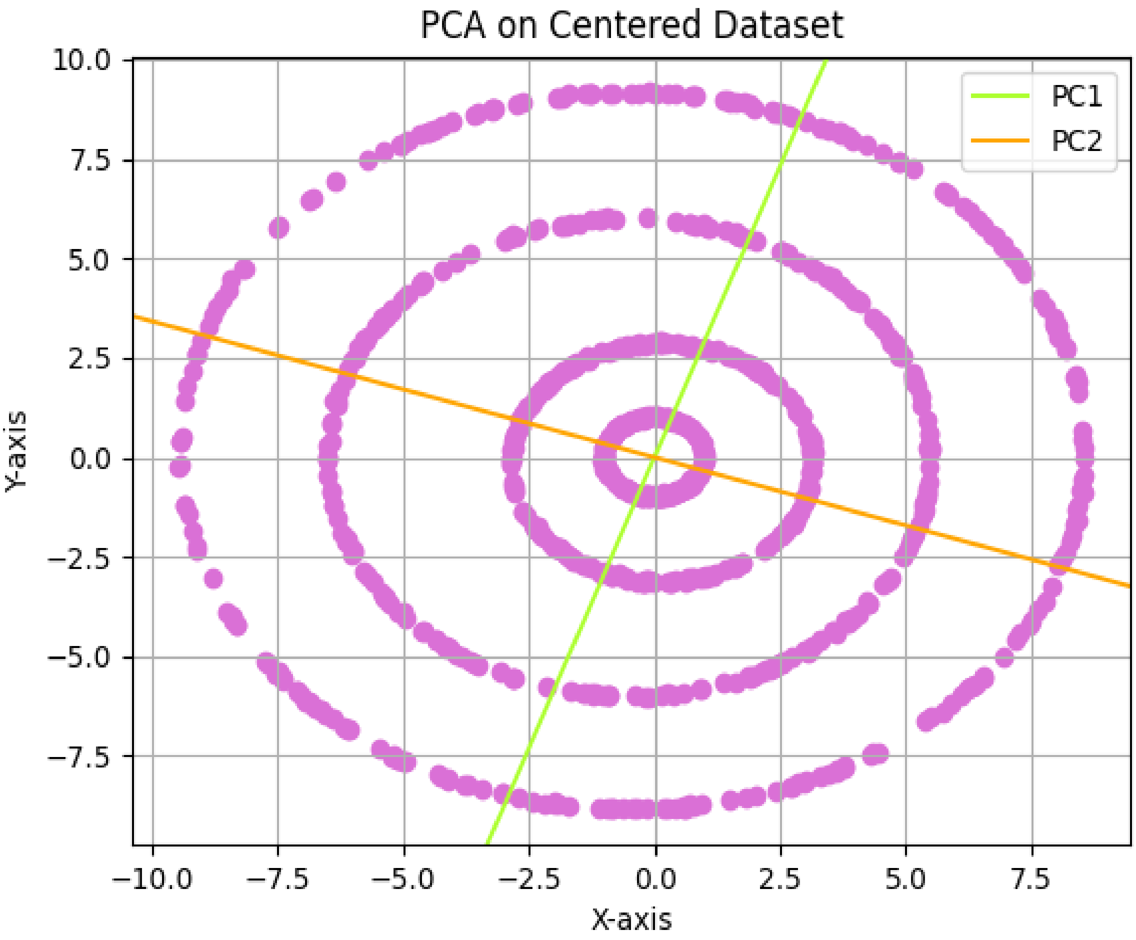
Q1.(i)

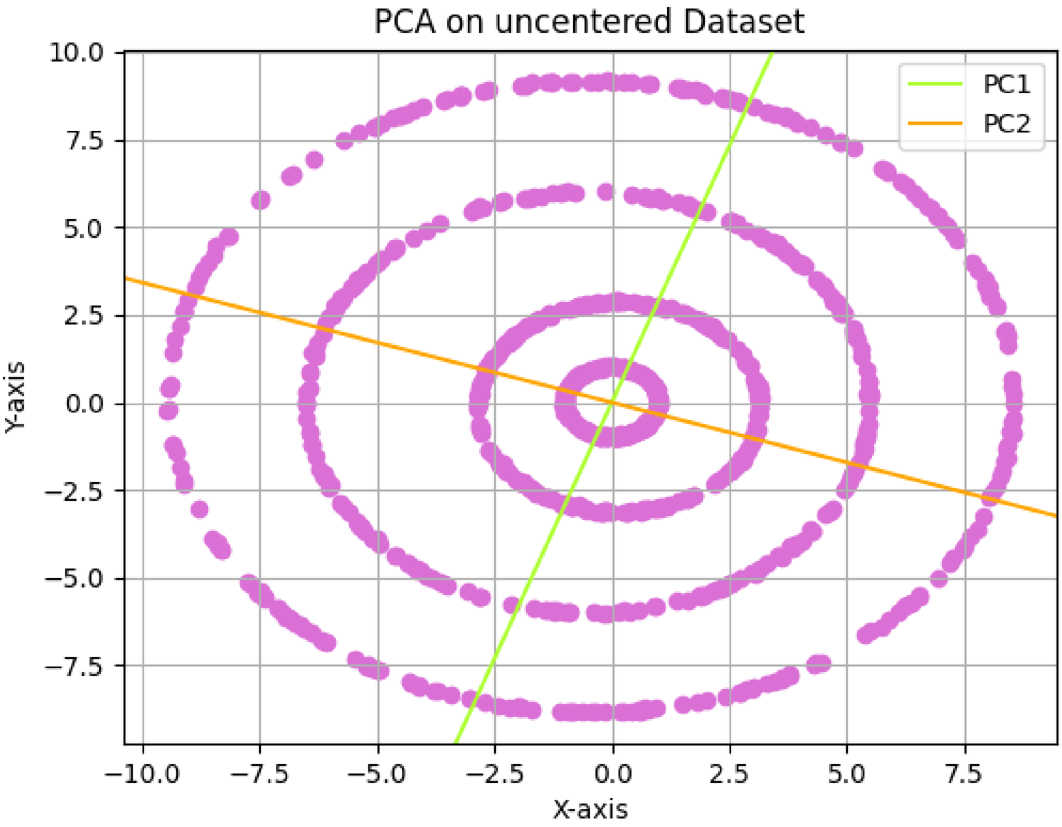
The variance in the dataset explained by each principal component is :-
Variance along principal component 1 = **54.178 %**
Variance along principal component 2 = **45.822 %**



Q1(ii)

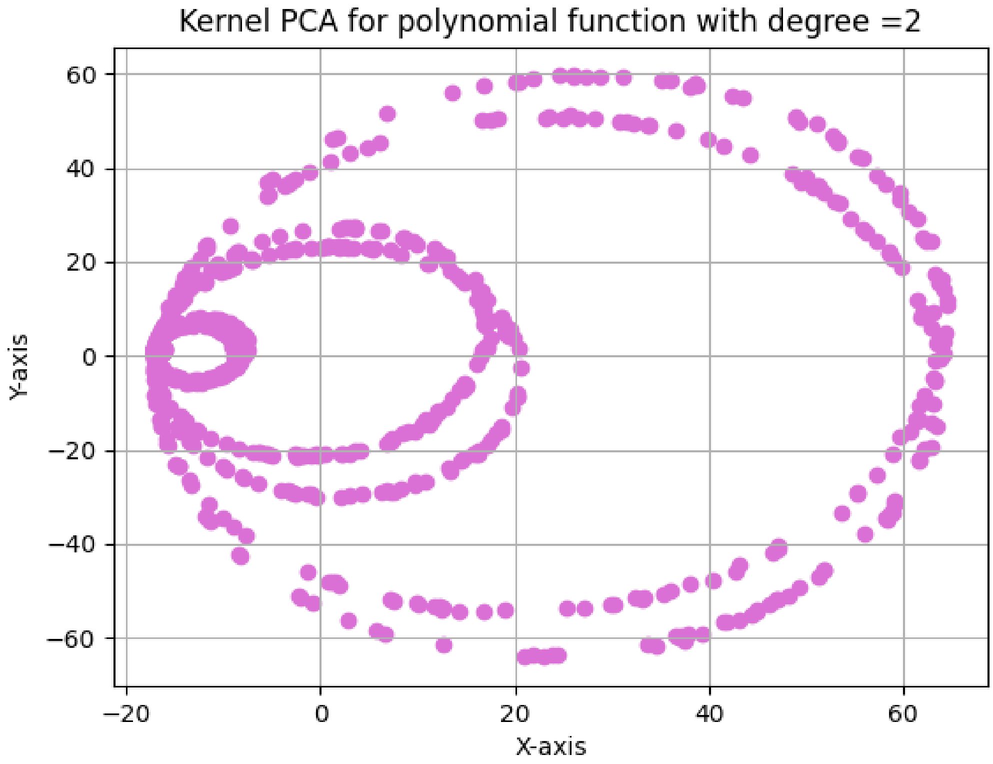
After performing PCA on dataset without centering :-
The variance in the dataset explained by each principal component is :-
Variance along principal component 1 = **54.178 %**
Variance along principal component 2 = **45.822 %**

Since the mean of the given dataset ie [4.075e-07 2.227e-07] is very small and close to (0,0) so there is negligible effect of centering the dataset and performing pca ie. the results before and after centering the dataset remains the same .So centering does not help in case of given dataset.

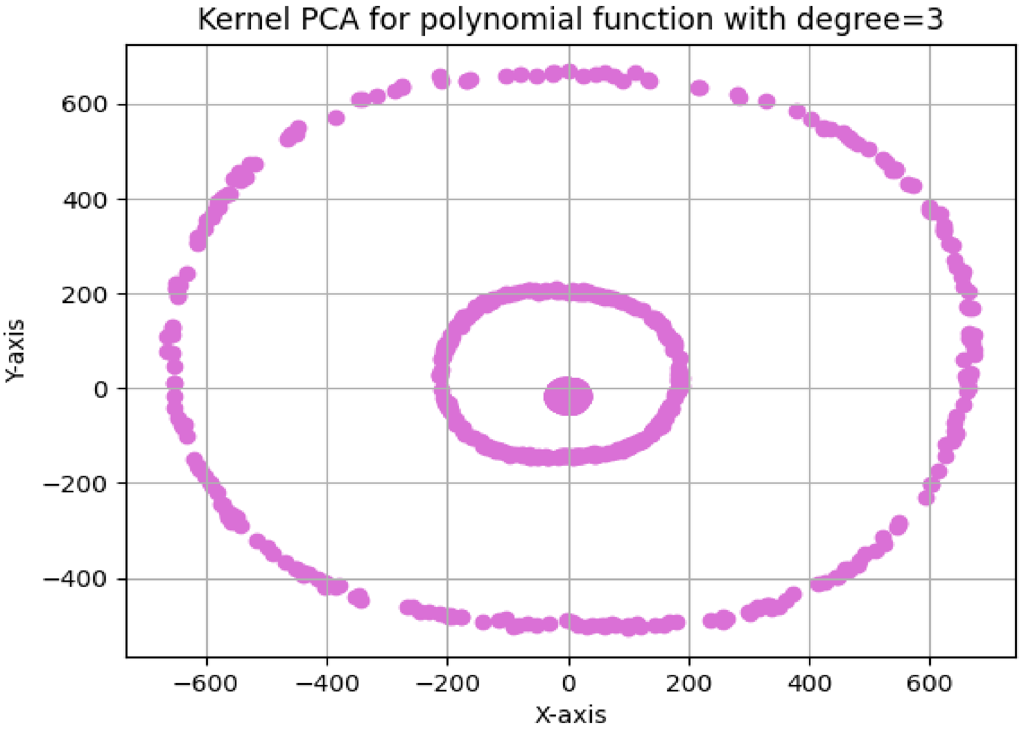


Q1 (iii)

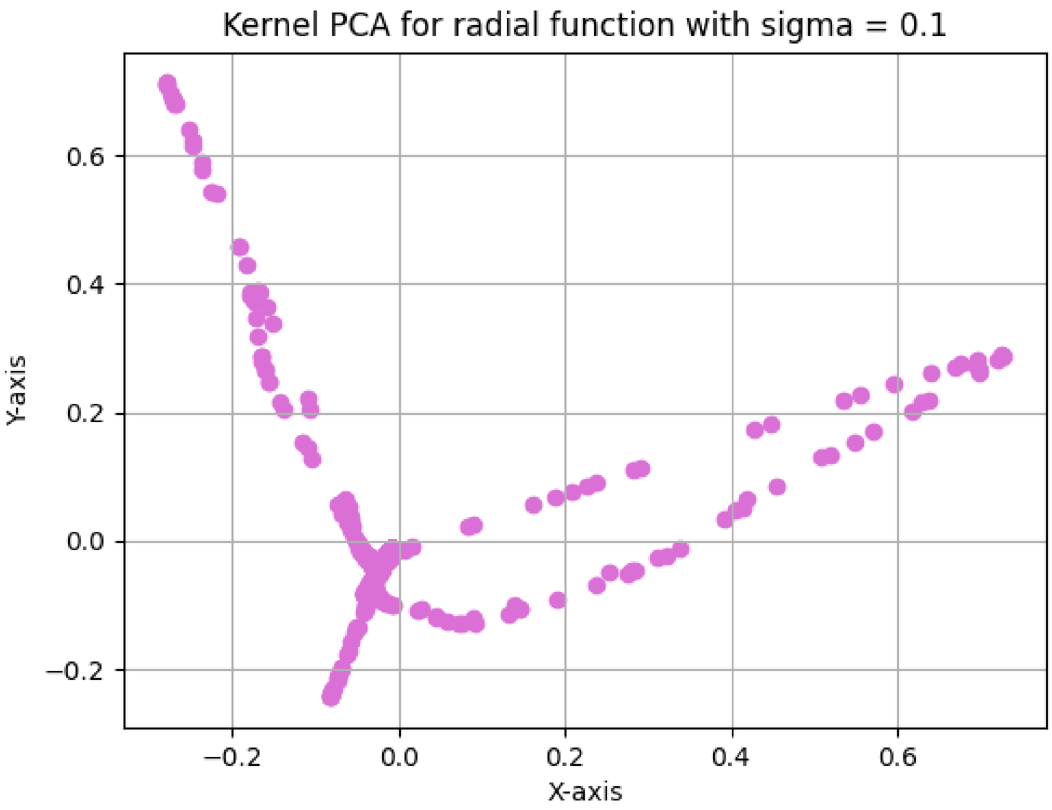
A.i. Plot for polynomial kernel function for degree 2 :



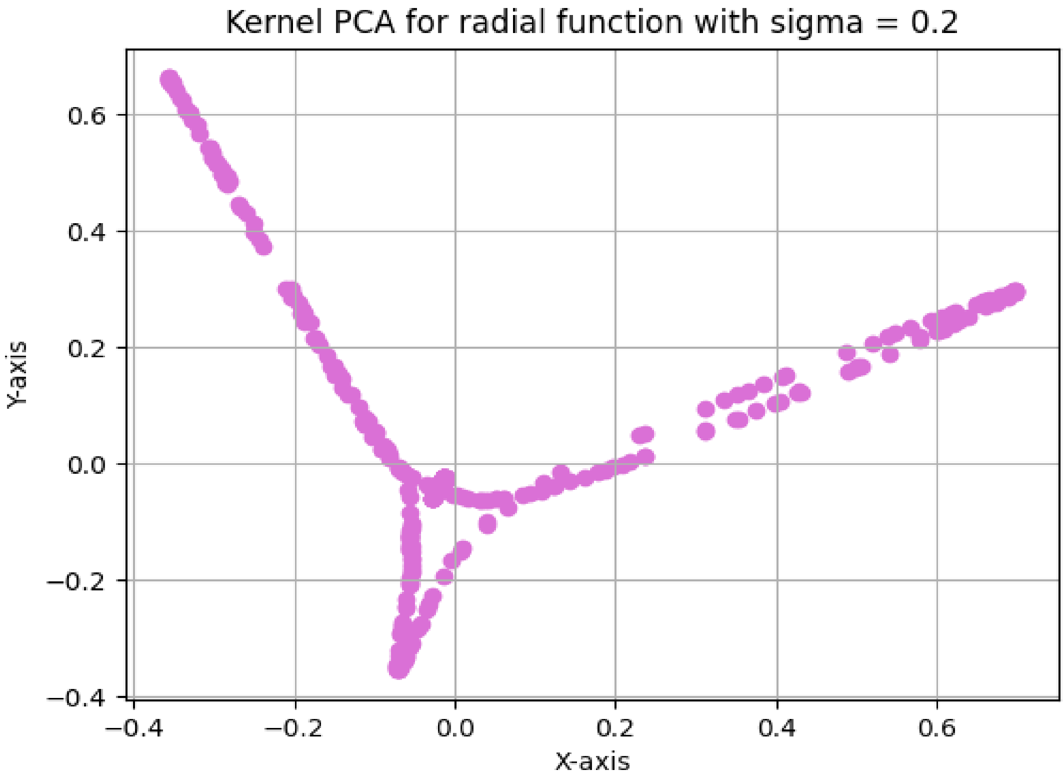
A.ii. Plot for polynomial kernel function for degree 3 :



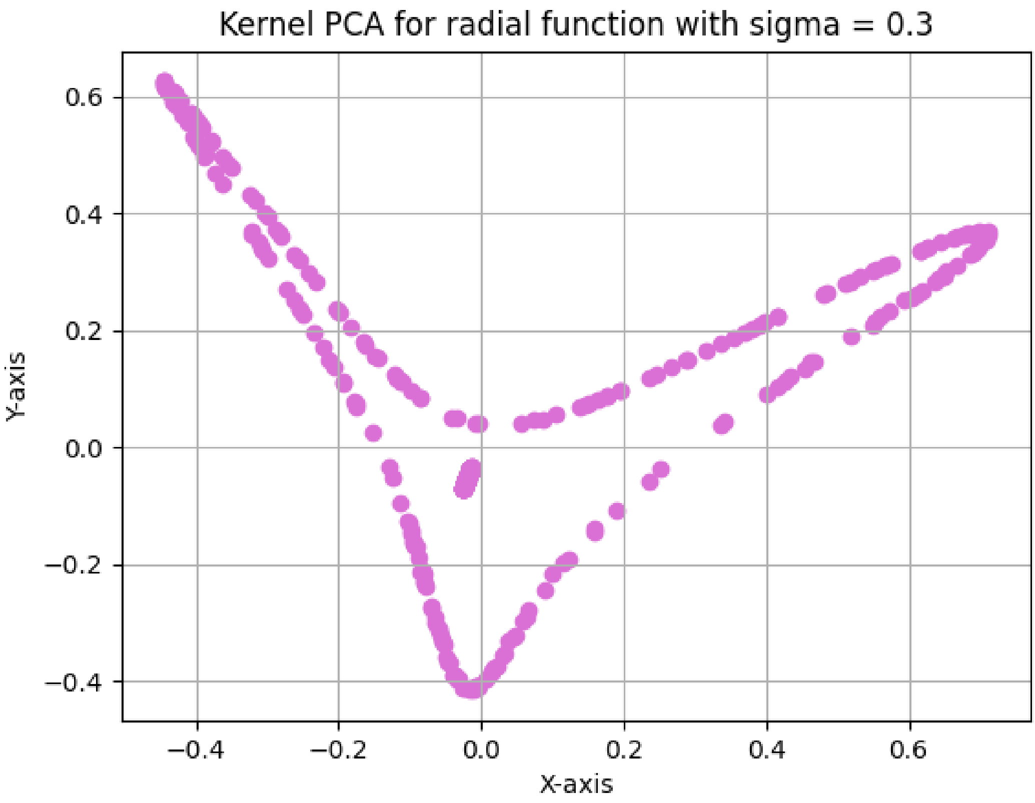
B.i. Plot for Radial Basis function for sigma=0.1 :



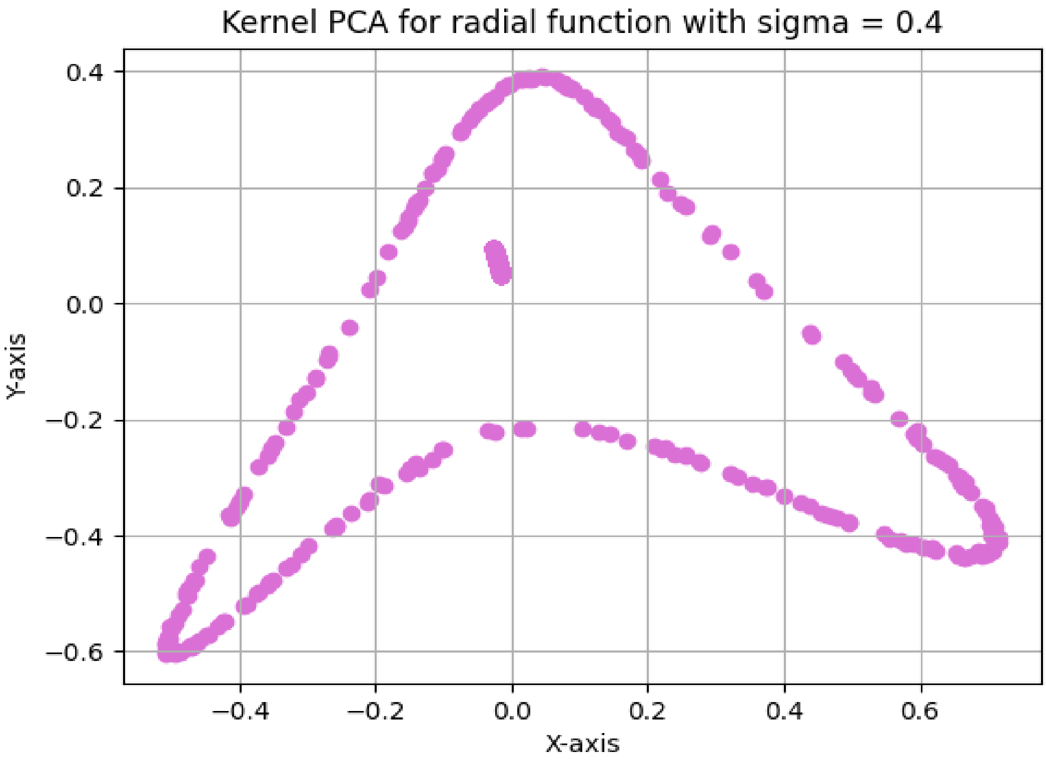
B.ii. Plot for Radial Basis function for sigma=0.2 :



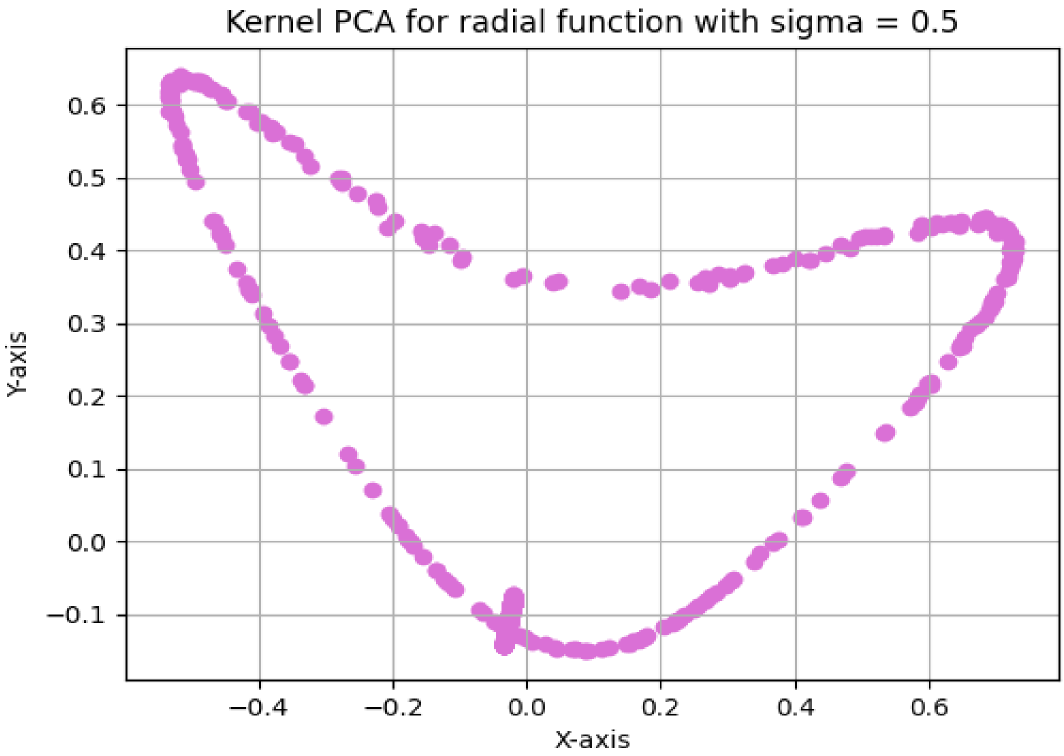
B.iii. Plot for Radial Basis function for sigma=0.3 :



