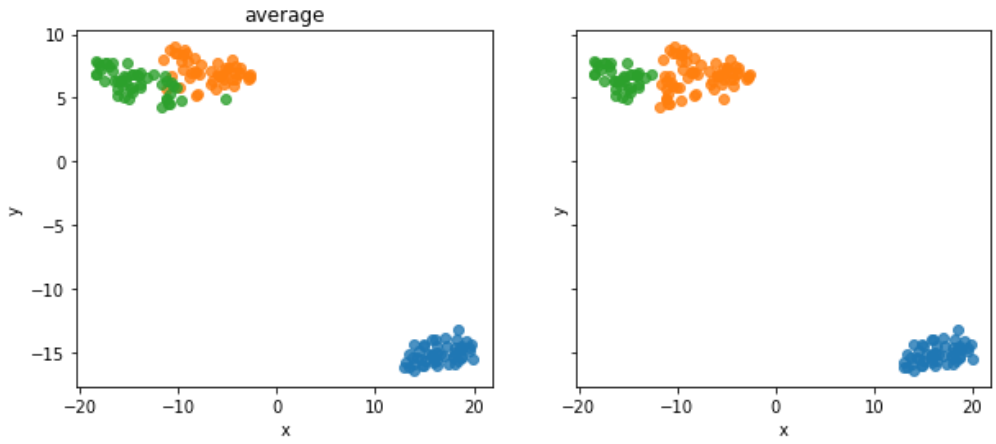
Digits vs Iris vs Credit Card Fraud Detection

Although agglomerative clustering performs well on both digits as well as iris dataset, we see that it performs better on iris. This is because in iris, Iris Virginica and Iris Versicolor have quite similar attributes which leads to the Iris dataset having only 2 separable clusters. However, in hierarchical clustering, we can divide a single cluster into various sub-clusters. This can be interpreted as follows: Iris Virginica and Iris Versicolor belong to the same “superclass”. Superclasses are captured really well by hierarchical clustering and so agglomerative clustering performs the best on Iris dataset.

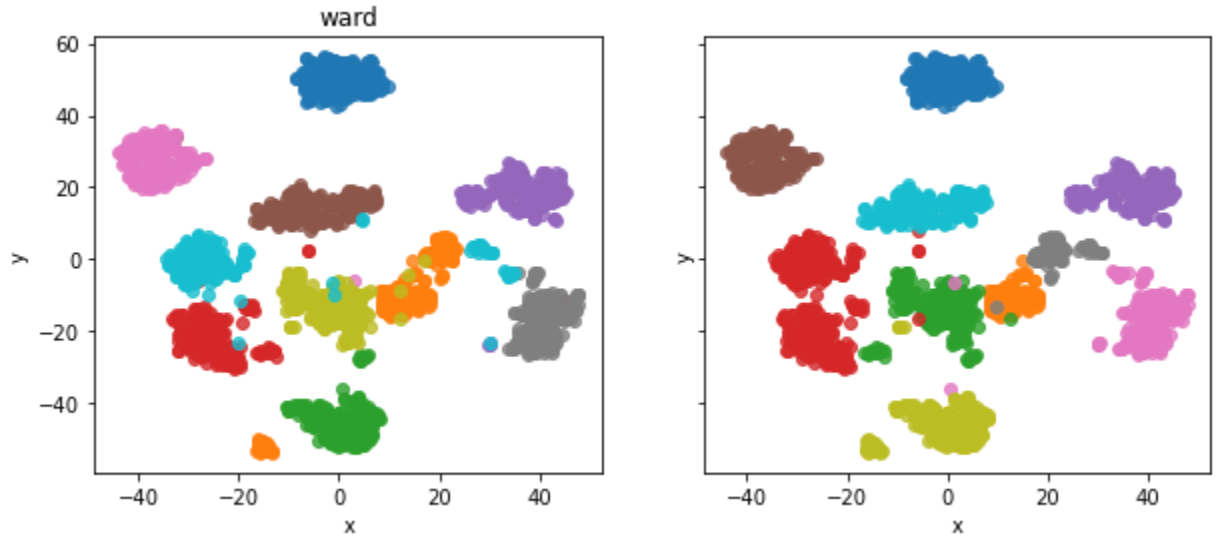
Digits dataset on the other hand has no such “superclass” and so agglomerative clustering does not offer any advantage over other clustering methods.

It is the same case with credit card fraud detection dataset – there is no superclass which is further divided into subclasses. As is evident from the empirical measures as well as the TSNE plots, agglomerative clustering does not perform very well on credit card fraud detection dataset.

