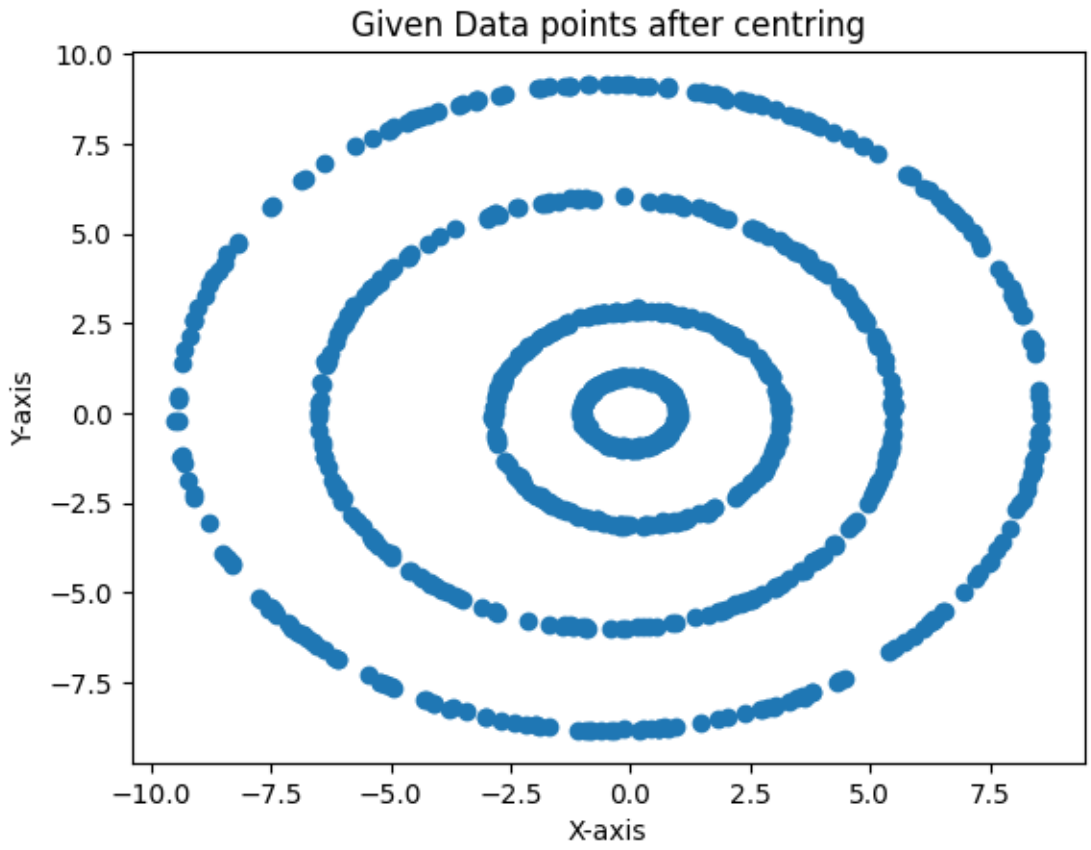
Q1.i) PCA with centering

Step 1 :-

To center data we have to calculate mean and subtract it from all points.

Centered Data= Original data - mean



Step 2 :- Calculate covariance matrix for centered data.

Step 3 :- Calculate eigen vector and eigen values for covariance matrix.

Sort eigen values and corresponding eigen vectors in descending order.

Step 4 :- The top two eigen values obtained are as follows

Eigen value 1 = 17.149063465

Eigen value 2 = 14.504108858

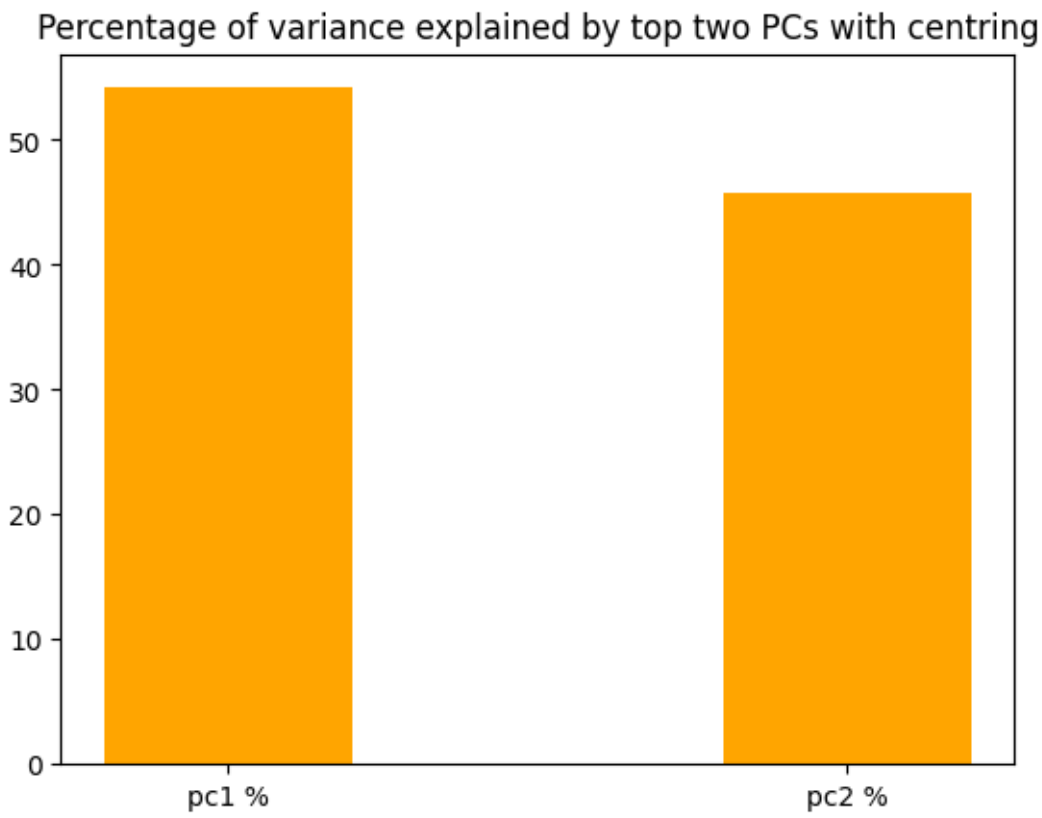
Percentage of variance given by principal component