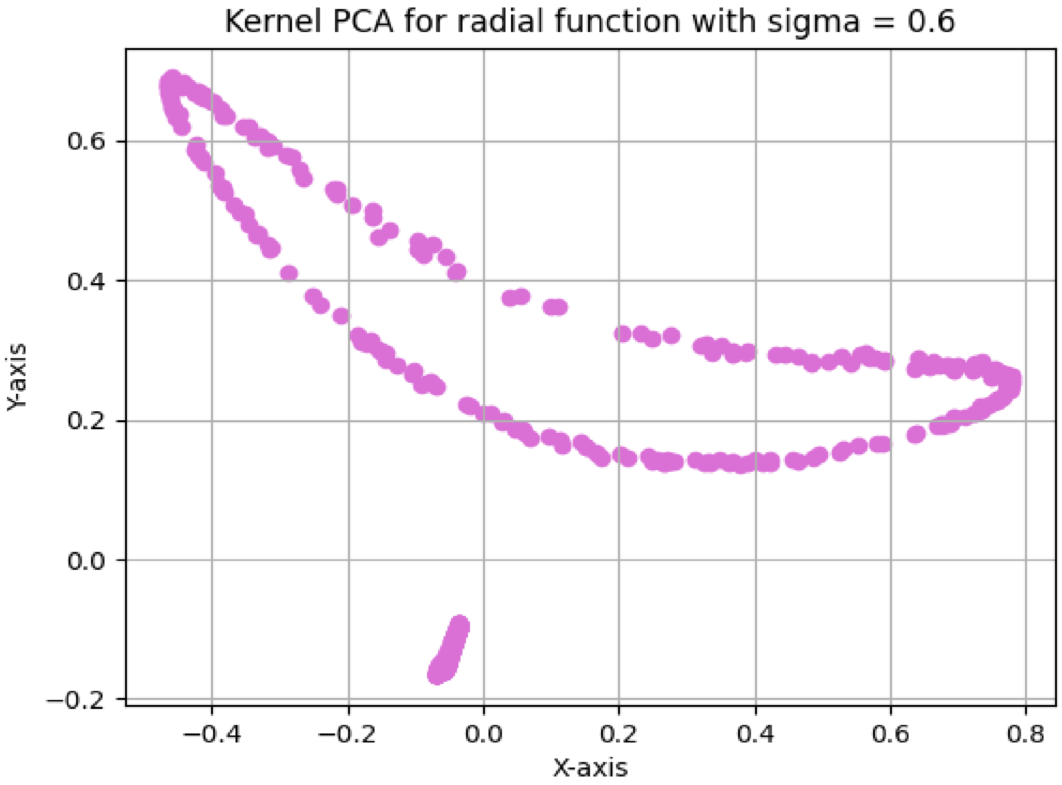
B.iv. Plot for Radial Basis function for sigma=0.4 :



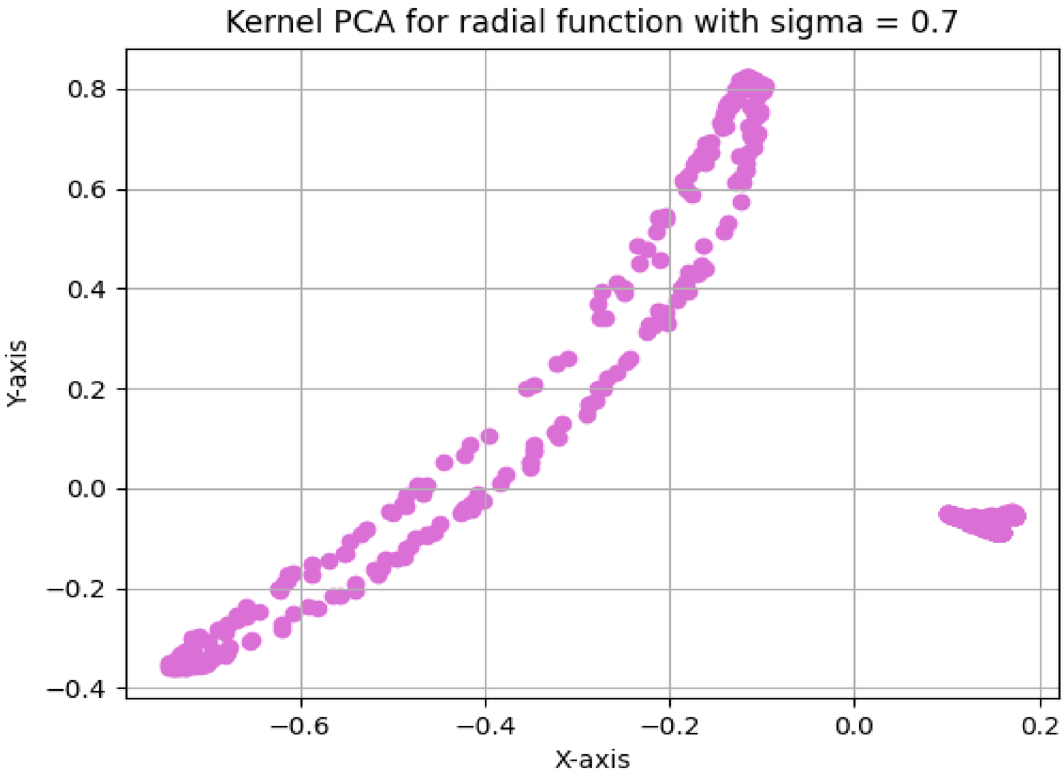
B.v. Plot for Radial Basis function for sigma=0.5 :



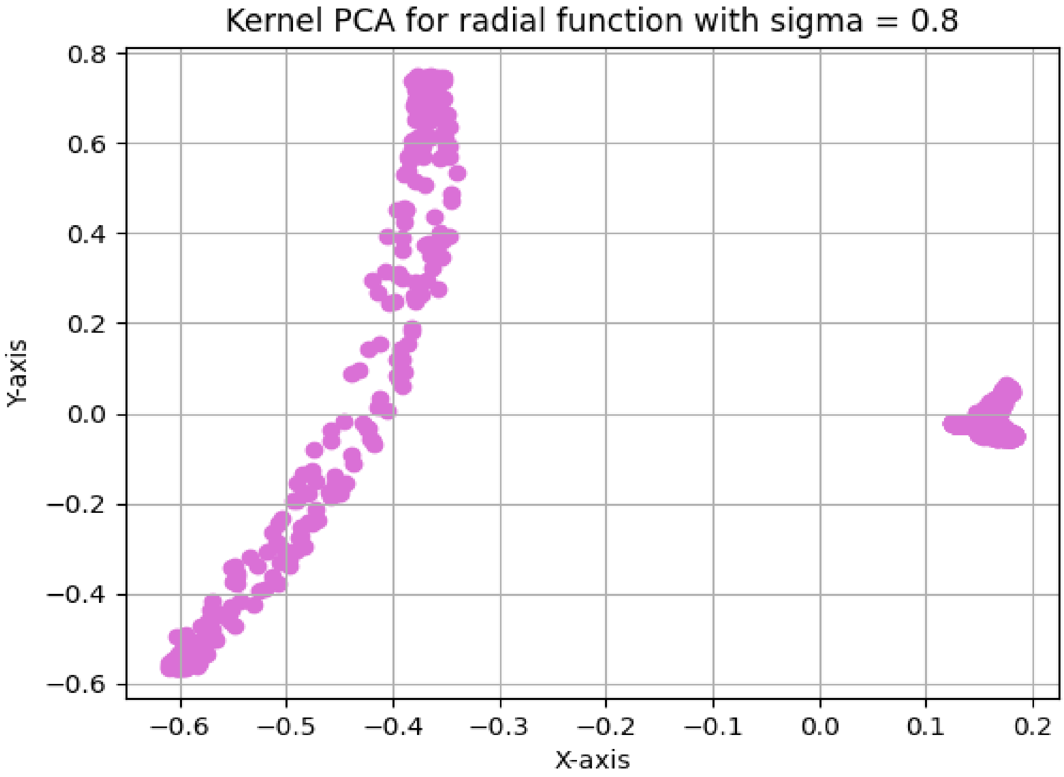
B.vi. Plot for Radial Basis function for sigma=0.6 :



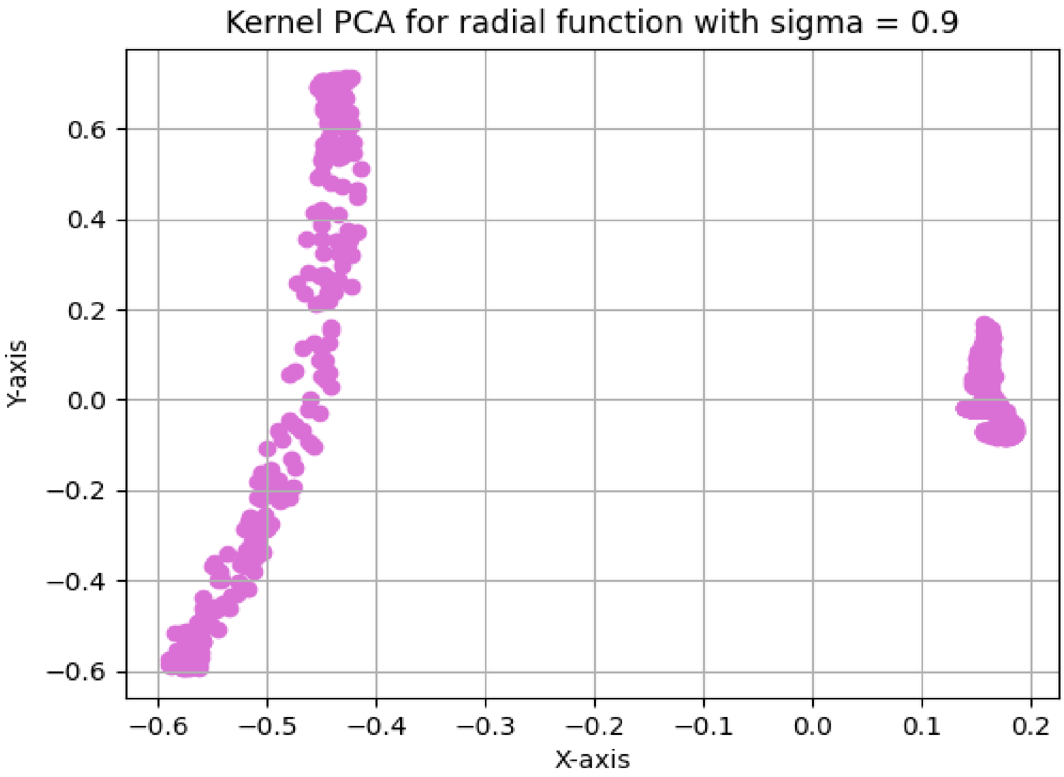
B.vii. Plot for Radial Basis function for sigma=0.7 :



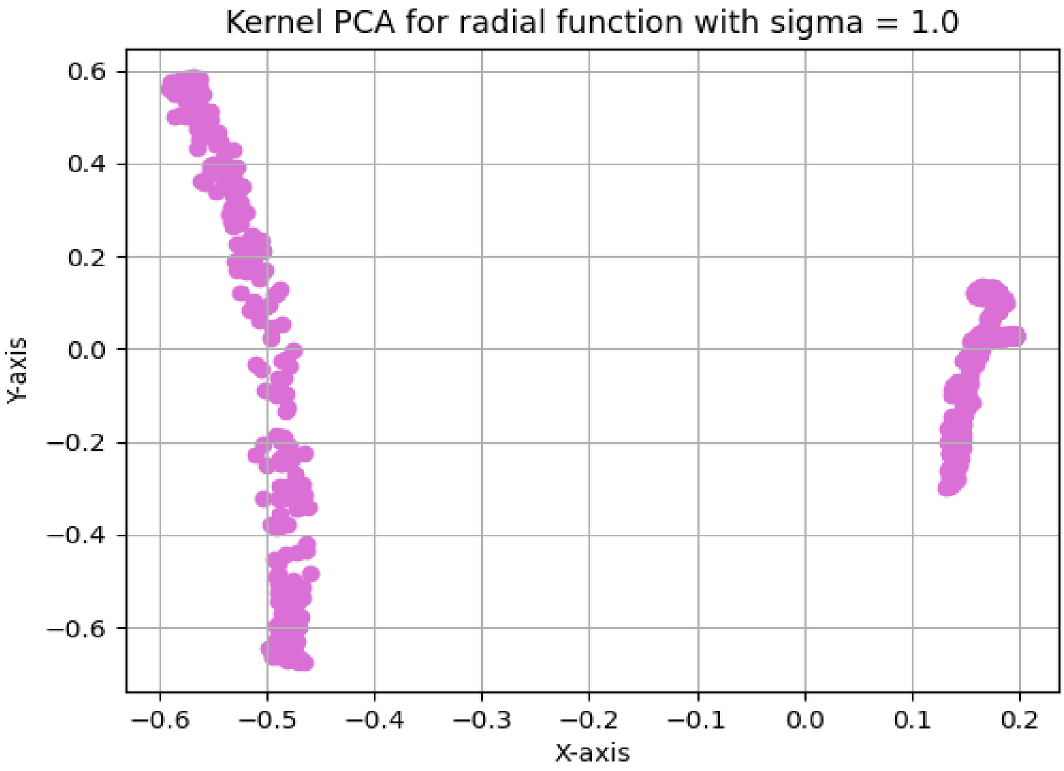
B.viii. Plot for Radial Basis function for sigma=0.8 :



B.ix. Plot for Radial Basis function for sigma=0.9 :



B.x. Plot for Radial Basis function for sigma=1.0 :



Q1.(iv)

Polynomial kernel function is better than radial basis kernel function for the given dataset. When we plot the projection of each point on the top 2 components of each kernel we are able to reconstruct the original dataset upto a certain limit in case of polynomial kernel function but in case of radial basis function the plots obtained by projecting the dataset on the top two components of radial basis kernel function does not reconstruct the original dataset in any way which means that the variance of the top two components of radial kernel function is not enough to reconstruct original dataset and so it starts forming clusters of data so polynomial kernel function is better.

Variance obtained by top two components is given by:-

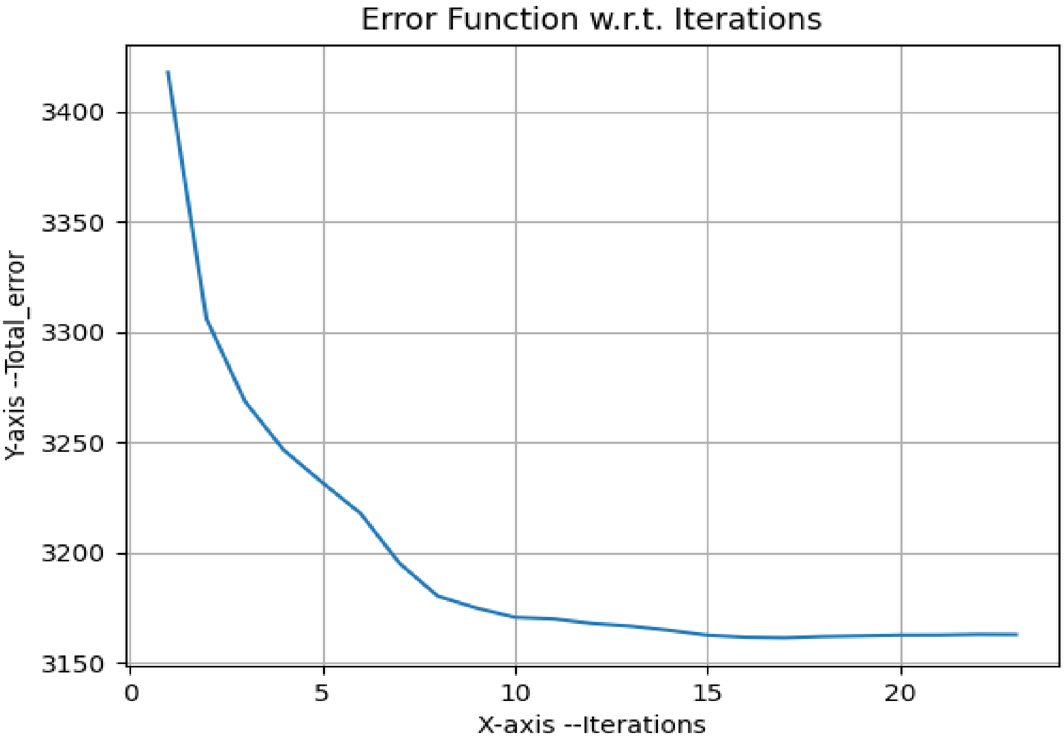
Sum of top two eigenvalues of K/sum of all eigenvalues of K

For polynomial kernel function with d=2:
The variance retained by the top two components is nearly 68.5%
For polynomial kernel function with d=3:
The variance retained by the top two components is nearly 73.4%
For radial basis kernel function the max variance retained by the top two components among all values from 0.1 to 1.0 is for sigma=1.0 and the is 15.6%

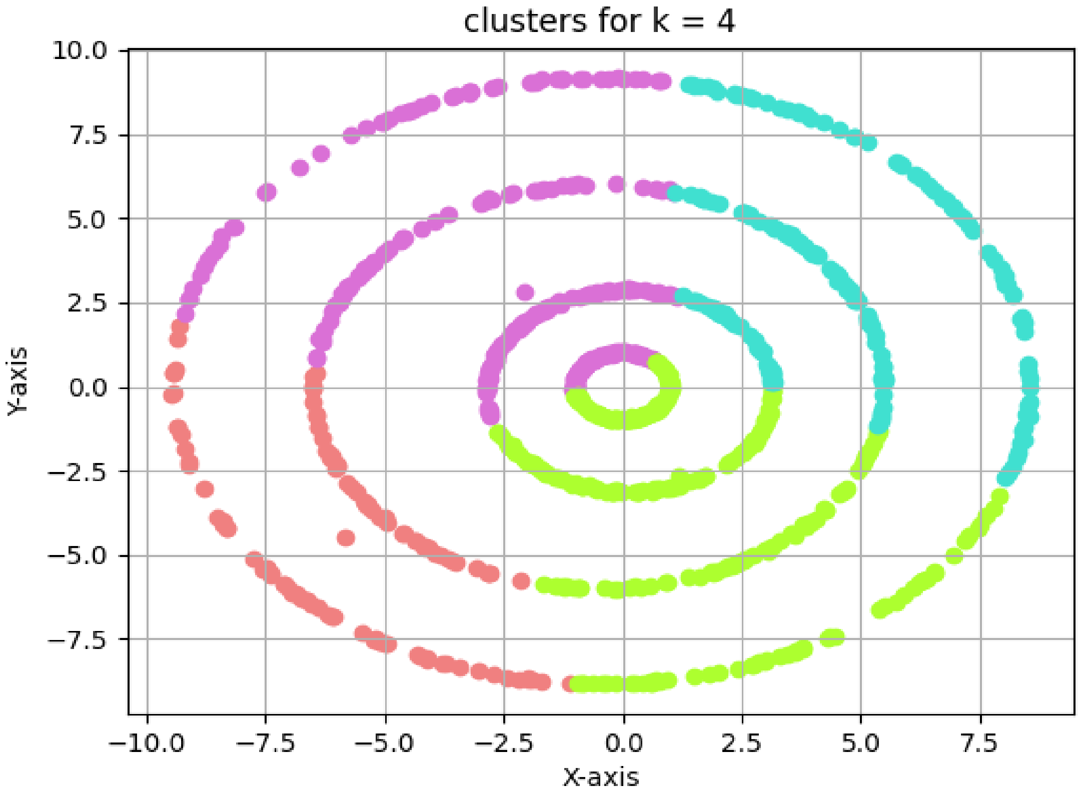
Therefore, from the above argument we can infer that the polynomial kernel with d=3 is better as compared to any other given kernel functions.

Q2 (i)

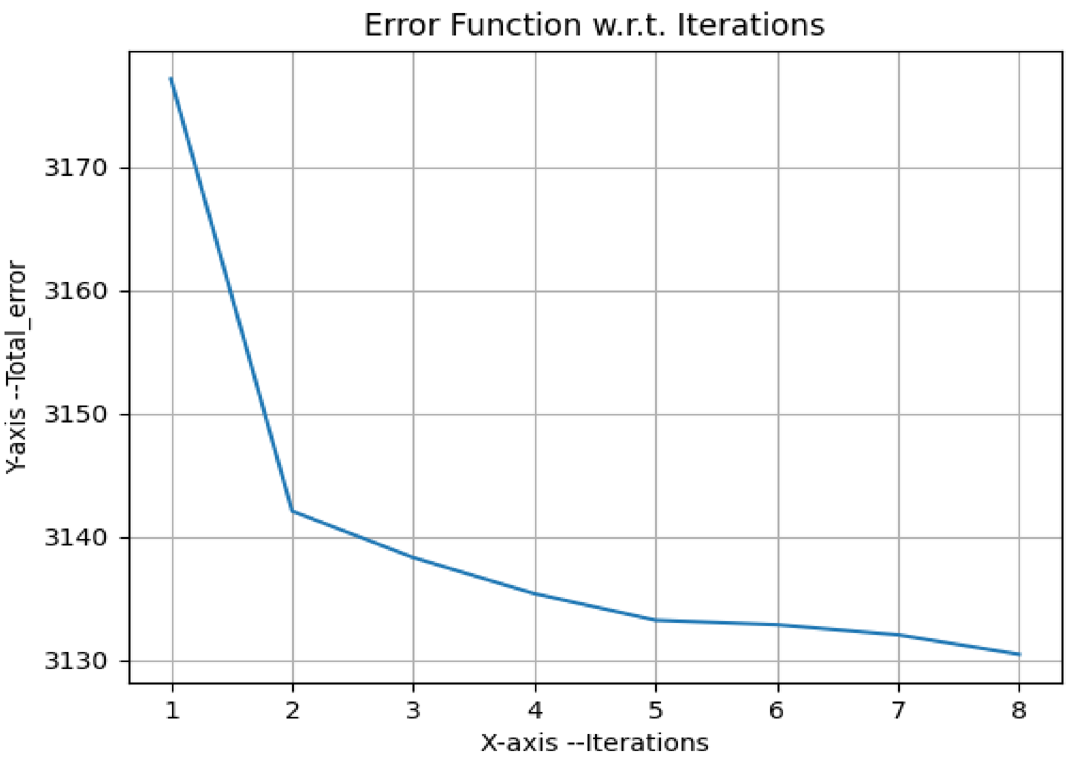
- **Error Function w.r.t. Iterations for random initialization no.1-**



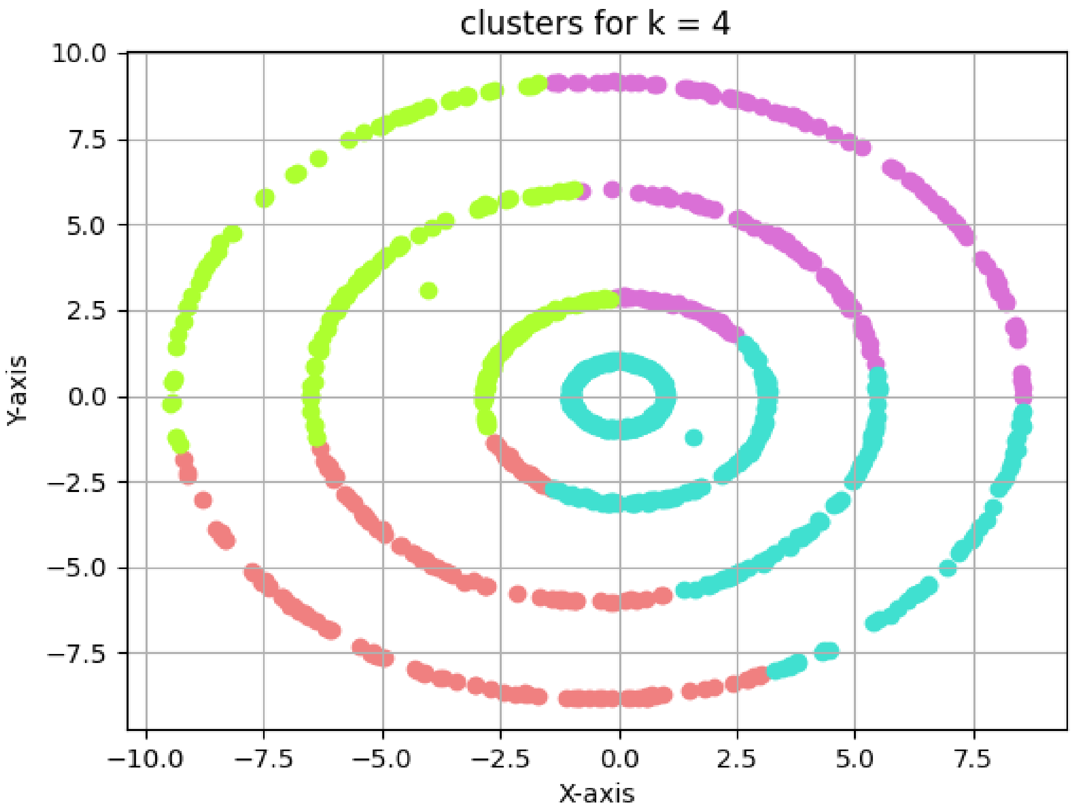
- **Clusters obtained for random initialization no.1-**