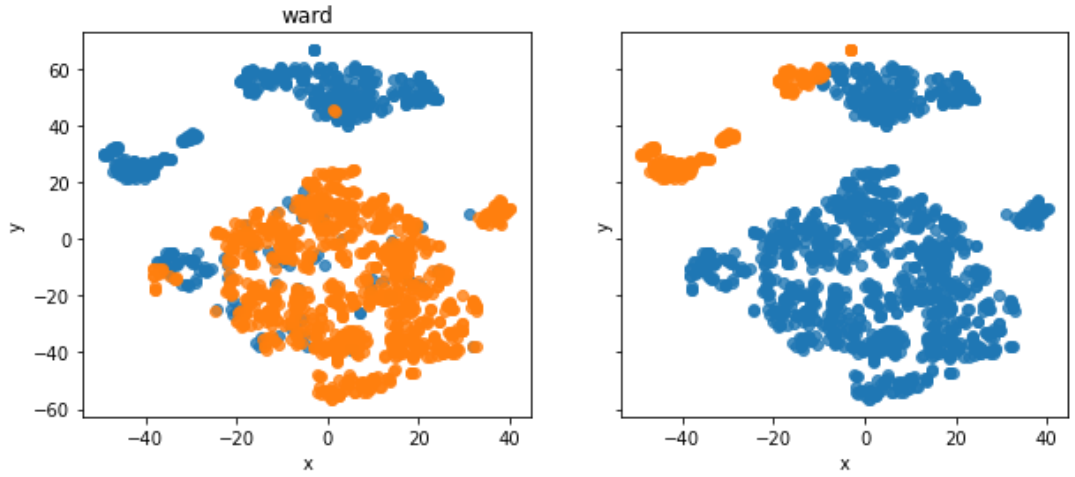
Clustering on Iris dataset:



Clustering on Digits dataset



Clustering on credit card fraud detection



Note that it usually doesn't make much sense to compare empirically using intrinsic and extrinsic measures over different datasets. This is because certain metrics are unnormalized and thus can give very high measures for datasets with more points. However, it makes sense to use empirical measures to compare clusterings on the same dataset (as have been done while finding the best linkage for each dataset).

PART D

As we see in the previous part, the best results for agglomerative clustering are obtained on Iris dataset which is the one we chose for this clustering. This is because the Iris dataset contains a "superclass" cluster which is further divided into 2 more classes which are Iris Virginica and Iris Versicolor. Hierarchical clustering works best with this kind of data and thus performs better on it than the other datasets thus confirming our hypothesis.

PART E

To compare our clusterings with random data, we assign random clusters to each data point in the dataset. We then compute all the measures for this random assignment of clusters. Then, we compare the results obtained for the random assignment of clusters with those obtained using specific algorithms.

Doing this ensures that our algorithm gives some defined clusters and not just random ones.

Iris Dataset

The best result for this dataset were obtained on average linkage. The TSNE plots for both average linkage and random clustering are already shown in part C. Their results are as follows:

For average linkage:
{'adjusted_rand_index': 0.7591987071071522, 'adjusted_mutual_information': 0.8032287370935434}
{'silhoutte_coefficient': 0.5541608580282851, 'calinski_harabasz_index': 556.8795419179529, 'davies_bouldin_index': 0.658444278322429}

For random clustering:
{'adjusted_rand_index': -0.011522586950724426, 'adjusted_mutual_information': -0.010882017797028453}
{'silhoutte_coefficient': -0.030443408695375224, 'calinski_harabasz_index': 0.24690327044645627, 'davies_bouldin_index': 20.995734132300644}

Lower davies_boulding_index implies better clustering. Clearly, agglomerative clustering performs better than random clustering according to all the measures thus ensuring that the clustering is valid.

Digits dataset

The best result for this dataset is obtained on ward linkage. The TSNE plots have already been plotted in part C. The results are as follows:

For ward linkage:
{'adjusted_rand_index': 0.7940031835568753, 'adjusted_mutual_information': 0.8668321489750319}
{'silhoutte_coefficient': 0.17849659940596496, 'calinski_harabasz_index': 161.20475281721804, 'davies_bouldin_index': 1.8889885974945109}

For random clustering:
{'adjusted_rand_index': 0.0007665017919979328, 'adjusted_mutual_information': 0.001891693228082355}

{'silhoutte_coefficient': -0.014596159360139938, 'calinski_harabasz_index': 0.9269625875770197, 'davies_bouldin_index': 25.479874841558097}

Again, agglomerative clustering performs better than random clustering in all measures thus ensuring that the clustering is valid.

Credit Card Fraud Detection Dataset

The best result for this dataset is obtained on ward linkage. The TSNE plots have already been plotted in part C. The results are as follows:

For ward linkage:
{'adjusted_rand_index': 0.27076842211221824, 'adjusted_mutual_information': 0.2854562141596948}
{'silhoutte_coefficient': 0.616058771099793, 'calinski_harabasz_index': 1465.1122546374725, 'davies_bouldin_index': 0.7303820743456186}

For random clustering:
{'adjusted_rand_index': -0.00041681381259574583, 'adjusted_mutual_information': -0.00030804572088447793}
{'silhoutte_coefficient': -0.0002415660842069361, 'calinski_harabasz_index': 1.046248933009503, 'davies_bouldin_index': 31.671008429850033}

Clearly all measures are better in agglomerative clustering than in random clustering thus ensuring that agglomerative clustering gives valid clusters.

Question 4

Part (A)

- k-means is an unsupervised clustering algorithm that divides the data into ‘k’ clusters, where each cluster is represented/defined by a prototype in terms of a ‘centroid’ (mean of the points in that cluster). However, it has the following limitations:

    - **Clustering outliers**
      Outliers are those datapoints in the dataset that deviate from the other majority data. These are abnormal, anomalous observations in the dataset whose value significantly differs from the data distribution. In k-means, the cluster center (centroid) is chosen by taking the mean of the cluster observations. An outlier tends to affect the mean value by pulling it in its direction thus, in the case of

k-means the centroids can be dragged towards the outliers. It may also be the case that these outliers points become the centroids in some cases and get a cluster of their own. This will influence the k-means algorithm.

