CS5691: Pattern recognition and machine learning Assignment 1

Name and Signature
Hanumantappa Budihal

1. **Question 1 :**

   (a) 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?

   Solution:
   The code for PCA is in the attached code file named question **Q1.py.**

   Variance along first principal component is **54.178025%** of total variance.
   Variance along the second principal component **is 45.821975%** of total variance.

   The total of variance observed by data along principal components is = **31.621519151774**

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

   Solution: By running the PCA on the given data without centering, the result obtained are as follows:

   Total of variance along principal components : **31.621519151774983**
   Variance along first principal component : **54.178025%**
   Variance along second principal component : **45.821975%**

   Form (a) and (b) we can see that the percentage of variance observed by each of principal components are the same in case of centered and non-centered data.

   Hence , we can say that for given data centering does not help, below figure is shown for the centered data

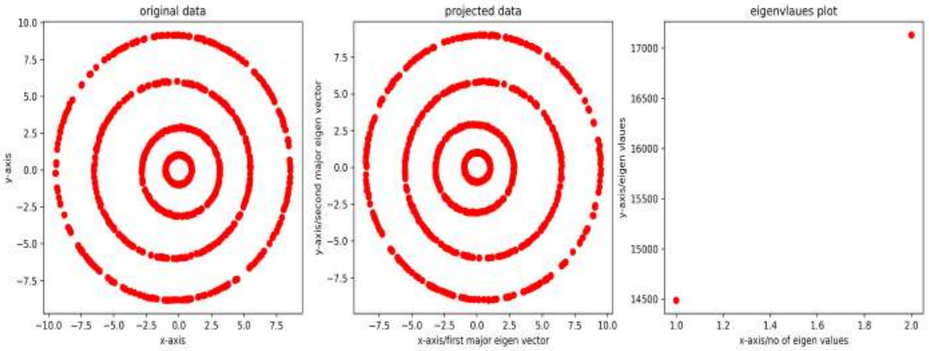

Figure1 : Result of centered data

Below figure shows the results for non-centered data

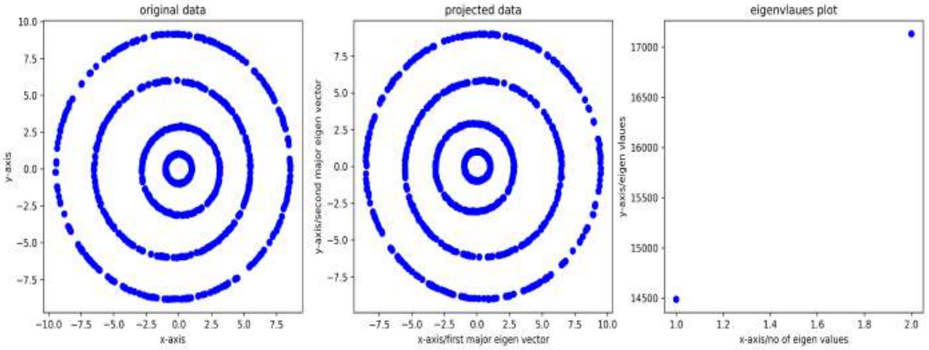

Figure2 : Result of non centered data

As the given data is already centered so centering does not plays any role here. From figure it is clear that PCA is not able to do any helpful transformation on data, because, after projecting data onto eigen space we are obtaining the same data.

Also we are having the initial total variance of given data = total variance of projected data. as the data is circular around the origin, any pair of perpendicular vectors can become principal components directions.

(c) Write a piece of code to implement the Kernel PCA algorithm on this dataset. Use the following kernels :

