CS584 HOMEWORK 4

Name: Sreeram Bangaru

Minername: sreerambangaru

G#: G01406291

IRIS_partA_rank: 56 V-measure: 0.95

IMAGE_partB_rank: 9 V-measure: 0.86

Given problem statement is to apply the k-means clustering algorithm on unsupervised data and use metrics to check homogeneity and completeness of clusters and how well the kmeans algorithm can optimize the clusters. We have used the kmeans algorithm on two datasets namely the Iris data set and another an image dataset where each vector is a collection of pixels as datapoints.

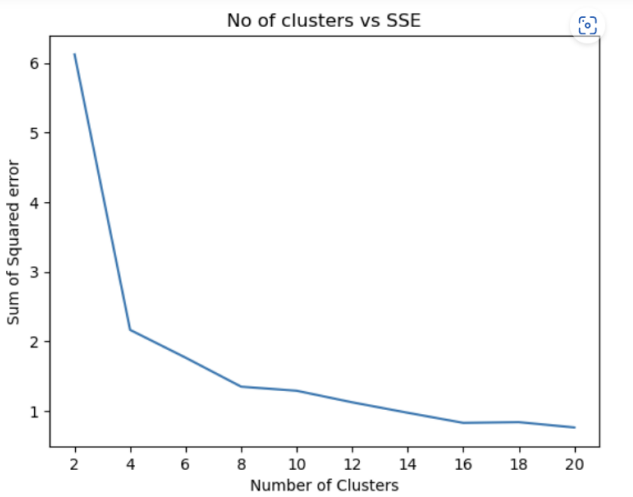
Part A Iris Dataset:

By looking at the dataset we can see that there is a little variance between the datapoints so we have used scaling techniques such as MinMaxScaler, RobustScaler and MaxAbsScaler before using the kmeans algorithm. Also, to check the homogeneity of clusters we have used v-measure output given by miner for cluster count 3 and have also considered metric sum of squared error to check the distance between the cluster centroids and the datapoints. We have used the internal metrics on clusters by generating centroids using kmeans where all the centroids are generated at random. The results are tabulated below where cluster count is 3 and we have used two different distance measures to see how close each datapoint is to the finalized cluster centroids:

K-MEANS		
Scaling Method, Distance metric	Sum of Squared Error	V-measure from miner
No Scaler with Euclidean distance	145.27	0.69
MinMaxScaler with Euclidean distance	6.99	0.68
RobustScaler with Euclidean distance	73.5	0.61
MaxAbsScaler with Euclidean distance	2.83	0.86
MaxAbsScaler with Cosine Similarity	6.06	0.81
No Scaler with Cosine Similarity	78.9	0.95

From the above data we can see that we obtained very good v-measure for MaxAbsScaler with Euclidean distance. So, from this we can say that by considering the absolute values of the datapoints we were able to result in better clustering performance. But we cannot completely depend on this scaling technique as it is sensitive to outliers or extreme values. Also, we can see that cosine similarity is giving great results of **0.95** from which we can infer that the data is already normalized or centered. Adding to this cosine similarity measures the angle between two vectors, rather than the magnitude of the vectors themselves. So now we can justify that choosing cosine similarity as the distance measure when used with kmeans give better results of clustering.

A graphical representation is plotted below where the X-axis shows the number of clusters and the Y-axis show the sum of squared error for each cluster count when using kmeans.



In the above graph we can see an elbow point at 3-4 clusters which shows that this is one optimal cluster count as from that point further there is a less significant decrease in the sum of squared error.

Part B Image dataset:

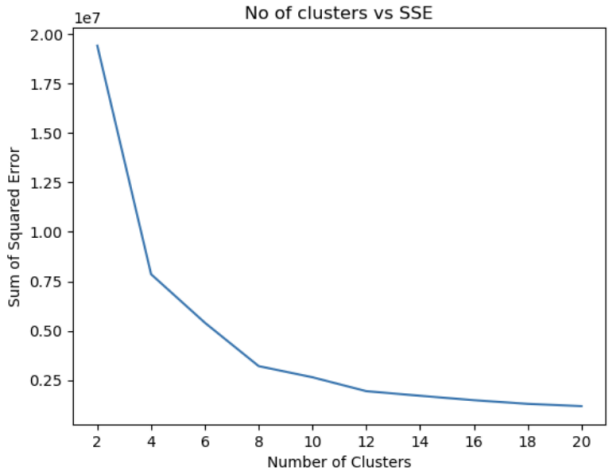
By looking at the dataset, we can say that the it has many features relative to the number of observations. So, it is important for use of dimensionality reduction techniques to do feature reduction to increase the model performance as large datasets with high dimensionality can be computationally expensive to perform analysis. For checking which dimensionality reduction technique gives more optimal clustering solution. We are considering scaling with dimensionality reduction when using distance measure as Euclidean distance and cosine similarity. Also, we are taking max iterations as 300 but break out of the iterations whenever the new centroid and the old centroid are generated at the same datapoint.

First, we are considering the dimensionality reduction technique t-SNE. So, while using this technique it is important to tune the hyperparameters carefully. t-SNE is a non-linear technique that can capture complex relationships between variables and preserve the local structure of the data, making it well suited for datasets having non-linear structure. The hyperparameters we are considering are n_components and perplexity.

