Iris and Image Clustering

Name: Yaswanth Reddy Manukonda
Email: ymanukon@gmu.edu
Miner2 Id: ymanukon
Mason Id: ymanukon

Score:

Part 1	0.90
Part 2	0.67

Rank:

Part 1	61
Part 2	91

Problem Statement: develop a K means algorithm that clusters the given Iris data and Image data.

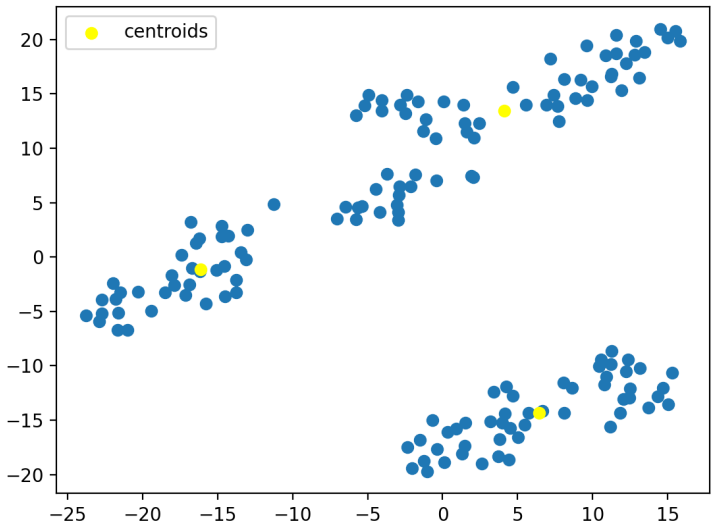
Solution:

1. Load the given data and do the pre-processing like removing the null values in the data
2. Normalize the data so that the whole data range from 0 to 1
3. Reduce the dimensions using TSNE (t-distributed stochastic neighbor embedding) or PCA (Principal component analysis)
4. Choose some points as random centroids from the given dataset
5. Identify which points in the actual dataset is closest to which centroid
6. Now that we have figured out which data point is associated with which centroid. Now for each centroid we calculate the average of all these associated points and move the centroid to that average point
7. As K means is unsupervised learning, we don't exactly know the number of clusters needed. So, we go for elbow analysis or silhouette score and determine the k value.

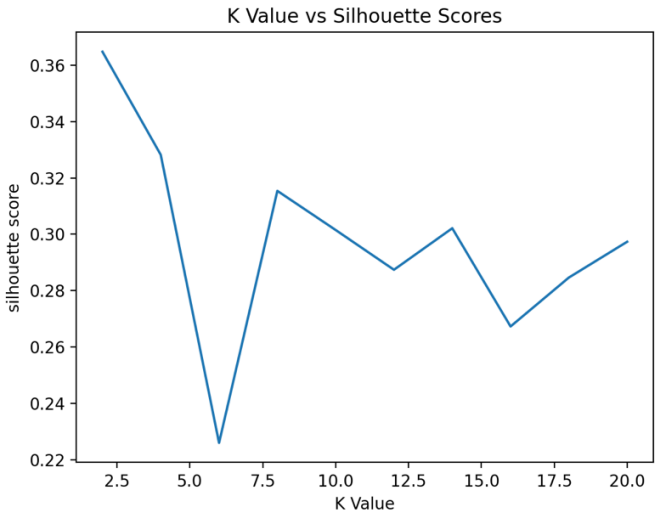
Explanation:

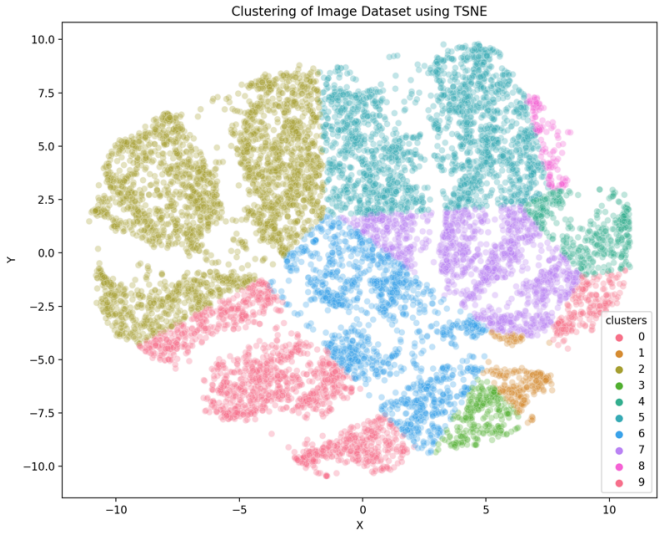
- For loading the data, I have used read_csv function from pandas library, after reading the files I have checked for the null values in the data set and removed them.
- I have normalized the dataset using the custom function ***normalizeData()*** which converts the given data into the range of 0 to 1
- As there are 784 dimensions in the Image data, I have applied **dimension reduction technique** TSNE and PCA and found that TSNE has better performance, accuracy so decided to move ahead by using TSNE. Applying the same technique for Iris data also helped me increase the score
- This TSNE calculates the probability of similarity of points in high dimension and then calculates probability of similarity in the corresponding low dimensions. It takes the below parameters
    - ***N_components:*** it's the dimensions number.
    - ***Perplexity:*** related number of neighbors used.
    - ***N_tier:*** number of iterations for optimization.
- **Iris Data:** The code for this is present in the file ***HW3_Part1.py***
    - I have chosen 3 random centroids out of the normalized and preprocessed data

