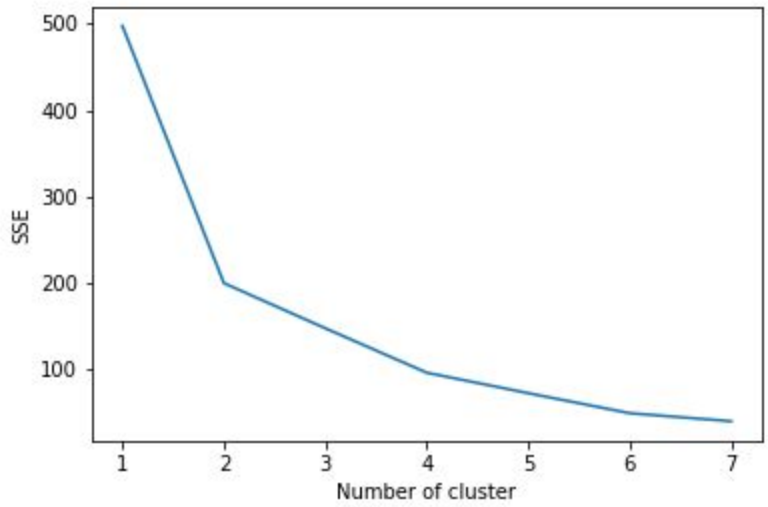
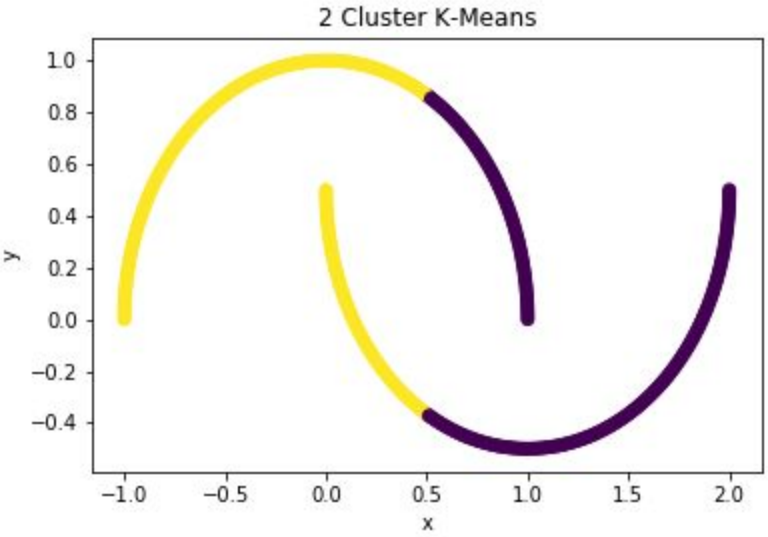
Que-3 (Clustering)

a) K-means clustering on dataset 1

- Firstly I find the optimal k, at which squared sum error is minimum
- Graph between sse and k we get for dataset 1 is

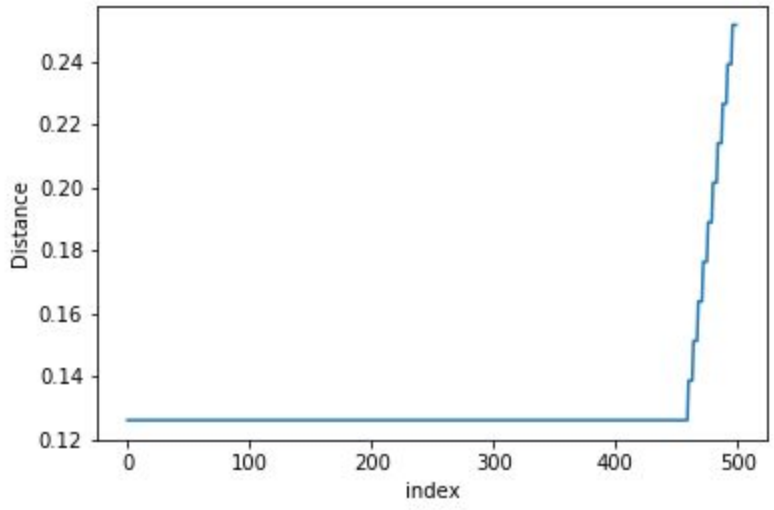


- Clearly from the above graph we can see that at k=2 a sharp turn is there so, 2 is the optimal value for k
- Now when we put n_component=2 in Kmeans function then the cluster formation for dataset 1 will look like