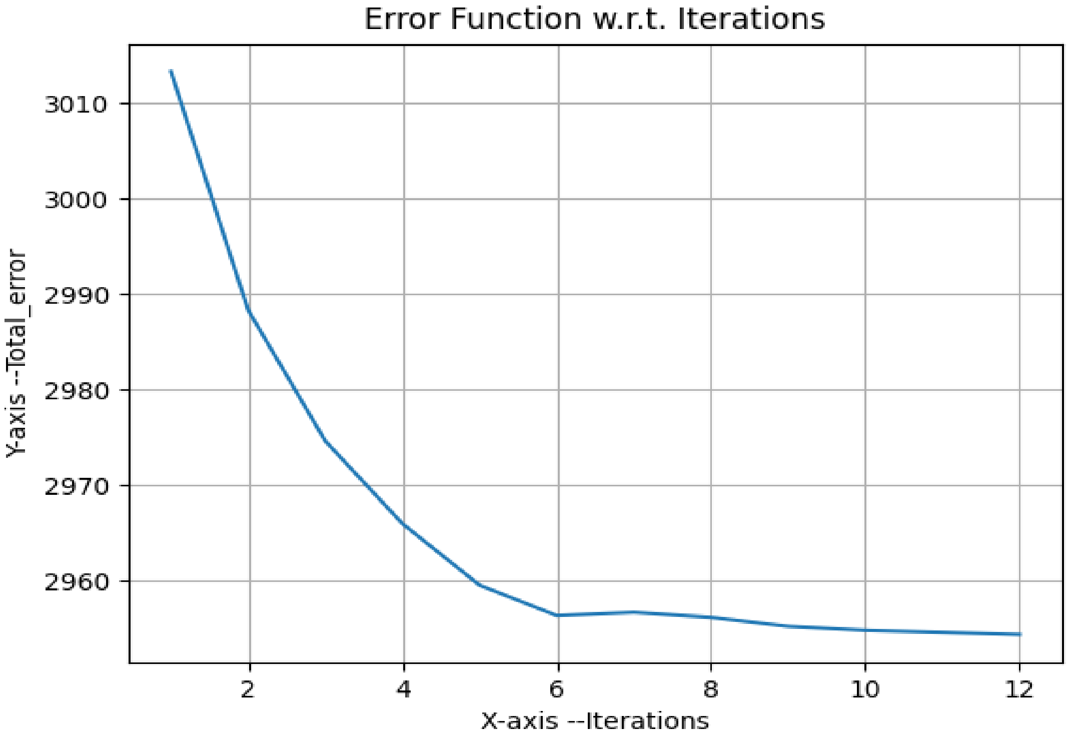
- **Error Function w.r.t. Iterations for random initialization no.2-**



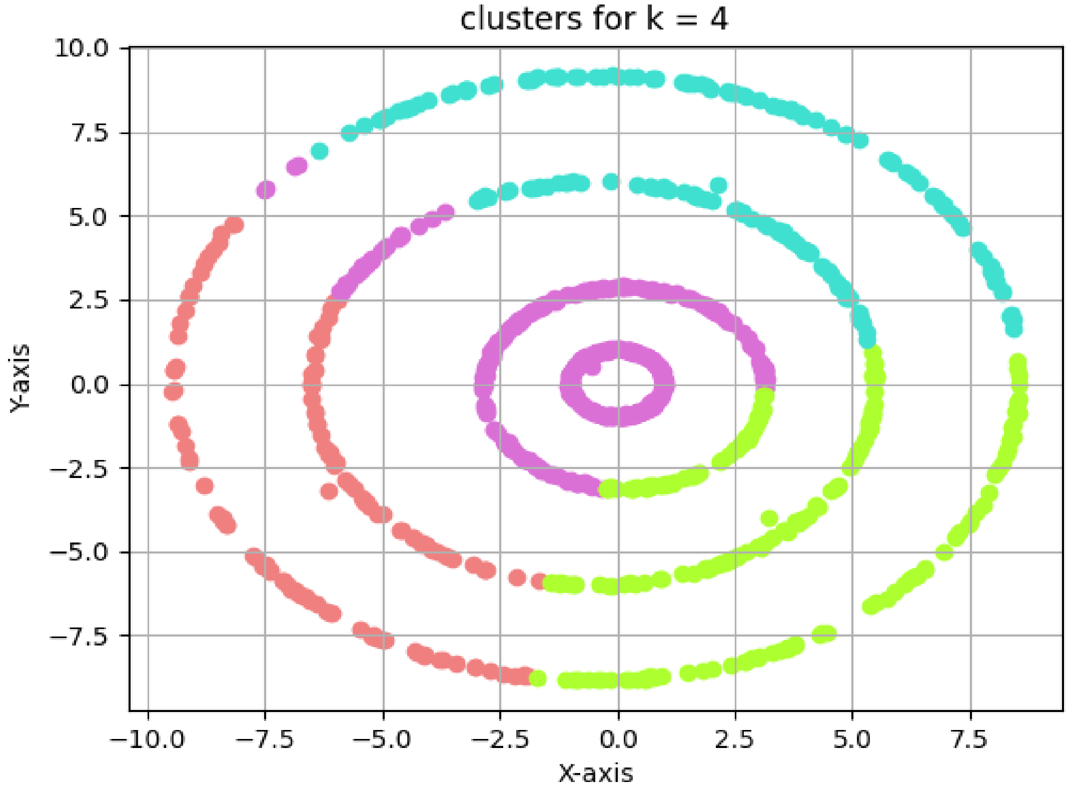
- **Clusters obtained for random initialization no.2-**



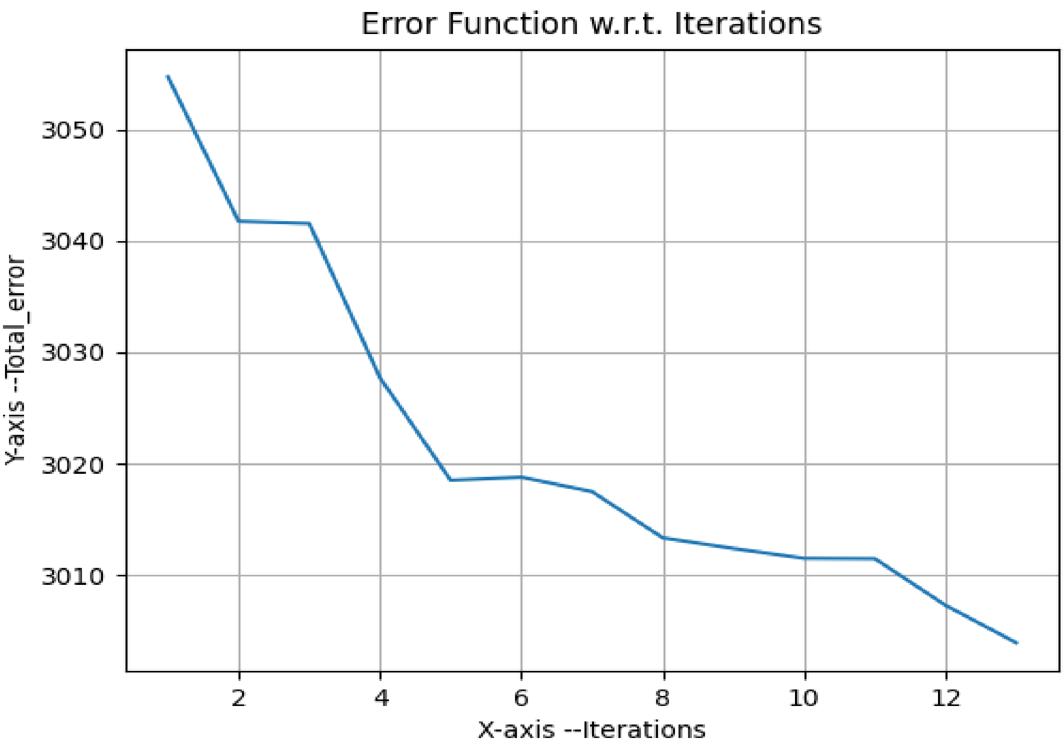
- **Error Function w.r.t. Iterations for random initialization no.3-**



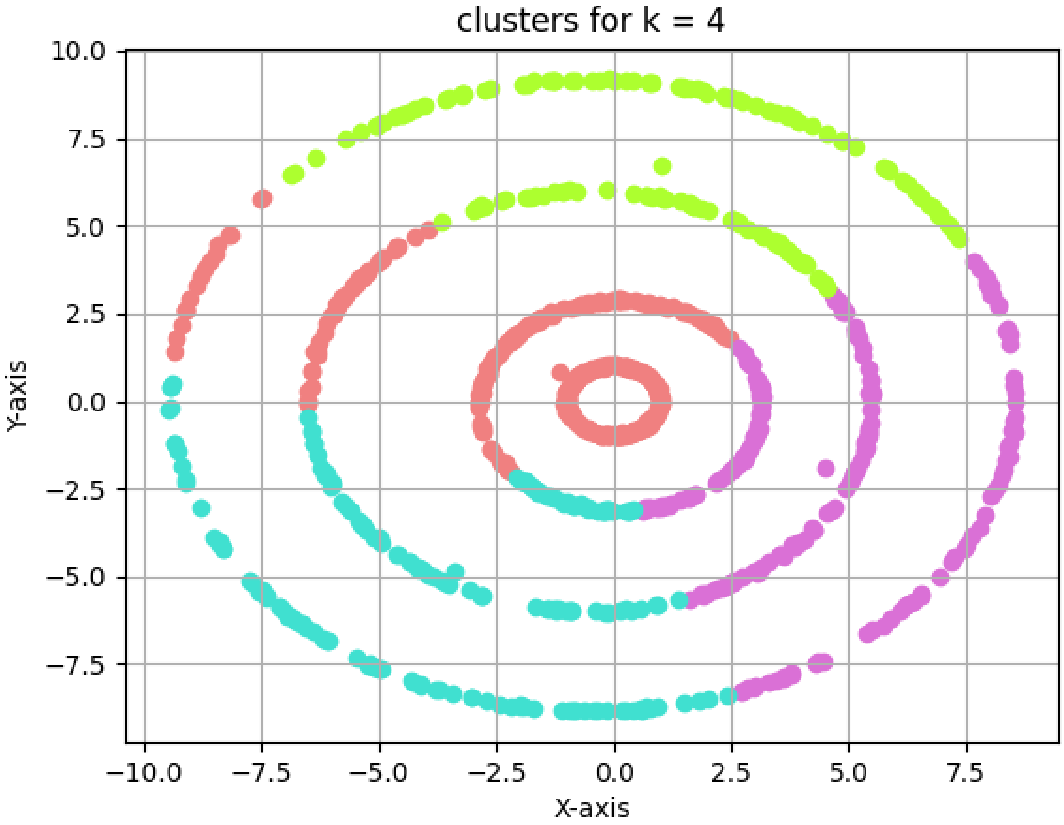
- **Clusters obtained for random initialization no.3-**



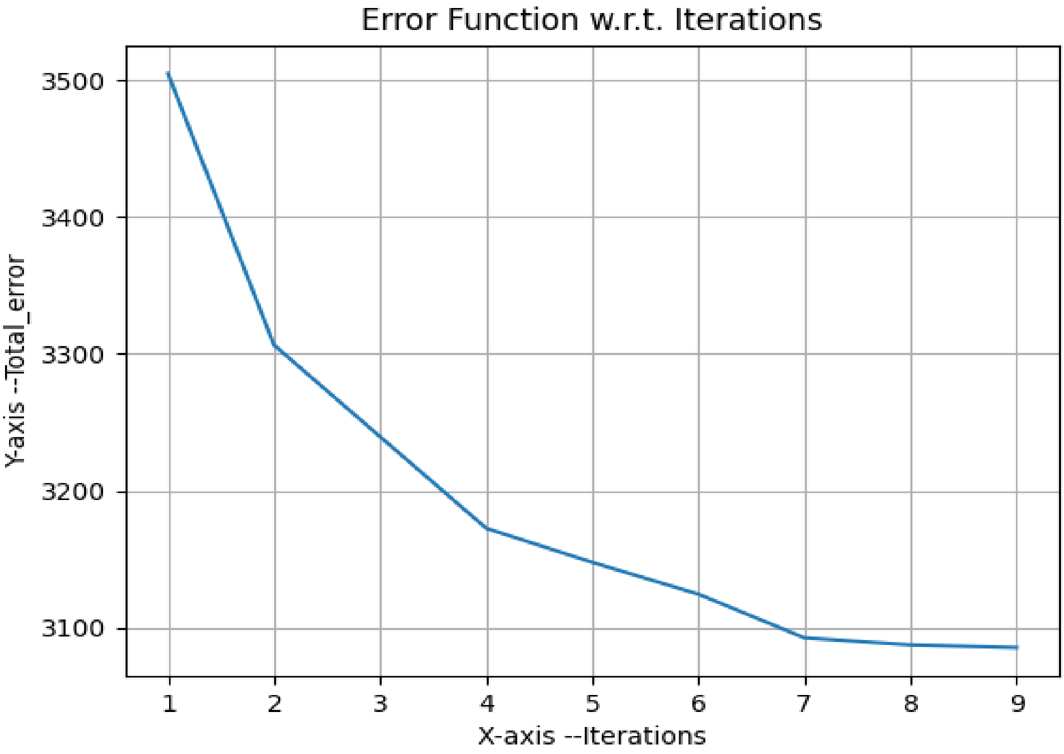
- **Error Function w.r.t. Iterations for random initialization no.4-**



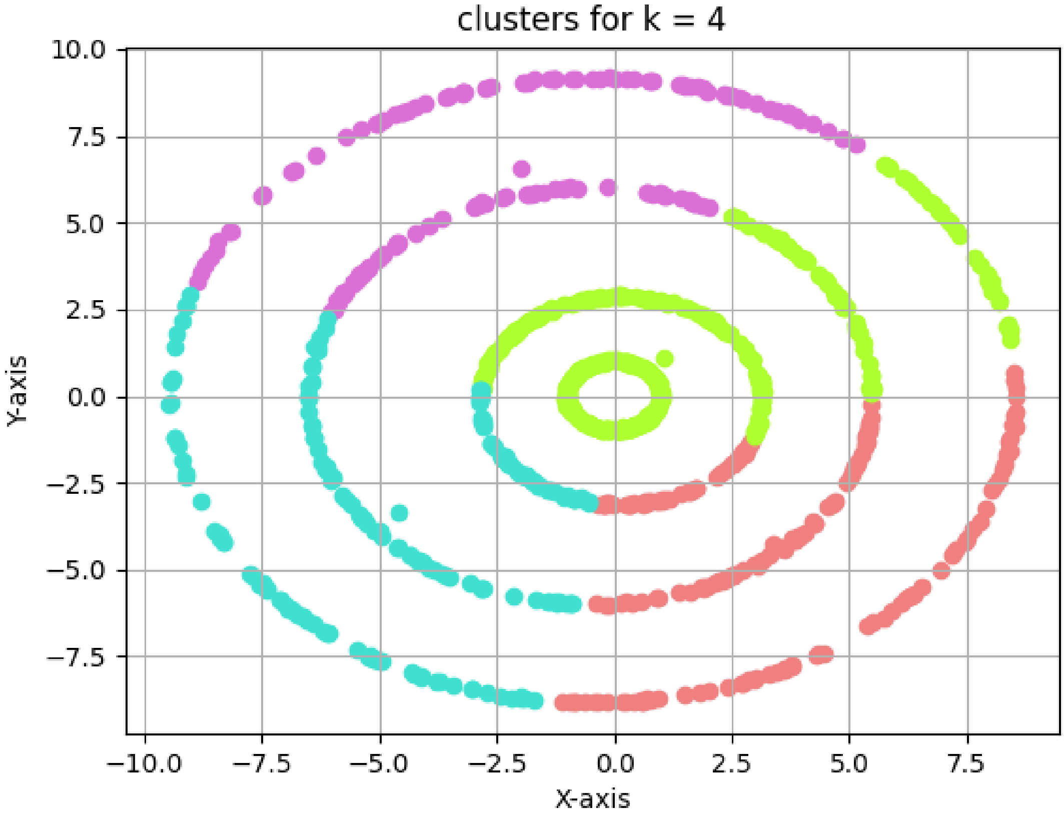
- **Clusters obtained for random initialization no.4-**



- **Error Function w.r.t. Iterations for random initialization no.5-**

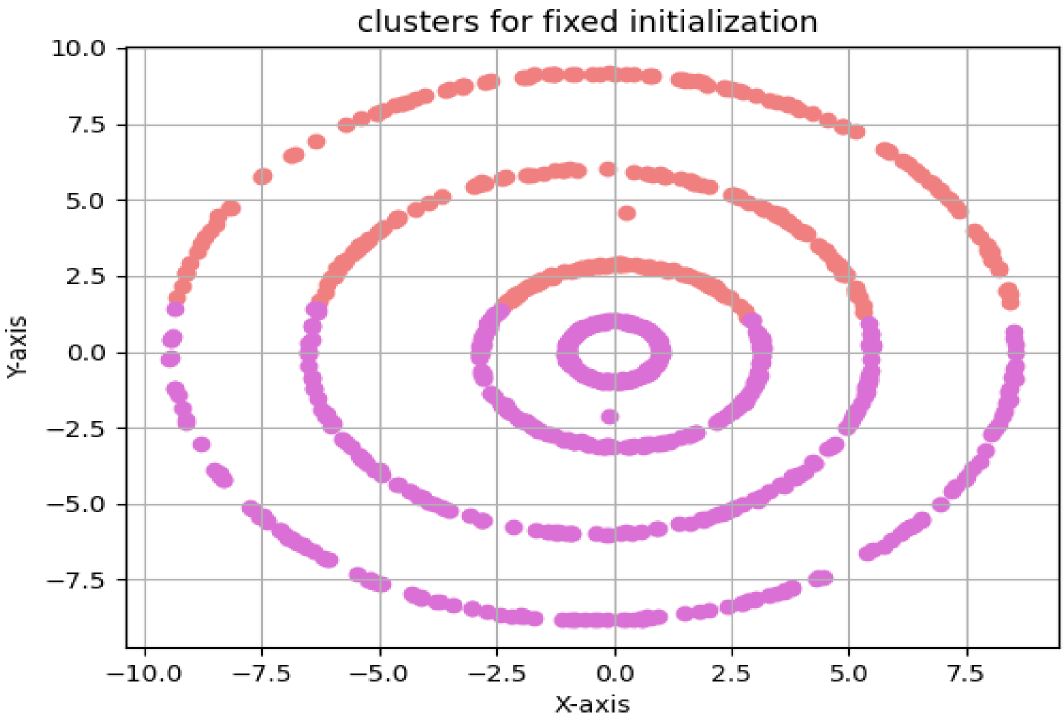


- **Clusters obtained for random initialization no.5-**

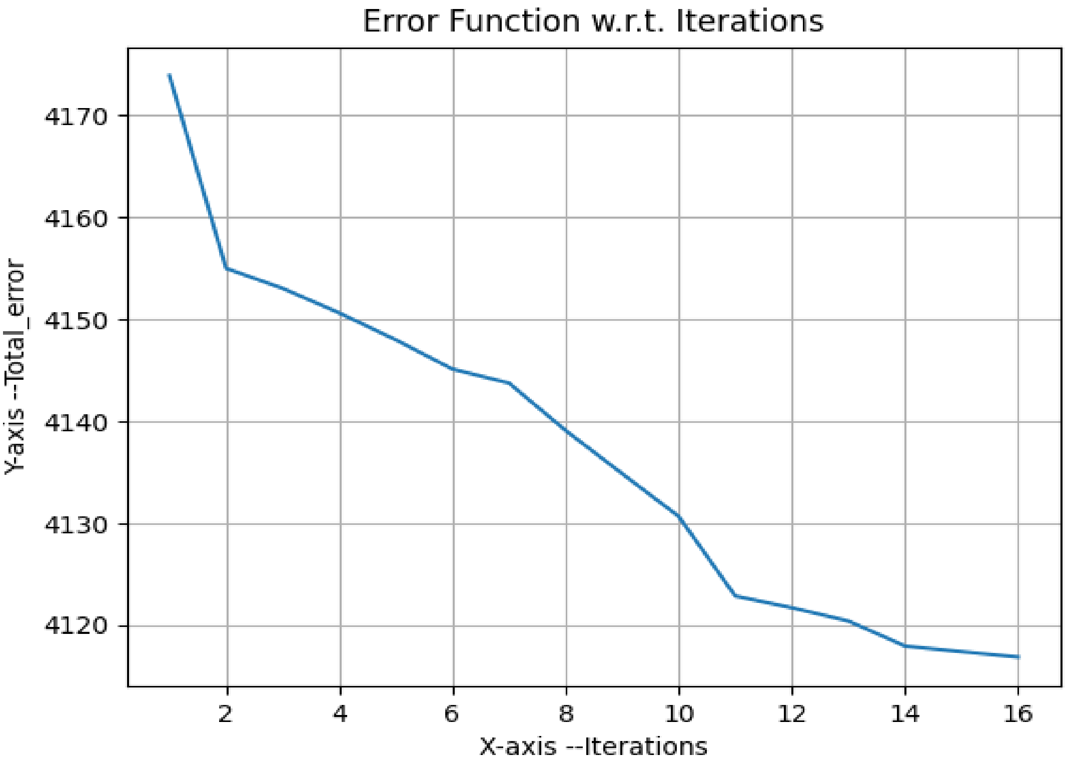


Q.2 (ii)

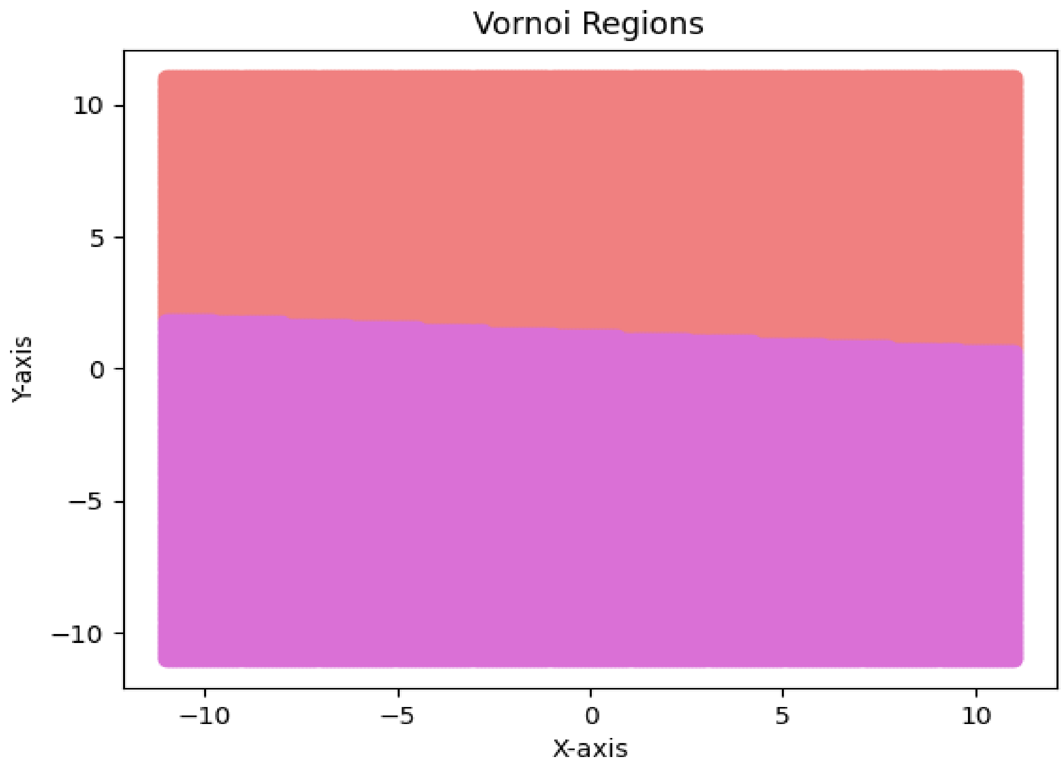
- **Clusters for a fixed random initialization with k=2:**