A. $k(x, y) = (1 + x^T y)^d$ for d = 2,3

B. $k(x, y) = exp\frac{-(x-y)^T(x-y)}{2\sigma^2}$ for $\sigma = 0.1, 0.2, ..., 1$

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

Solution: Plots for kernel $k(x, y) = (1 + x^T y)^d$ **are shown below**

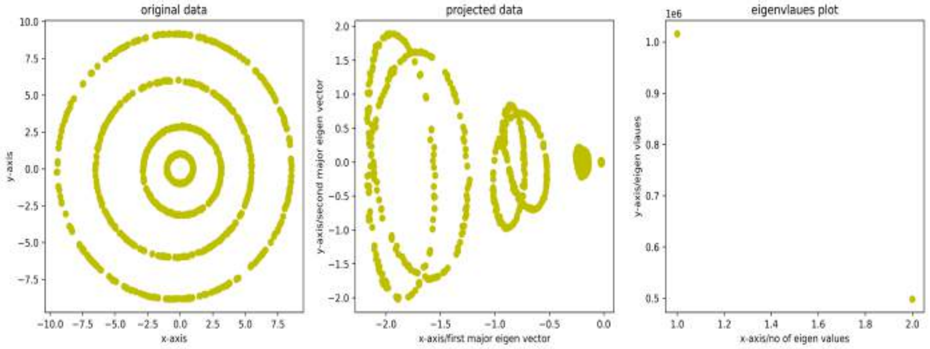

Figure 3 : for d=2

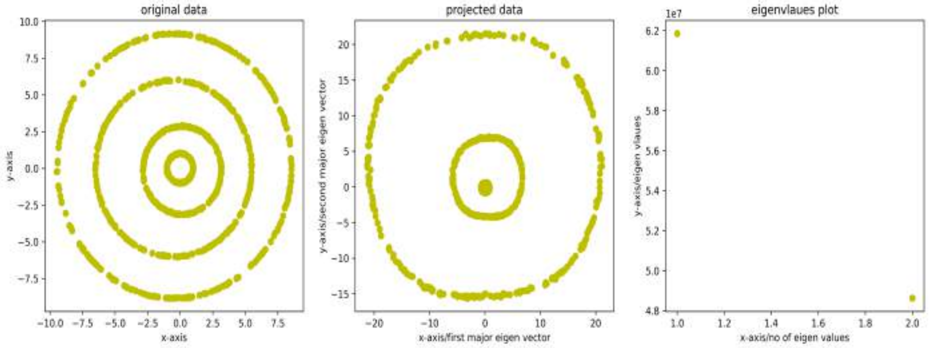

Figure 4 : for d=3

