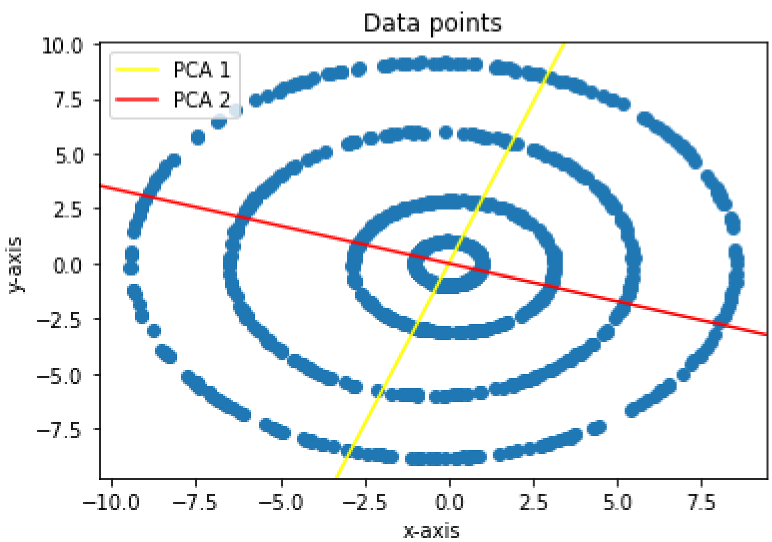
CS5691: Pattern Recognition and Machine Learning
Assignment 1
Course Instructor : Arun Rajkumar.

Name: Amishi Panwar
Roll No. : CS22M009

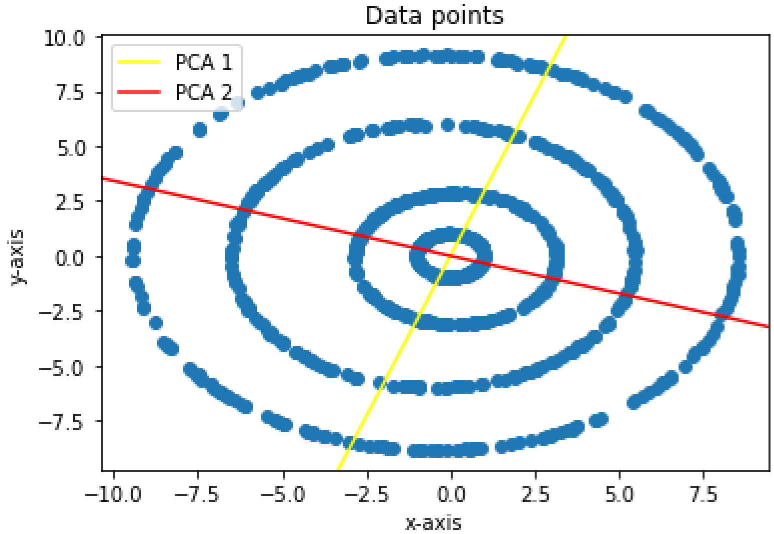
Que. 1 You are given a data-set with 1000 data points each in R2 .
i. Write a piece of code to run the PCA algorithm on this data-set. How much of the variance in the data-set is explained by each of the principal components?



Variance along principal component 1: 54.17802452885222
Variance along principal component 2: 45.82197547114777

- Eigen vectors of covariance matrix gives principal components in PCA
- Eigen values shows the variance along that principal component.
- Percentage of variance explained by principal components = eigen value of that component / sum(all eigen values)
- As we see data is not lineary distributed normal PCA failed to represent it and it might not recover original data points from projected points, therefore we'll use kernel pca.

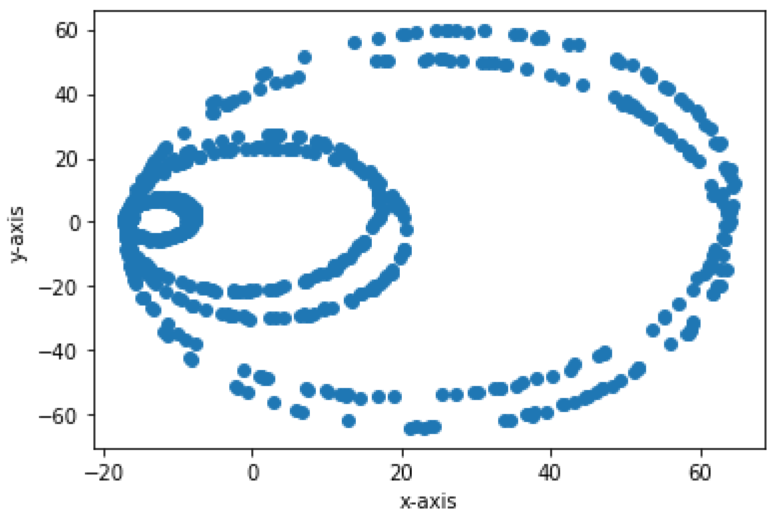
ii. Study the effect of running PCA without centering the data-set. What are your observations? Does Centering help?



As the mean of data i.e. [4.075e-07 2.227e-07] is very close to [0,0], there is very little difference between data centered and original data, therefore principal components are the same. Centering is not that useful in this dataset.

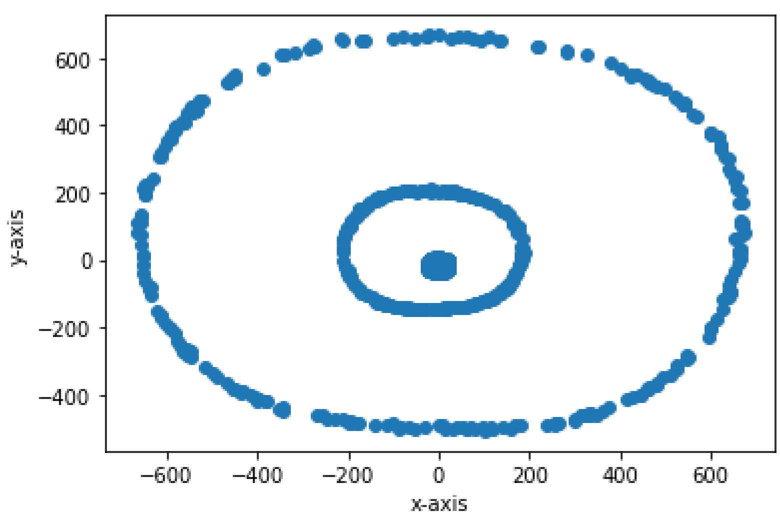
iii. Write a piece of code to implement the Kernel PCA algorithm on this dataset. Use the following kernels :
A. Polynomial kernel (d = 2,3)
B. Radial Basis Kernel (σ = 0.1, 0.2, 0.3….1)
Plot the projection of each point in the dataset onto the top-2 components for each kernel. Use one plot for each kernel and in the case of (B), use a different plot for each value of σ.