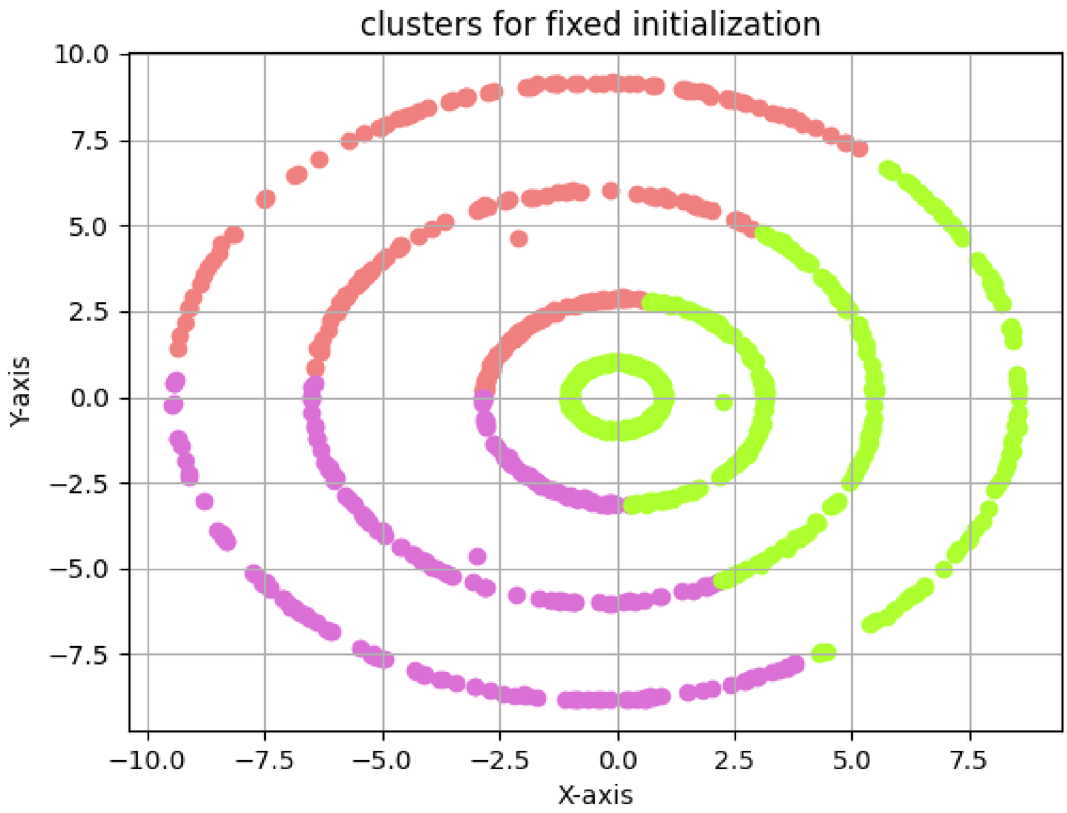
- **Error function w.r.t. Iterations for a fixed random initialization with k=2:**



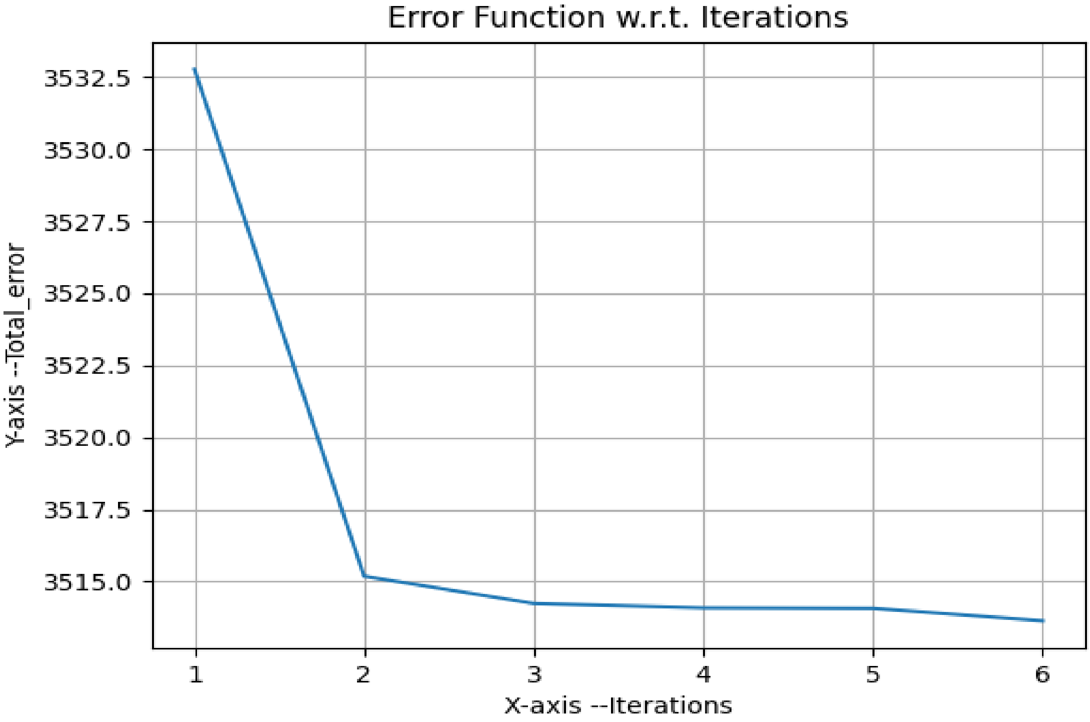
- **Voronoi regions for a fixed random initialization with k=2:**



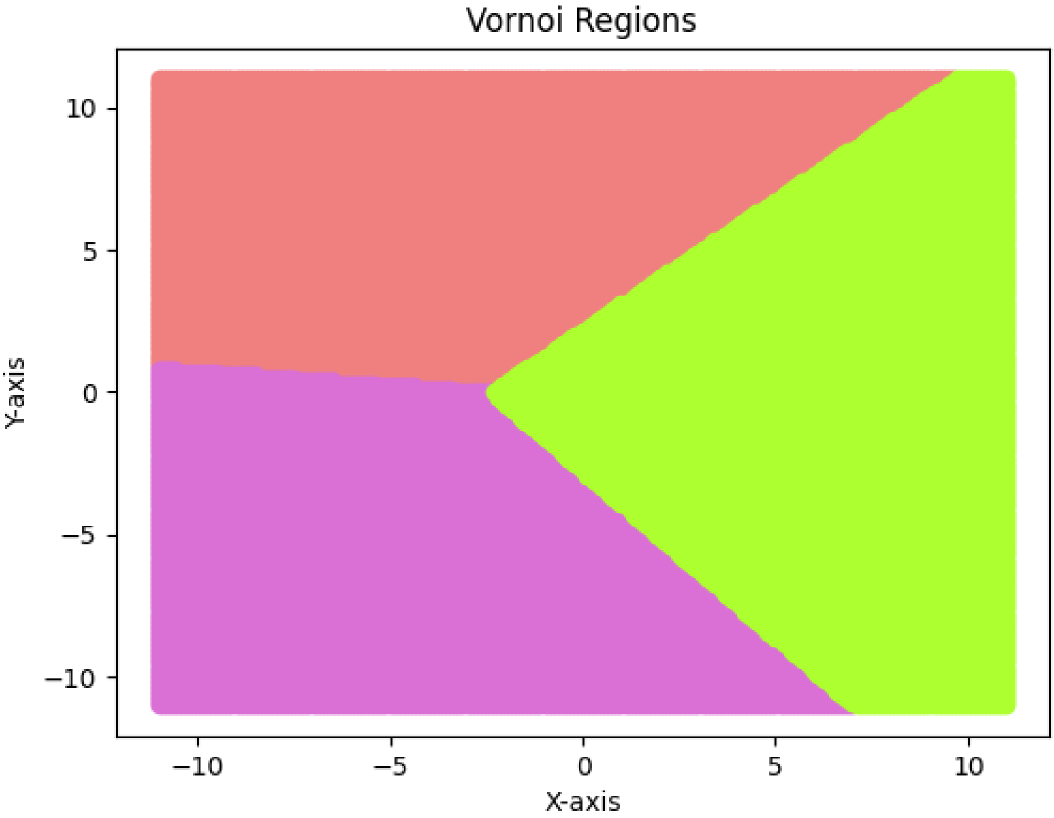
- **Clusters for a fixed random initialization with k=3:**



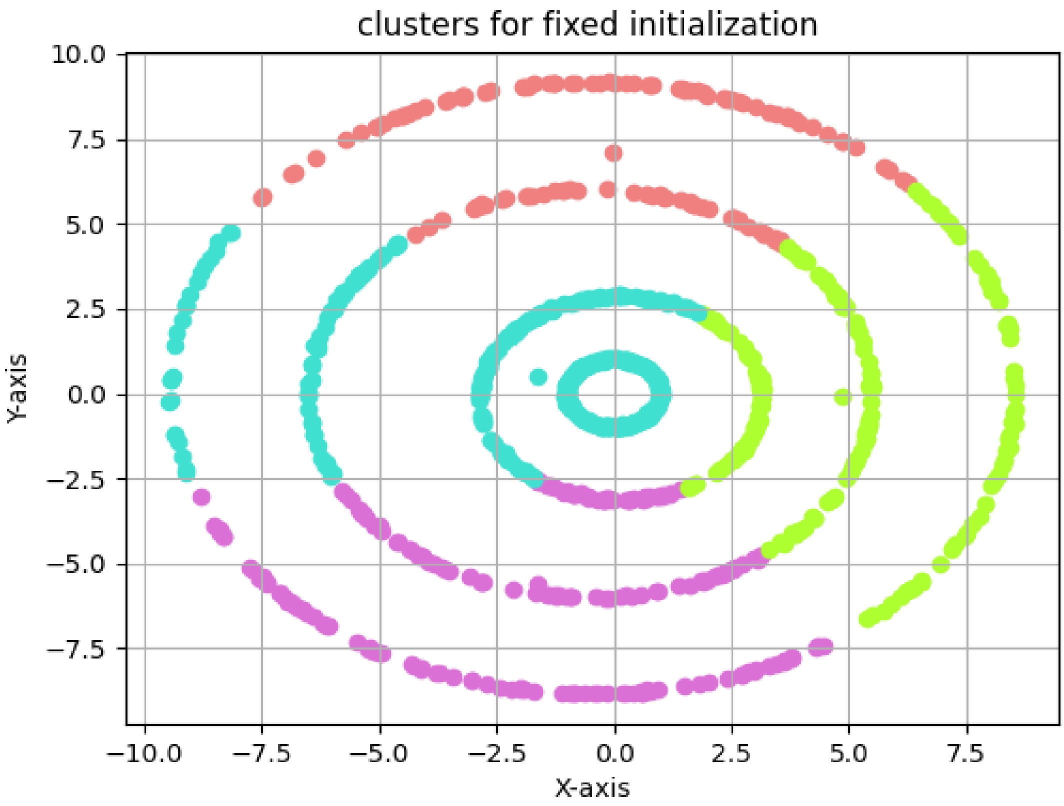
- **Error function w.r.t. Iterations for a fixed random initialization with k=3:**



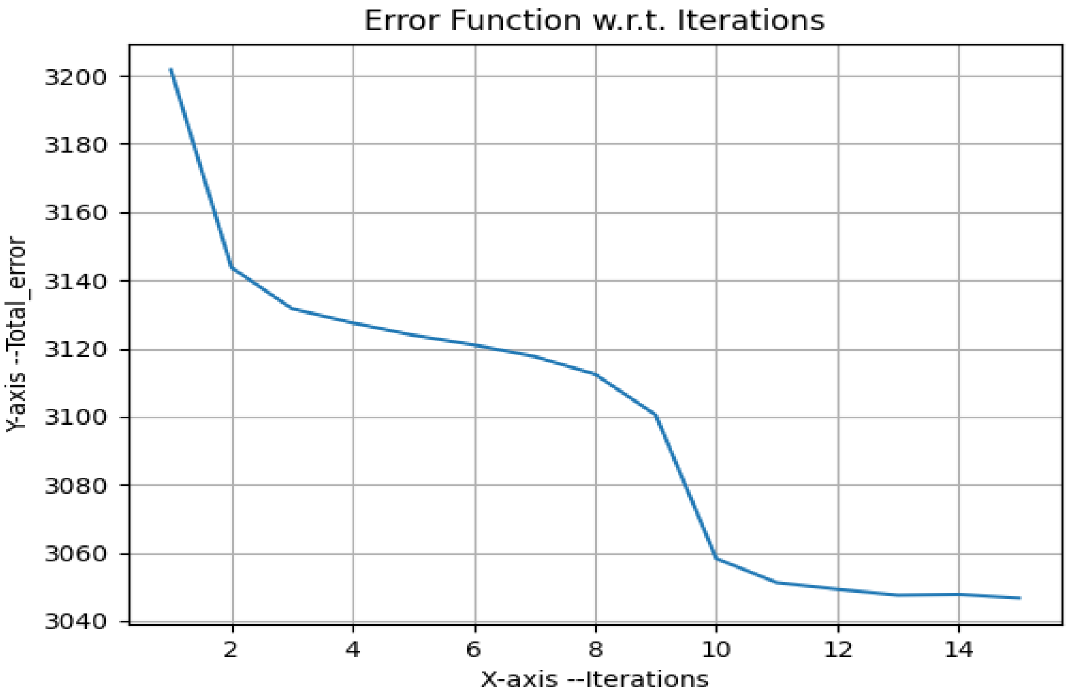
- **Voronoi regions for a fixed random initialization with k=3:**



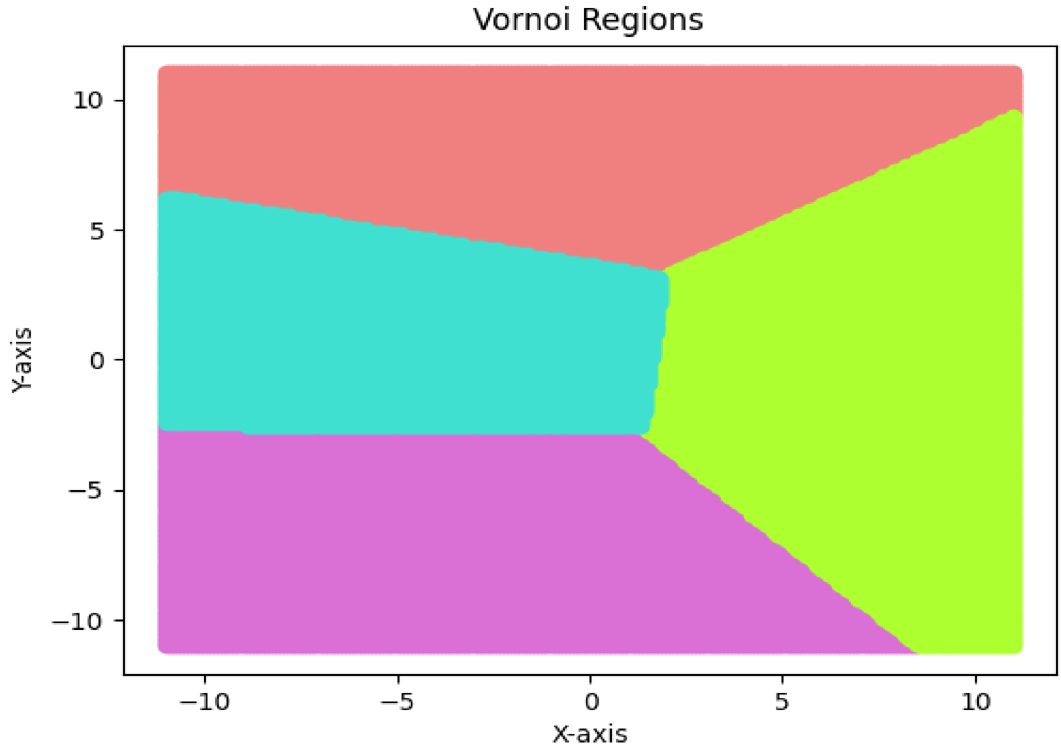
- **Clusters for a fixed random initialization with k=4:**



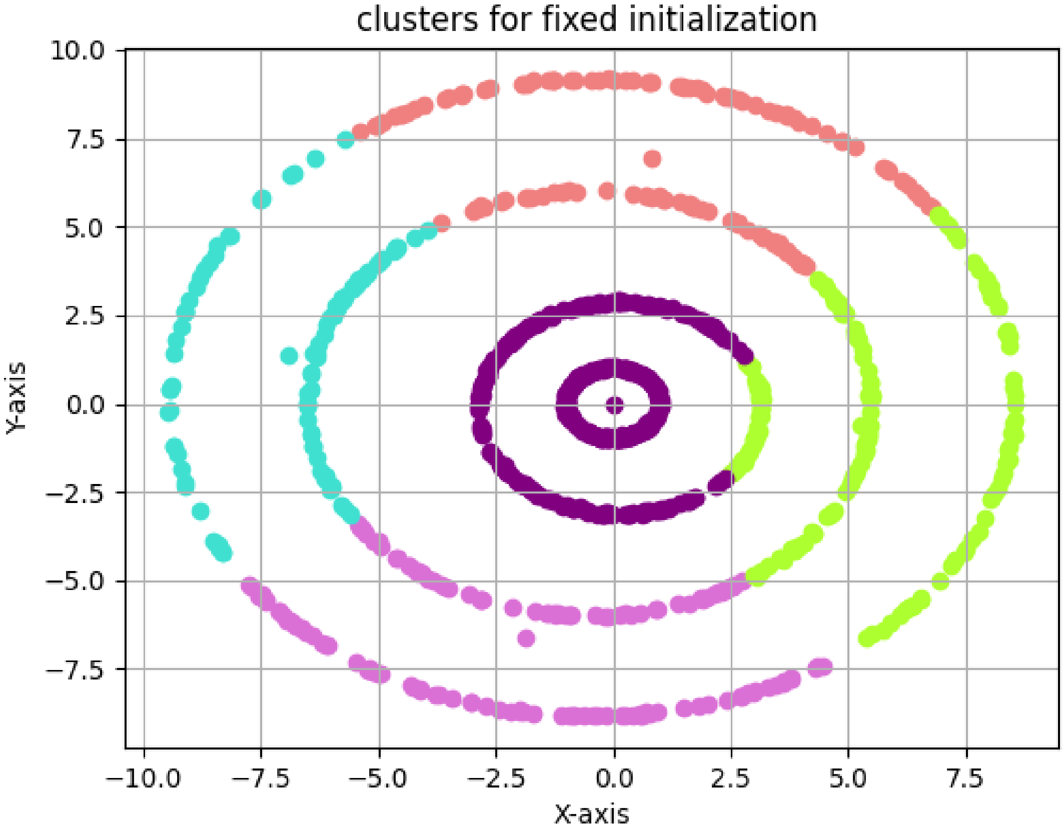
- **Error function w.r.t. Iterations for a fixed random initialization with k=4:**



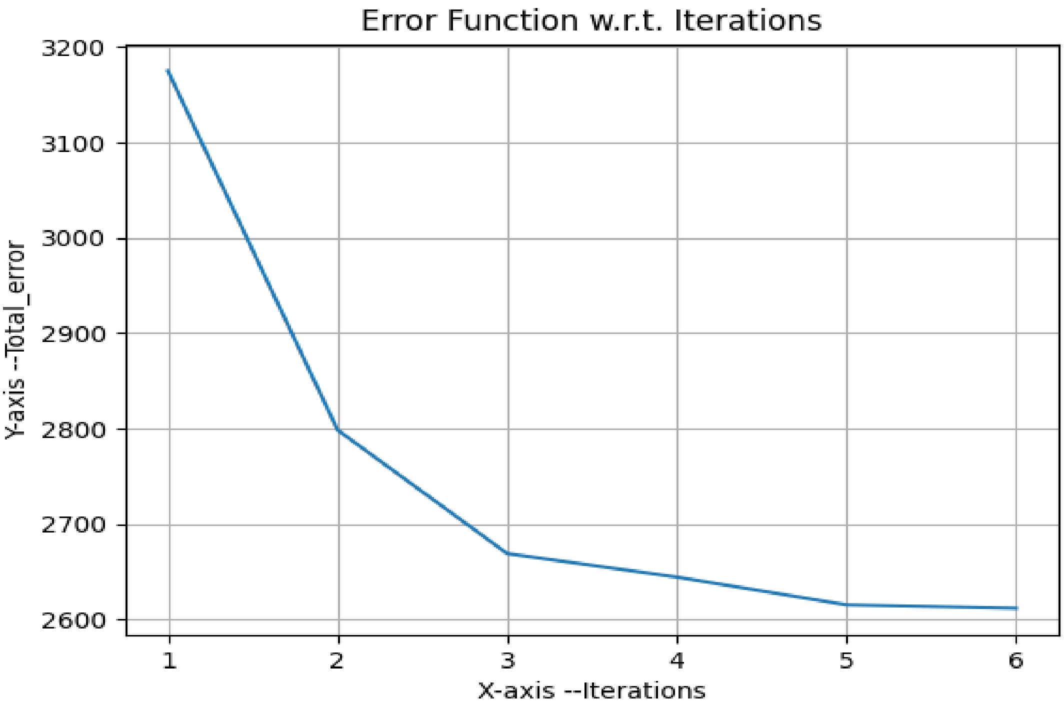
- **Voronoi regions for a fixed random initialization with k=4:**



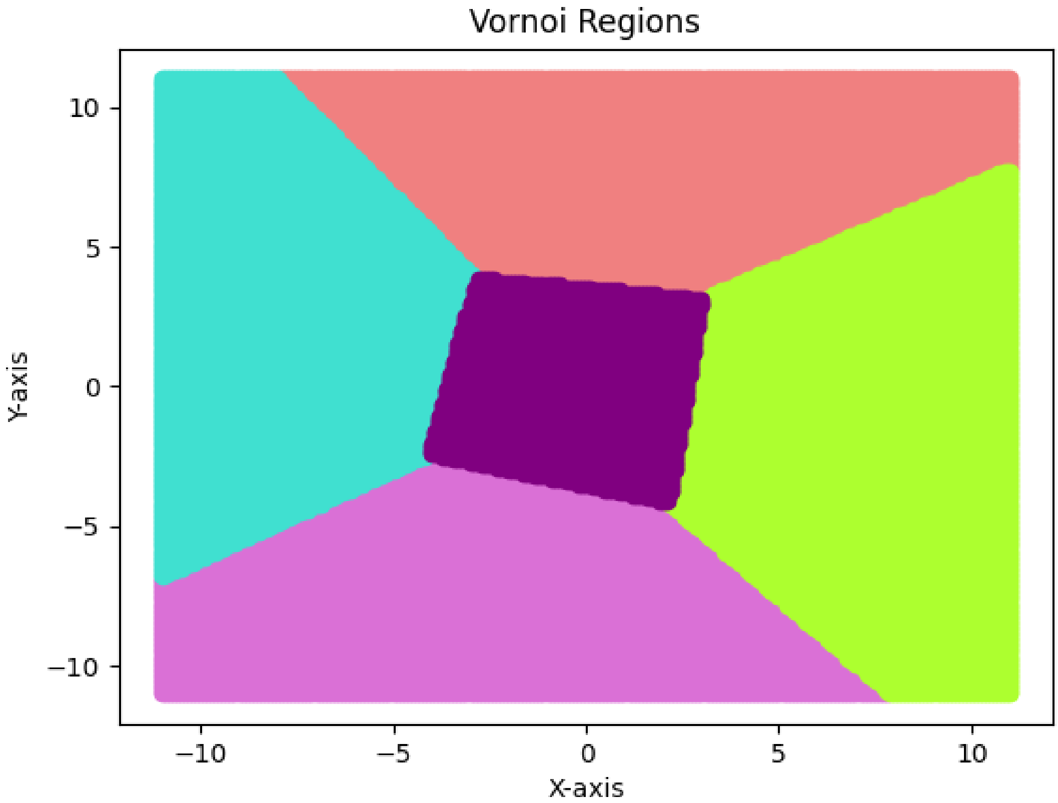
- **Clusters for a fixed random initialization with k=5:**



- **Error function w.r.t. Iterations for a fixed random initialization with k=5:**



- **Voronoi regions for a fixed random initialization with k=5:**

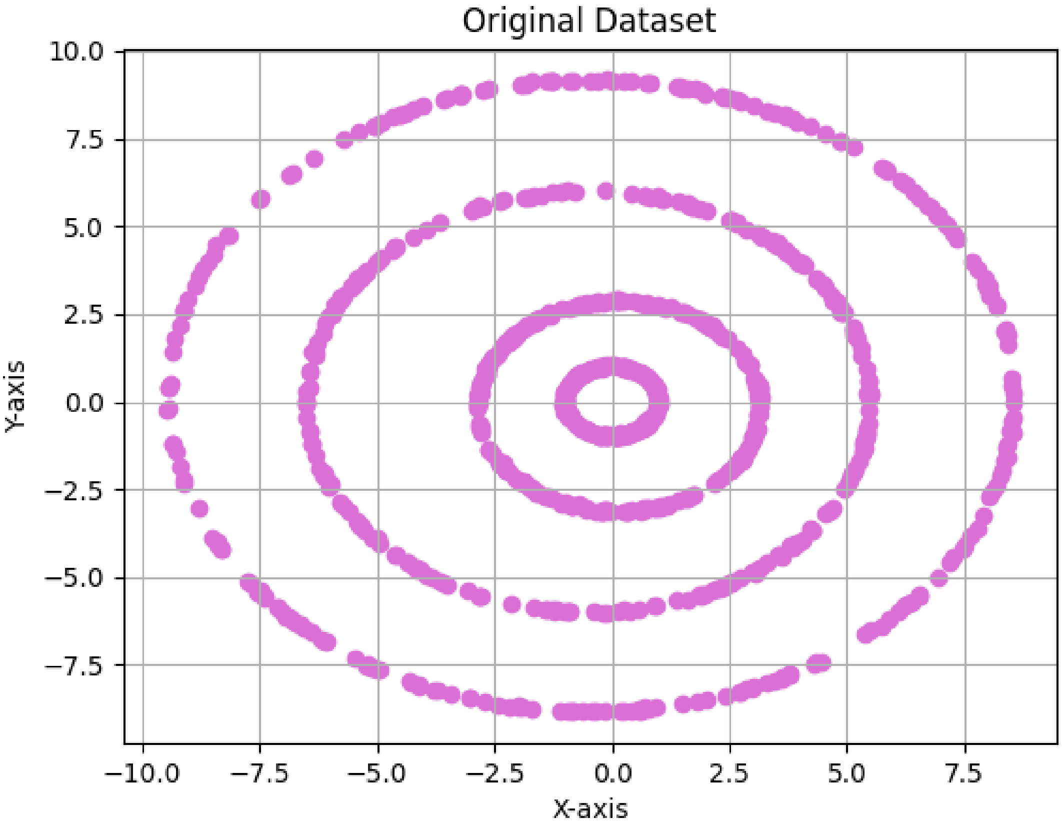


Q.2(iii)