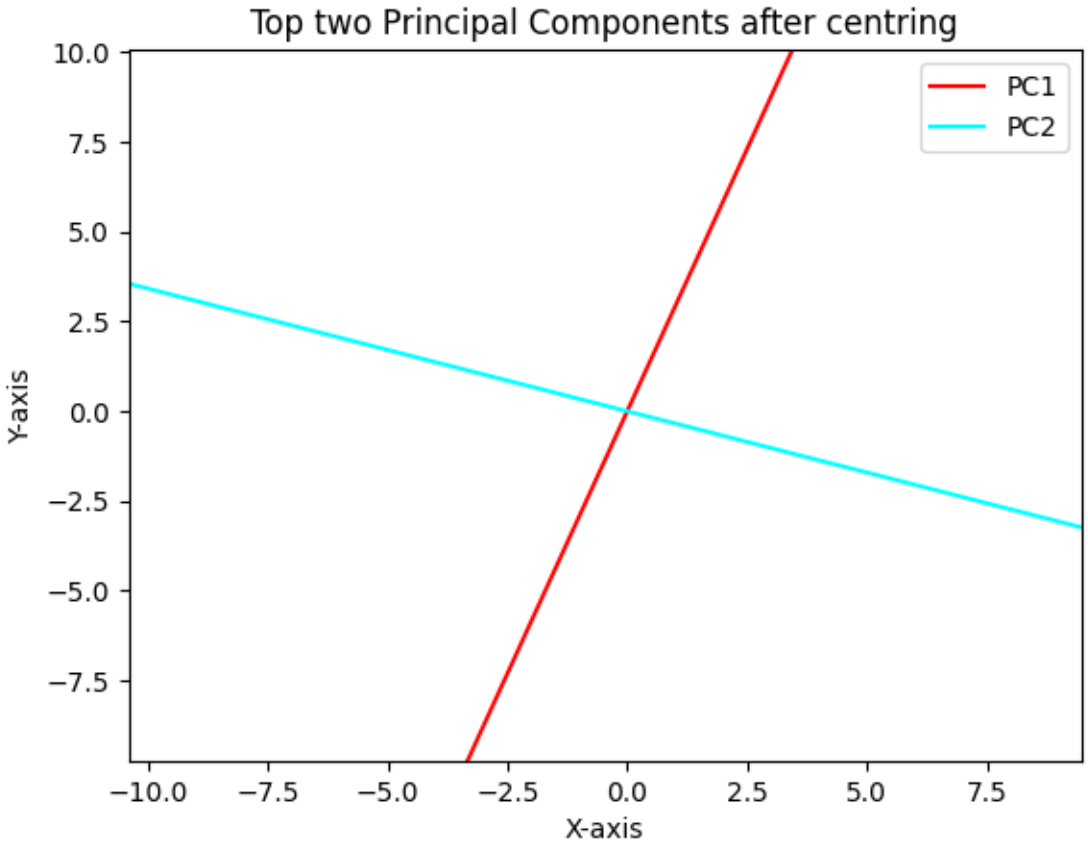
= corresponding eigen value * 100 /(sum of all eigen values)

Percentage variance explained by

Principal component 1 = 54.1780245288 %

Principal component 2 = 45.8219754711 %


Percentage of variance explained by top two PCs with centring
50
40
30
20
10
0
pc1 %
pc2 %


Top two Principal Components after centring
PC1
PC2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis

Q1.ii) PCA without centering

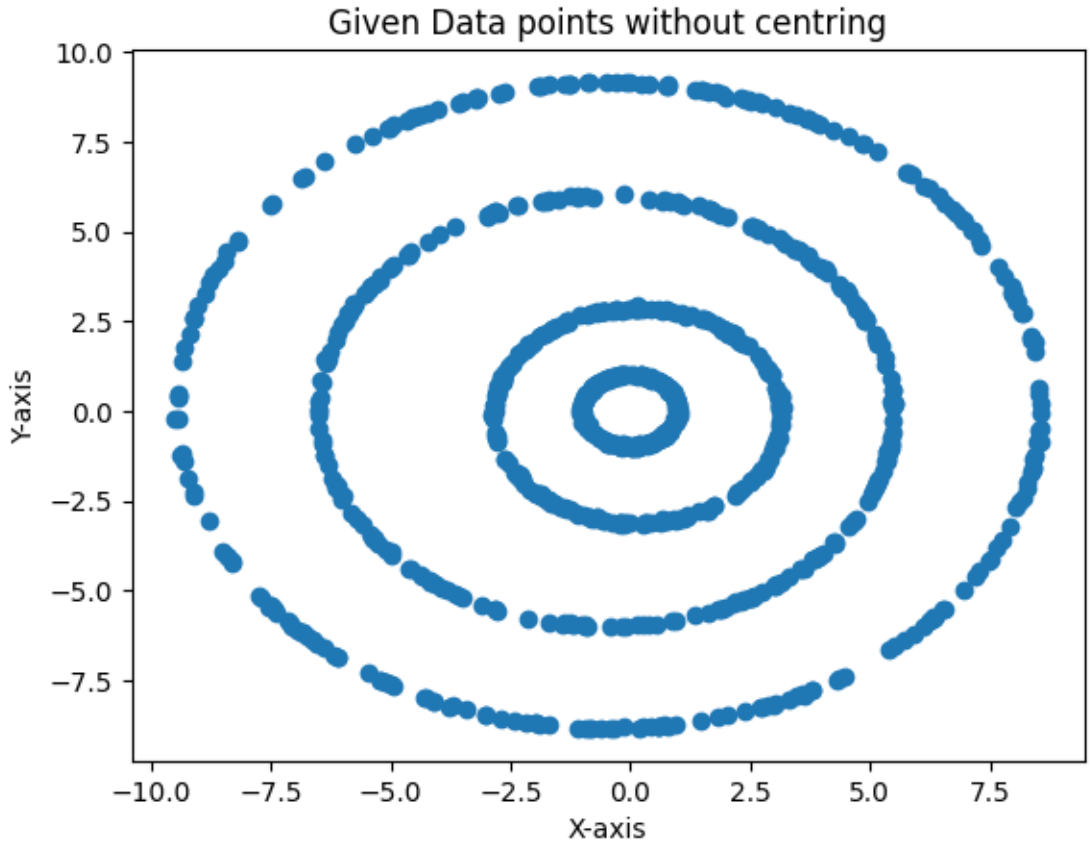
The steps are same as mentioned in above example except centering is not done in this example.

In this example , the data given is already centered. That is the mean calculated for data is almost (0,0) origin.

Calculated mean = (4.075 * e ^-7 , 2.227 * e^-7)

So the PCA without centering has similar results to the PCA with centering.

Centering has no effect for given data set to run PCA algorithm.