A (i) Polynomial kernel (d = 2)



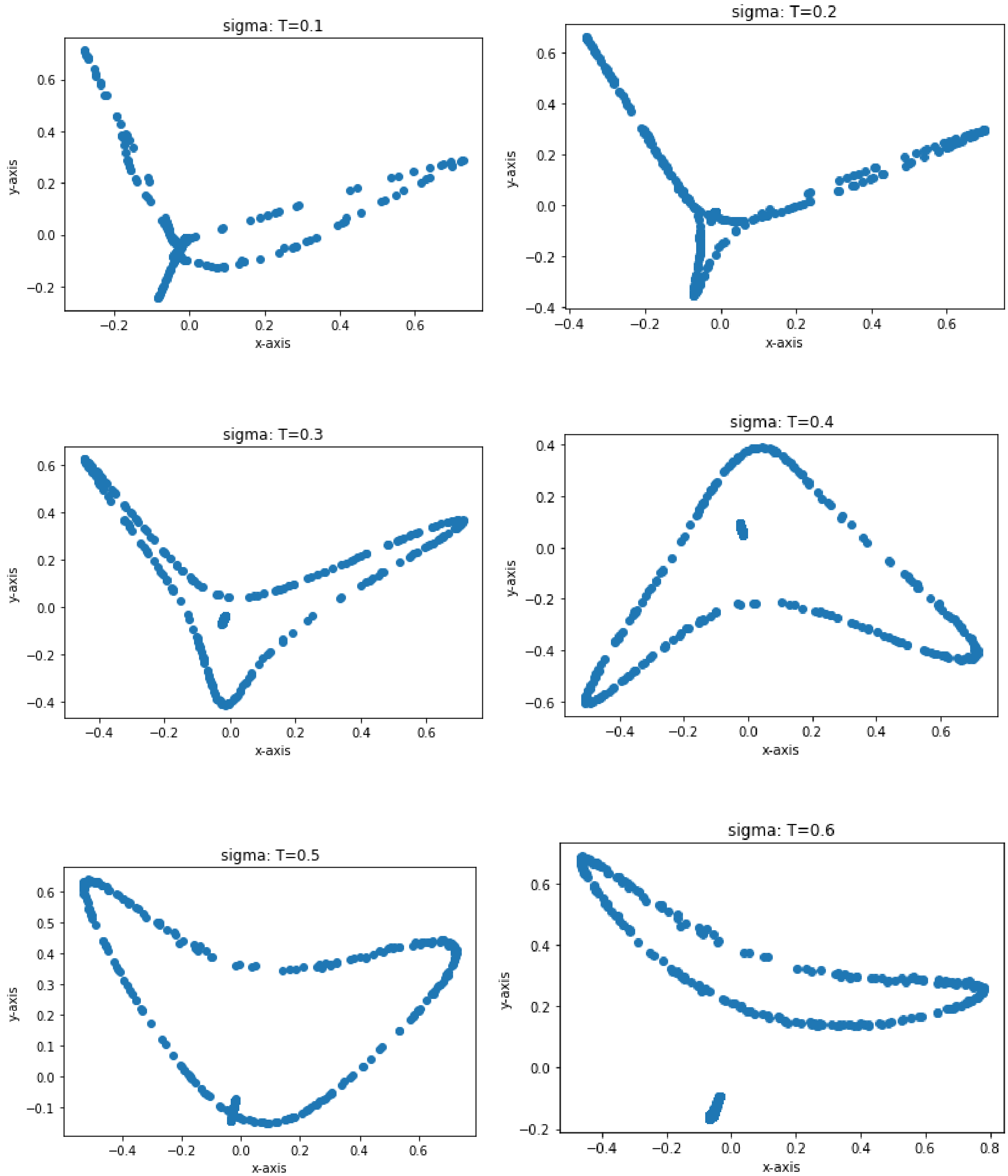
Using a polynomial kernel with degree 2, we get circular projections and the data is also concentric circles, but we can see most of the projections coincide with each other and are more on the left side, thereby showing lesser variance.

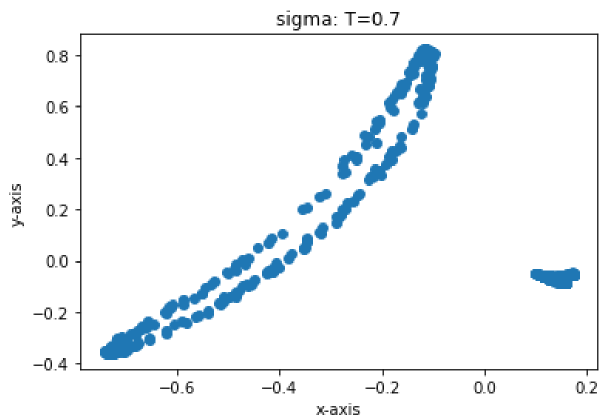
A (ii) Polynomial kernel (degree = 3)

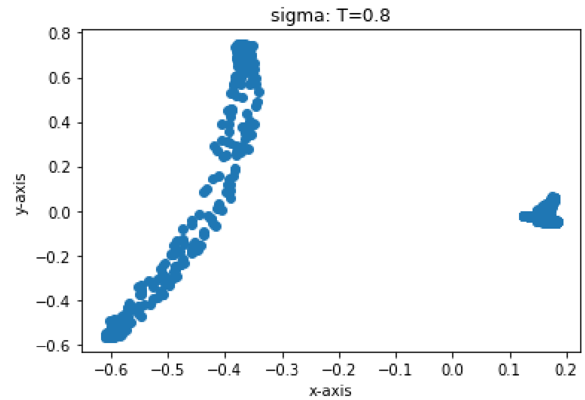


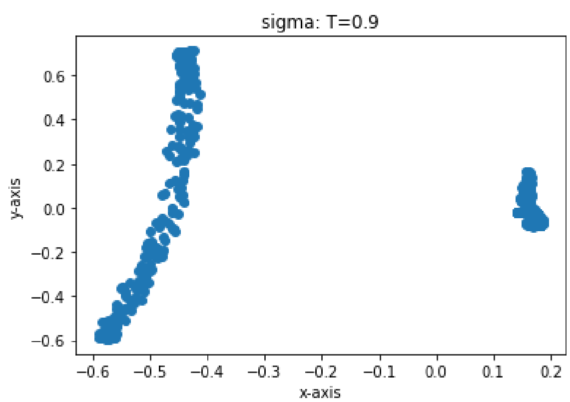
Using a polynomial kernel with degree 3, we get concentric circles similar to our original data so it seems to be a good kernel for this dataset.

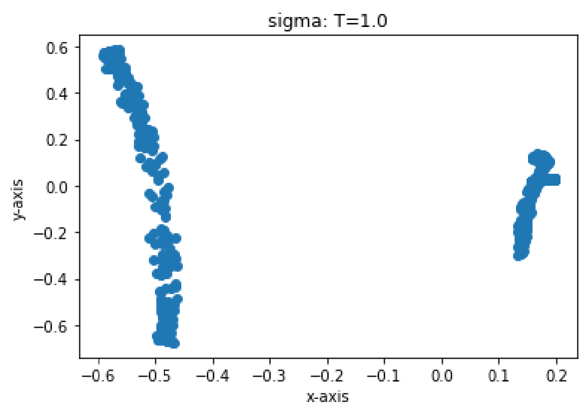
B (i) Radial Basis Kernel for different sigmas











iv. Which Kernel do you think is best suited for this dataset and why?
Polynomial kernel with degree 3 is best suited for this dataset because as we see from the plots polynomial kernel with degree 3 is giving projections which are similar to our dataset therefore gives less error.

Que. 2 You are given a data-set with 1000 data points each in R2 .

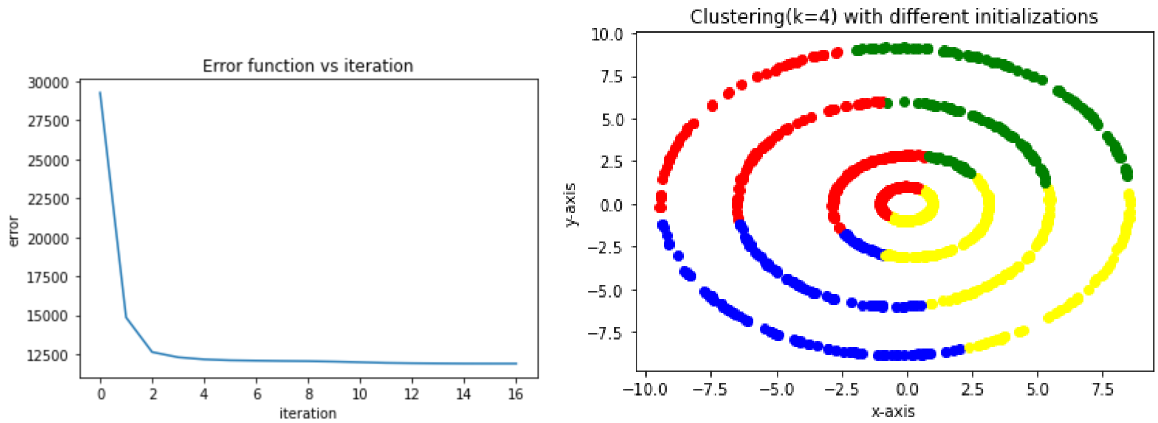
i. Write a piece of code to run the algorithm studied in class for the K-means problem with k = 4 . Try 5 different random initialization and plot the error function w.r.t iterations in each case. In each case, plot the clusters obtained in different colors.