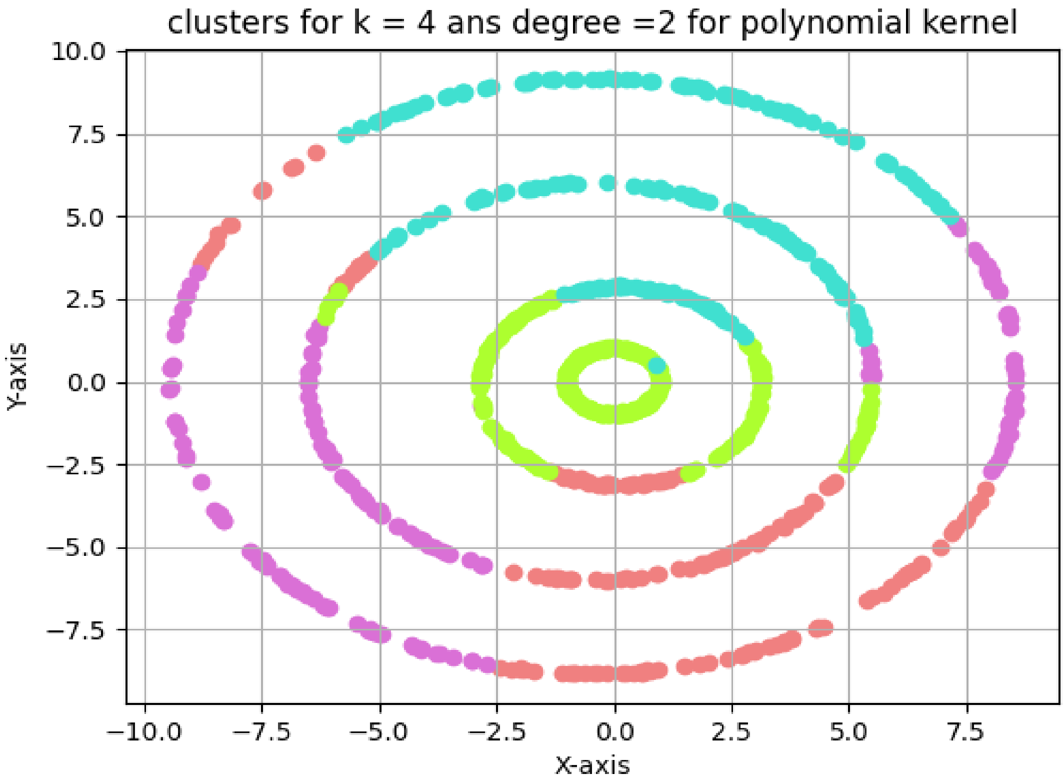
In spectral clustering we expect every ring to be a cluster but the polynomial and radial kernel functions used by me does not give the expected output.

The function closest to expected output is the polynomial kernel function with degree=2. So according to me the choice of kernel function would be polynomial kernel function with degree=2.

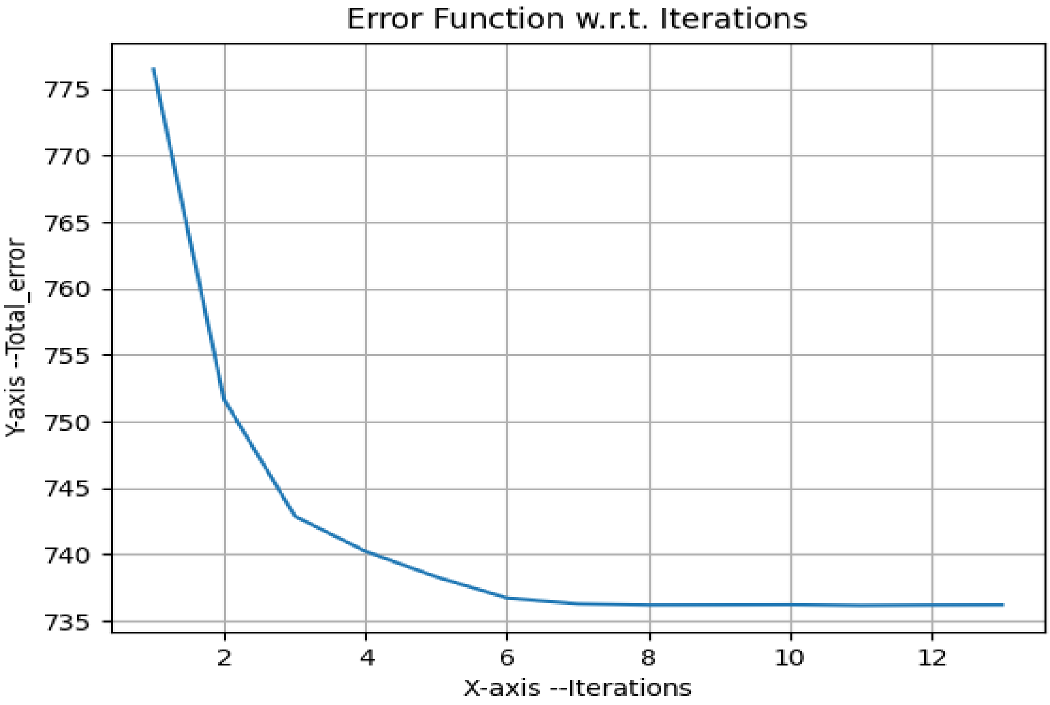
Original Dataset:



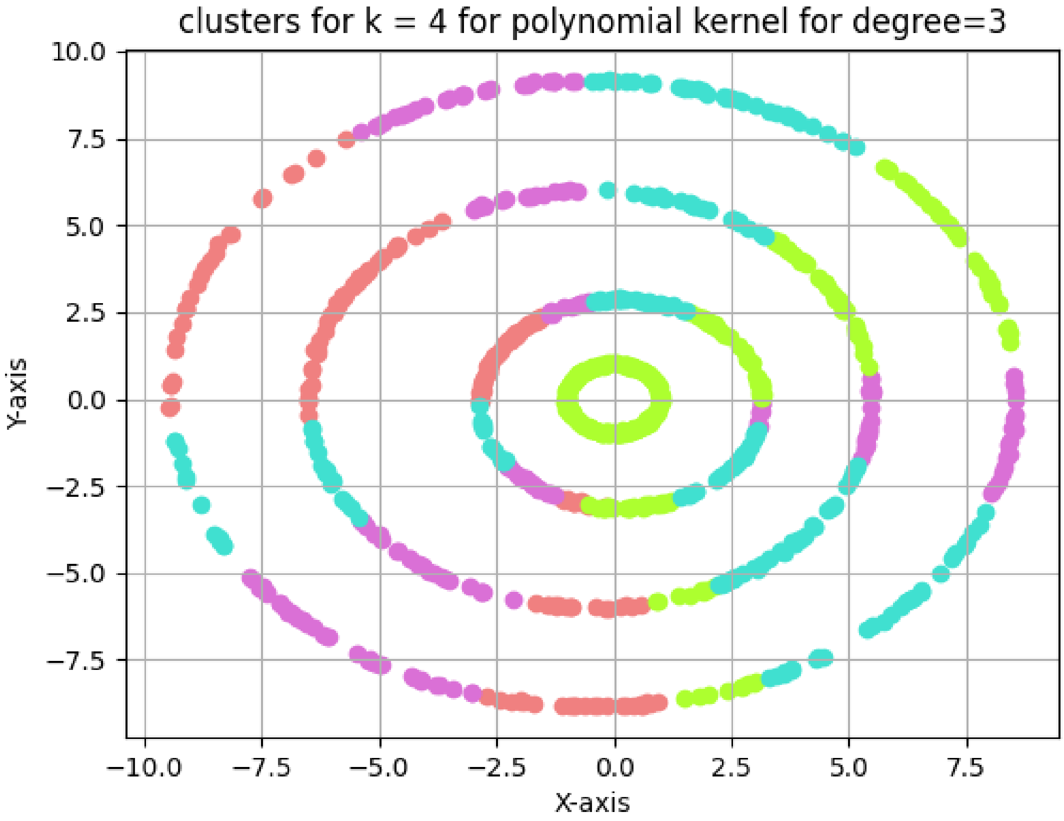
- **Clusters for polynomial kernel function with degree=2:**



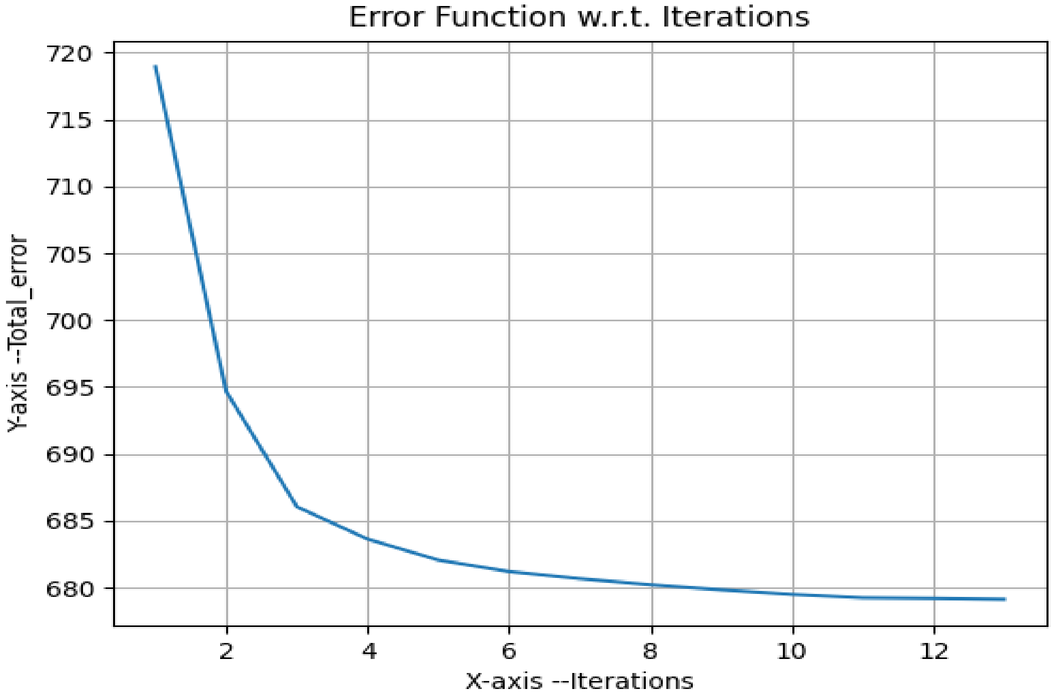
- **Error function w.r.t iterations for polynomial kernel function with degree=2:**



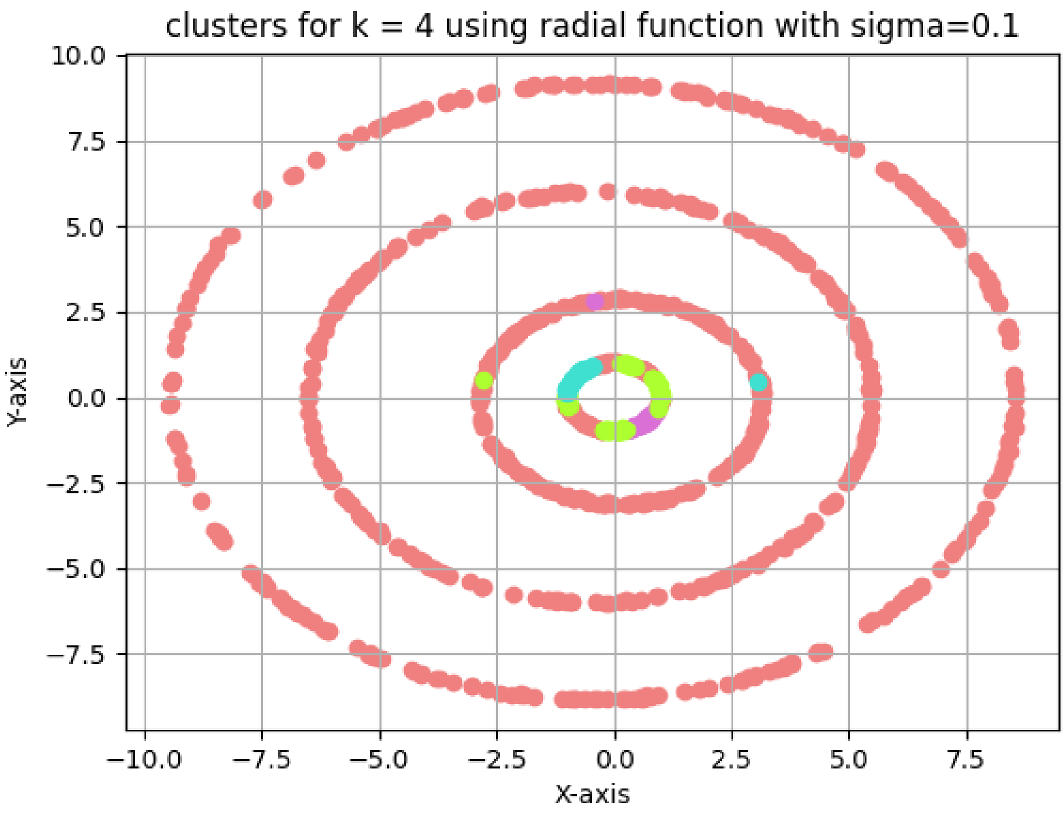
- **Clusters for polynomial kernel function with degree=3:**



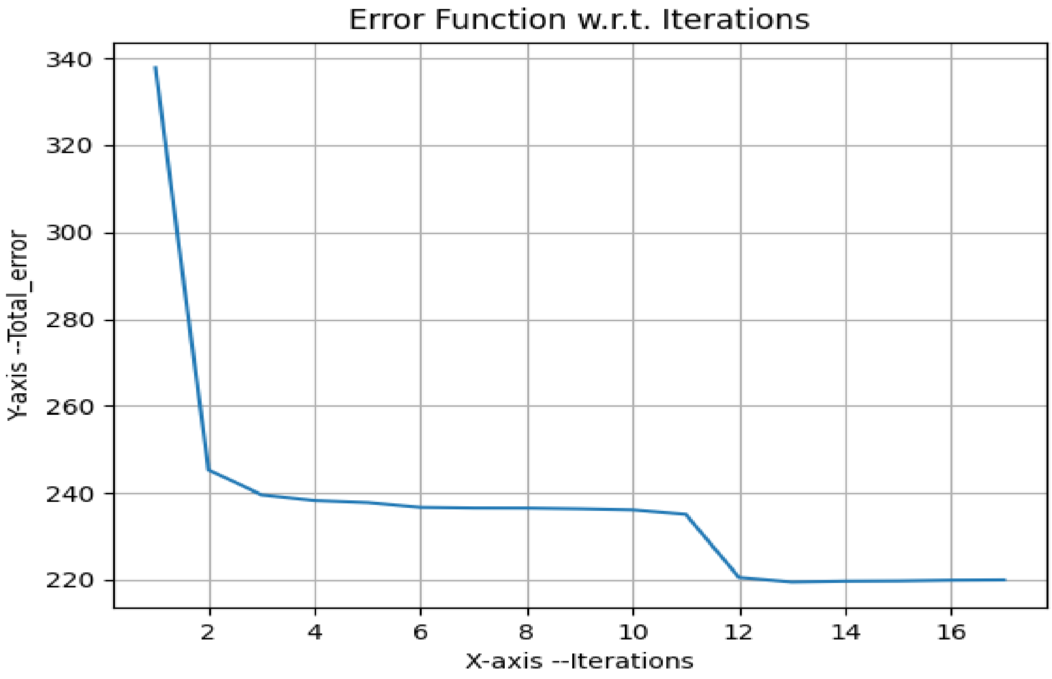
- **Error function w.r.t iterations for polynomial kernel function with degree=3:**



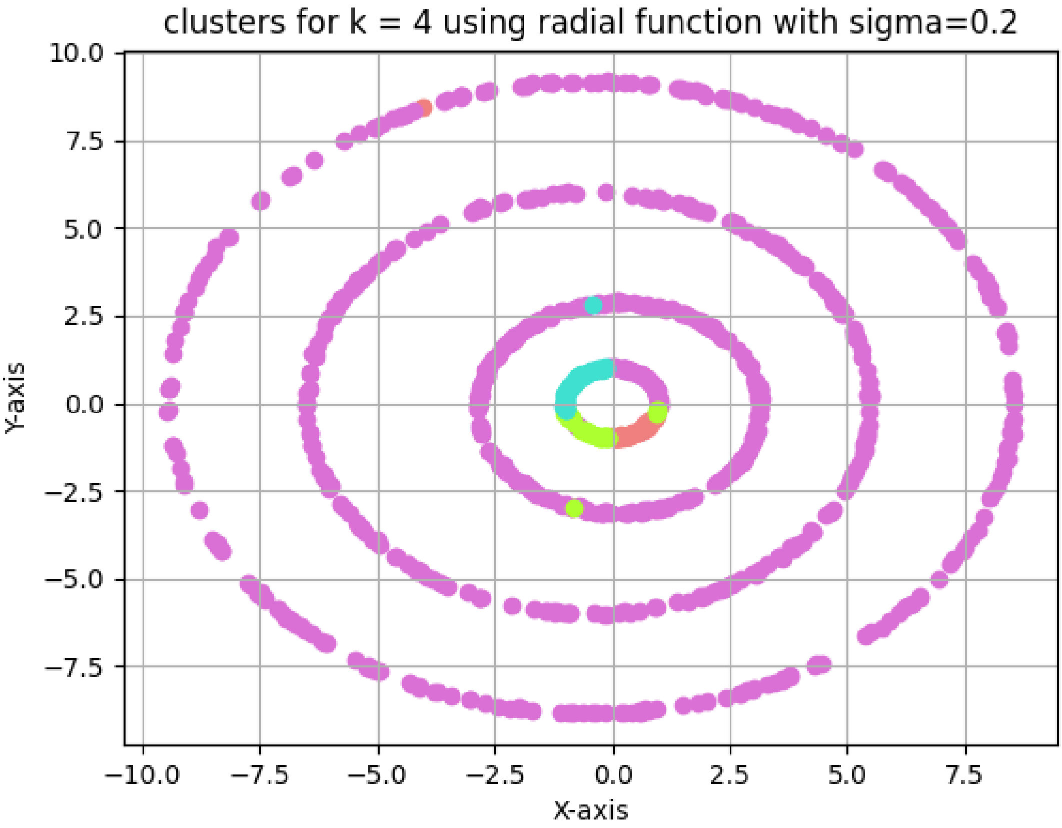
- **Clusters for radial basis kernel function with sigma=0.1:**



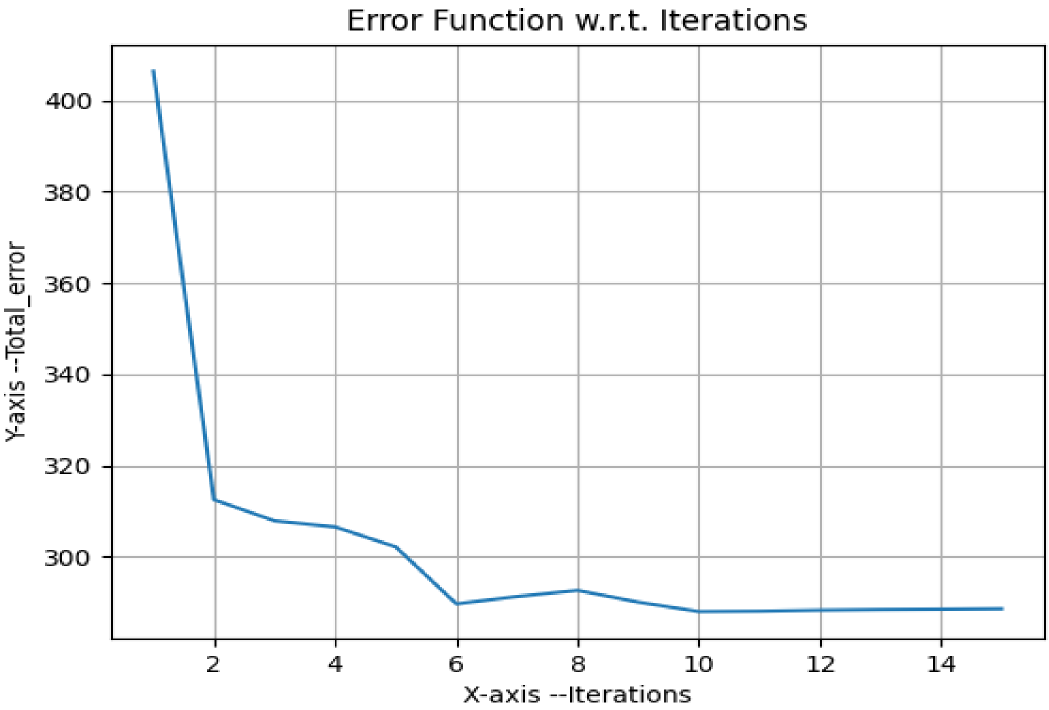
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.1:**



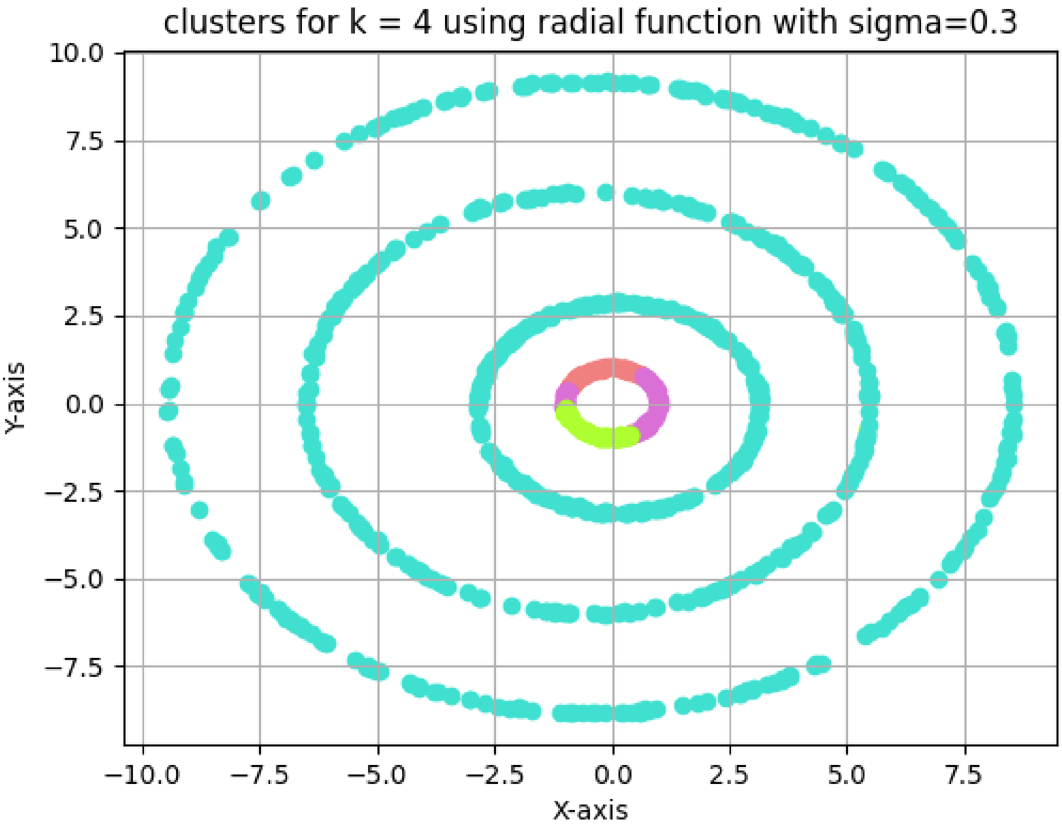
- **Clusters for radial basis kernel function with sigma=0.2:**