The top two eigen values obtained are as follows

Eigen value 1 = 17.149063465

Eigen value 2 = 14.504108858

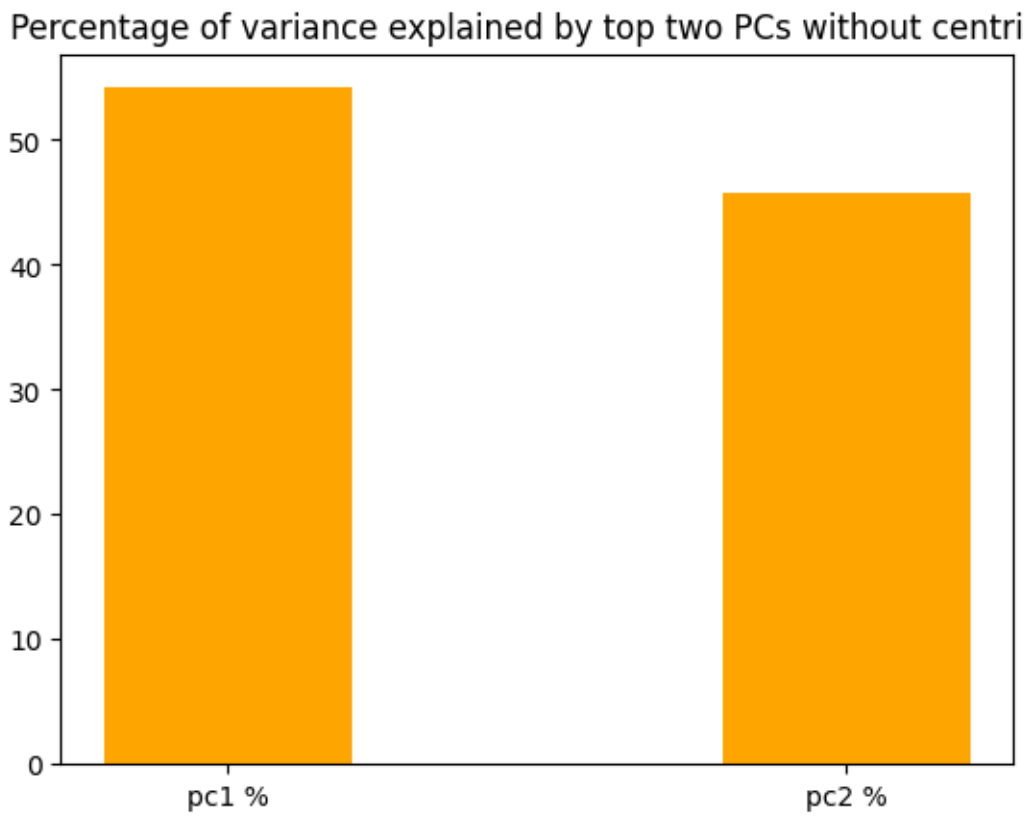
Percentage of variance given by principal component

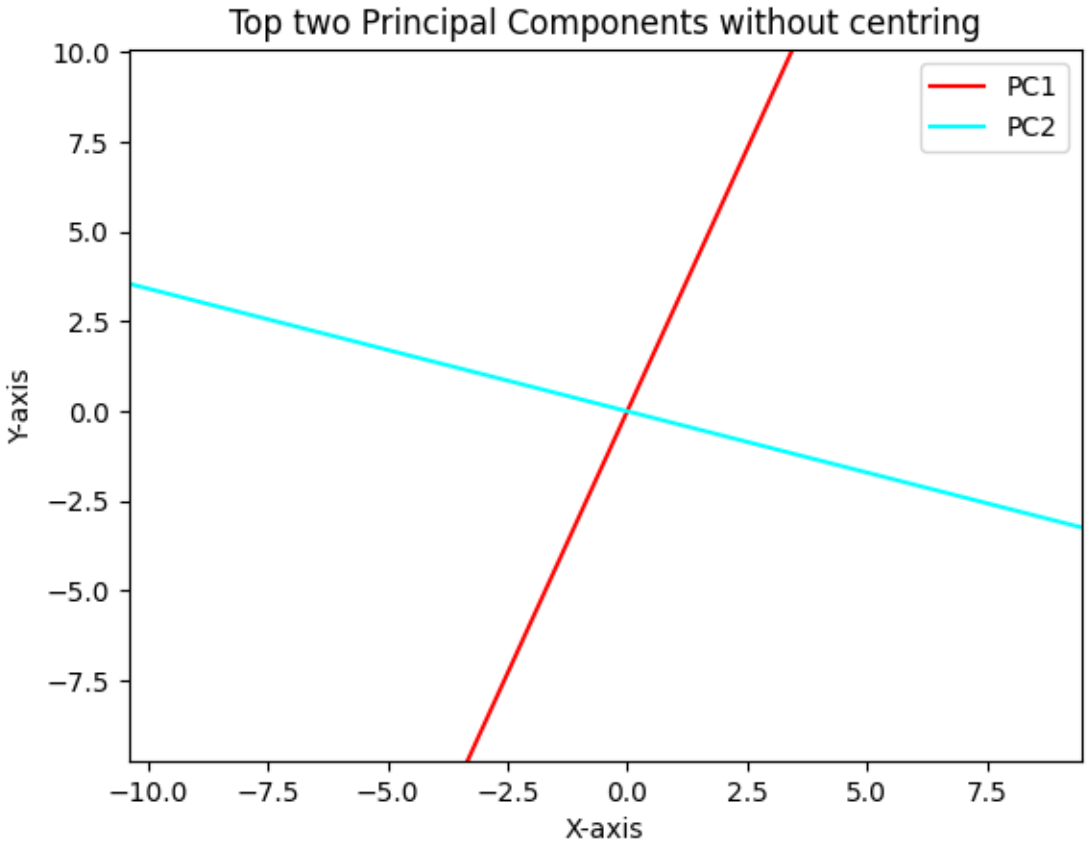
= corresponding eigen value * 100 /(sum of all eigen values)

Percentage variance explained by

Principal component 1 = 54.1780245288 %

Principal component 2 = 45.8219754711 %


Percentage of variance explained by top two PCs without centring
50
40
30
20
10
0
pc1 %
pc2 %


Top two Principal Components without centring
PC1
PC2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis

Q1.iii A)

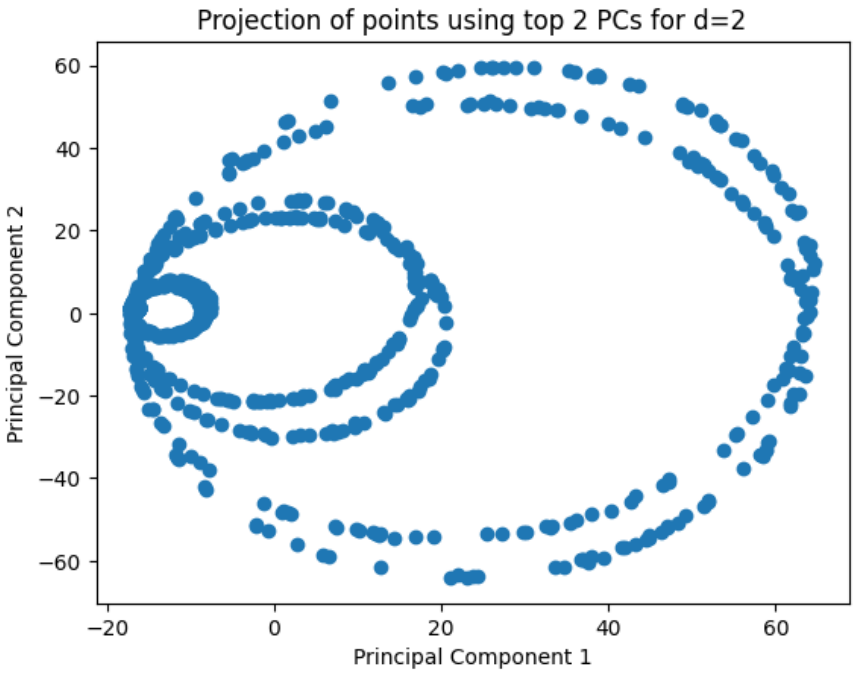
Step 1:- Calculate kernel matrix using given kernel function