- We can see from the **error function vs iteration** plot that it is a decreasing function and after some iterations error doesn't change, at that point our algorithm converges and we get minimum error.

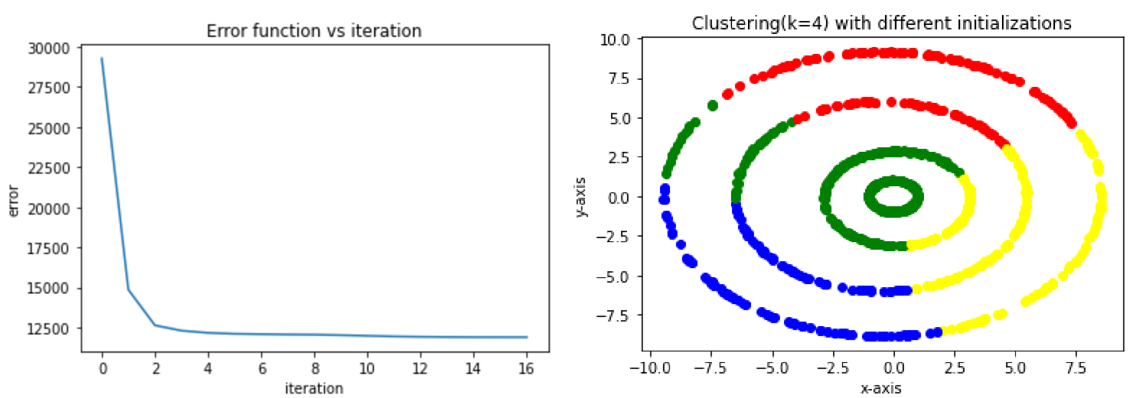
- By trying 5 different initializations we can see that only sometimes we obtain perfect clusters depending on the initialization.

That means good initialization is very important in clustering.

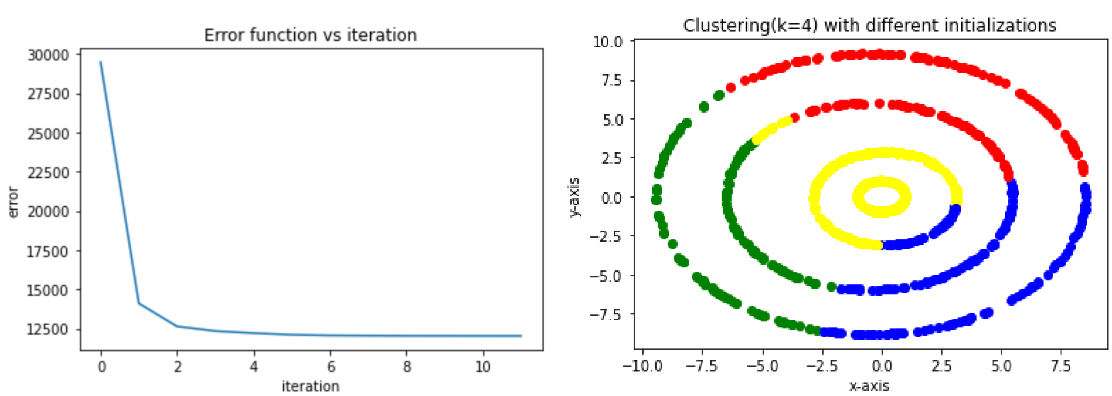
Initialization 1



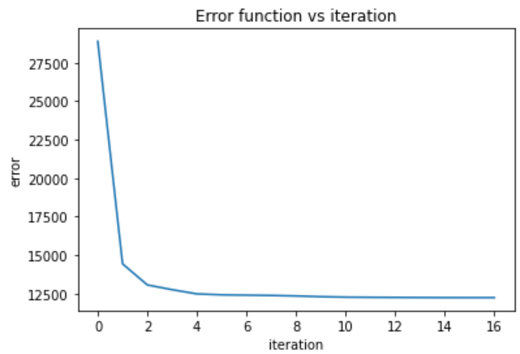
(a) Initialization 2

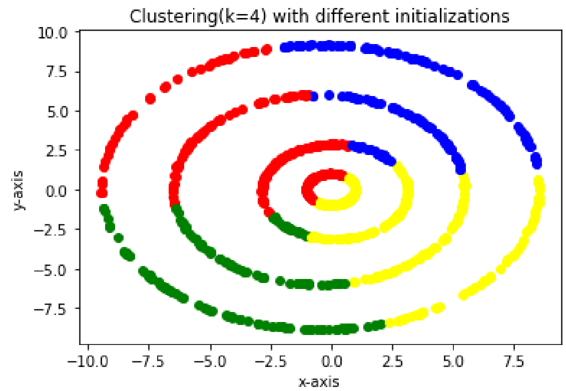


(c) Initialization 3

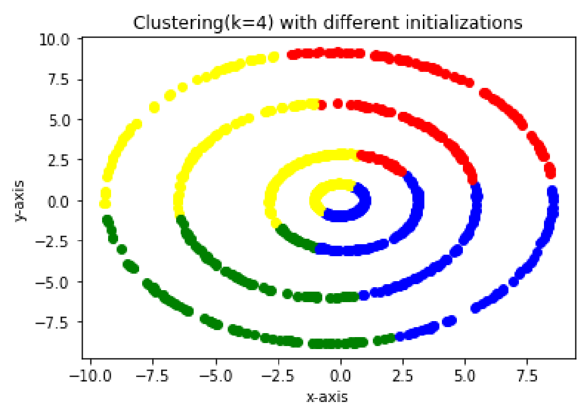


(d) Initialization 4





(e) Initialization 5

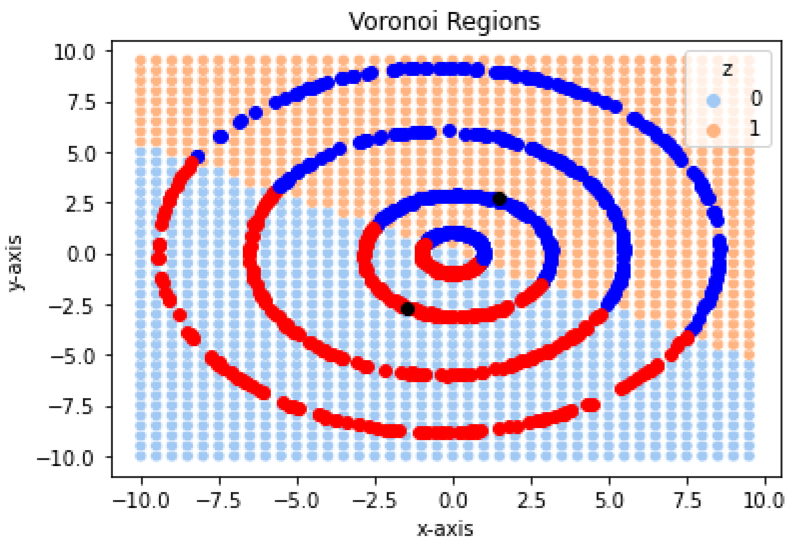


ii. Fix a random initialization. For K = {2, 3, 4, 5}, obtain cluster centers according to K-means algorithm using the fixed initialization. For each value of K, plot the Voronoi regions associated to each cluster center. (You can assume the minimum and maximum value in the data-set to be the range for each component of R2).