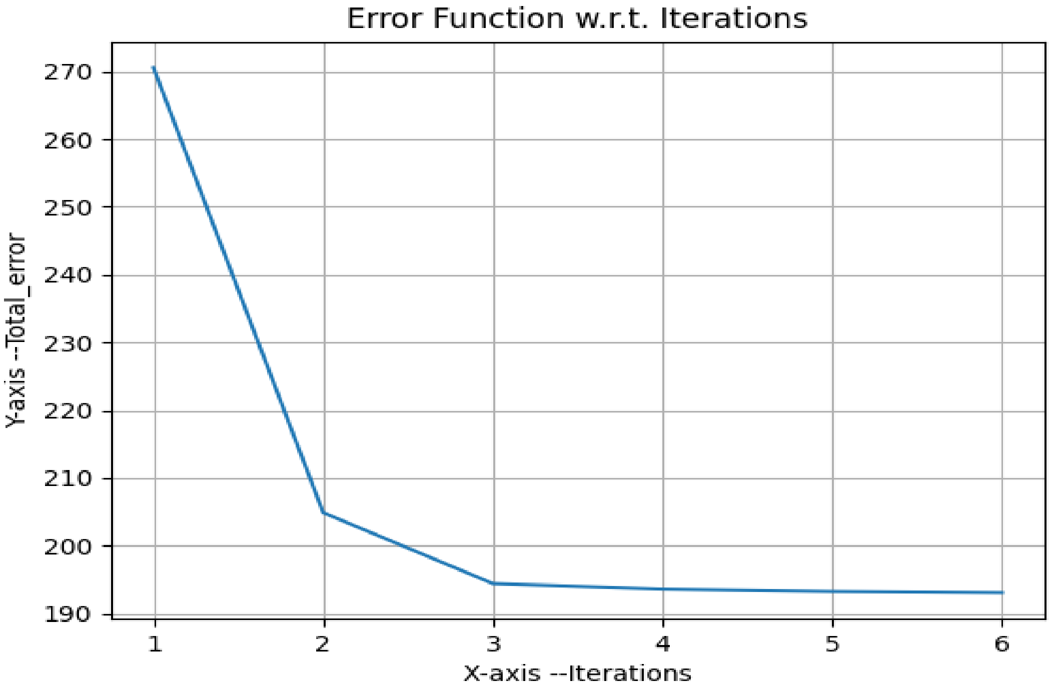
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.2:**



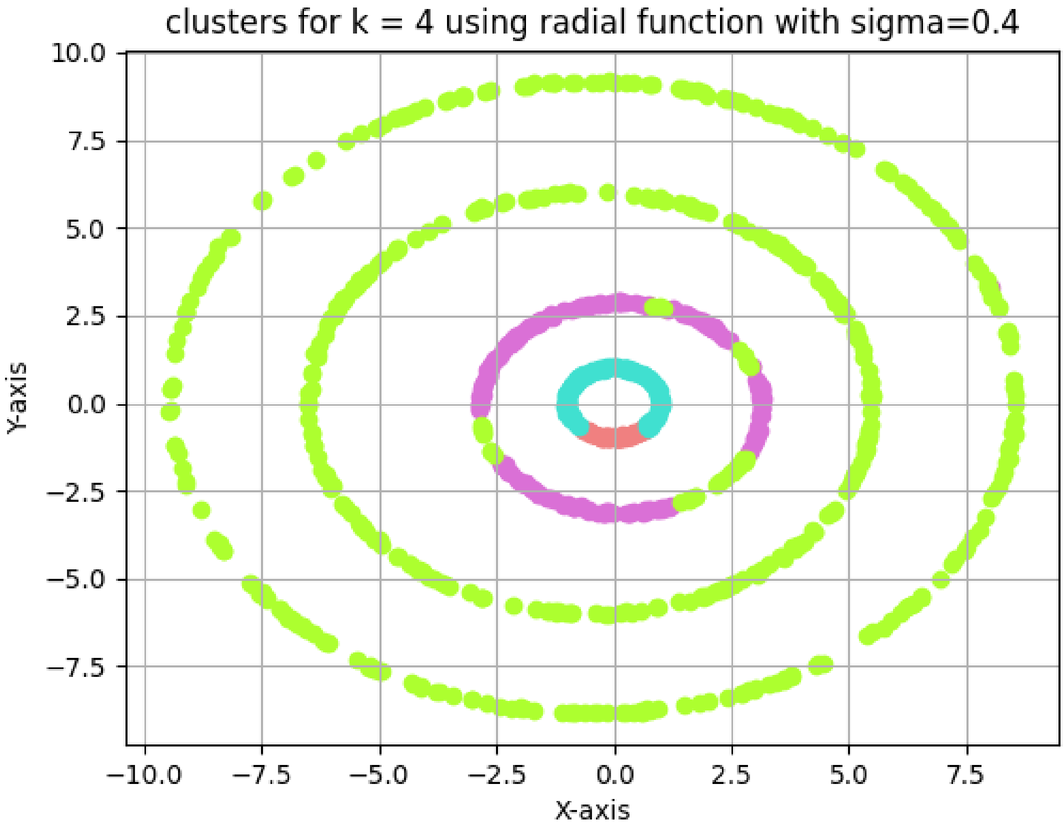
- **Clusters for radial basis kernel function with sigma=0.3:**



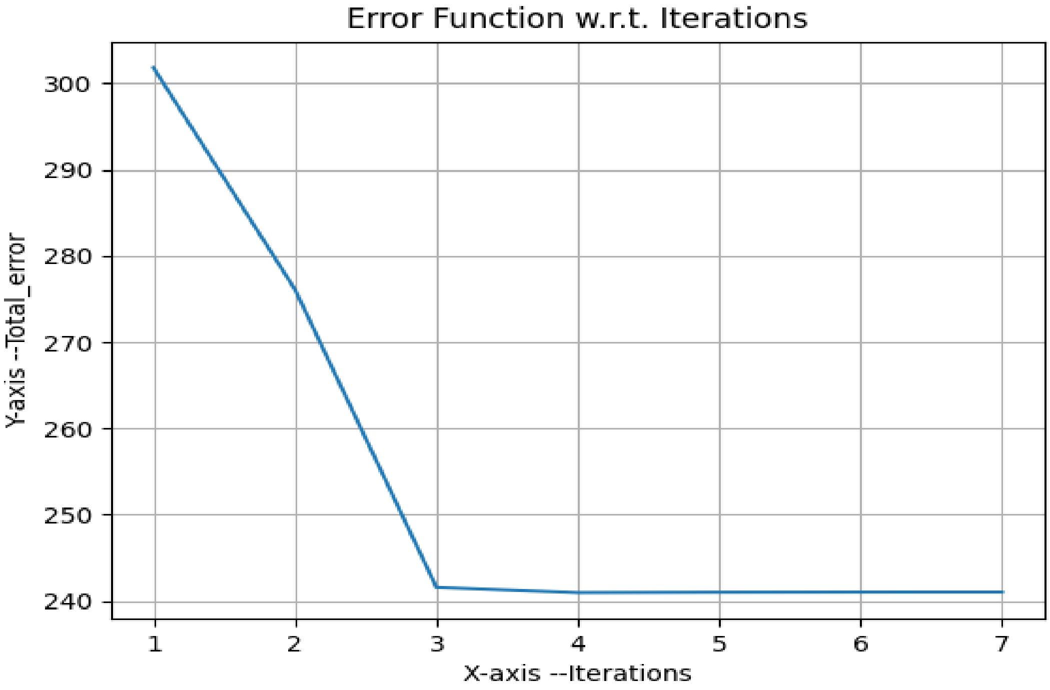
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.3:**



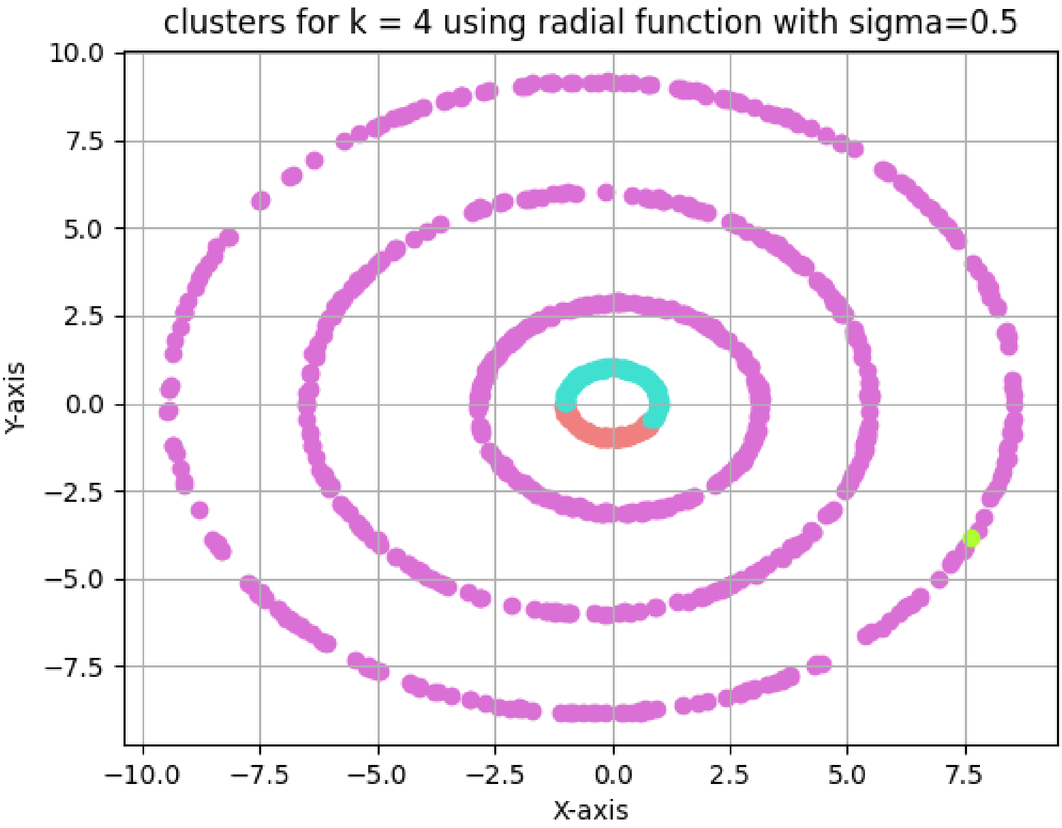
- **Clusters for radial basis kernel function with sigma=0.4:**



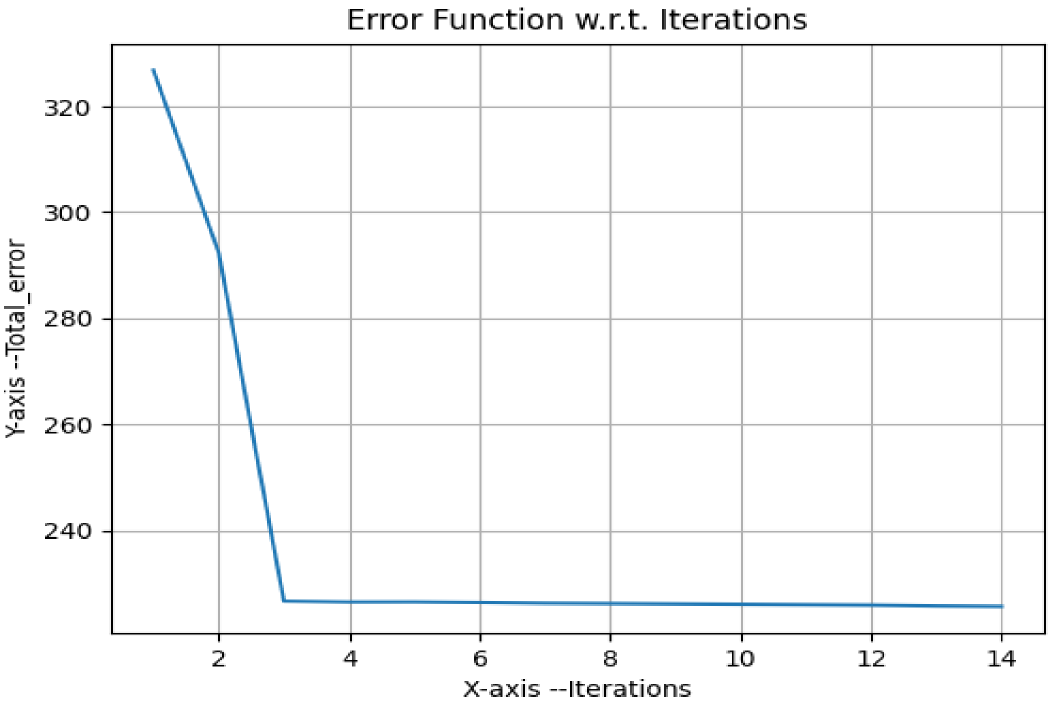
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.4:**



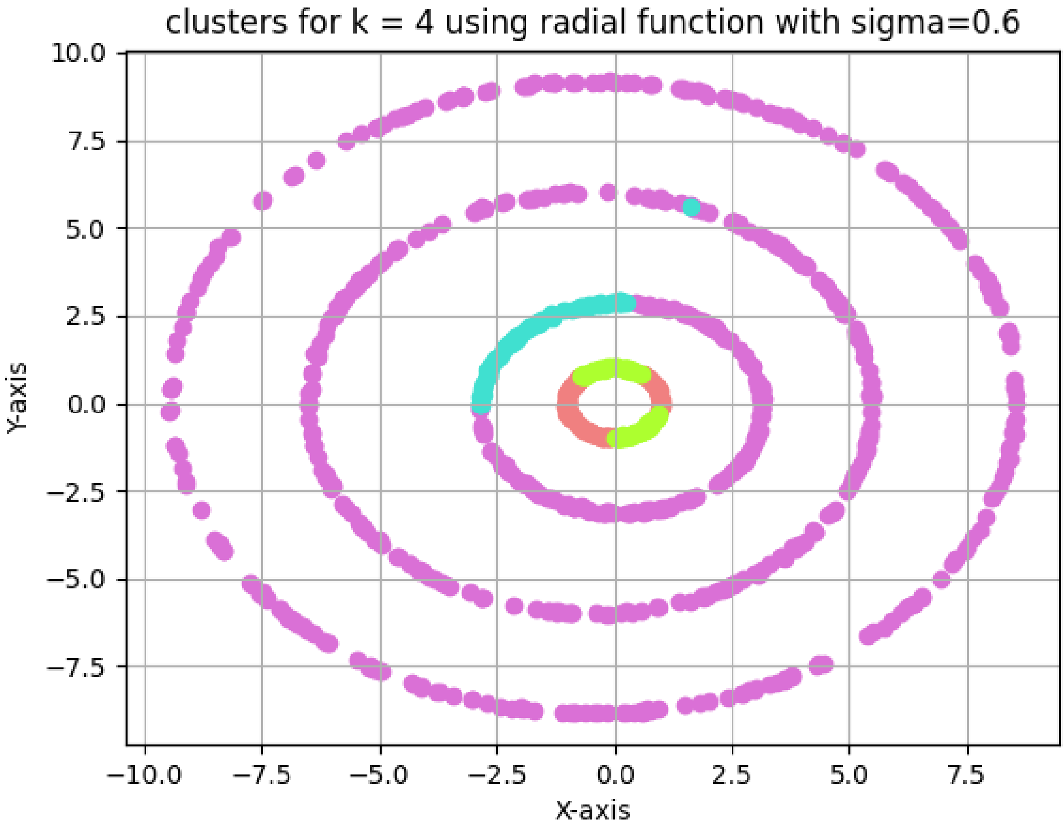
- **Clusters for radial basis kernel function with sigma=0.5:**



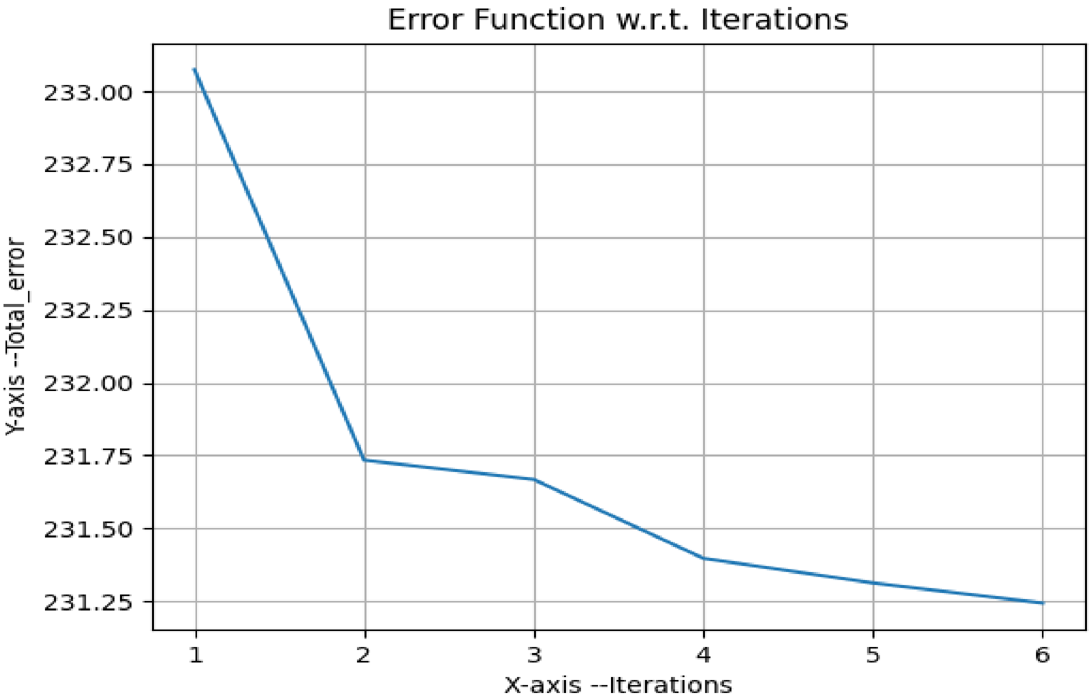
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.5:**



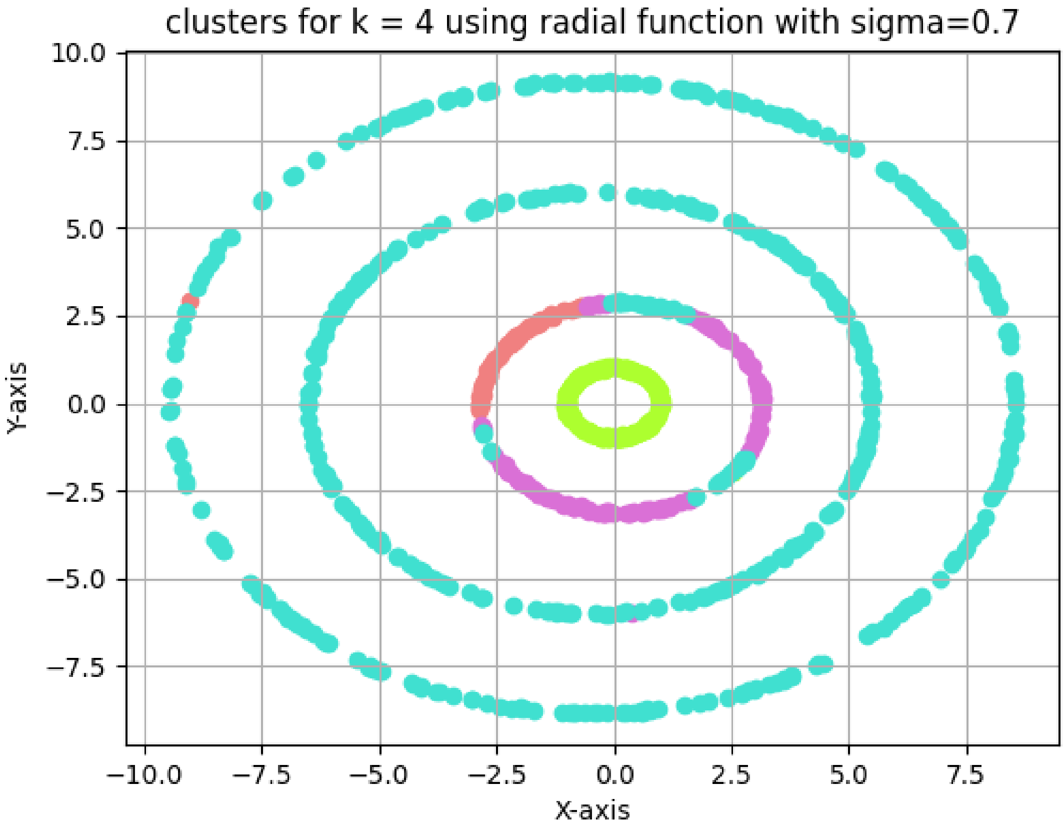
- **Clusters for radial basis kernel function with sigma=0.6:**



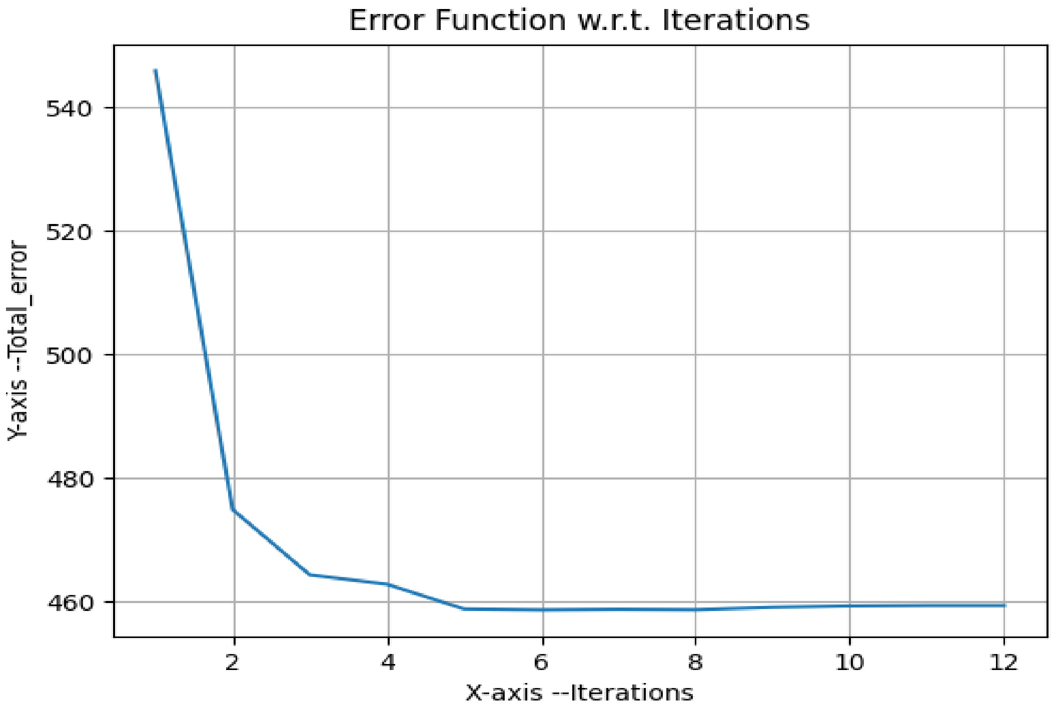
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.6:**



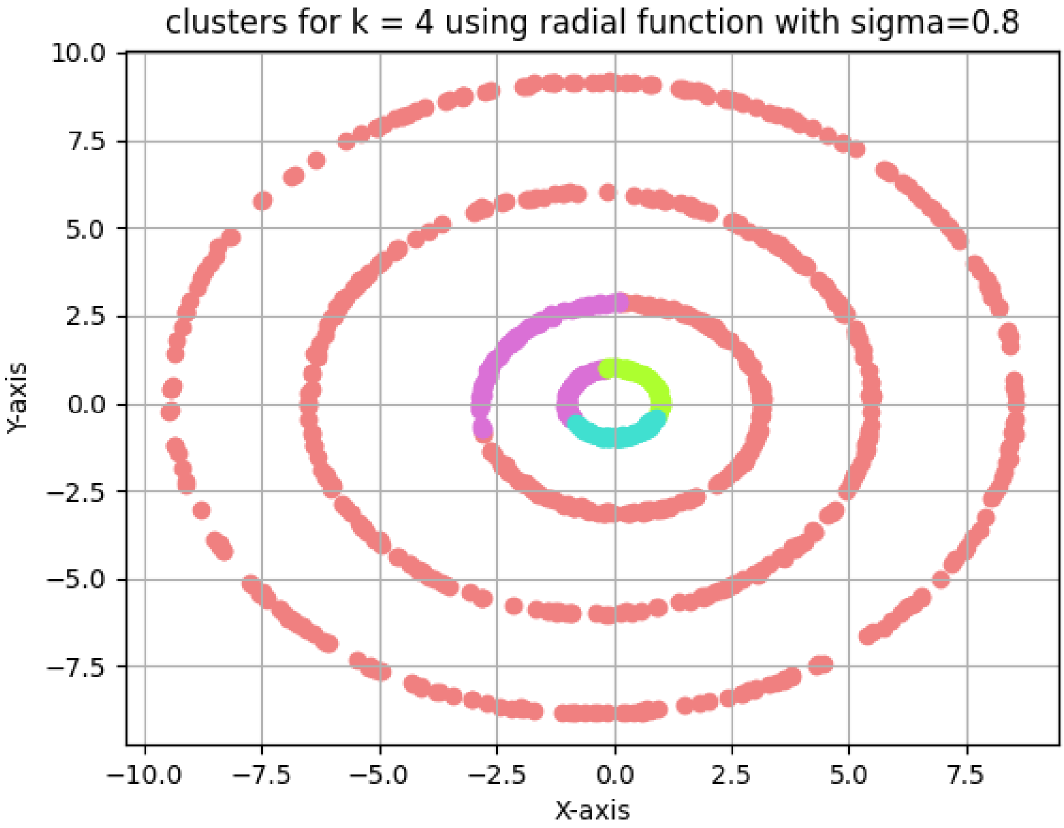
- **Clusters for radial basis kernel function with sigma=0.7:**