Plots for kernel $k(x, y) = (1 + x^T y)^d$ **are shown below**

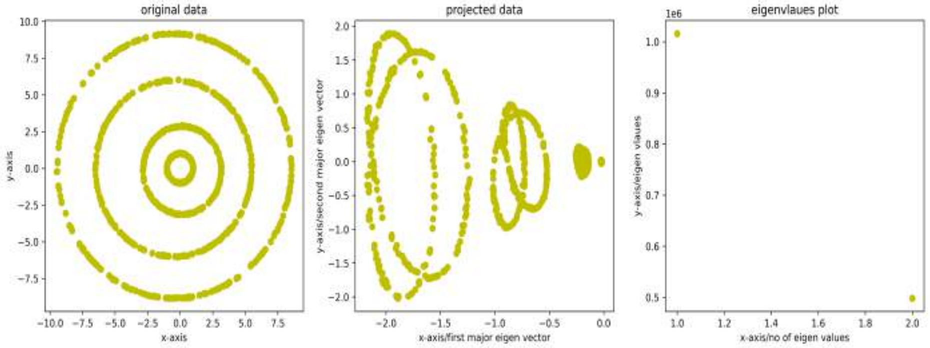

Figure 5: for α=0.1

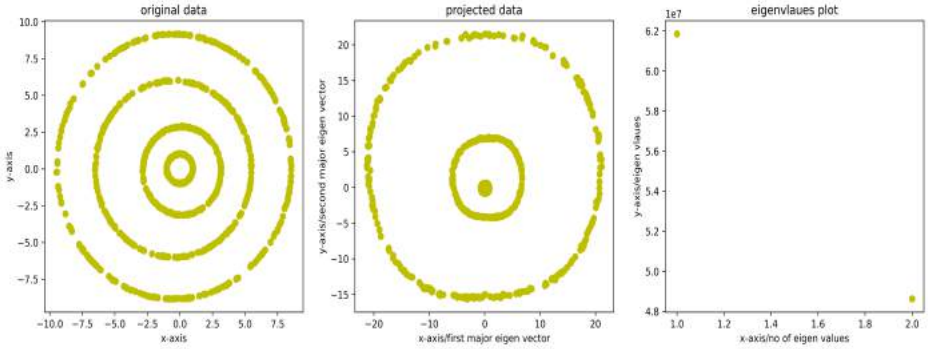

Figure 6 : for α=0.2

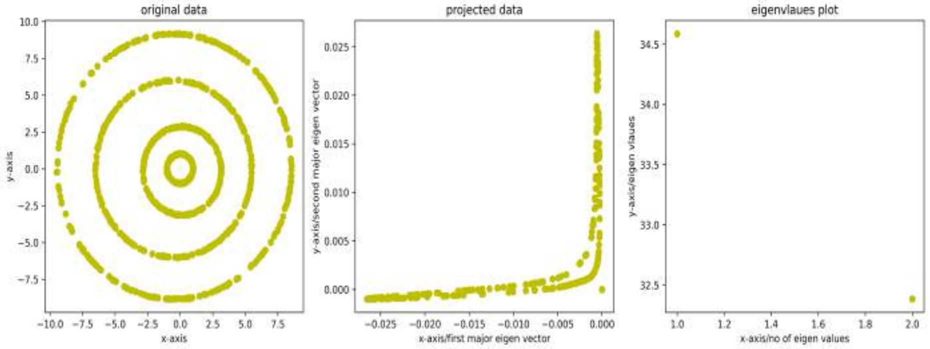

Figure 7 : for α=0.3

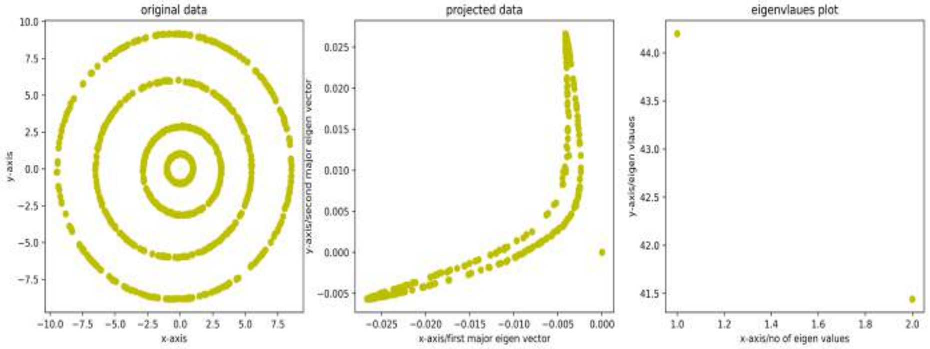

Figure 8 : for α=0.4

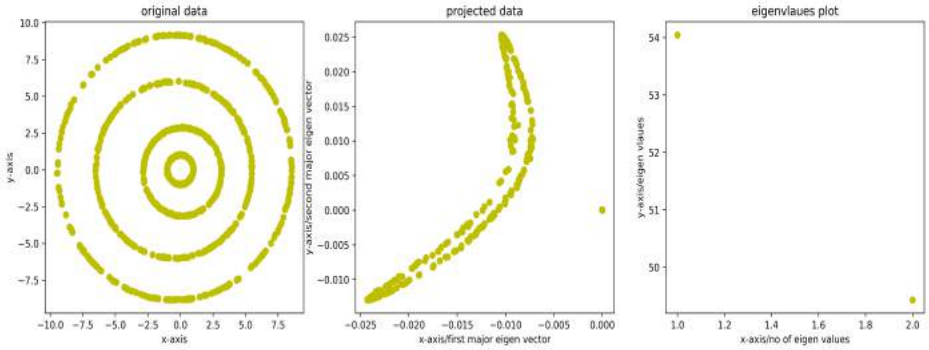

Figure 9 : for α=0.5

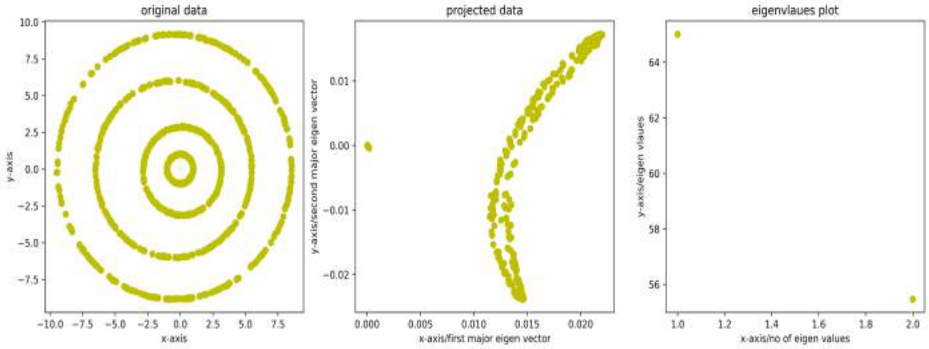