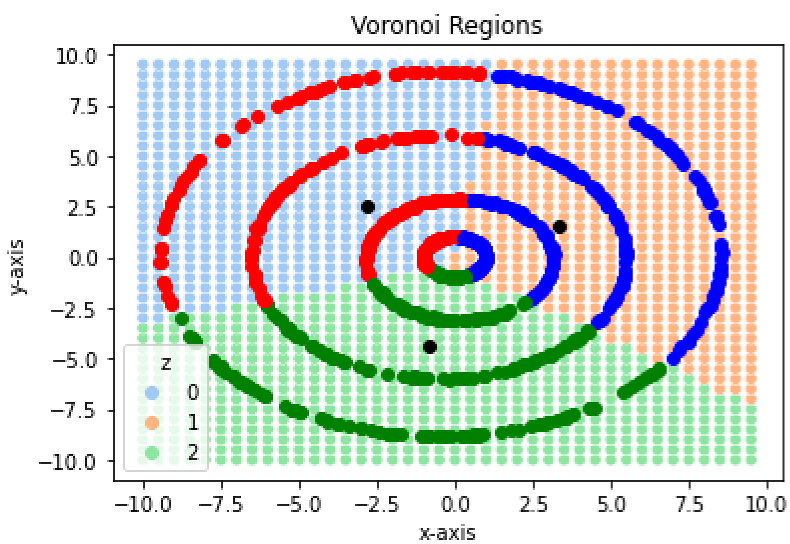
- Voronoi regions as shown in the plots separating the space into clusters K = {2, 3, 4, 5},
- We can see that point in the grid that belongs to a particular voronoi region is closest to its cluster centroid.

- Data points from our dataset that lie in the particular voronoi region belong to one cluster.

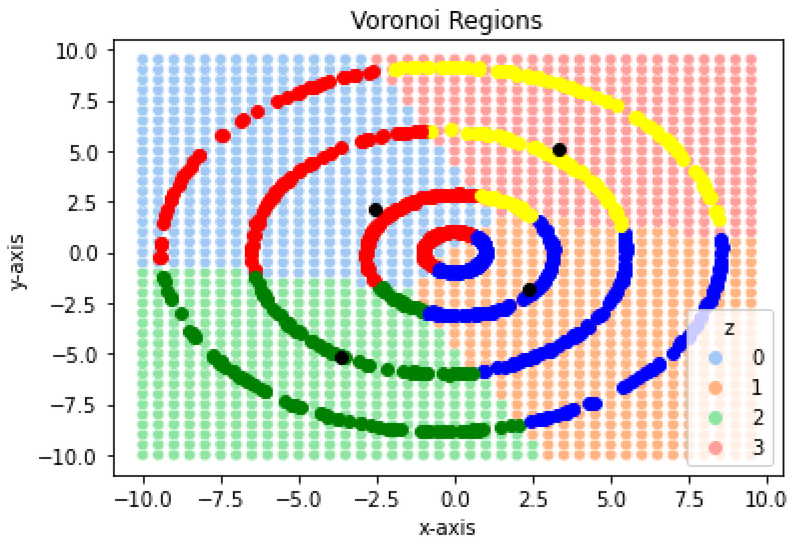
(a) K=2



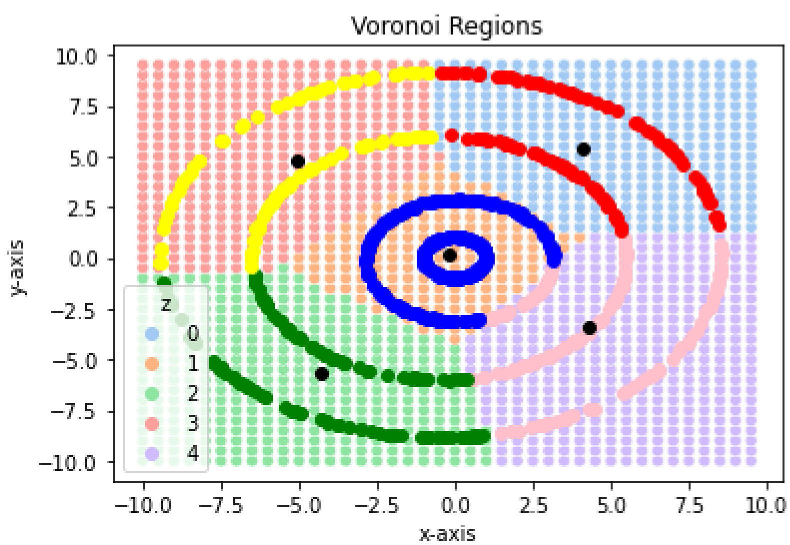
(b) K=3



(c) K=4



(d) K=5

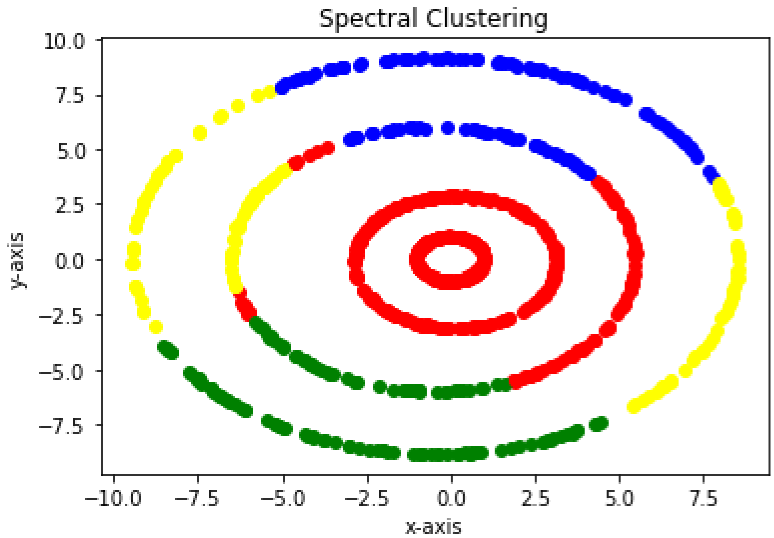


iii. Run the spectral clustering algorithm (spectral relaxation of K-means using Kernel- PCA) k = 4. Choose an appropriate kernel for this data-set and plot the clusters obtained in different colors. Explain your choice of kernel based on the output you obtain.

After testing all the kernels, Radial Basis Function (σ=0.5) is giving the least errorSum after convergence.
And from the plot of Radial Basis Function (σ=0.5), we can assume that data points that are in the same cluster may lie in same plane and other clusters in different plane.