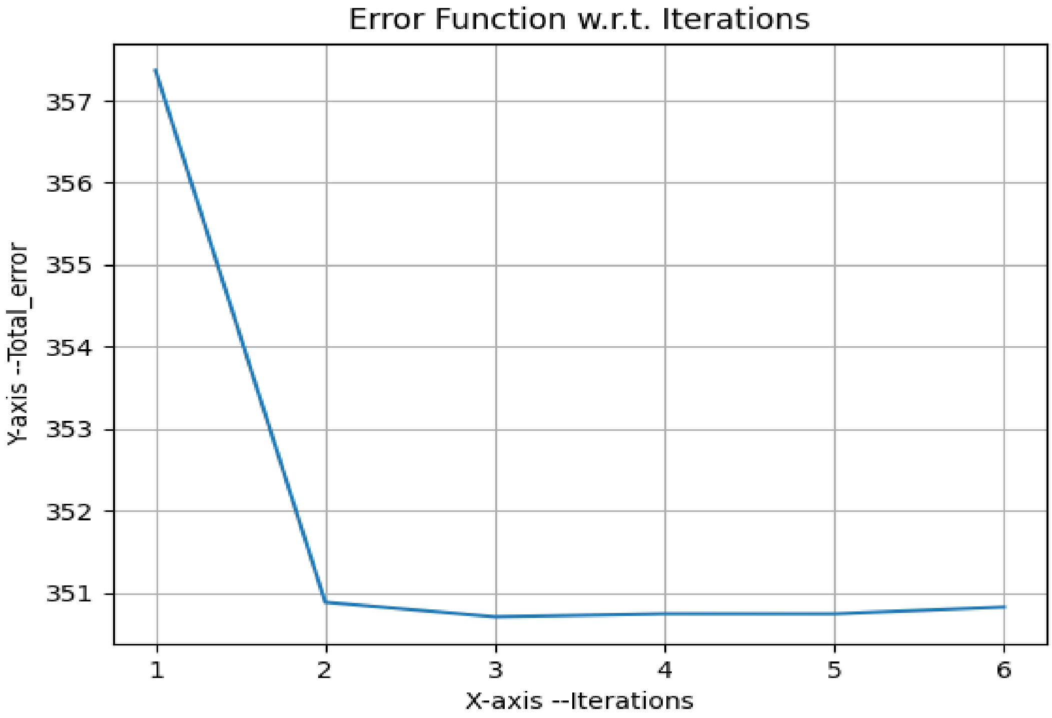
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.7:**



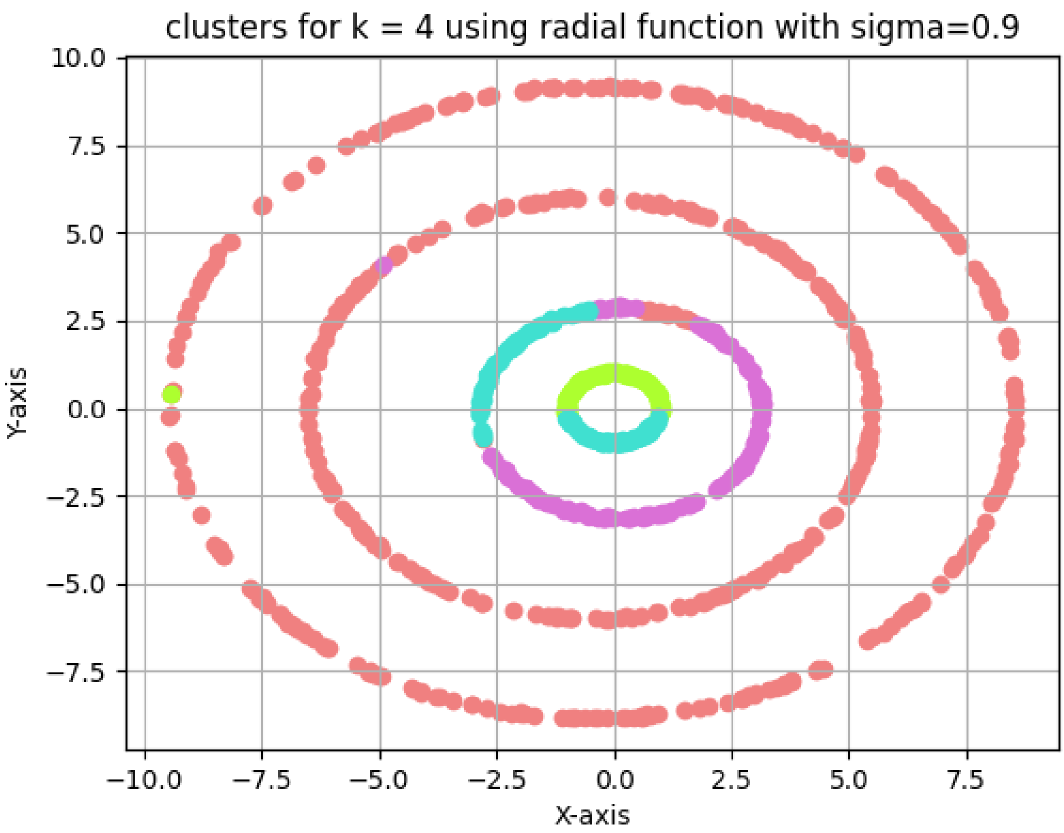
- **Clusters for radial basis kernel function with sigma=0.8:**



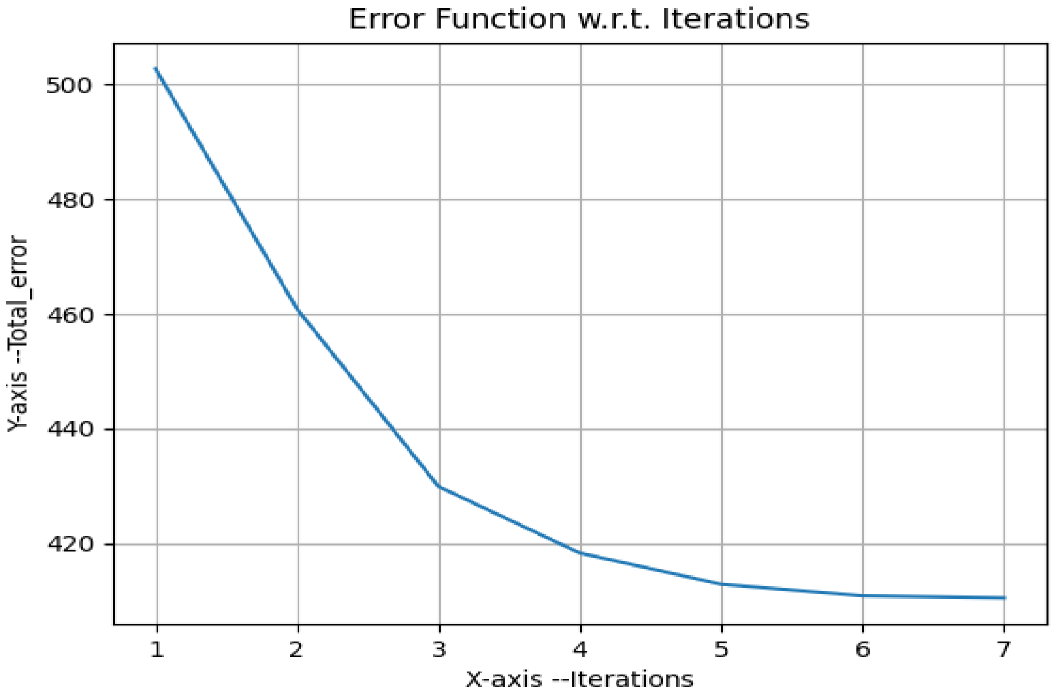
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.8:**



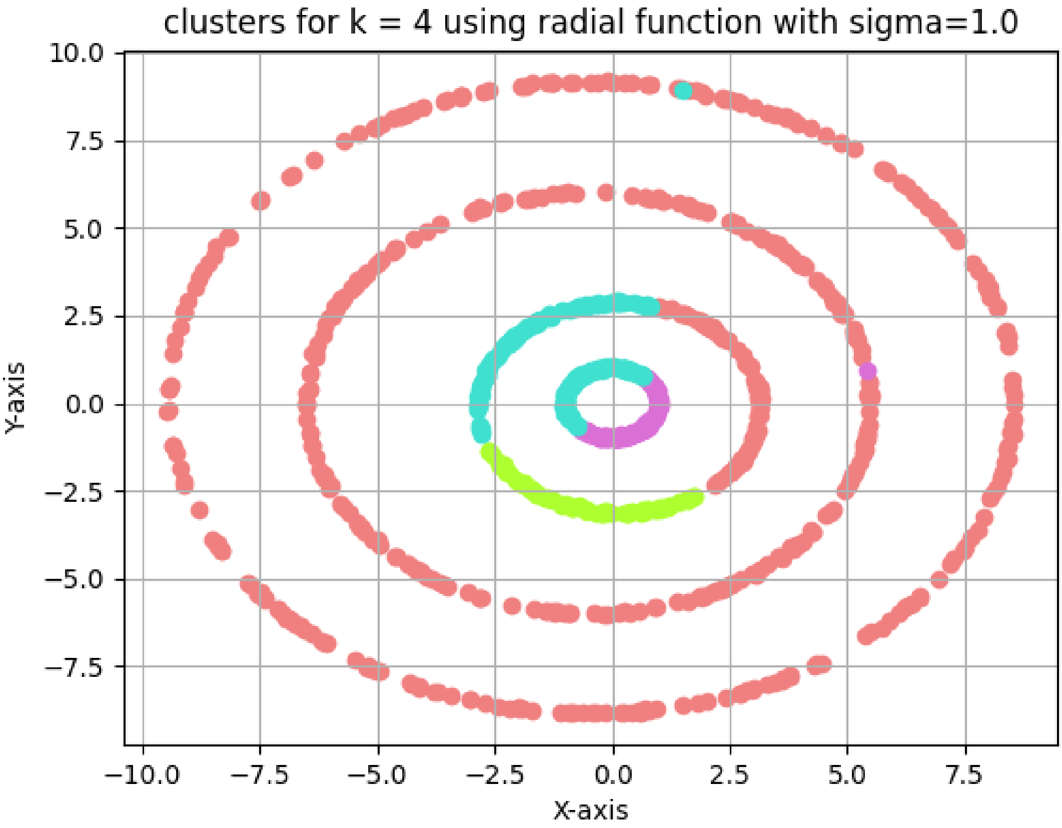
- **Clusters for radial basis kernel function with sigma=0.9:**



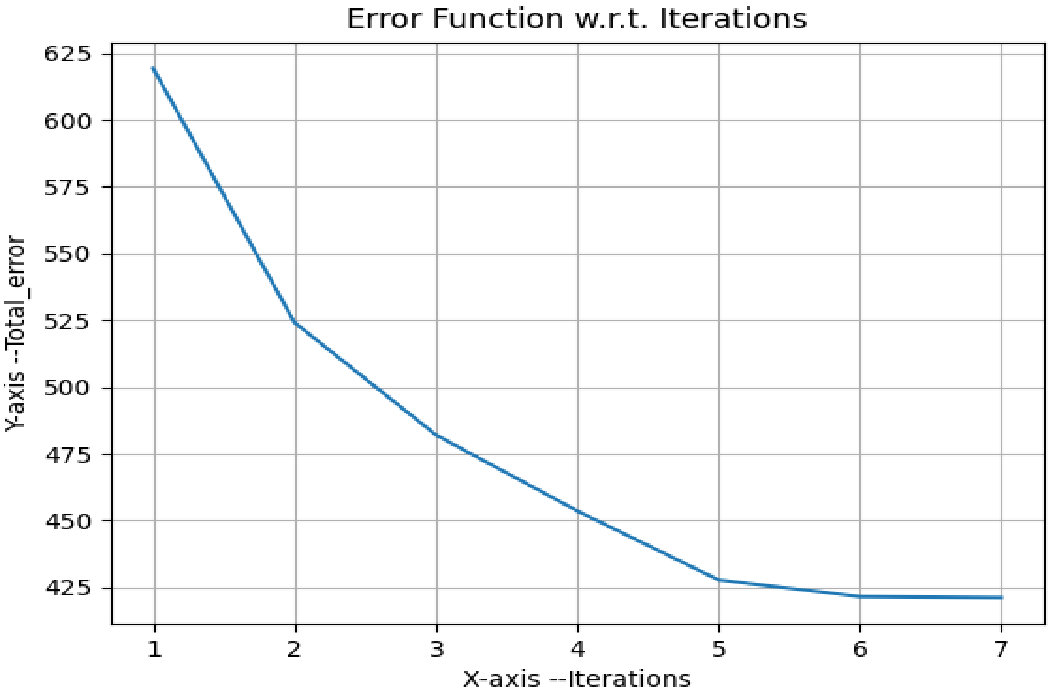
- **Error function w.r.t iterations for radial basis kernel function with sigma=0.9:**



- **Clusters for radial basis kernel function with sigma=1.0:**



- **Error function w.r.t iterations for radial basis kernel function with sigma=1.0:**

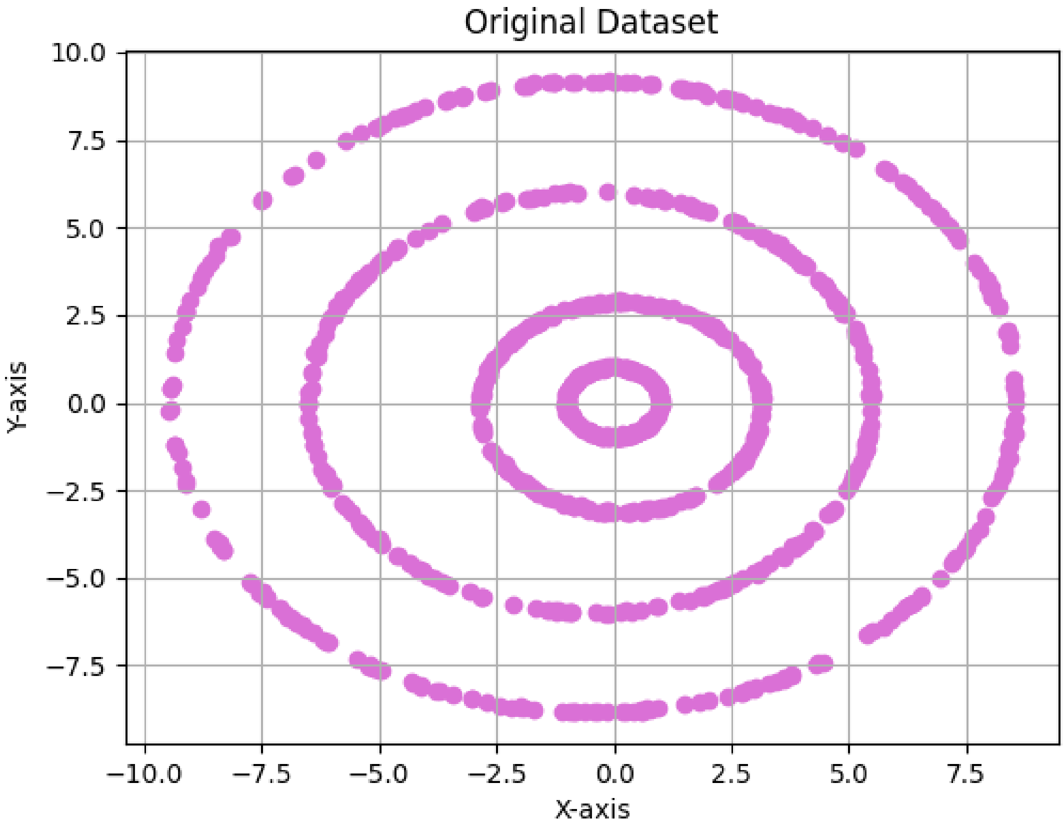


Q2(iv)

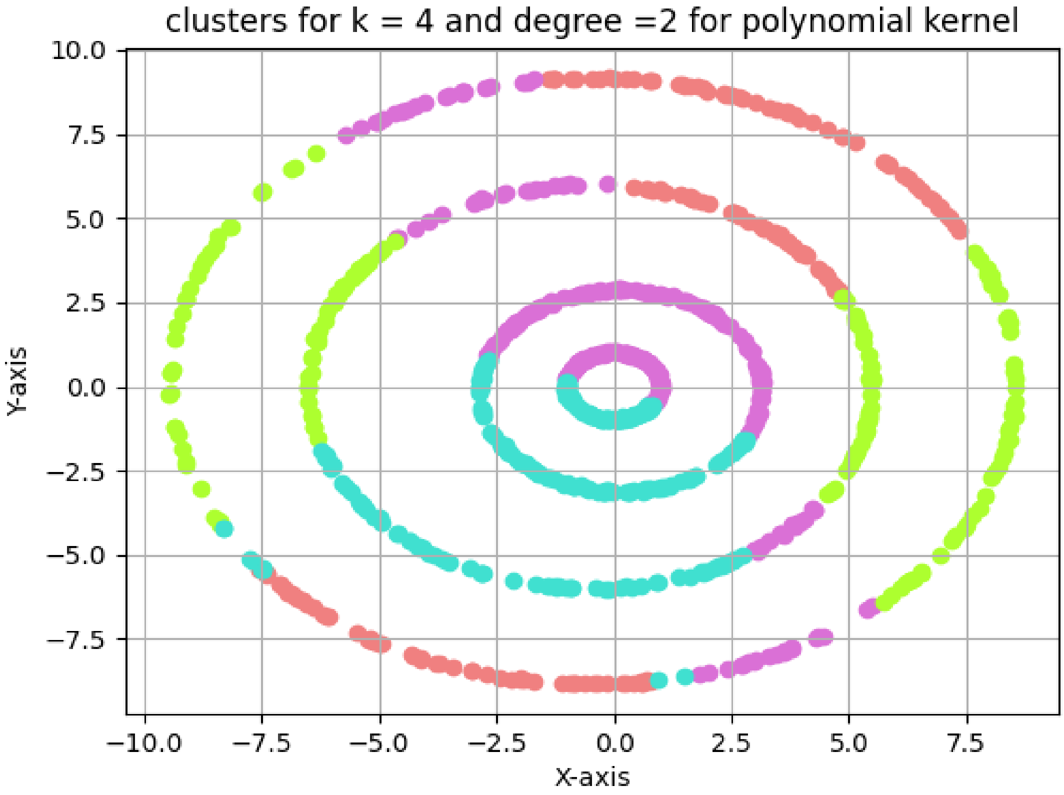
The above algorithm is not better than lloyd's algorithm using the normalised eigenvector approach as the data but it is better than k-means with random initialization.
The reason is that in the above algorithm we consider the H[1000][k] matrix that contains the top k eigenvectors of K and it chooses cluster based on the maximum value in a particular row considering it to be the most probable cluster to which datapoint belongs instead of randomly initialising clusters

If a datapoint has maximum H[n][k] value then that datapoint n belongs to kth cluster.

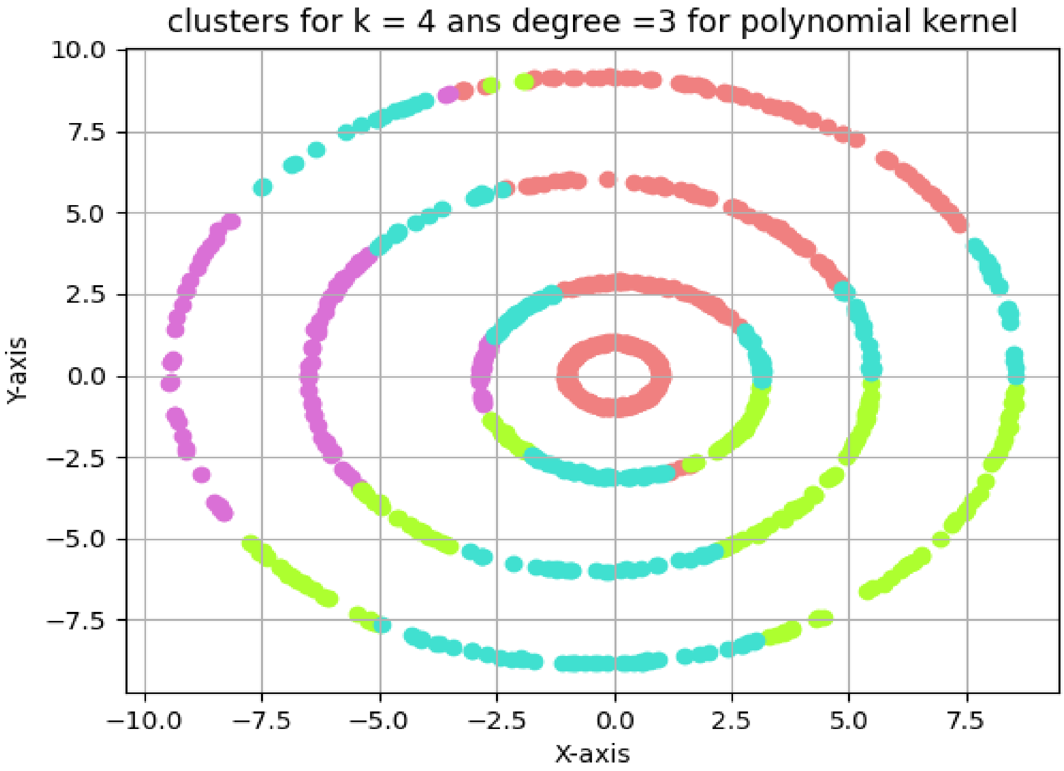
Original Dataset:



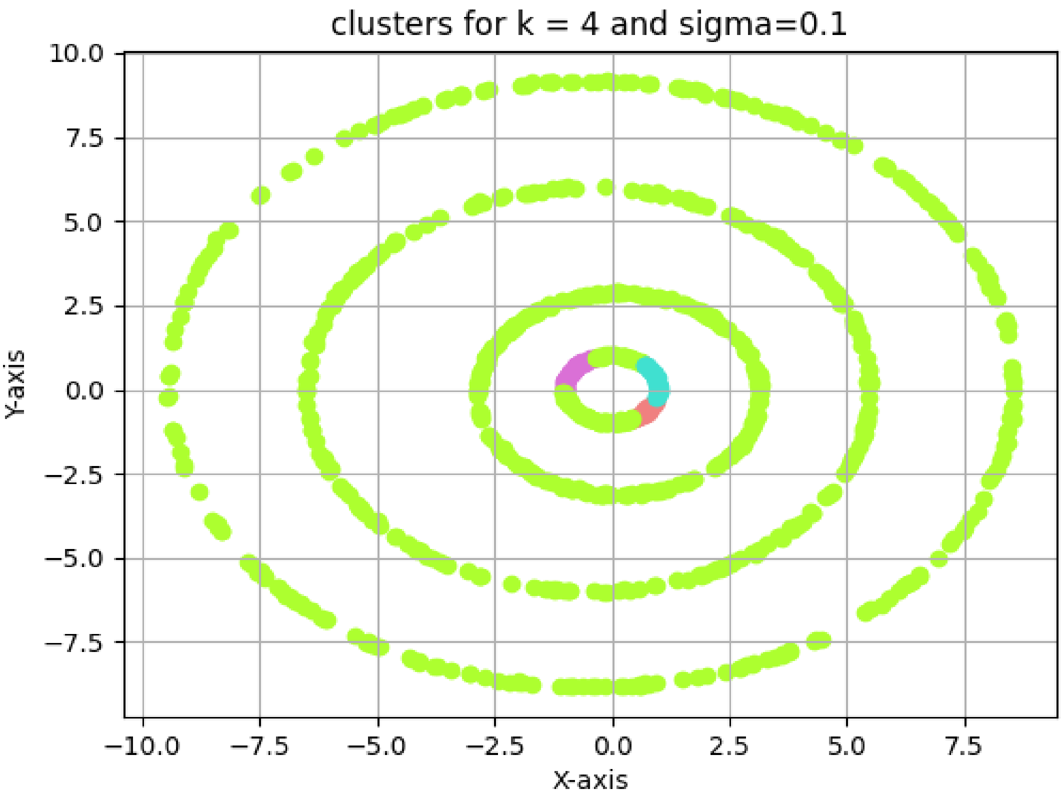
- **Clusters for polynomial kernel function with degree=2:**