- Using these 3 points I have clustered the given data by calculating the sum of squared distance between each point and the centroids. The implementation for this can be found in the function ***assignClusters()***
  - ***data:*** this is the pre-prpocessed data
  - ***centroids:*** this is the list of previous centroids
- Now, I have calculated the average of all the points in their respective clusters and made these average points as the new centroids. Implementation can be found in the function ***updateCentroids()***. This function takes 3 parameters and returns the updated centroids
  - ***data:*** this is the pre-prpocessed data
  - ***centroids:*** this is the list of previous centroids
  - ***assignedClusters:*** the array of the cluster numbers for all the data points
- the above process is iterated for 100 times and got best results of clustering
- I have plotted the cluster visualization with centroids using scatter plot. Below is the plot I obtained for the given data

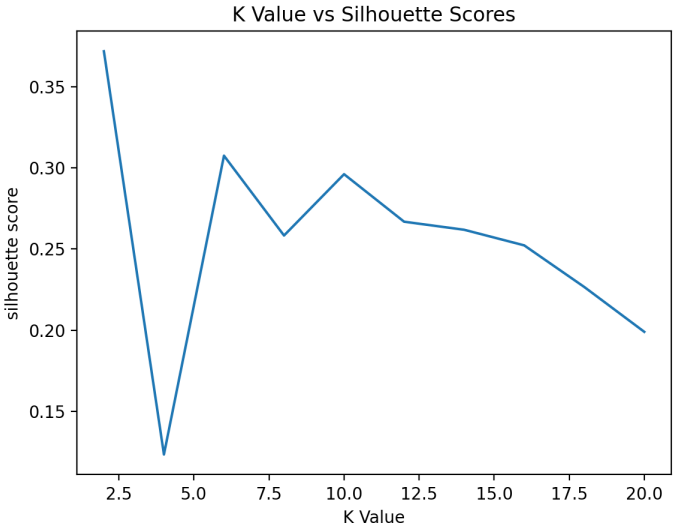


- **<u>Image Data</u>:** for this dataset I have used both PCA and TSNE as dimension reduction techniques. However, like the iris data TSNE gave me the best results
  - ***iterations():*** this function takes the data and k value as parameters and returns the new clusters. If the centroids are changed this whole process of assigning clusters and recomputing the centroids based on the mean value of the clusters continues till the centroids doesn't change or the silhouette score is less than 0.25 where max iterations are 100
  - ***assignClusters():*** is the function used to assign the cluster number for all the given input datapoints
  - ***updateCentroids():*** is the function used to change the centroids upon calculating the mean of the sum of squares of the distance between the points in that cluster
  - ***visualize():*** is the function used to plot the clusters from 0 to 9 as the image data is represented from 0 to 9. It also has code for the line plot between K value ranging from 2 to 20 and the silhouette score.





- The same plots are made with the PCA as dimensionality reduction technique and is clear that using TSNE has high silhouette score



References:

- Sklearn.metrics.silhouette_score. scikit. (n.d.). Retrieved October 13, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.
- Sklearn.manifold.TSNE. scikit. (n.d.). Retrieved October 13, 2021, from https://scikitlearn.org/stable/modules/generated/sklearn.manifold.TSNE.html.
- Bhardwaj, A. (2020, May 27). Silhouette coefficient: Validating clustering techniques. Medium. Retrieved October 13, 2021, from https://towardsdatascience.com/silhouettecoefficient-validating-clustering-techniques-e976bb81d10c.
- 3.3. metrics and scoring: Quantifying the quality of predictions. scikit. (n.d.). Retrieved October 13, 2021, from https://scikit-learn.org/stable/modules/model_evaluation.html.