- **Scaling with number of dimensions**
  High dimensional data is generally sparse and the distance metric used for k-means clustering loses (like euclidean distance, manhattan distance etc) significance in such high dimensions. The average distance between any two samples will increase in a high number of dimensions. In a high dimensional setting, distance-based measures of similarity between any two samples converges to a constant value as the number of dimensions rises.

Part (B)

K-medoids clustering

It is an unsupervised clustering algorithm that uses medoids as the cluster centers instead of taking the means for finding the centroid for the cluster. Medoids are those points in the cluster that have the minimum sum of dissimilarity with all the points in the cluster. They are considered a better representative point for the cluster as compared to the means because medoids are the actual data points in the clusters, whereas in k-means the centroid is not necessarily an actual data point. This also makes it more robust to outliers as well as noise, because the centroids being medoids and not means, they are not affected by outliers greatly as compared to effect of outliers on means(explained in part (A)).

Spectral Clustering

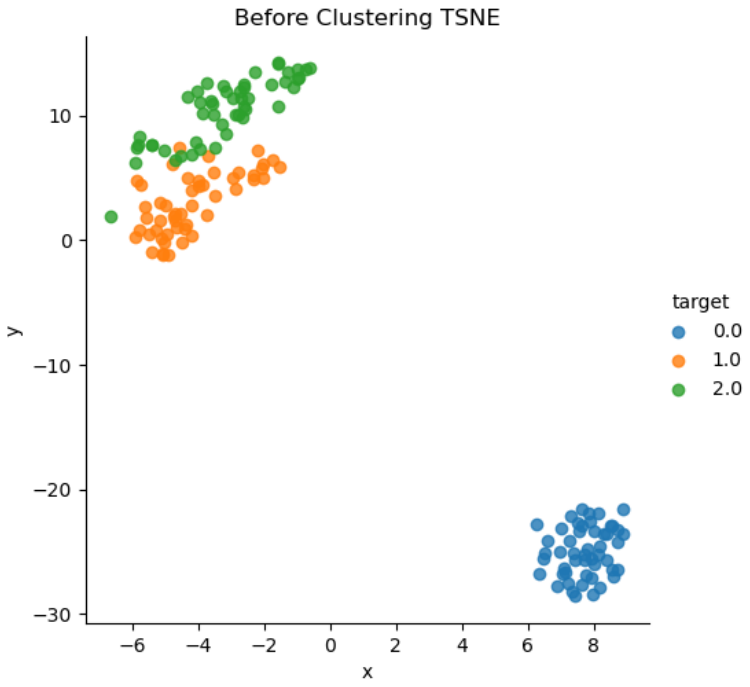
Spectral clustering is a clustering algorithm which uses concepts of graph theory to figure out clusters in the data. It is suitable for graph and non graph data. Spectral clustering comes under the umbrella of graph based clustering which are essentially used to deal with high dimensional data.

Spectral clustering uses eigenvalues (find number of clusters) and eigenvectors (find cluster labels) which are derived from the graph laplacian of the data. The problem associated with high dimensional data is the fact that the distance metrics such as euclidean distance becomes meaningless as the dimensions go on increasing. This is because as the dimensions increase the distance between two data points keeps increasing. This keeps happening until the closest and the farthest point from let's say a point A becomes similar such that the ratio between the distance to the closest and the farthest point from A is close to 1. Here spectral clustering makes clusters by using the number of shared neighbors in making the clusters.

Part (C) + (D)

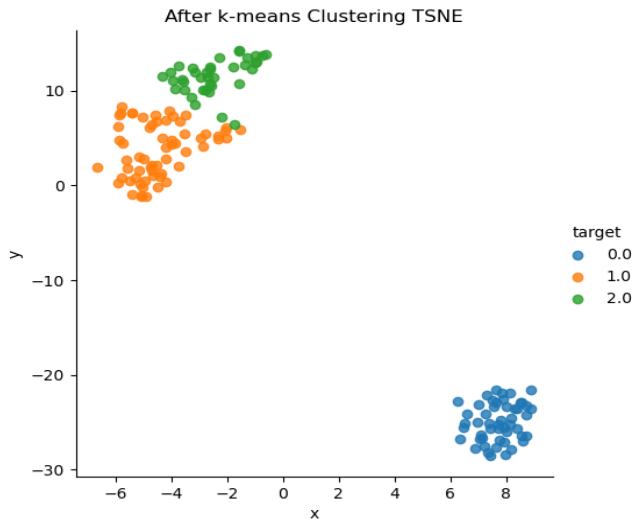
Iris Dataset

TSNE Before Clustering



K-Means

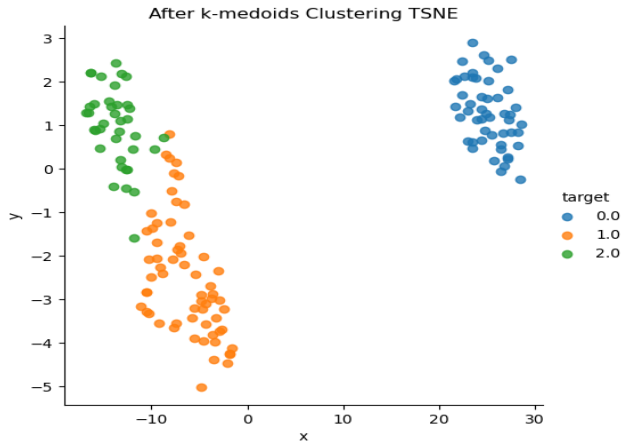
Best Hyperparameters : n_clusters = 3, init = ‘random’, n_init = 10, max_iter = 30