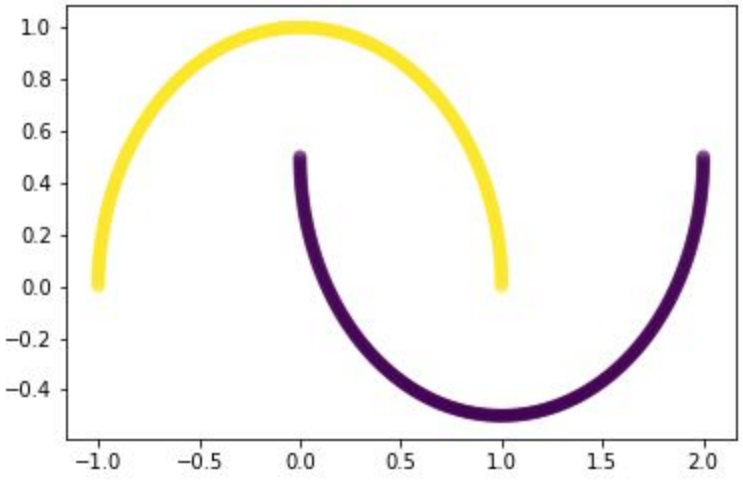


b) Results for my own DBSCAN implementation

- Dataset 1 exploration
- Graph between index and distance



- So from the above graph we can see that the stip curve is at distance 0.13 So we select epsilon=0.13 and minimum_points=20
- We have taken the 20th column of distance matrix so min_points=20
- Clusters form by DBSCAN look like



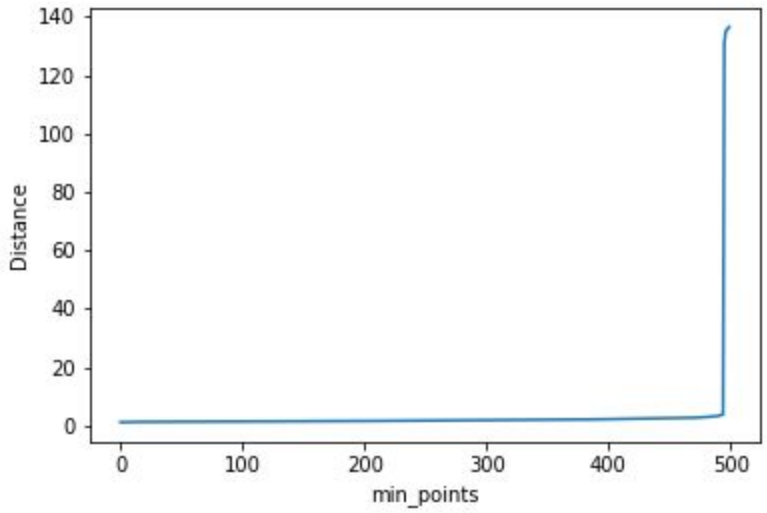
- X axis represent the x coordinate
- Y axis represent the y coordinate

c) Differences that I observed for K-means and DBSCAN from the above graphs are :-

- Density clustering algorithms use the concept of reachability i.e. how many neighbors has a point within a radius whereas K-means see the distance from a central clustering point
- In K-means final clusters depends on initial selection of mean point whereas in DBSCAN for a fixed epsilon and minimum number of points within a radius the final clusters will be same
- For this dataset DBSCAN is performing better than K-means, as you can see the above cluster plots it may be because as we know DBSCAN works well when the clusters in the dataset have similar densities

d) Dataset 2

- Data exploration
- Graph between index and distance for dataset 2

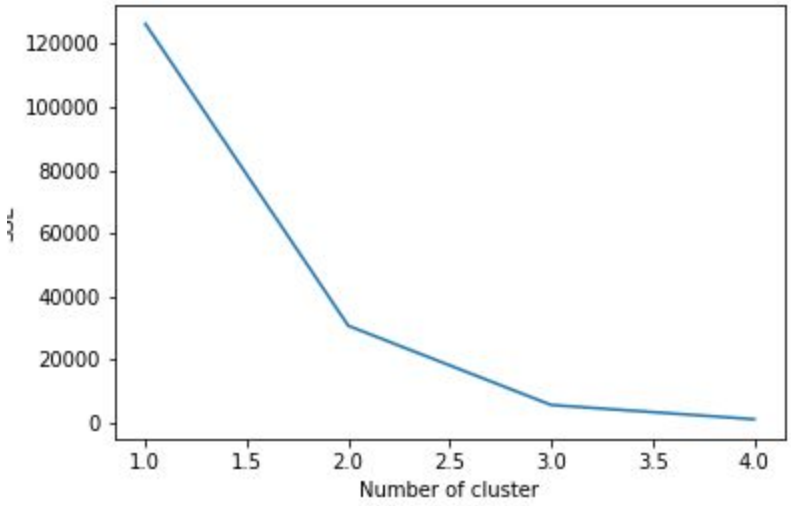


- So from the above graph we can see that the stip curve is at distance 4 So we select epsilon=10 and minimum_points=100
- We have taken the 100th column of distance matrix so min_points=100