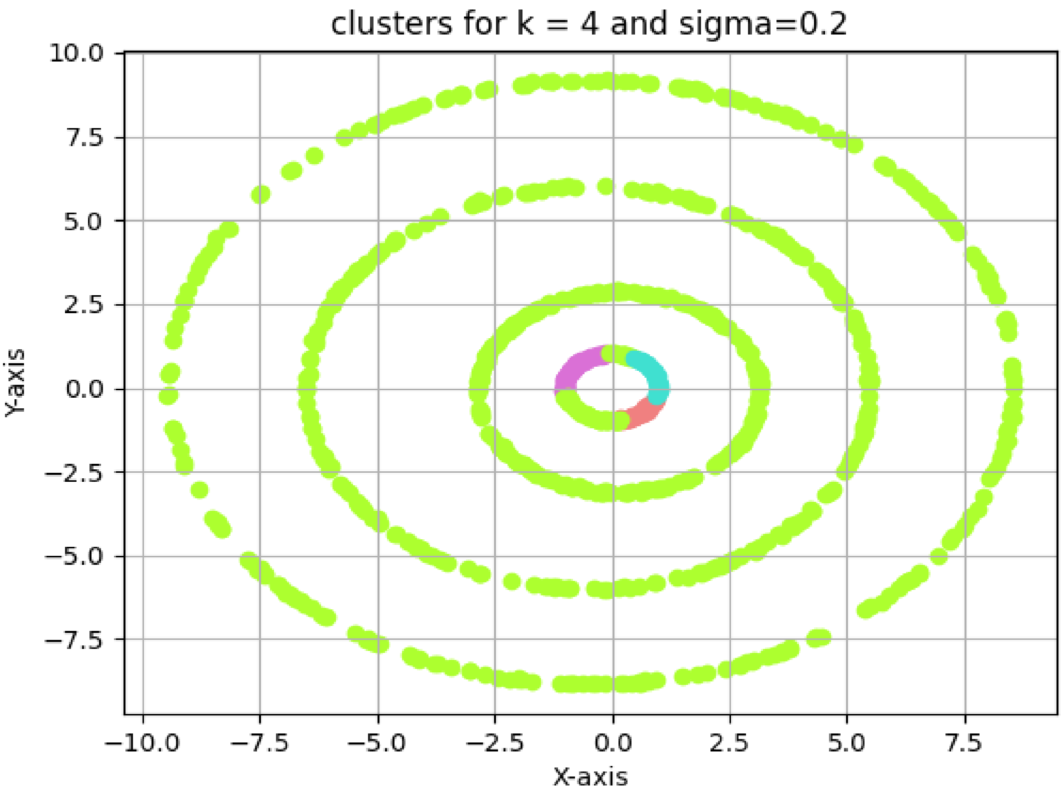
- **Clusters for polynomial kernel function with degree=3:**



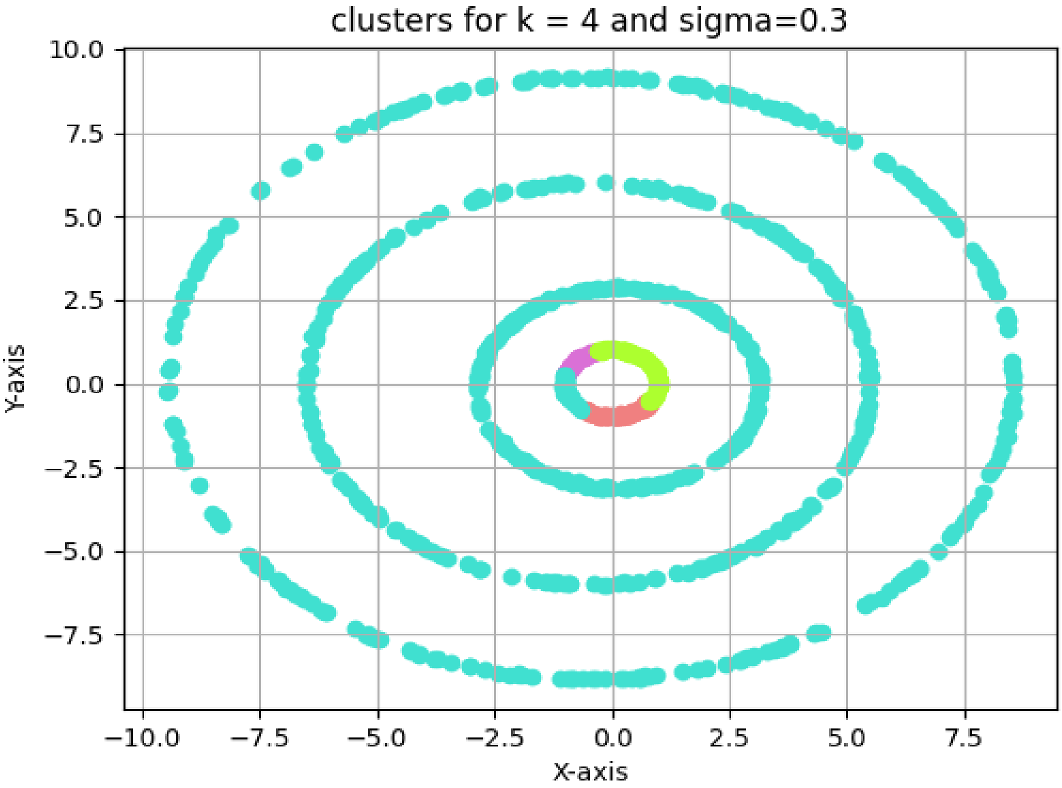
- **Clusters for radial basis kernel function with sigma=0.1:**



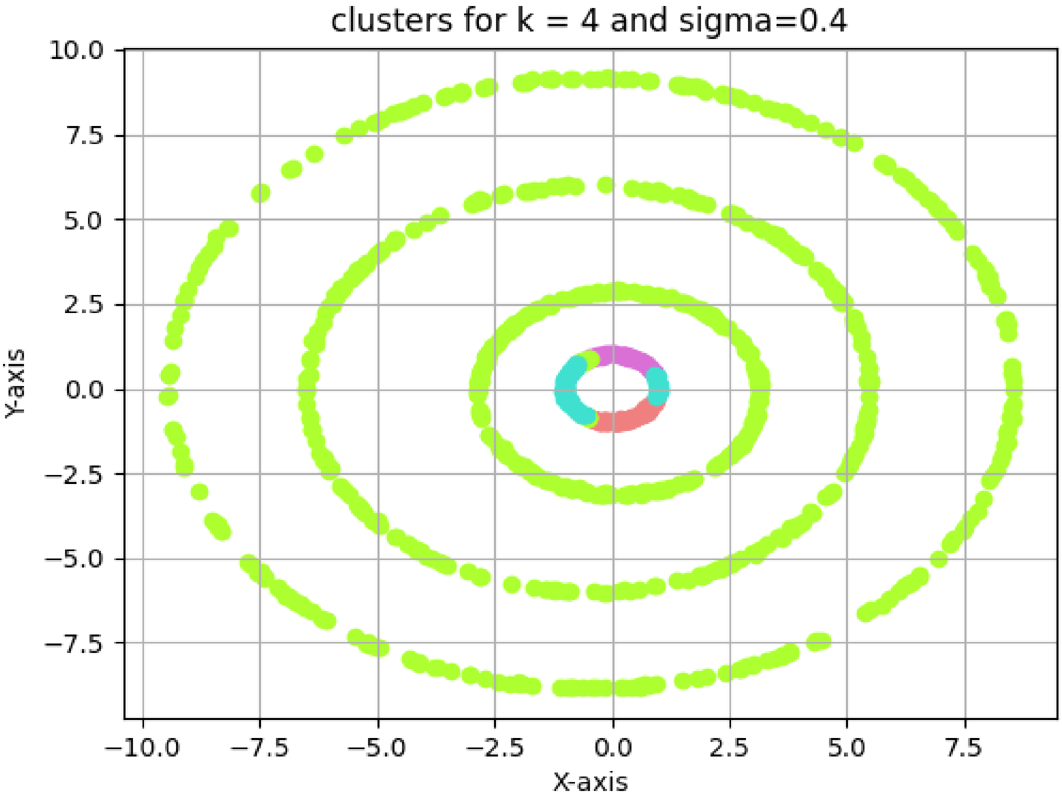
- **Clusters for radial basis kernel function with sigma=0.2:**



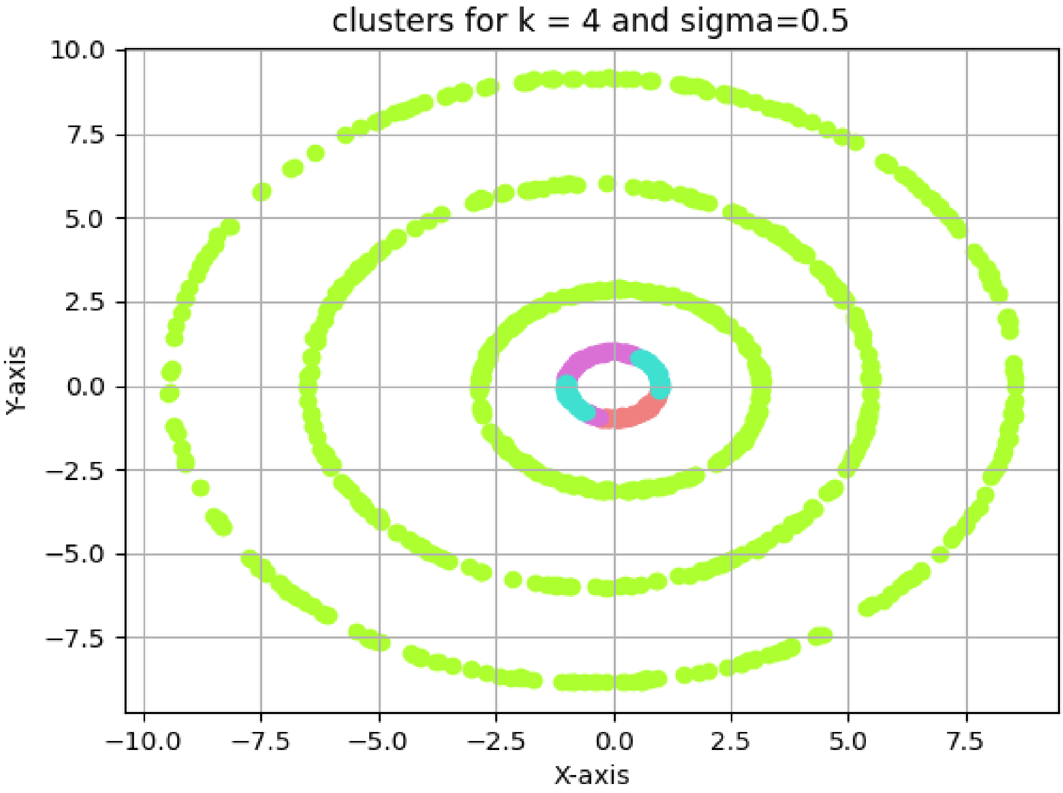
- **Clusters for radial basis kernel function with sigma=0.3:**



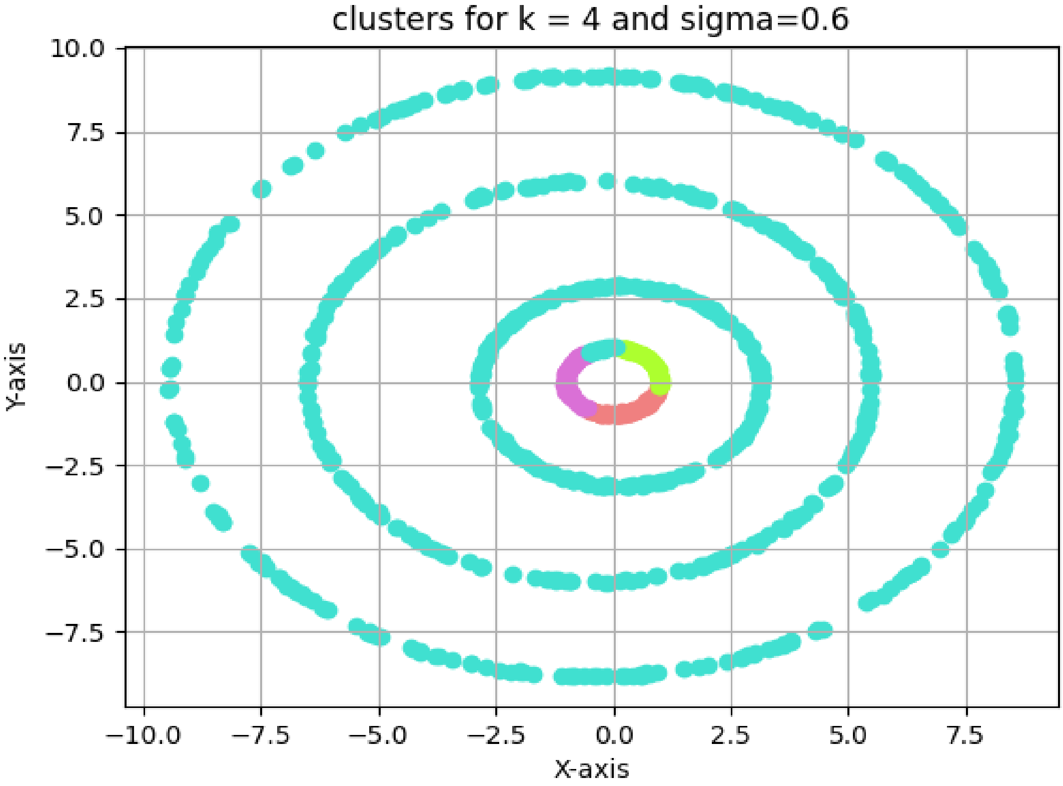
- **Clusters for radial basis kernel function with sigma=0.4:**



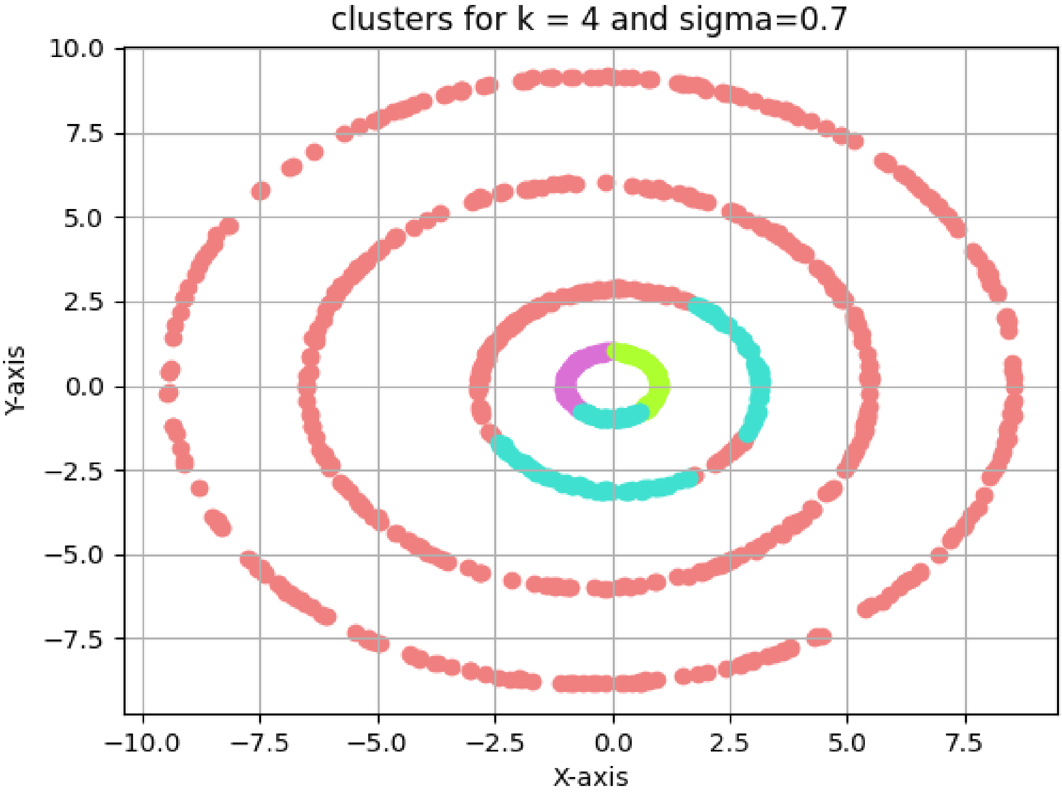
- **Clusters for radial basis kernel function with sigma=0.5:**



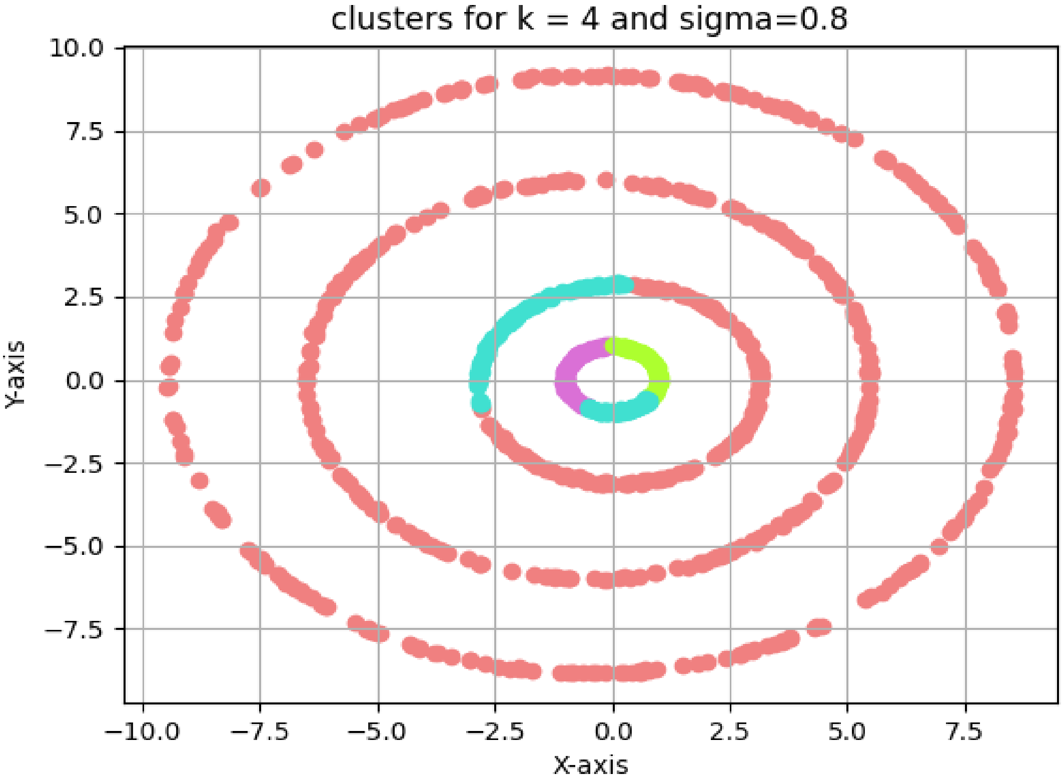
- **Clusters for radial basis kernel function with sigma=0.6:**



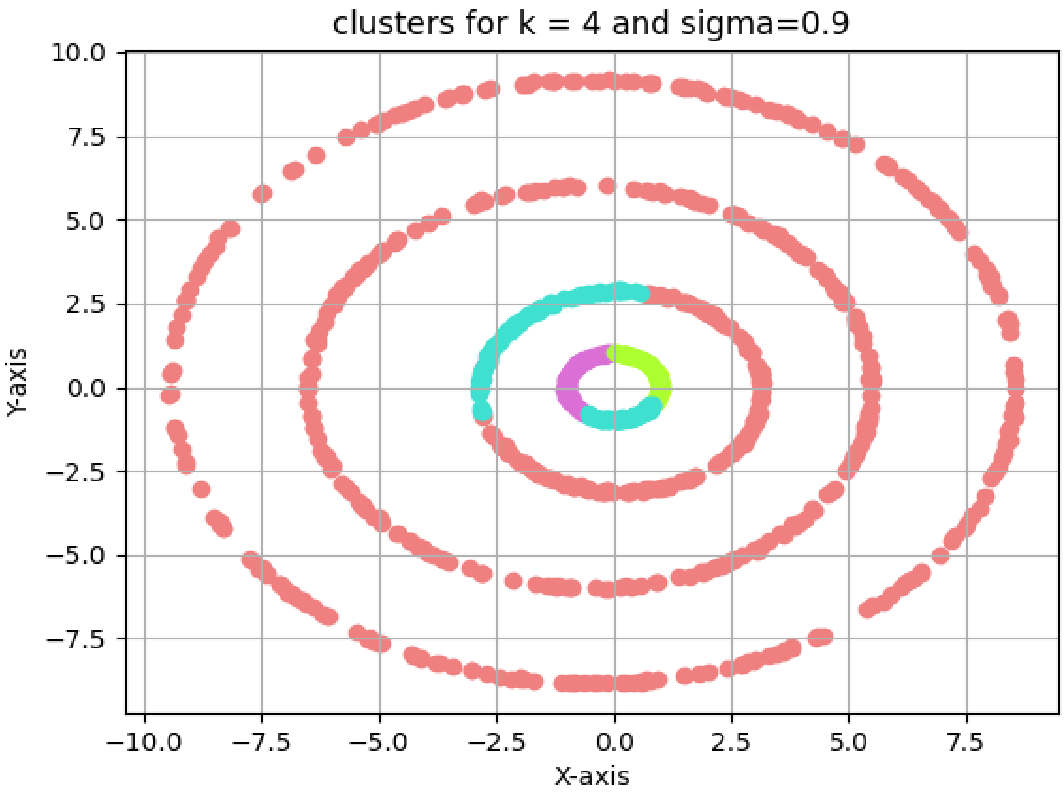
- **Clusters for radial basis kernel function with sigma=0.7:**



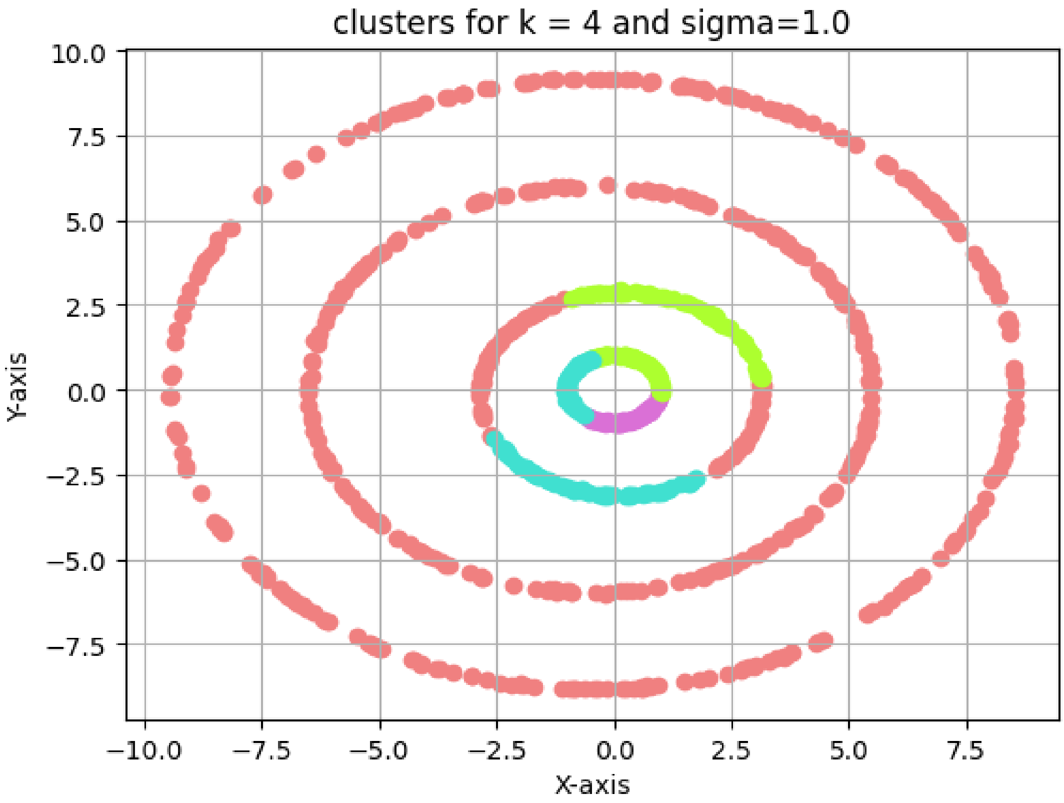
- **Clusters for radial basis kernel function with sigma=0.8:**



- **Clusters for radial basis kernel function with sigma=0.9:**



- **Clusters for radial basis kernel function with sigma=1.0:**


clusters for k = 4 and sigma=1.0
Y-axis
X-axis