{'adjusted_rand_index': 0.7302382722834697, 'adjusted_mutual_information': 0.7551191675800483}
{'silhoutte_coefficient': 0.5528190123564095, 'calinski_harabasz_index': 561.62775662962, 'davies_bouldin_index': 0.6619715465007484}

K-Medoids

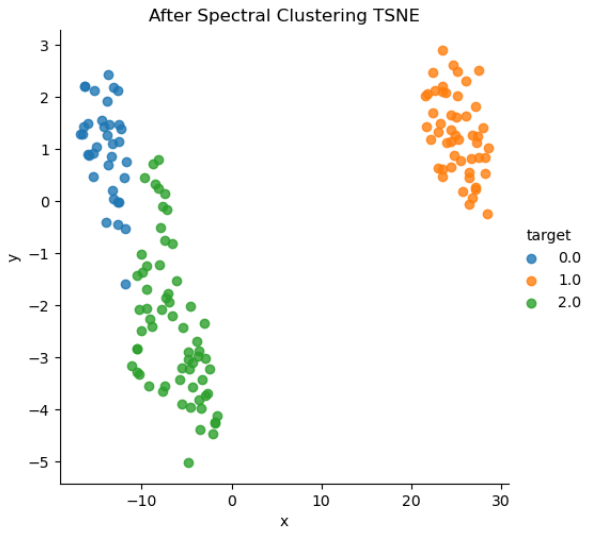
Best Hyperparameters : n_clusters = 3, metric = ‘euclidean’, init = ‘k-medoids++’, method = ‘pam’



{'adjusted_rand_index': 0.7302382722834697, 'adjusted_mutual_information': 0.7551191675800483}
{'silhoutte_coefficient': 0.5528190123564095, 'calinski_harabasz_index': 561.62775662962, 'davies_bouldin_index': 0.6619715465007484}

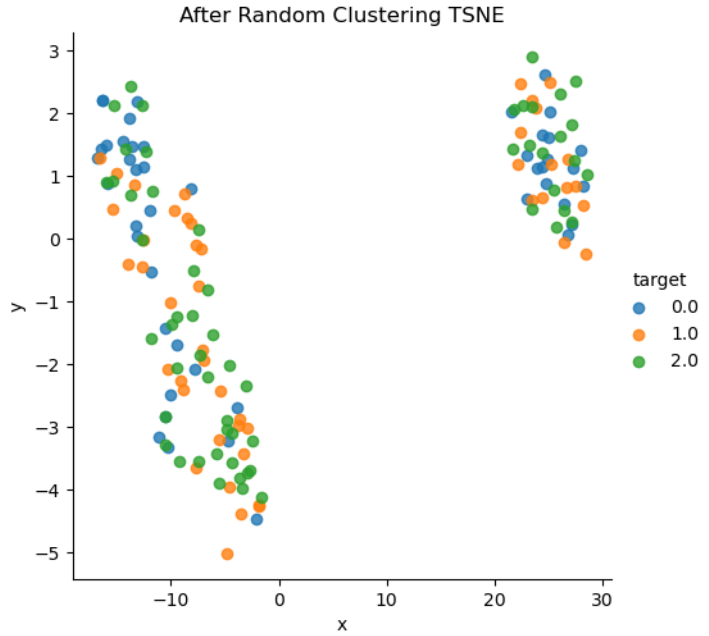
Spectral

Best Hyperparameters : n_clusters = 3, affinity = ‘nearest_neighbors’, n_components=3,
n_init = 10



{'adjusted_rand_index': 0.7591987071071522, 'adjusted_mutual_information': 0.8032287370935435}
{'silhoutte_coefficient': 0.5541608580282851, 'calinski_harabasz_index': 556.8795419179528, 'davies_bouldin_index': 0.6584442783224304}

Random Clustering for Iris Dataset



{'adjusted_rand_index': 0.04378690355509054, 'adjusted_mutual_information': 0.048434206502902026}
{'silhoutte_coefficient': -0.03856209479593506, 'calinski_harabasz_index': 0.5187659517665726, 'davies_bouldin_index': 35.39887442498115}