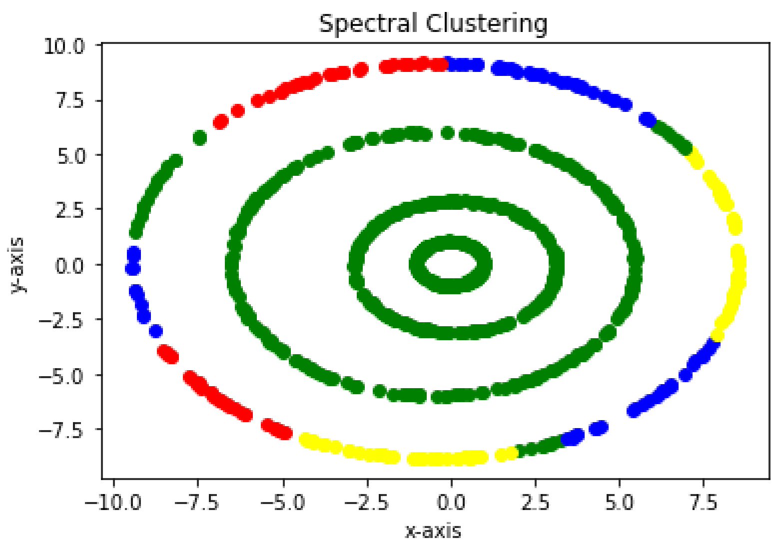
Polynomial d=2

Error: 1.7753212264093747



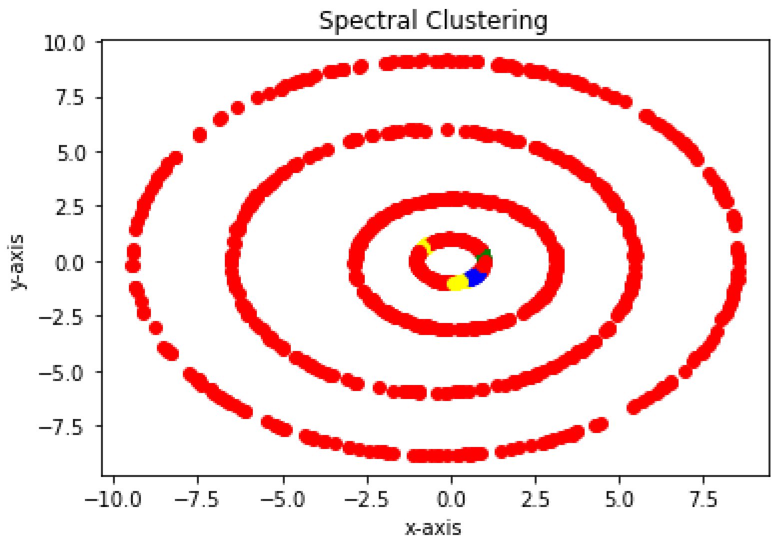
Polynomial d=3

Error: 2.7008595439802563

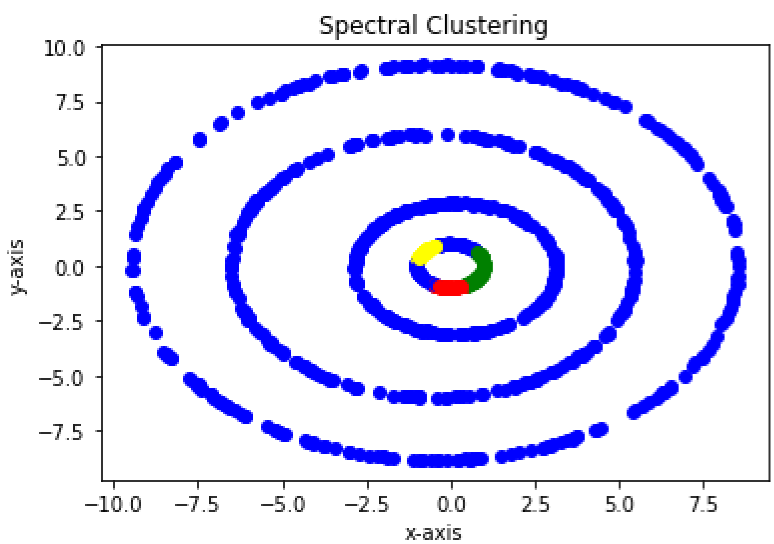


Radial Basis Function (σ=0.1)

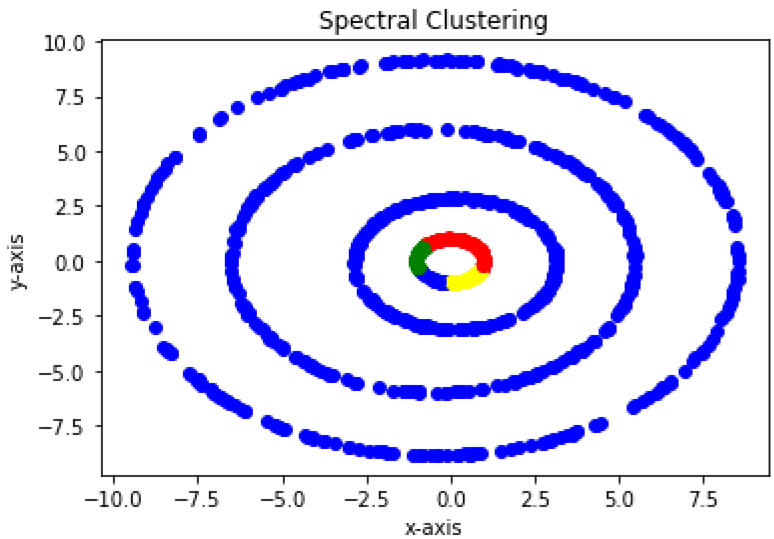
Error: 1.5200317622278834



Radial Basis Function (σ=0.2)
Error: 1.4268755265849082

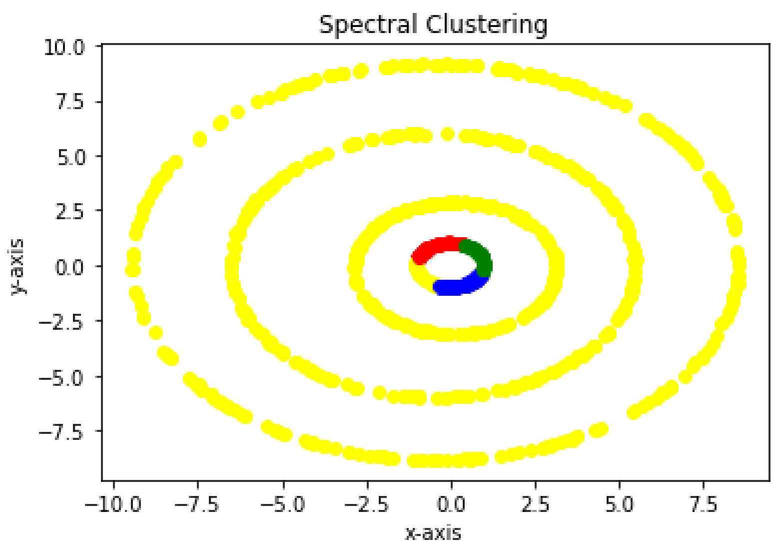


Radial Basis Function (σ=0.3)
Error: 1.569399940097189



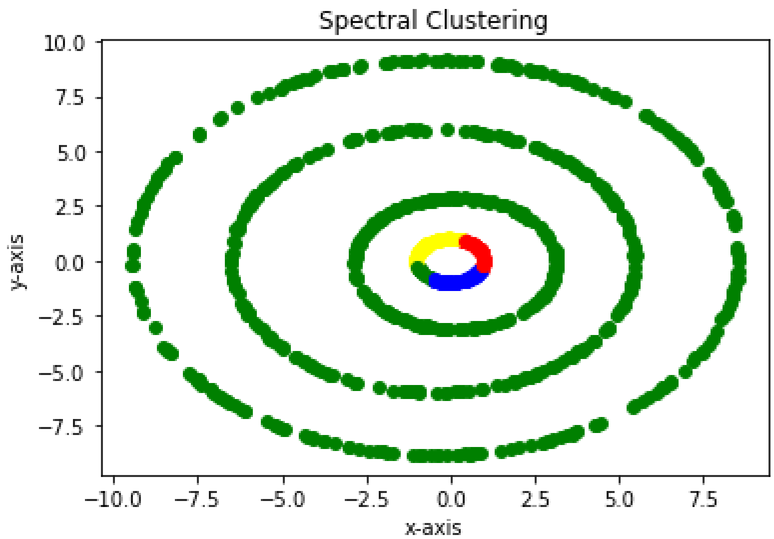
Radial Basis Function (σ=0.4)