Figure 10 : for α=0.6

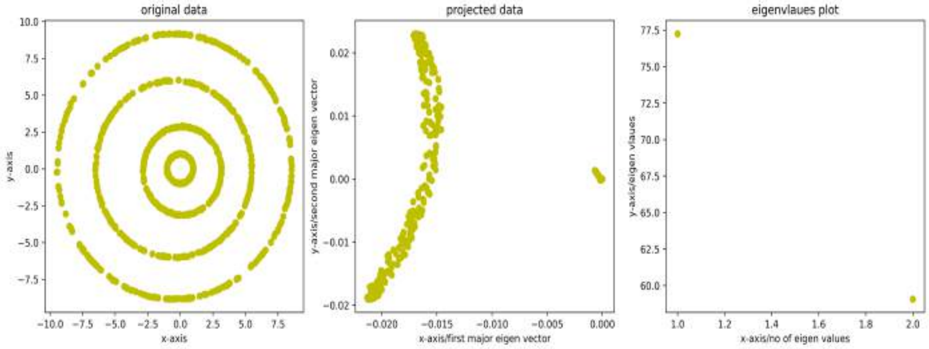

Figure 11 : for α=0.7

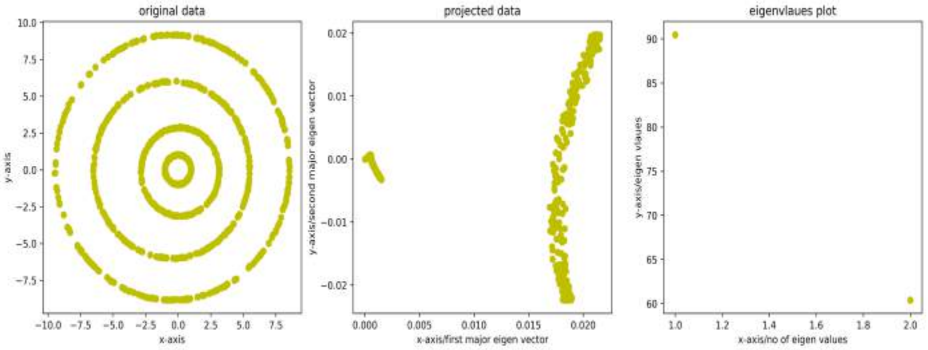

Figure 12 : for α=0.8

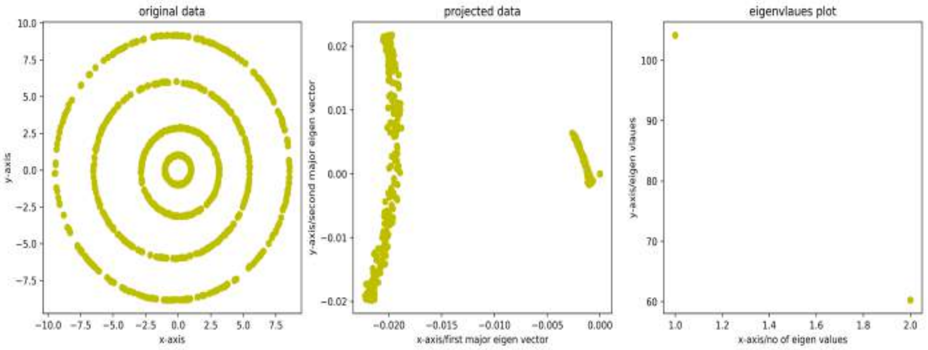

Figure 13 : for α=0.9

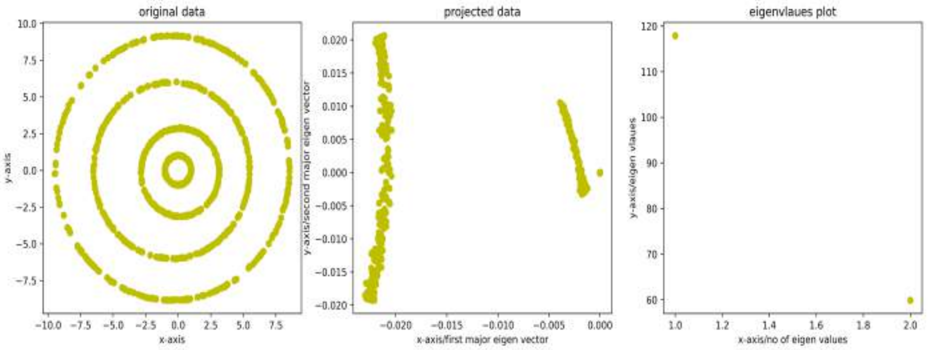

Figure 14 : for α=1.0

(d) Which Kernel do you think is best suited for this dataset and why?

Solution: The gaussian kernel is best suited for this dataset with as 1,because it is capturing maximum variance among all other values of . The polynomial kernel is not able to properly de-correlate the data; it can be seen from figure 3 and figure 4 as compared to the gaussian kernel.

2. .

(a) Write a piece of code to run the algorithm studied in class for the K-means problem with k = 4 . 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.

Solution:

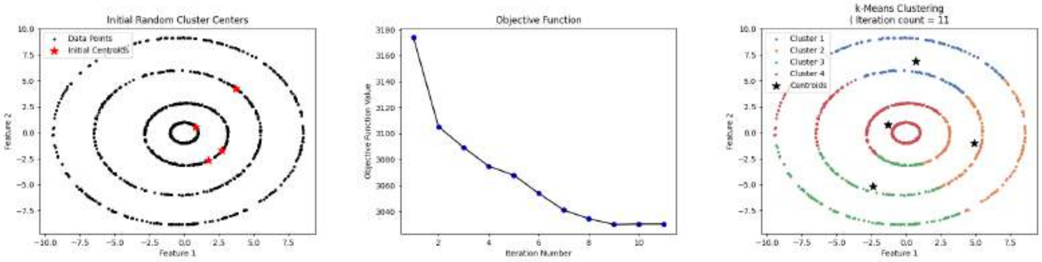

Figure 15 : Run 1 : no of clusters = 4

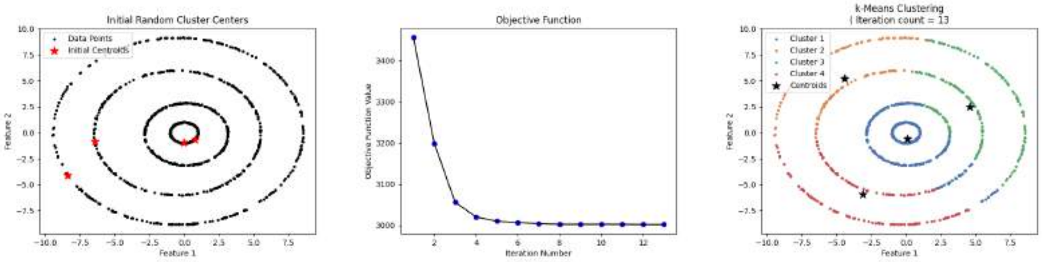

Figure 16 : Run 2 : no of clusters = 4

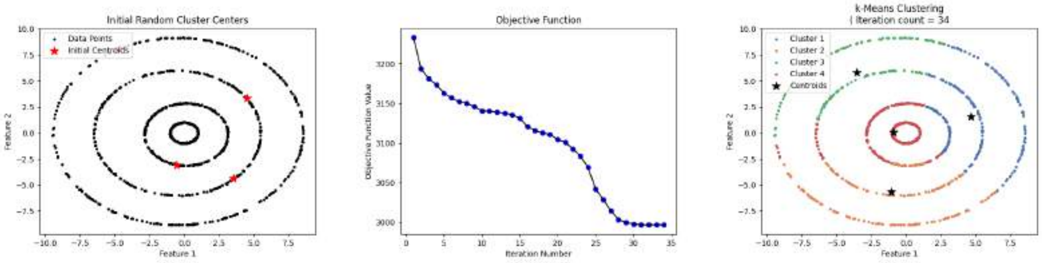

Figure 17 : Run 3 : no of clusters = 4

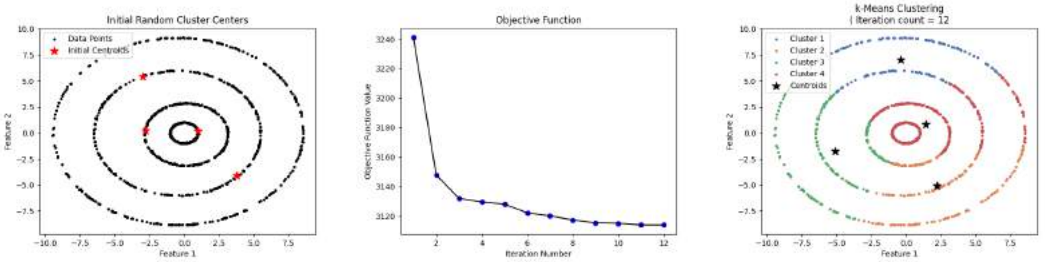

Figure 17 : Run 4 : no of clusters = 4

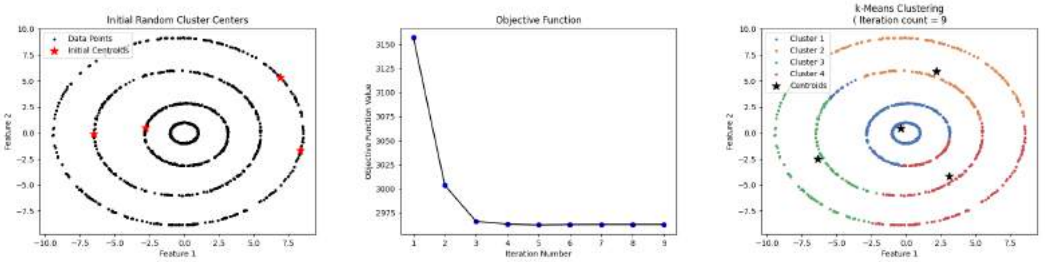

Figure 17 : Run 5: no of clusters = 4

(b) 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).