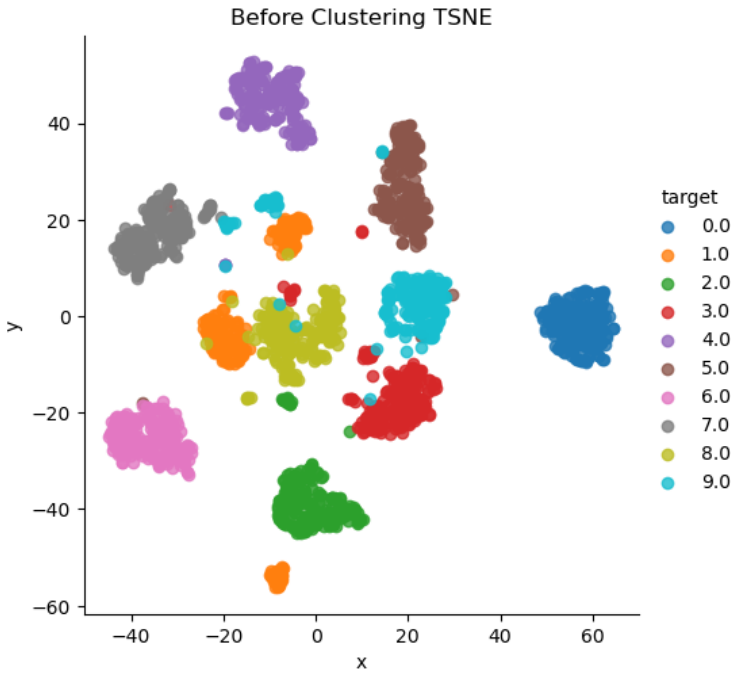
Comparison

Here, we can see the spectral clustering performs the best followed by k-medoids and k-means. Spectral clustering has higher extrinsic scores than both of them and has only slightly lesser intrinsic scores. K-means and k-medoids have similar intrinsic and extrinsic scores.

In general, all the algorithms perform well on iris dataset. This is also evident from the TSNE plots since they overlap quite well with true labels.

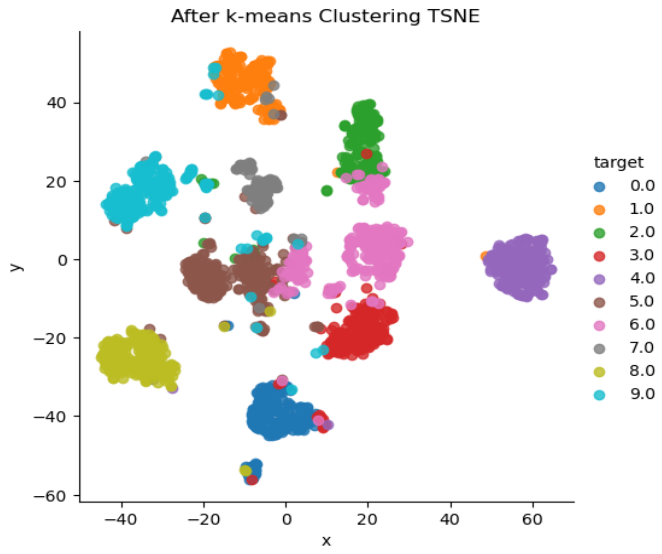
Digits Dataset

TSNE Before Clustering



K-Means

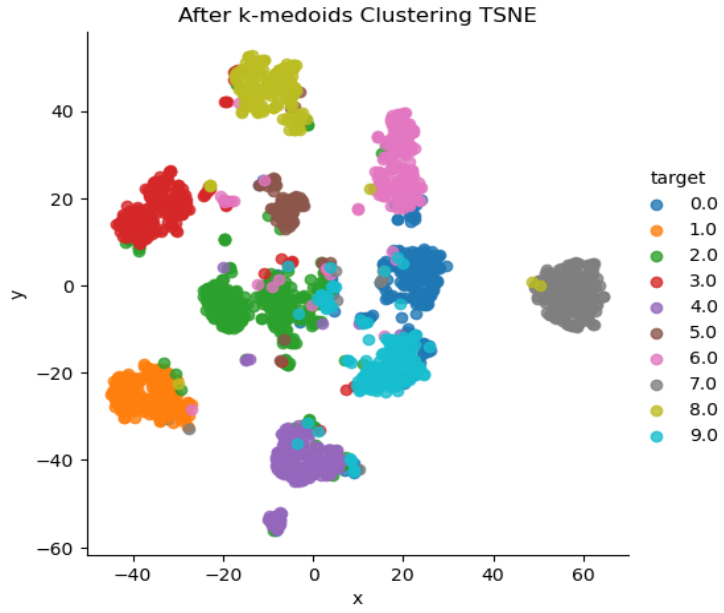
Best Hyperparameters : n_clusters = 10, init = ‘random’, n_init = 10, max_iter = 30



{'adjusted_rand_index': 0.6503471906636054, 'adjusted_mutual_information': 0.7187681191581738}
{'silhoutte_coefficient': 0.17364792838075688, 'calinski_harabasz_index': 163.1851613529295, 'davies_bouldin_index': 1.9809685214176604}

K-Medoids

Best Hyperparameters : n_clusters = 10, metric = ‘euclidean’, init = ‘k-medoids++’, method = ‘pam’

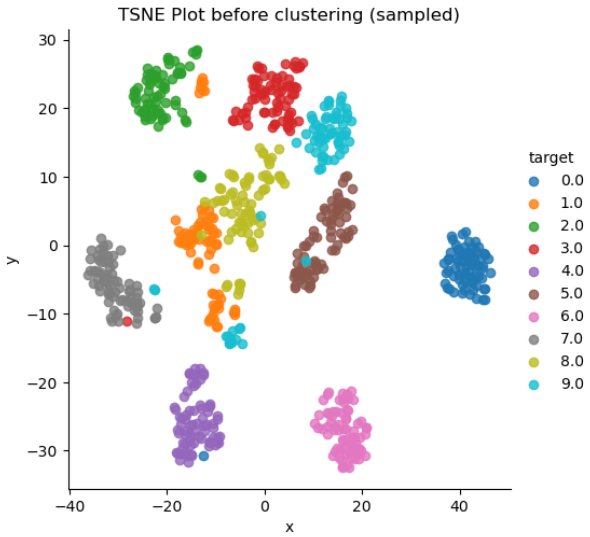


{'adjusted_rand_index': 0.6689567035719545, 'adjusted_mutual_information': 0.7413450081295662}