Error: 1.368044113374649

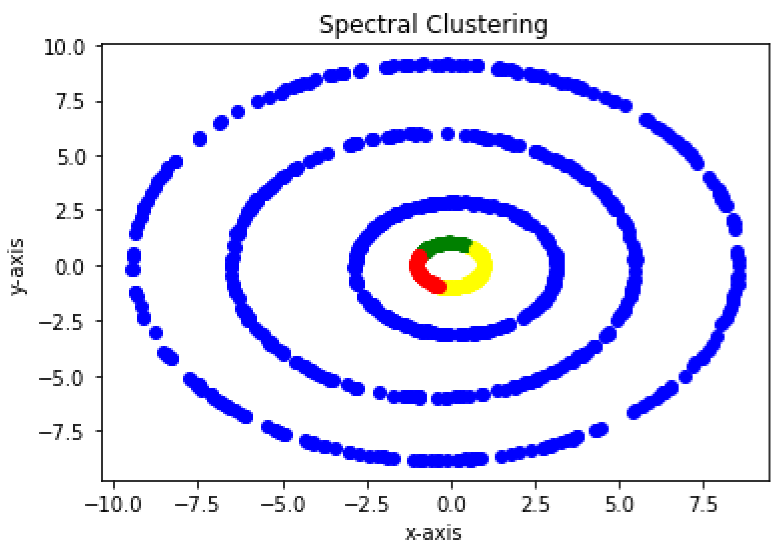


Radial Basis Function (σ=0.5)

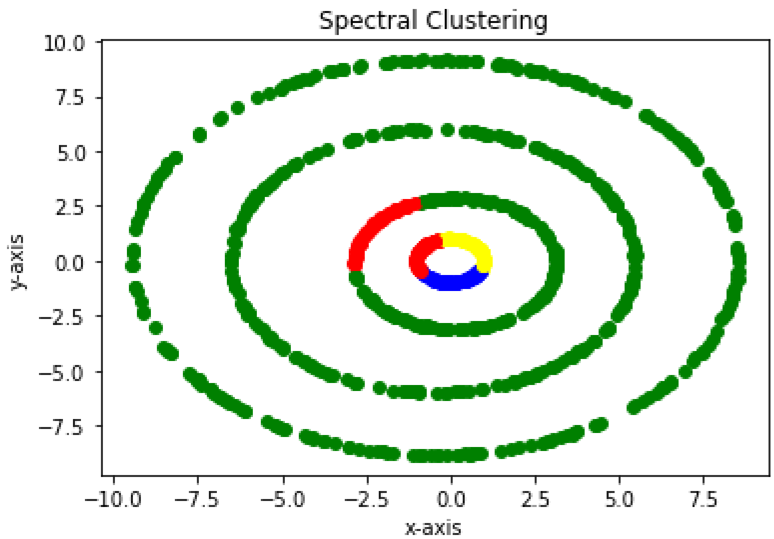
Error: 1.3309398291750631



Radial Basis Function (σ=0.6)
Error: 1.405523039127826

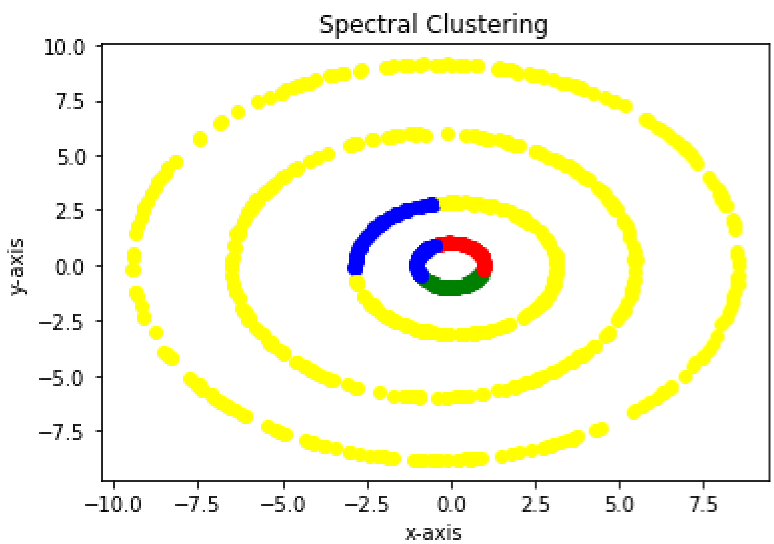


Radial Basis Function (σ=0.7)
Error: 1.4735067184314117



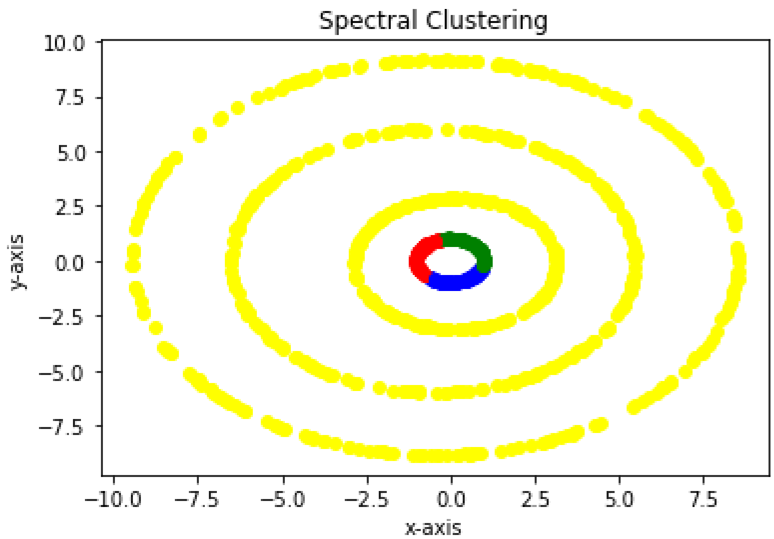
Radial Basis Function (σ=0.8)

Error: 1.4943096284268194



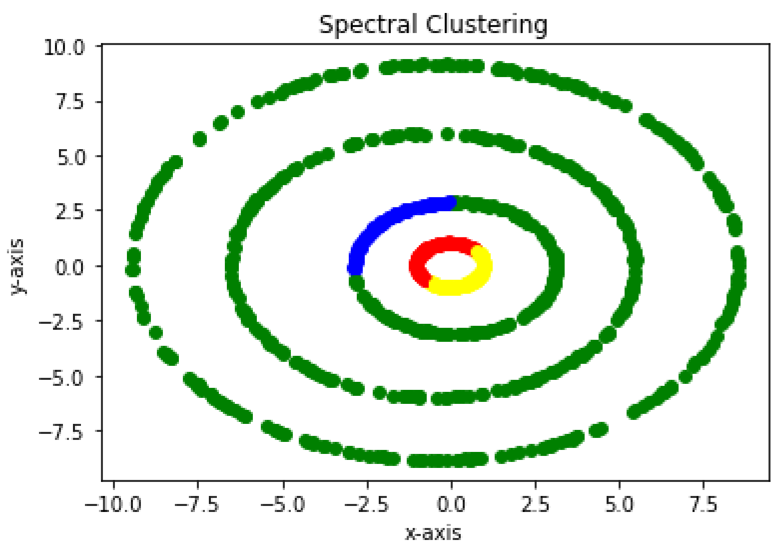
Radial Basis Function (σ=0.9)

Error: 1.5194933647832367



Radial Basis Function (σ=1)