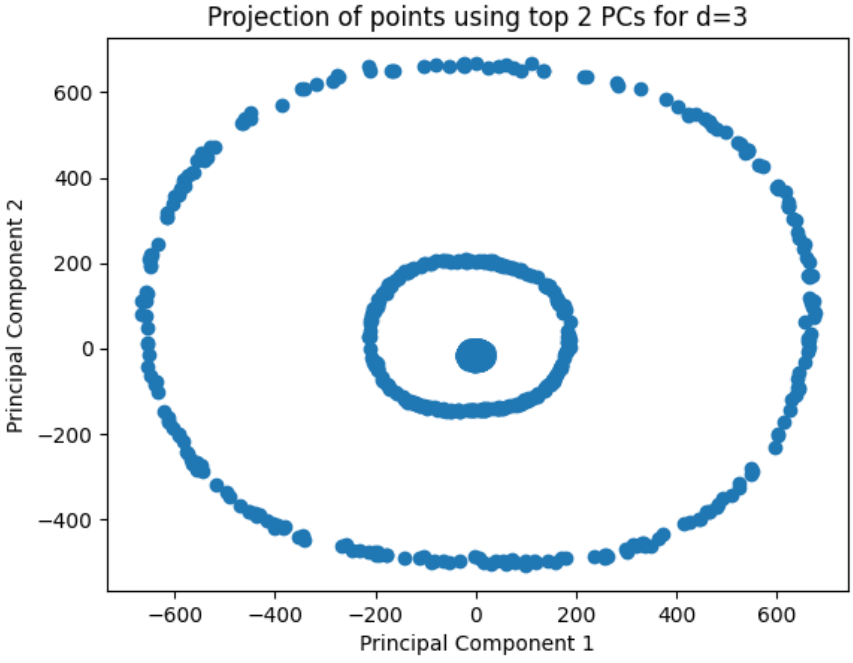
$$K(x, y) = (1 + x^T y)^d \text{ for } d = \{2, 3\}$$

Step 2:- Centered the kernel matrix and calculate eigen vectors sort them in decreasing order with corresponding eigen values.

Step 3 :- Take only real part of eigen values and eigen vectors and divide each eigen vector by lambda. Where lambda is square root of corresponding eigen value.

Step 4 :- Project all the points on top two principal components.





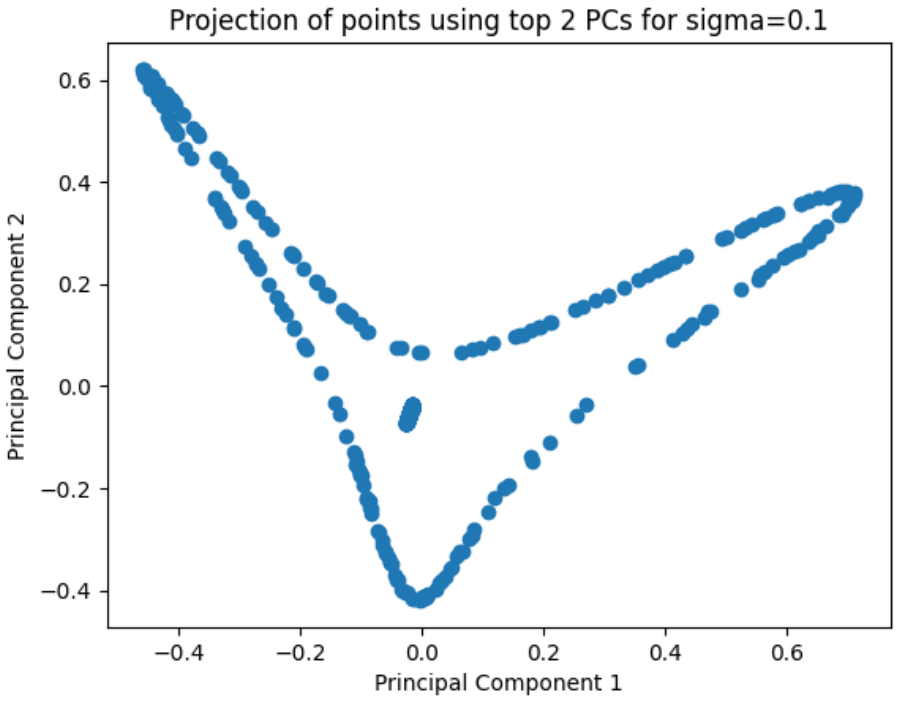
Q1.iii B)

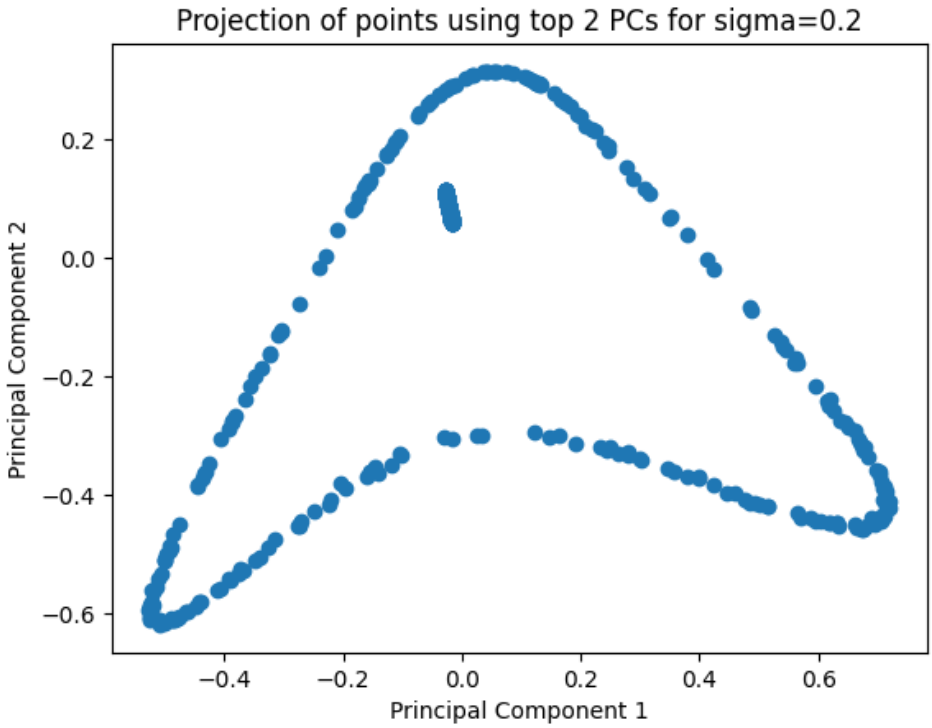
Calculate kernel matrix using given kernel function