Solution: Fixed a random initialization and $K = 2, 3, 4, 5$,

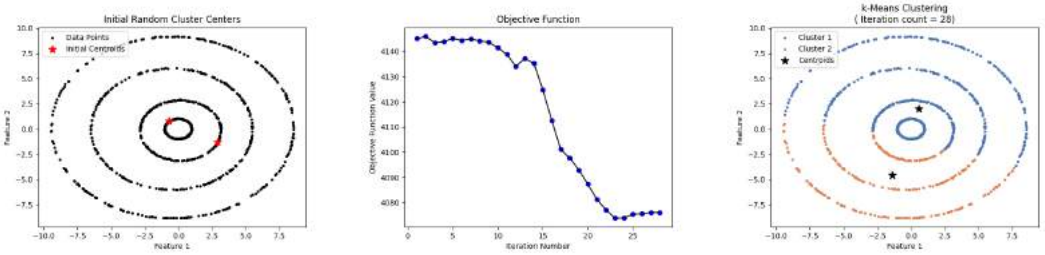

Figure 18 : Run 1: no of clusters = 2

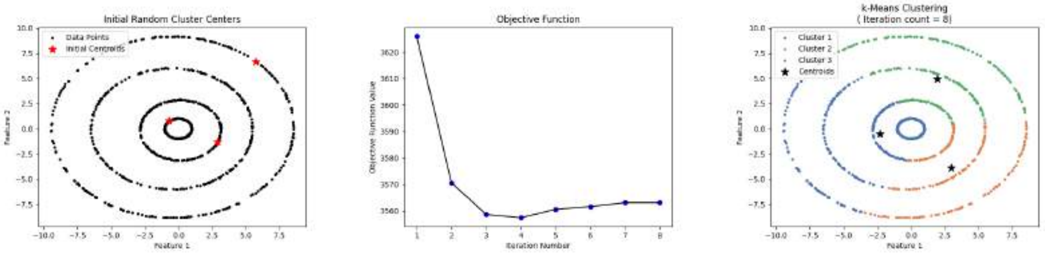

Figure 18 : Run 2: no of clusters = 3

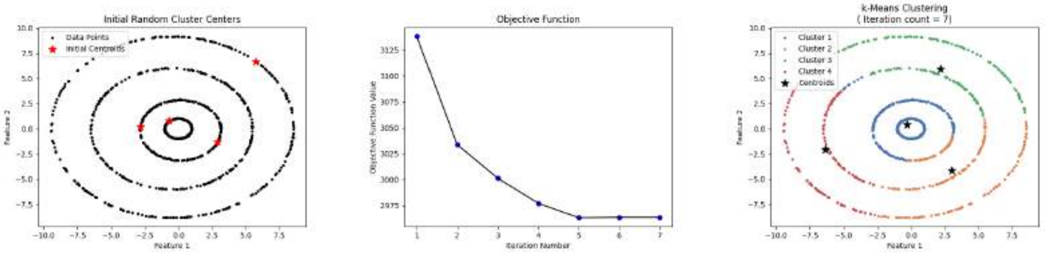

Figure 18 : Run 3: no of clusters = 4

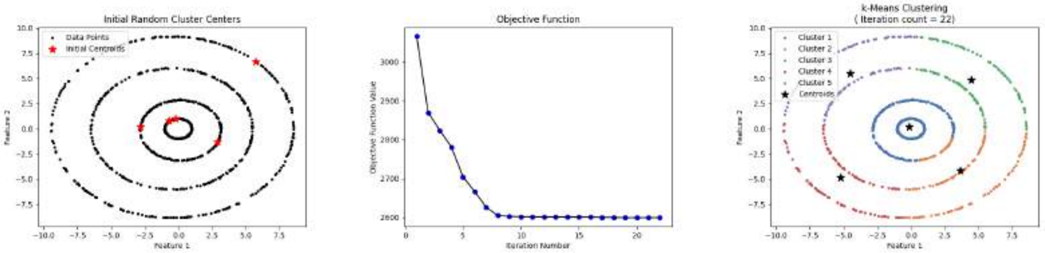

Figure 18 : Run 4: no of clusters = 5

(c) 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.

Solution:

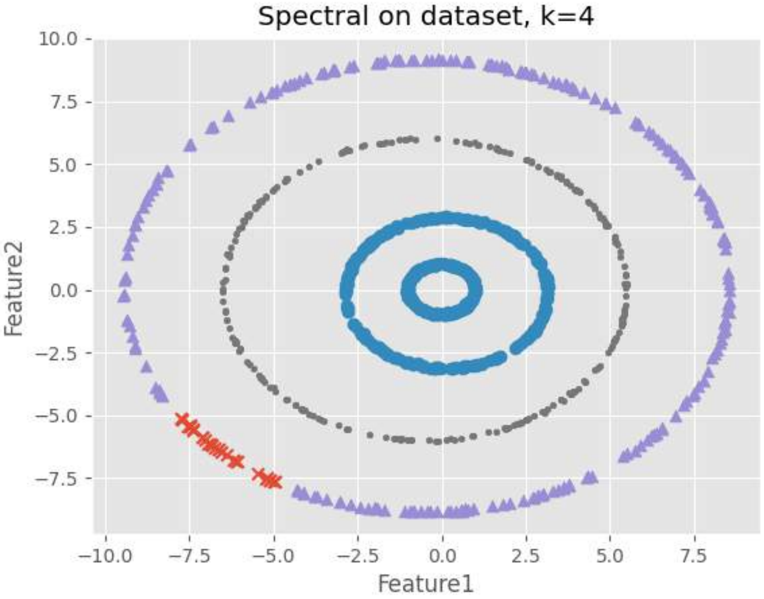

Figure 19 : Spectral Clustering , K=4

Used the Radical Basic function as function approximation. The idea behind the self tuning spectral clustering is determine the optimal number of clusters and also the similarity metric i used in the computation of the affinity matrix. We can achieve using the RBF.

(d) 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 l whenever

$$l = arg \max_{j=1,...k} v_i^j$$

where $v^j \in R^n$ 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.

Solution:
It is failed to converge it self or might not able to output good solution.