Error: 1.3926826216368942



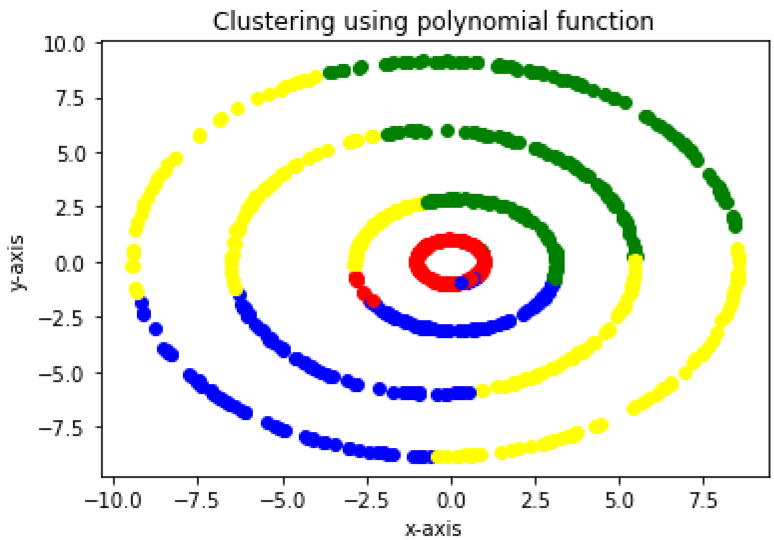
iv. Instead of using the method suggested by spectral clustering to map eigenvectors to cluster assignments, use the following method: Assign data point i to cluster whenever

L = arg max (j=1,...,k) vi^j

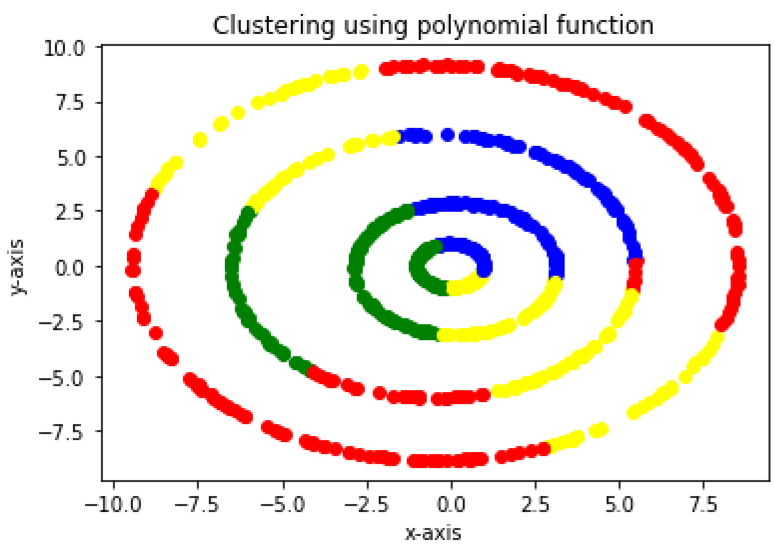
where v j ∈ Rn is the eigenvector of the Kernel matrix associated with the j-th largest eigenvalue. How does this mapping perform for this dataset?. Explain your insights.

As there is little change between the previous question(spectral clustering) and this one. This mapping seems to perform well on this dataset.

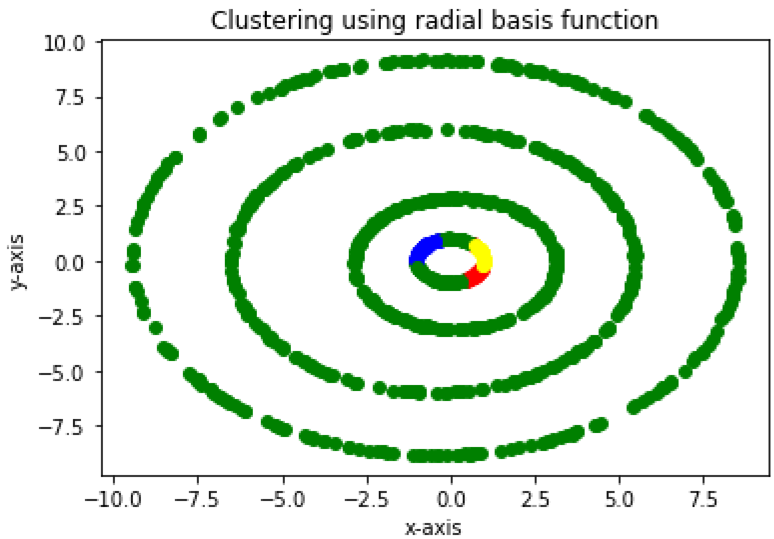
Polynomial d=2



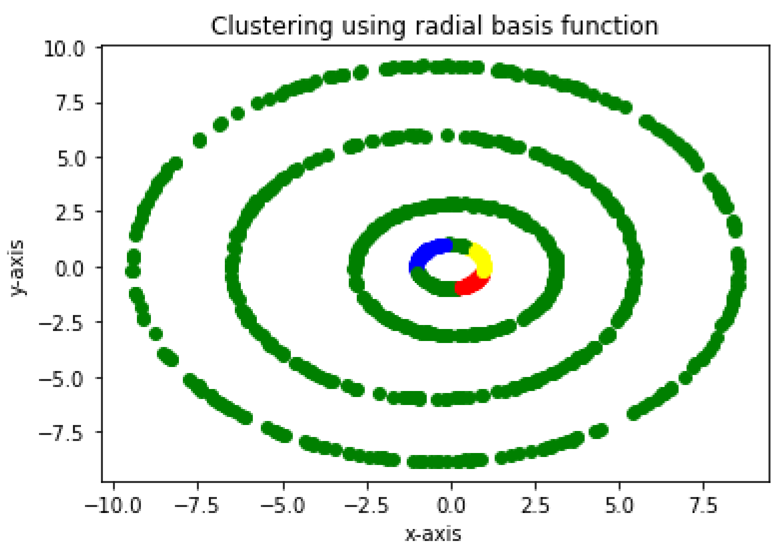
Polynomial d=3



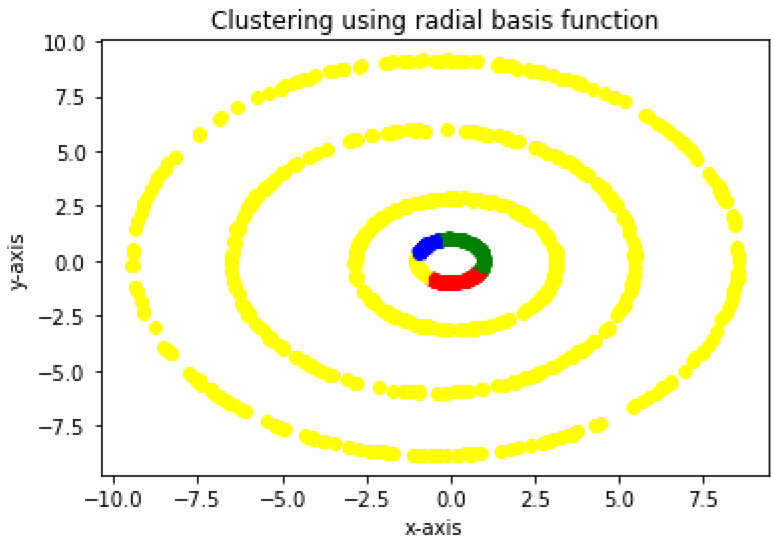
Radial Basis Function (σ=0.1)



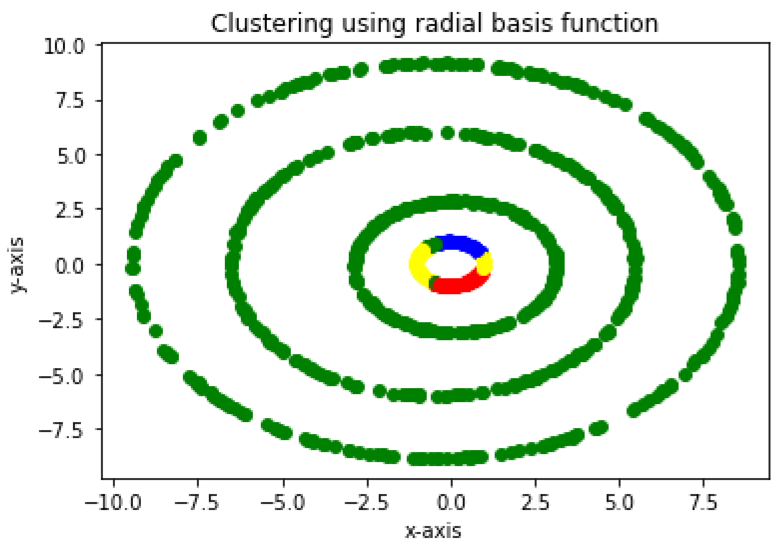
Radial Basis Function (σ=0.2)