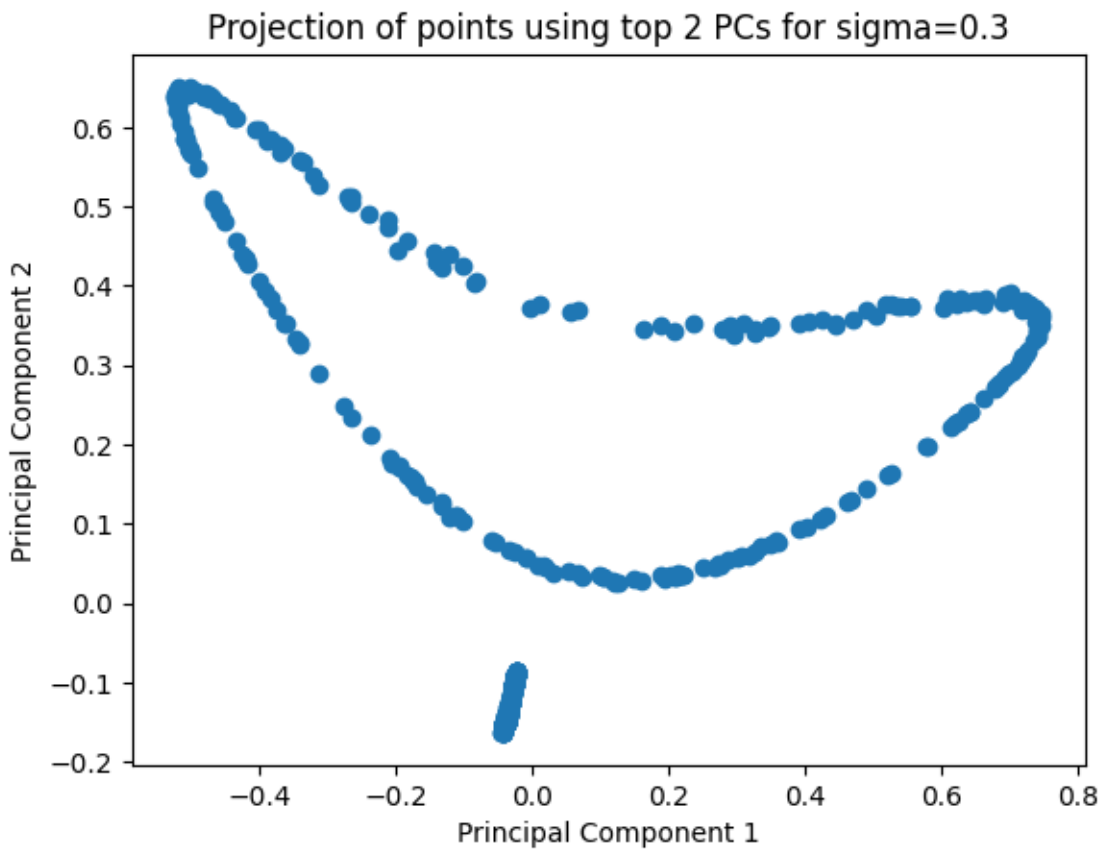
$$K(x, y) = \exp -(x-y)\ T\ (x-y)\ /2\sigma\text{^}2 \text{ for } \sigma = \{0.1, 0.2, \ldots, 1\}$$

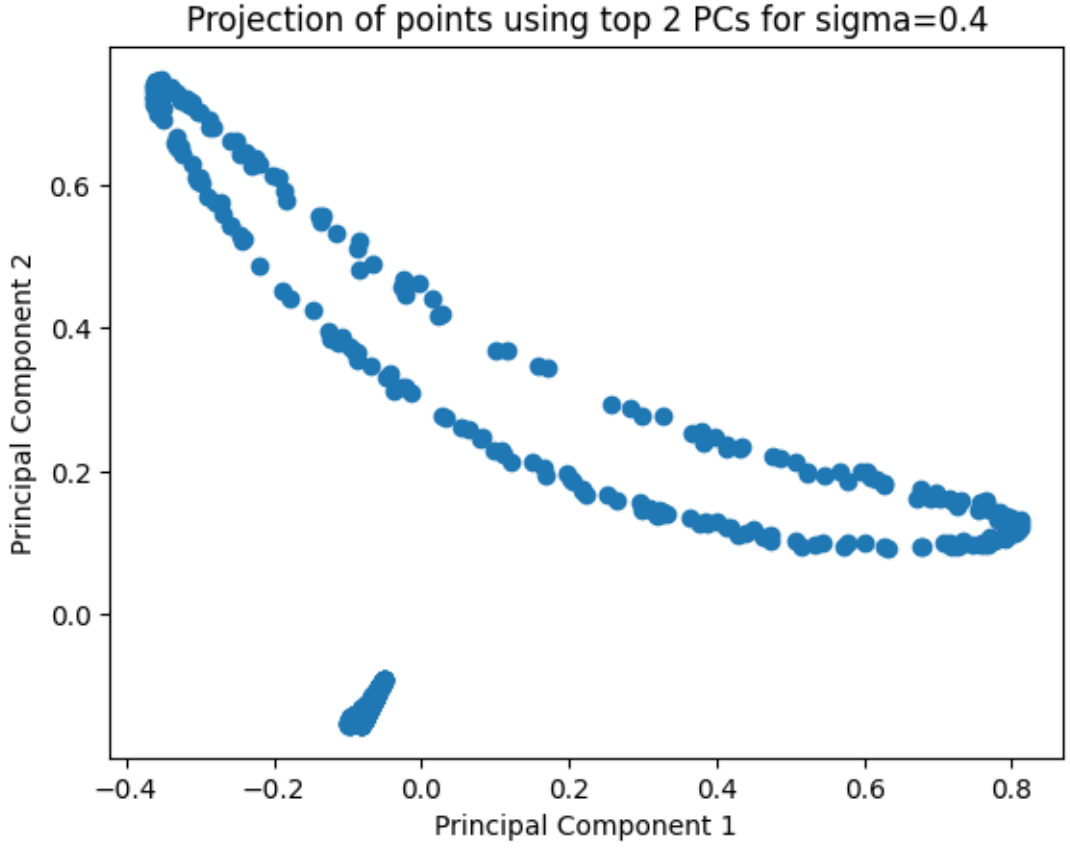
Steps are same as above example.

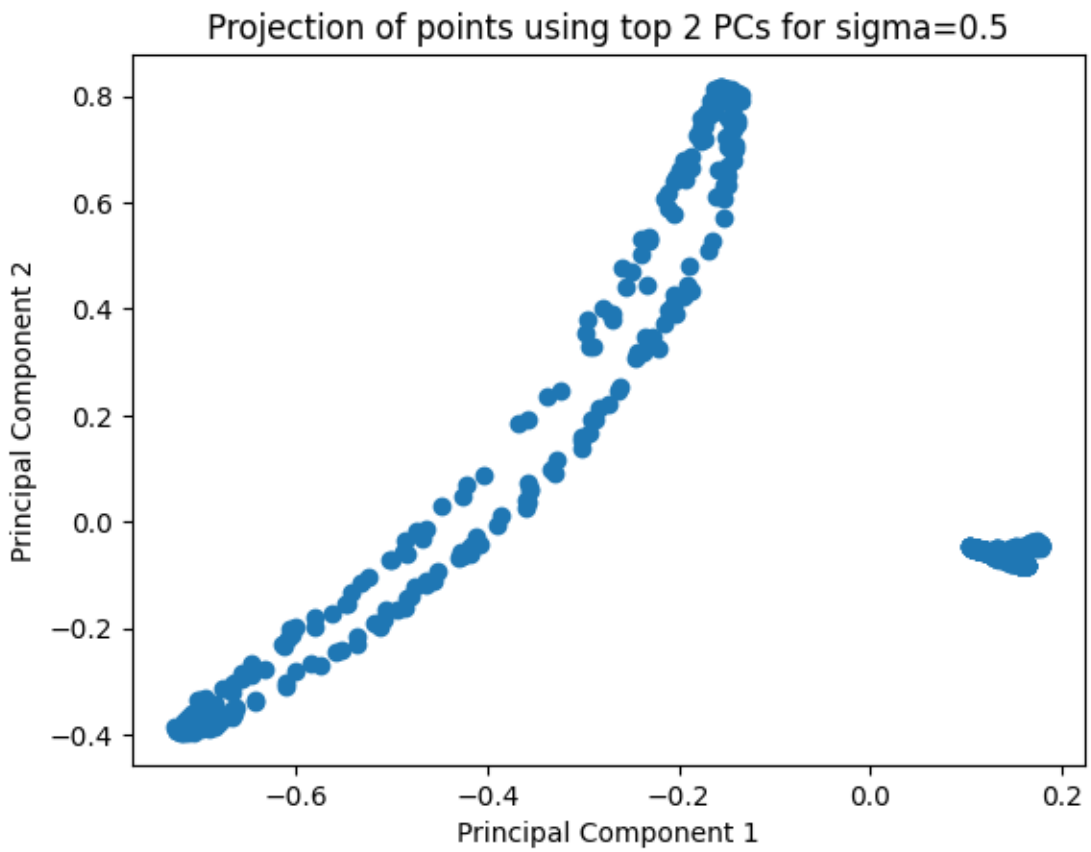
The results are as follows.