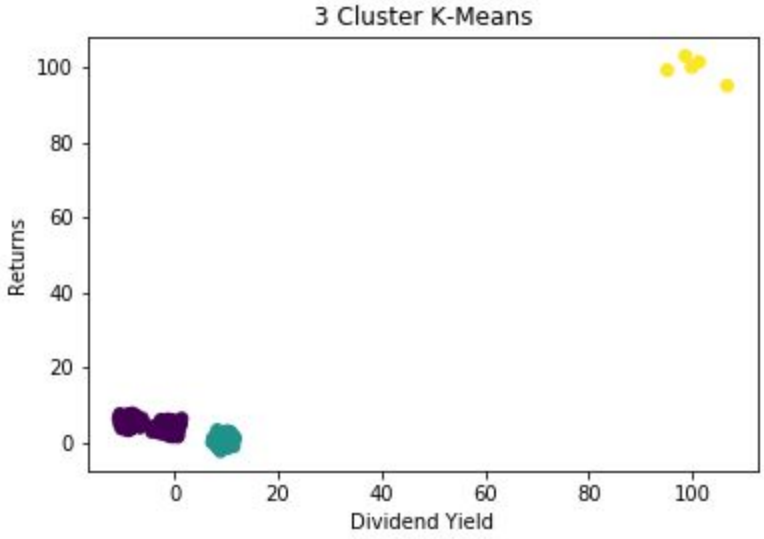
- If we apply k-means and DBSCAN on dataset 2 then the clusters formed will look like

K-Means

- Optimal k for dataset 2 will be 3, as shown in the figure



- Now when we put n_component=3 in Kmeans function then the cluster formation for dataset 2 will look like



Pros of K-means over DBSCAN -

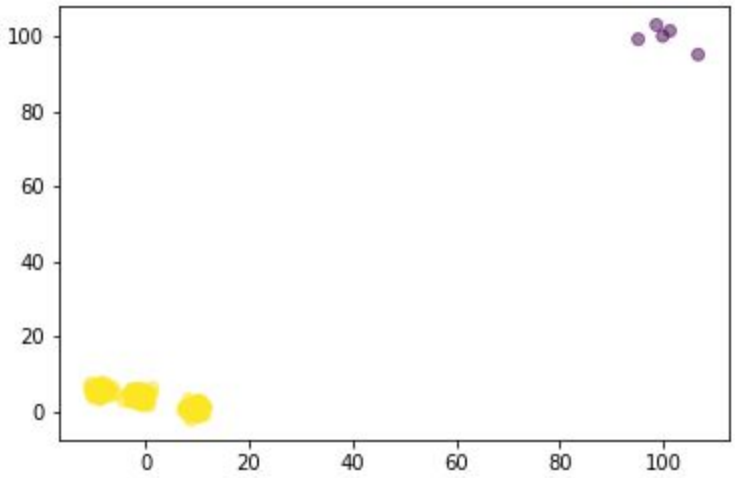
- As in the question itself number of cluster in the dataset is specified, so as we know that when number of clusters are already known to us then K-means works well and hence the above graph is the proof in which K-means can clearly identifying the three clusters of the given dataset
- KMeans is much faster than DBScan

Cons of K-means -

- It needs numbers of clusters that is hidden in the dataset
- To select initial points
- K-means clustering gives varying results on different runs of an algorithm and depends on initial value of mean point chosen

DBSCAN

- For dataset 2, DBSCAN will give cluster like



Pros of DBSCAN over K-means -

- DBScan doesn't need number of clusters

- Consistency- As the algorithm for a given epsilon and minimum number of points within a given radius gives same result irrespective of initial point selection

Cons of DBSCAN -

- DBScan doesn't work well over clusters with different densities
- DBScan needs a careful selection of its parameters

Conclusion-

- Clearly for this dataset k-means is giving good result as compared to DBSCAN because number of clusters is specified in the question and also density of clusters varies so much that is why may be DBSCAN is not giving good results