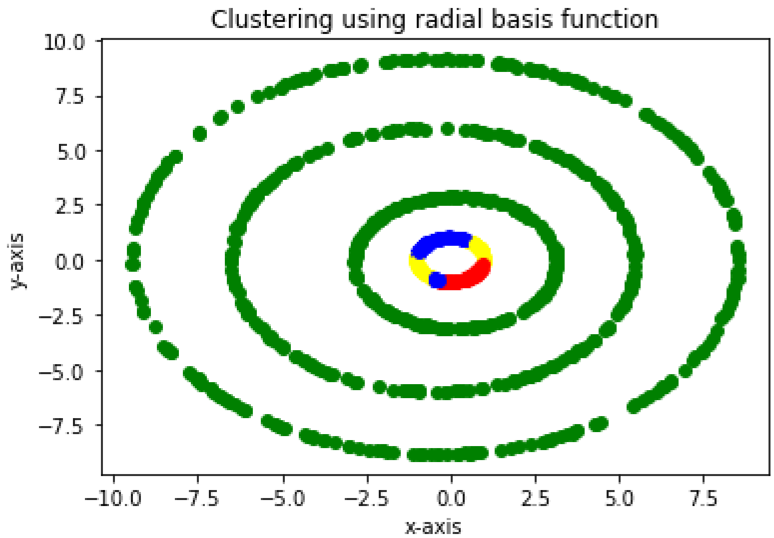
Radial Basis Function (σ=0.3)



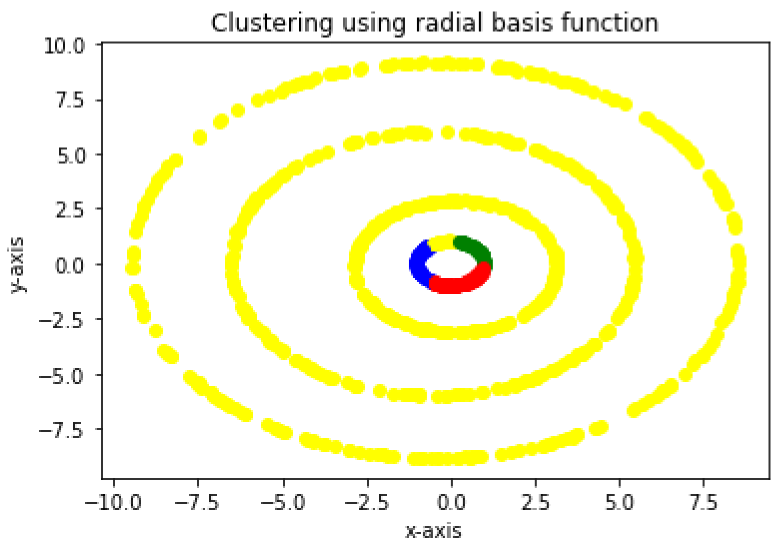
Radial Basis Function (σ=0.4)



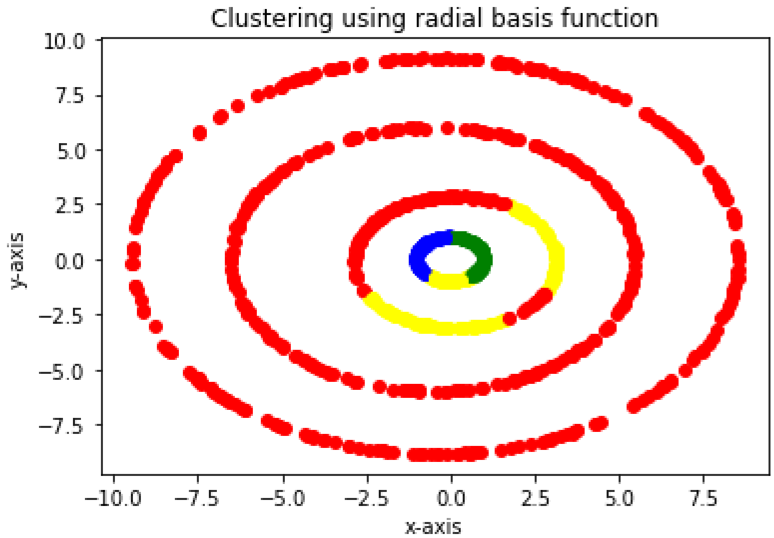
Radial Basis Function (σ=0.5)



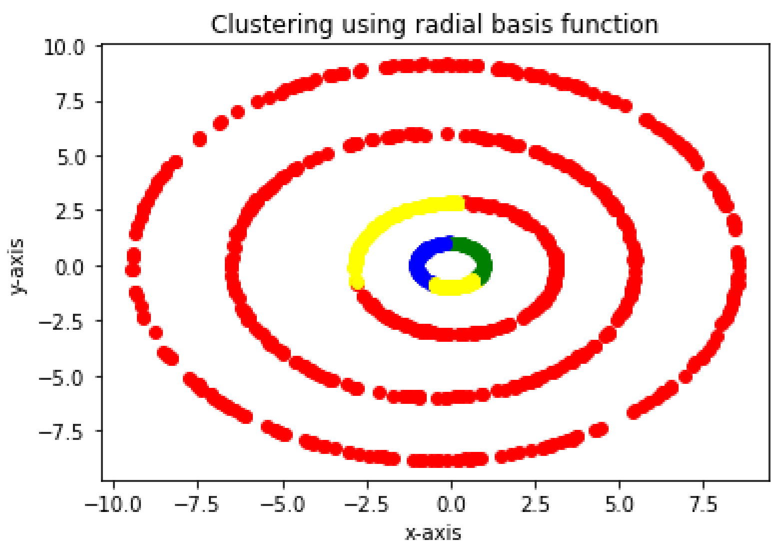
Radial Basis Function (σ=0.6)



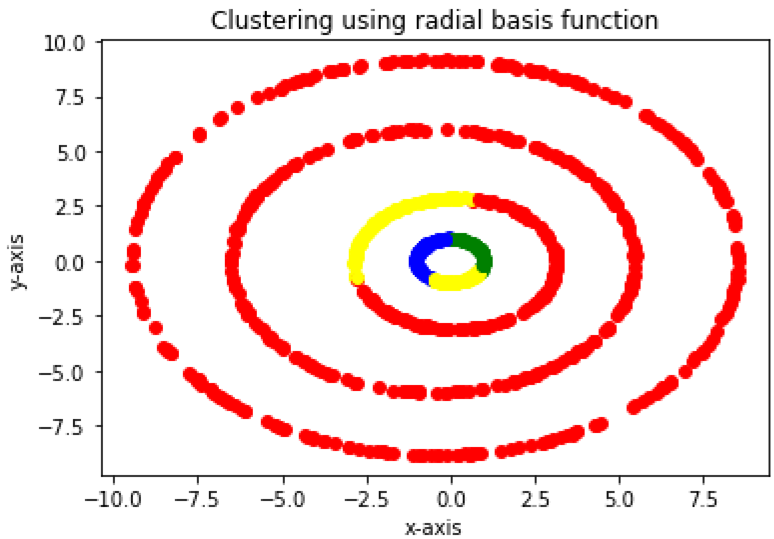
Radial Basis Function (σ=0.7)



Radial Basis Function (σ=0.8)



Radial Basis Function (σ=0.9)



Radial Basis Function (σ=1)