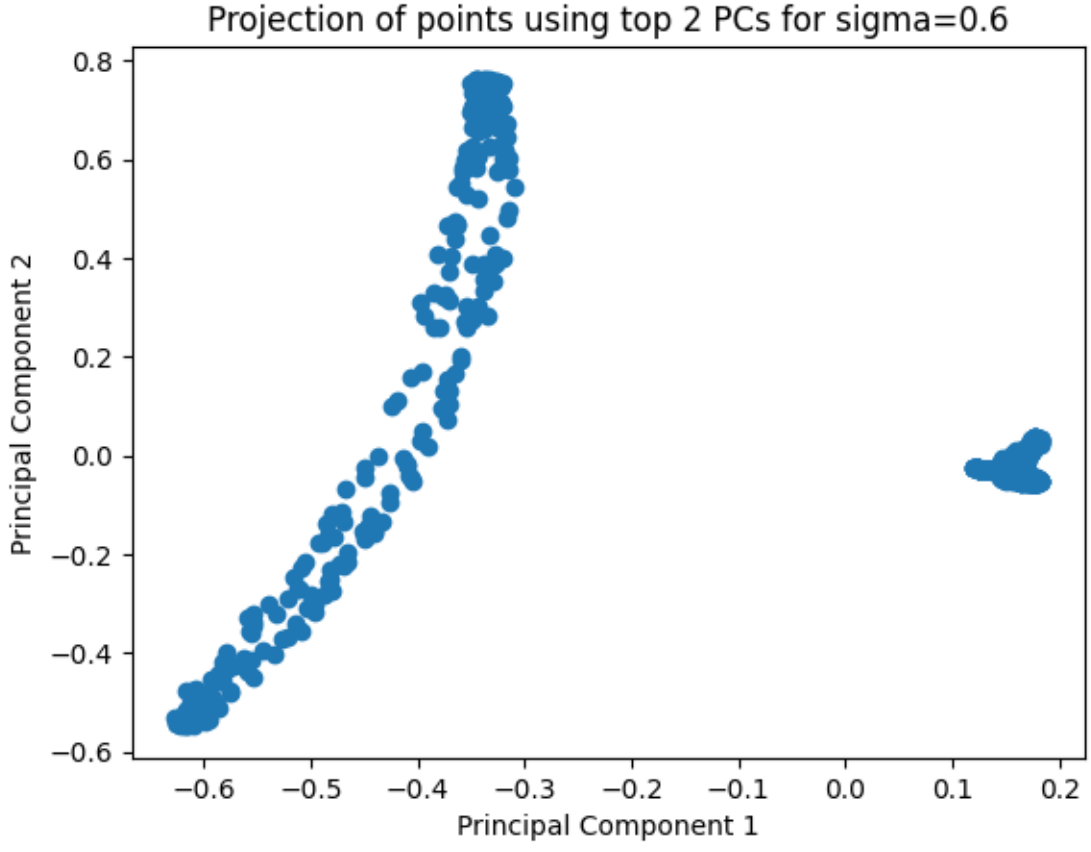


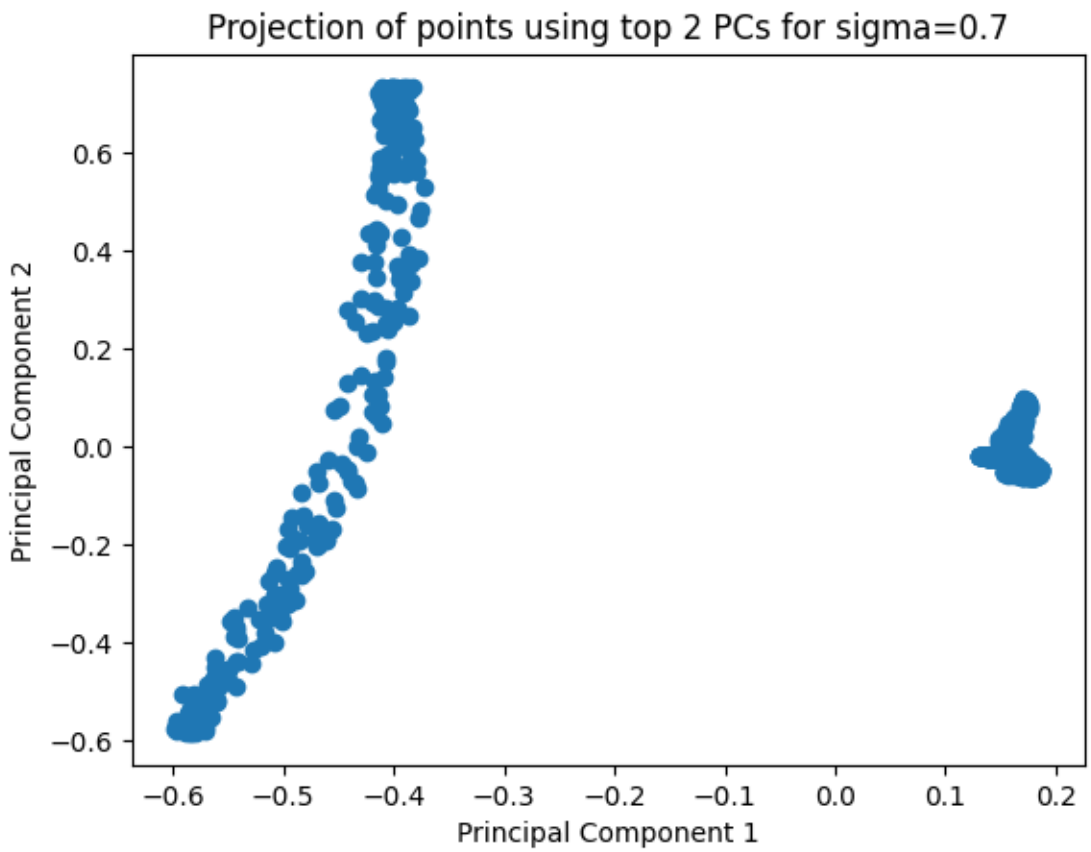


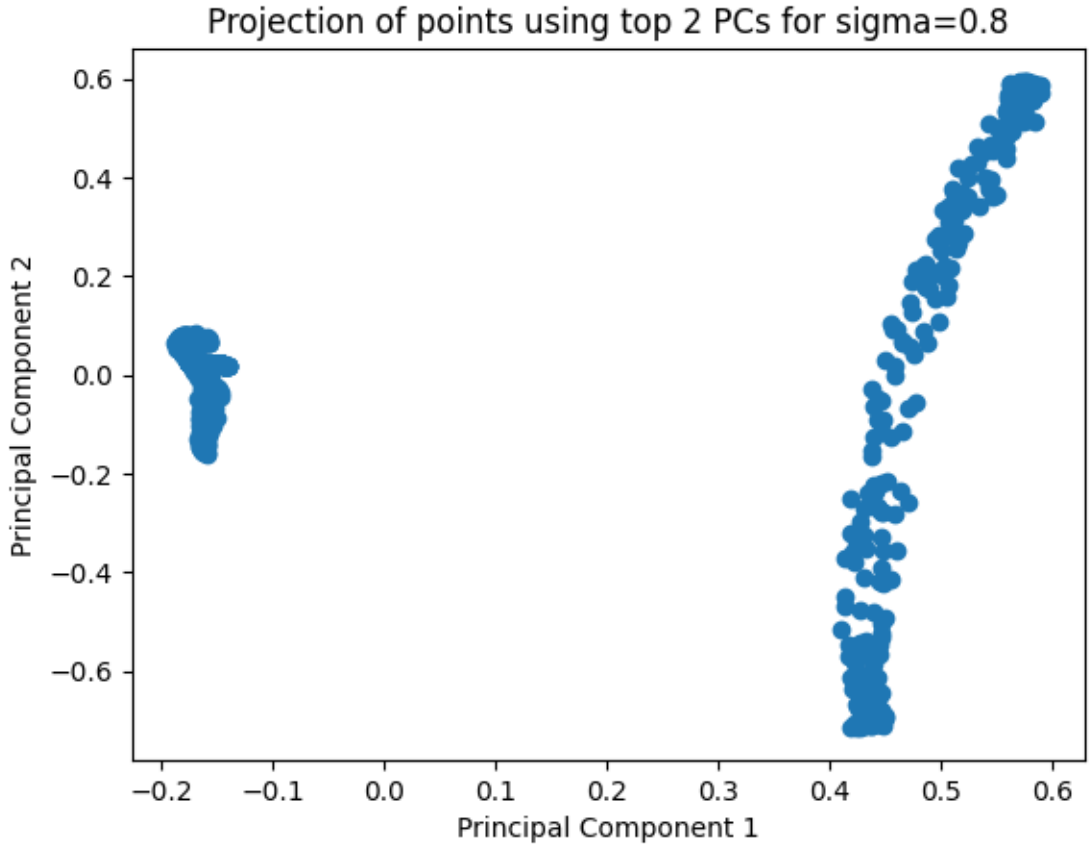

Projection of points using top 2 PCs for sigma=0.3
Principal Component 2
Principal Component 1