KMEANS USING t-SNE					
n_components	Perplexity	Scaling	distance measure	SSE	V-measure
2	30.0	No scaling	Cosine similarity	4.17	0.62
2	30.0	MaxAbs	Cosine similarity	4.18	0.63
3	30.0	No scaling	Cosine similarity	91.5	0.71
3	50.0	MaxAbs	Euclidean distance	411545	0.75
2	30.0	MaxAbs	Euclidean distance	242629	0.76

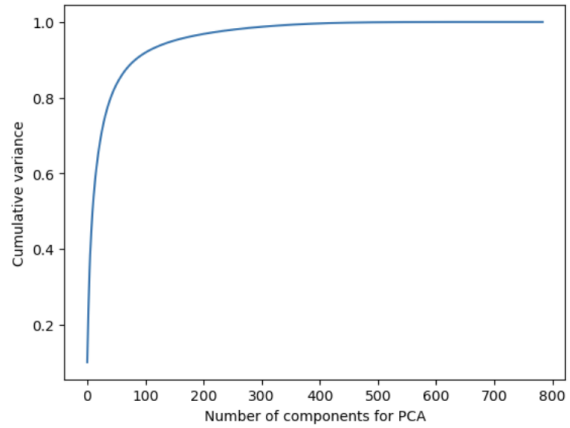
From the tabular results we can say that t-SNE is able to capture the underlying structure of the data and this complex data is giving more better results for non-linear transformation. Also, from the below graph

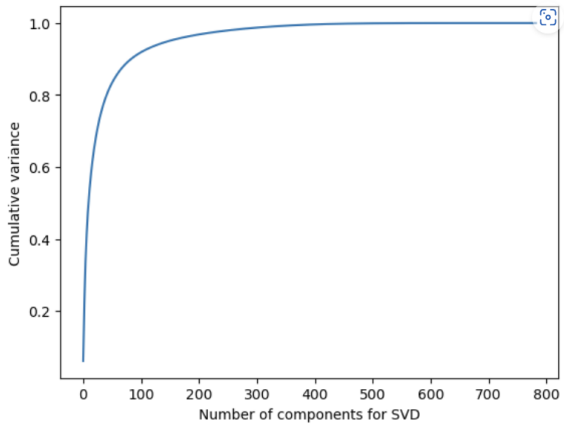
we can see that there as a drastic change in sum of squared error around 8 clusters where the optimal solution might lie as further from that point there is less significant decrease in sum of squared error till 20 clusters.



Perhaps we might get better clustering results by transforming the data using linear transformation techniques like **Principal Component Analysis (PCA) or Singular Value Decomposition (SVD)** and then follow up with non-linear transformation technique such as **UMAP** as this can potentially capture both global and local structure of the data.

For our dataset to find out the correct number of feature reduction when using PCA and SVD we plot a graph between the number of components and the explained variance ratio to get the optimal number of components where we have an elbow curve. Explained variance ratio is basically a measure that indicates the proportion of the total variance in the data that is explained by each principal component obtained through dimensionality reduction techniques. It is the ratio for each principal component's variance captured to the total variance of the data. Before plotting the graph, we scale the features using MaxAbs Scaling.



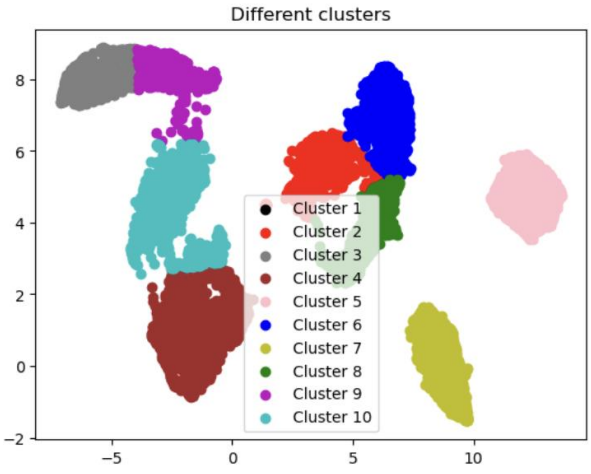


By looking at the above graphs we can say that the elbow curve is around 50 to 100 and the optimal feature reduction value might lie in this range. Hence, we do feature reduction on scaled data by trial and error from the above range and then applying UMAP where we reduce to 2 components followed by the kmeans

algorithm to find out the sum of squared error and v-measure. The below table represents the metrics for cluster count 10,

KMEANS USING PCA, SVD, UMAP			
Dimensionality reduction technique	Scaling method, Distance metric	SSE	V-measure
PCA (n=50), UMAP(n=2)	Standard scaling, Euclidean	957.18	0.75
PCA(n=100), UMAP(n=2)	Standard scaling, Euclidean	1233.89	0.83
SVD(n=100) UMAP(n=2)	Standard scaling, Euclidean	1221.73	0.80
SVD(n=50) UMAP(n=2)	Standard scaling, Euclidean	675.64	0.86

From the above V-measure values we can see that we have achieved a very good value of 0.86 with SVD where we reduced the components to 50 followed by UMAP where we did feature reduction to 2 components and then using standard scaler for scaling and Euclidean distance as the distance measure. Hence, we plot the graphs for the best result. The below graphs are a representation of the different clusters formed by using the kmeans algorithm and the plot showing the relation between the cluster count and its sum of squared error value.



To summarize our results, we can say that the optimal cluster count value is around 8 clusters which can be said by looking at the elbow curve in the graph and the linear dimensionality reduction technique is able to capture the overall structure and main patterns in the data, while the non-linear dimensionality reduction technique is capturing more complex relationships and the structure that may not be capture alone by the linear techniques due to which using this combination gave a better result than just using either one of the techniques.