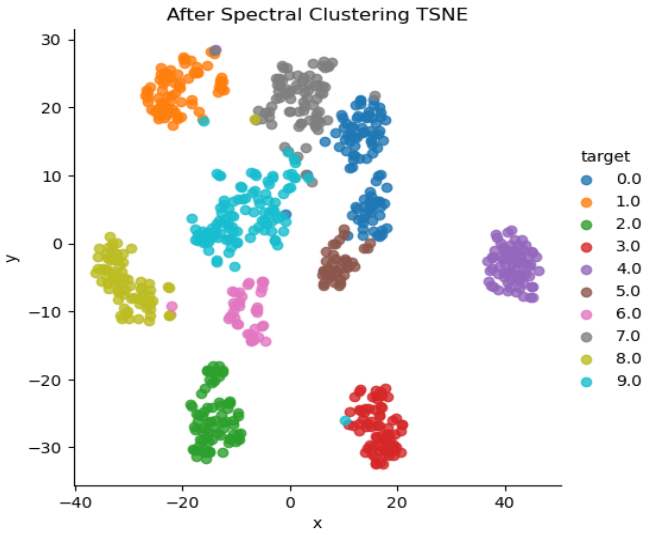
{'silhoutte_coefficient': 0.18242959371984133, 'calinski_harabasz_index': 169.3655418492578, 'davies_bouldin_index': 1.9223883446397978}

Spectral

- Note that for spectral clustering, we undersampled the dataset (719 samples) while keeping the distribution of classes in the subset the same (using stratify).
- The t-SNE plot for the subset is given below :

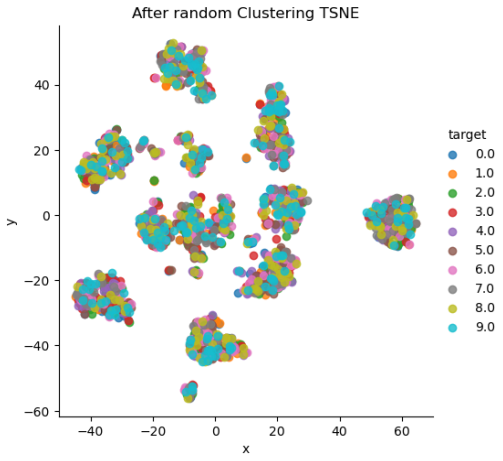


Best Hyperparameters: n_clusters = 10, affinity = 'nearest_neighbors', n_components=10, n_init = 10



{'adjusted_rand_index': 0.7337913275484396, 'adjusted_mutual_information': 0.8223242486341953}
{'silhoutte_coefficient': 0.17939546647882051, 'calinski_harabasz_index': 66.42548088914748, 'davies_bouldin_index': 1.939764314620943}

Random Clustering for Digits Dataset



{'adjusted_rand_index': -0.0004457871900047615, 'adjusted_mutual_information': -0.0010044616459200138}
{'silhoutte_coefficient': -0.018159166854004375, 'calinski_harabasz_index': 1.0944927830297015, 'davies_bouldin_index': 23.786742080402913}

In the digits dataset, we can clearly observe that random clustering is resulting in very poor t-sne plots with no clear unique clusters. Furthermore, the intrinsic and extrinsic measures too are very low in comparison to K-medoids, Spectral and K-means clustering algorithms results. (t-sne and metric values)

Comparison

In the digits dataset, spectral clustering performs the best as is expected. This is because digits dataset has 64 dimensions which is quite high. This can cause k-means to give poor clustering because the average distance between any two points is high in higher dimensions and the distances converge to a constant value (also explained in part A).

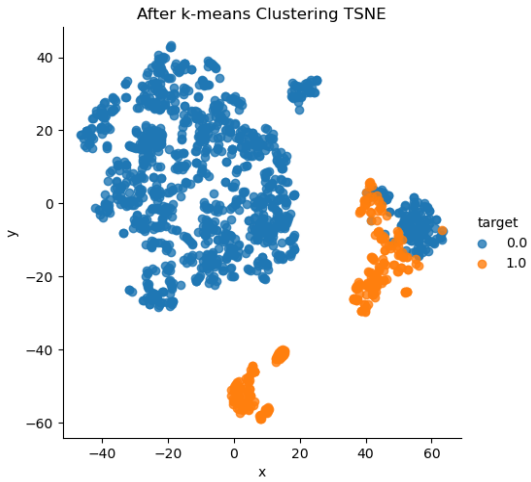
K-medoids also performs better than k-means. This is because there are a lot of outliers in the digits dataset which are not handled well by k-means since they can cause a lot of shift of k-means centers towards the outliers (also explained in part A).