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

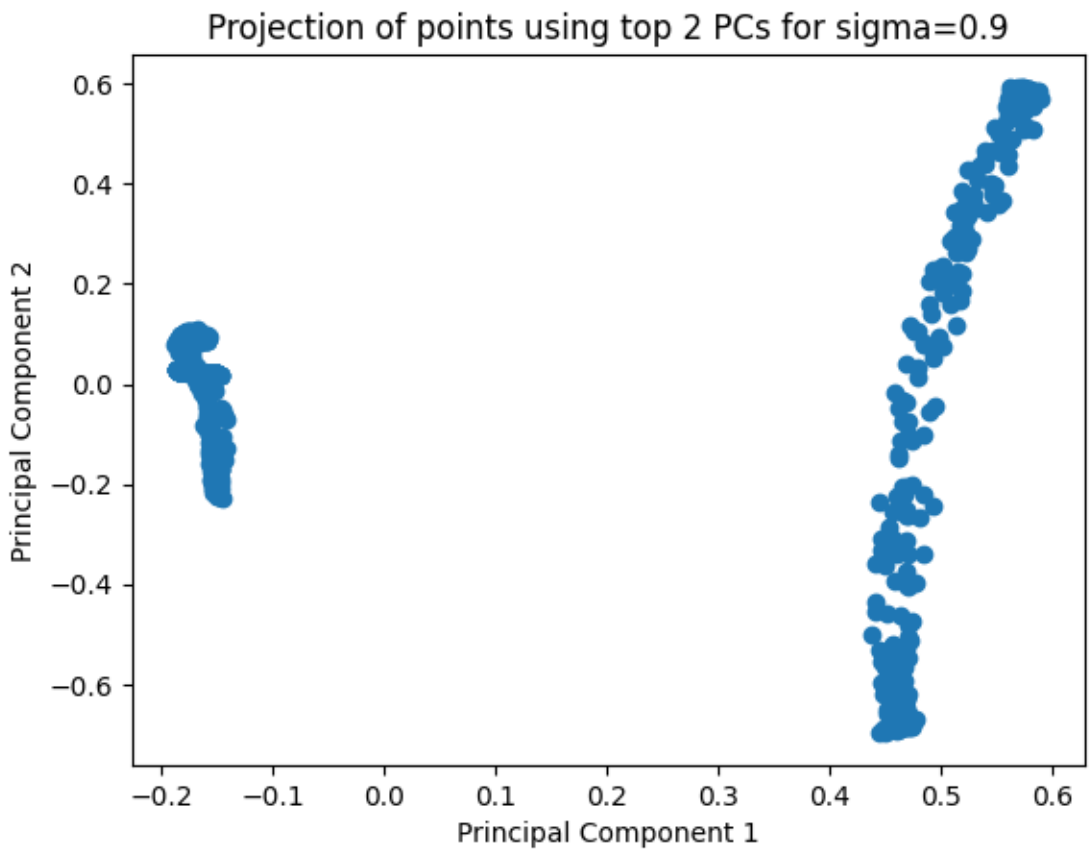

Projection of points using top 2 PCs for sigma=0.4
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