Credit Card Fraud Detection Dataset

TSNE Before Clustering

K-Means

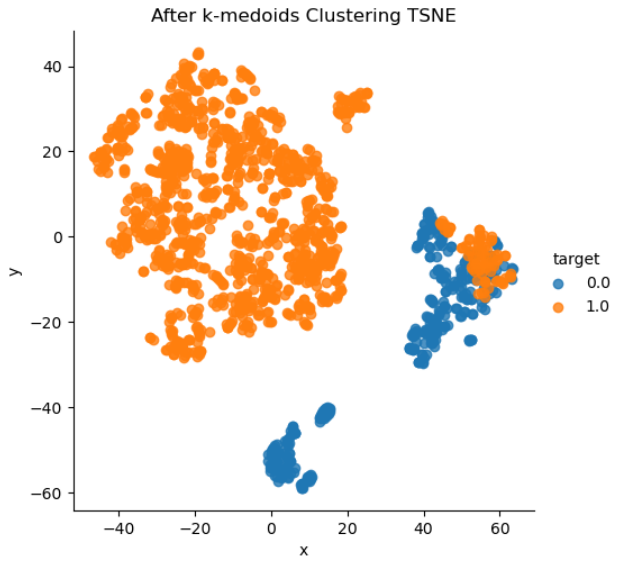
Best Hyperparameters: n_clusters = 2, init = ‘random’, n_init = 10, max_iter = 10



{'adjusted_rand_index': 0.46184172334067847, 'adjusted_mutual_information': 0.4389289923957658}
{'silhoutte_coefficient': 0.573864099656229, 'calinski_harabasz_index': 1590.1166434988945, 'davies_bouldin_index': 0.8484985732078886}

K-Medoids

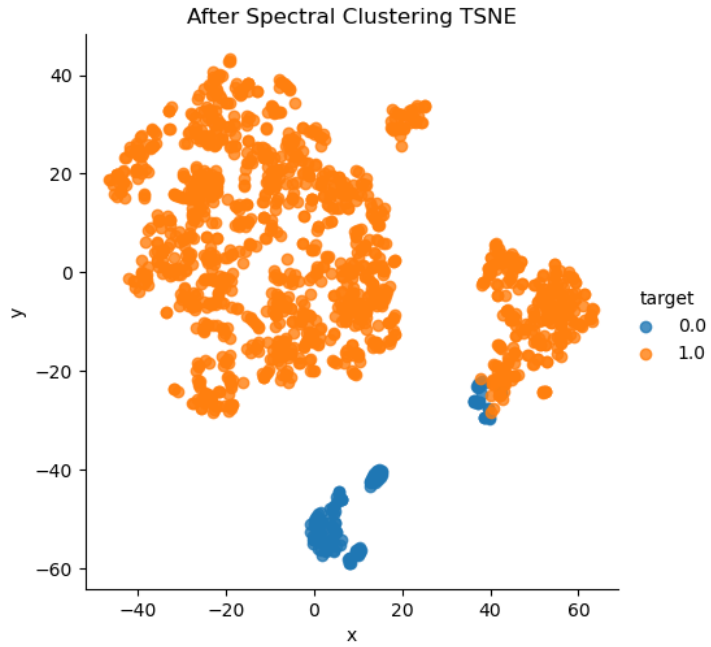
Best Hyperparameters: n_clusters = 2, metric = ‘euclidean’, init = ‘k-medoids++’, method = ‘pam’



{'adjusted_rand_index': 0.5625420367650639, 'adjusted_mutual_information': 0.5199368443874973}
{'silhoutte_coefficient': 0.5482828908906328, 'calinski_harabasz_index': 1522.5571839307056, 'davies_bouldin_index': 0.9020260323490058}

Spectral

Best Hyperparameters: n_clusters = 2, affinity = rbf, n_components=2, n_init = 10



{'adjusted_rand_index': 0.20515699100063214, 'adjusted_mutual_information': 0.22927479937184442}

{'silhoutte_coefficient': 0.6269726332300141, 'calinski_harabasz_index': 1301.819013374994, 'davies_bouldin_index': 0.669409991267859}

Random Clustering for Credit Card Fraud Detection

{'adjusted_rand_index': -0.0005773358865637621, 'adjusted_mutual_information': -0.0005057243226267288}
{'silhoutte_coefficient': 0.0003295400501856345, 'calinski_harabasz_index': 0.7563464752279336, 'davies_bouldin_index': 37.25114570550437}

Comparison

K-medoids performs the best in this case. This is because there are many outliers here as is visible clearly in the t-SNE plot of the data. Being robust to these outliers, the k-medoids algorithm performs better over here. K-means algorithm performs a bit poorer as compared to K-medoids due to the same reason.
Spectral clustering in this case performs better when we consider the intrinsic measures. This can be interpreted as the number of dimensions (28) is considerably high for simple k-means algorithm.

Part (E)

As we can see in the above parts, the prototype based algorithms perform best on the digits dataset. This performance can be seen in some cases by comparing the metrics but in other cases, we also rely on the clustering produced. In terms, of t-SNE plots, we can also see that in all the three algorithms (k-means, k-medoids, spectral) the clusters produced are globular and have a flat-geometry. Each cluster, which here represents a digit, is represented by a cluster prototype around which other similar data points are grouped into a single cluster. These clusters are also also well-separated and evenly sized making it appropriate use-ase for prototype based clustering algorithm. We can also see that, the high dimensionality of the dataset (64 dimensions) are handled pretty well when we make use of spectral clustering

method. Also, there are some outliers in the dataset which are being handled using the k-medoids algorithm and it is evident from the empirical as well as the visual results that the algorithm remains robust to these outliers in the dataset.

Part (F)

To compare our clusterings with random data, we assign random clusters to each data point in the dataset. We then compute all the measures for this random assignment of clusters. Then, we compare the results obtained for the random assignment of clusters with those obtained using specific algorithms.

Doing this ensures that our algorithm gives some defined clusters and not just random ones.

Credit-Card Fraud Detection Dataset

Random Clustering:-
{'adjusted_rand_index': -0.0005773358865637621, 'adjusted_mutual_information': -0.000505724322626728 8}
{'silhoutte_coefficient': 0.0003295400501856345, 'calinski_harabasz_index': 0.7563464752279336, 'davies_bouldin_index': 37.25114570550437}

K-means clustering
{'adjusted_rand_index': 0.46184172334067847, 'adjusted_mutual_information': 0.4389289923957658}
{'silhoutte_coefficient': 0.573864099656229, 'calinski_harabasz_index': 1590.1166434988945, 'davies_bouldin_index': 0.8484985732078886}

K-medoids clustering
{'adjusted_rand_index': 0.5625420367650639, 'adjusted_mutual_information': 0.5199368443874973}
{'silhoutte_coefficient': 0.5482828908906328, 'calinski_harabasz_index': 1522.5571839307056, 'davies_bouldin_index': 0.9020260323490058}

Spectral clustering
{'adjusted_rand_index': 0.20515699100063214, 'adjusted_mutual_information': 0.22927479937184442}
{'silhoutte_coefficient': 0.6269726332300141, 'calinski_harabasz_index': 1301.819013374994, 'davies_bouldin_index': 0.669409991267859}

In the credit-card fraud dataset, we can clearly observe that random clustering is resulting in very poor t-sne plots with no clear unique clusters. Furthermore, the intrinsic and extrinsic measures too are very low in comparison to K-medoids, Spectral and K-means clustering algorithms results. (t-sne and metric values).

Digits Dataset

Random clustering
{'adjusted_rand_index': -0.0004457871900047615, 'adjusted_mutual_information': -0.0010044616459200138}
{'silhoutte_coefficient': -0.018159166854004375, 'calinski_harabasz_index': 1.0944927830297015, 'davies_bouldin_index': 23.786742080402913}

K-means clustering
{'adjusted_rand_index': 0.6503471906636054, 'adjusted_mutual_information': 0.7187681191581738}
{'silhoutte_coefficient': 0.17364792838075688, 'calinski_harabasz_index': 163.1851613529295, 'davies_bouldin_index': 1.9809685214176604}

K-medoids clustering
{'adjusted_rand_index': 0.6689567035719545, 'adjusted_mutual_information': 0.7413450081295662}
{'silhoutte_coefficient': 0.18242959371984133, 'calinski_harabasz_index': 169.3655418492578, 'davies_bouldin_index': 1.9223883446397978}

Spectral clustering
{'adjusted_rand_index': 0.7337913275484396, 'adjusted_mutual_information': 0.8223242486341953}
{'silhoutte_coefficient': 0.17939546647882051, 'calinski_harabasz_index': 66.42548088914748, 'davies_bouldin_index': 1.939764314620943}

In the digits dataset, we can clearly observe that random clustering is resulting in very poor t-sne plots with no clear unique clusters. Furthermore, the intrinsic and extrinsic measures too are very low in comparison to K-medoids, Spectral and K-means clustering algorithms results. (t-sne and metric values)

Iris Dataset

K-means clustering
{'adjusted_rand_index': 0.7302382722834697, 'adjusted_mutual_information': 0.7551191675800483}

{'silhoutte_coefficient': 0.5528190123564095, 'calinski_harabasz_index': 561.62775662962, 'davies_bouldin_index': 0.6619715465007484}

K-medoids clustering
{'adjusted_rand_index': 0.7302382722834697, 'adjusted_mutual_information': 0.7551191675800483}
{'silhoutte_coefficient': 0.5528190123564095, 'calinski_harabasz_index': 561.62775662962, 'davies_bouldin_index': 0.6619715465007484}

Spectral clustering
{'adjusted_rand_index': 0.7591987071071522, 'adjusted_mutual_information': 0.8032287370935435}
{'silhoutte_coefficient': 0.5541608580282851, 'calinski_harabasz_index': 556.8795419179528, 'davies_bouldin_index': 0.6584442783224304}

Random Clustering
{'adjusted_rand_index': 0.04378690355509054, 'adjusted_mutual_information': 0.048434206502902026}
{'silhoutte_coefficient': -0.03856209479593506, 'calinski_harabasz_index': 0.5187659517665726, 'davies_bouldin_index': 35.39887442498115}

In the iris dataset, we can clearly observe that random clustering is resulting in very poor t-sne plots with no clear unique clusters. Furthermore, the intrinsic and extrinsic measures too are very low in comparison to K-medoids, Spectral and K-means clustering algorithms results. (t-sne and metric values)

Learnings:

- Learned visualization techniques like t-SNE plotting, scatterplots etc.
- Learned about various clustering algorithms like density based clustering, hierarchical clustering, k-prototpyes clustering.
- Learned about evaluation metrics like silhoutte score, calinski_harabasz_index, mutual information etc.
- Learned about techniques like hyperparameter tuning