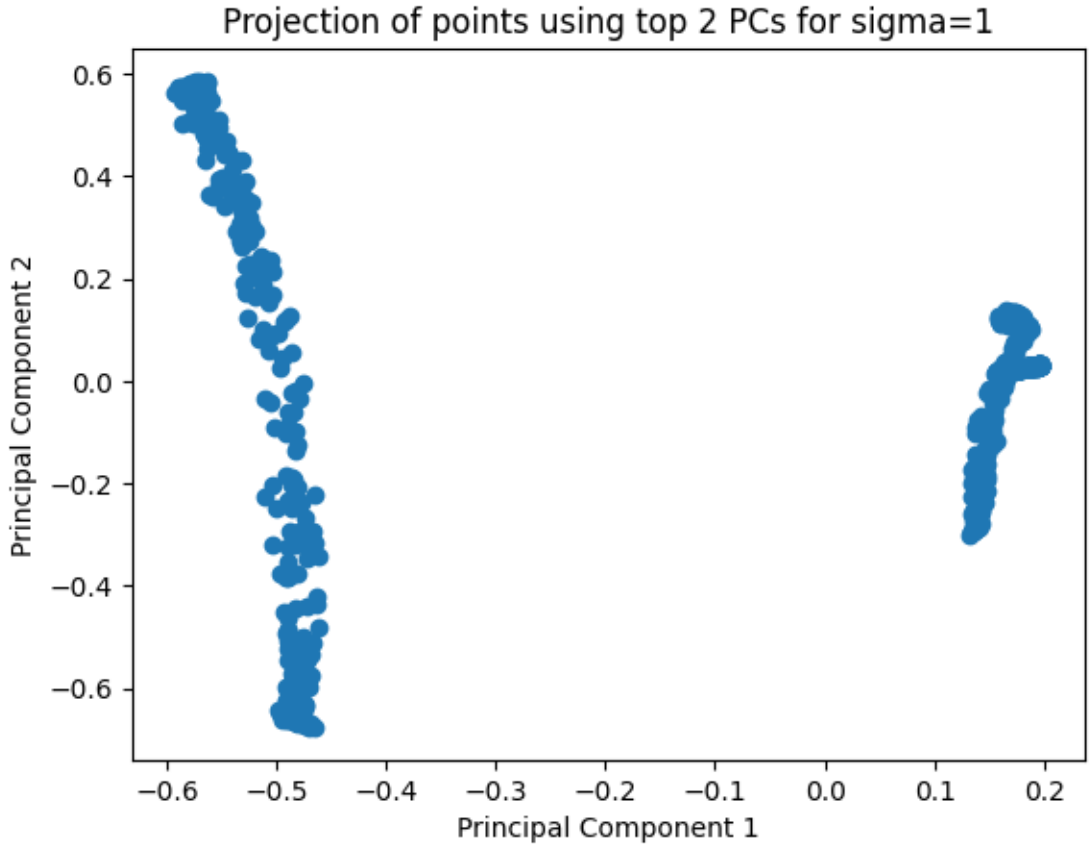

Projection of points using top 2 PCs for sigma=0.5
Principal Component 2
Principal Component 1
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.4
−0.2
0.0
0.2


Projection of points using top 2 PCs for sigma=0.6
Principal Component 2
Principal Component 1
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2


Projection of points using top 2 PCs for sigma=0.7
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2


Projection of points using top 2 PCs for sigma=0.8
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6


Projection of points using top 2 PCs for sigma=0.9
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6


Projection of points using top 2 PCs for sigma=1
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2

Q 1 iv)

Polynomial kernel for d=2 is best choice among given kernel choices because it gives more accurate clusters than the other kernels. We can see graph plotted in other questions which illustrate that polynomial kernel for d=2 is better choice.

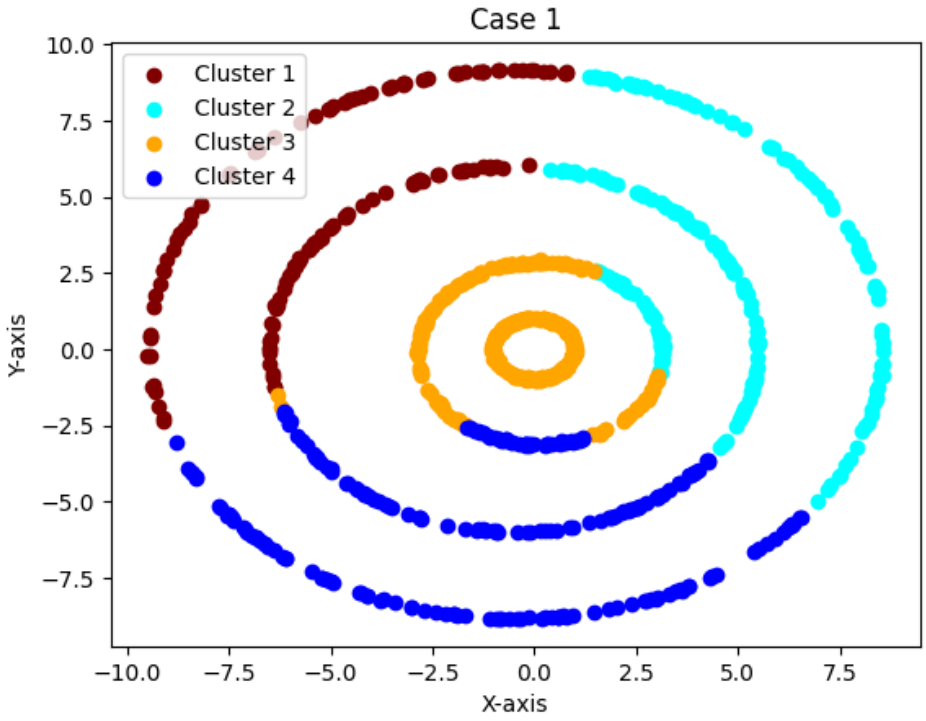
Q 2 i)

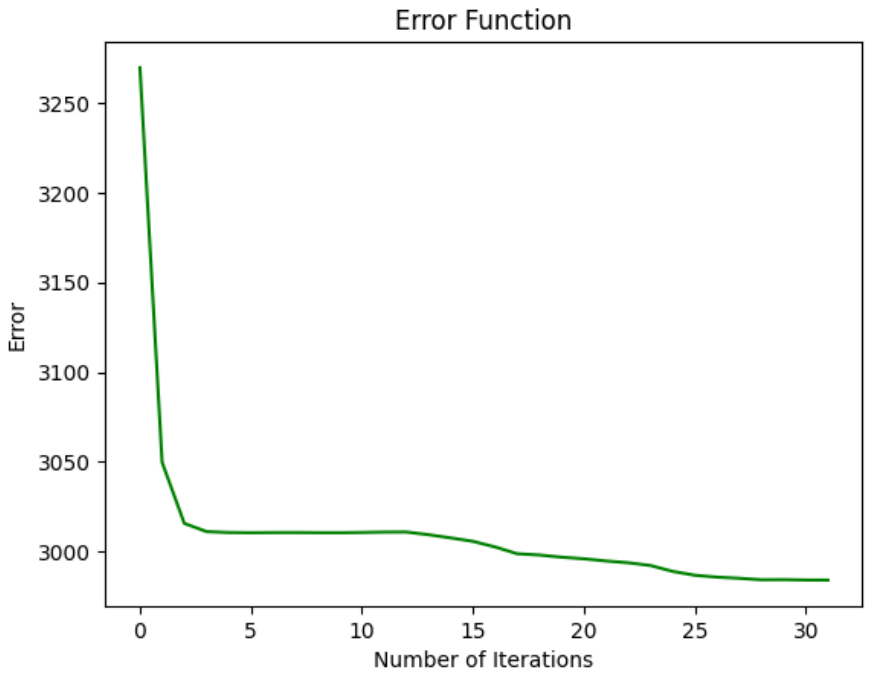
Step 1 :- Take random points as mean for each of four clusters and for each case.

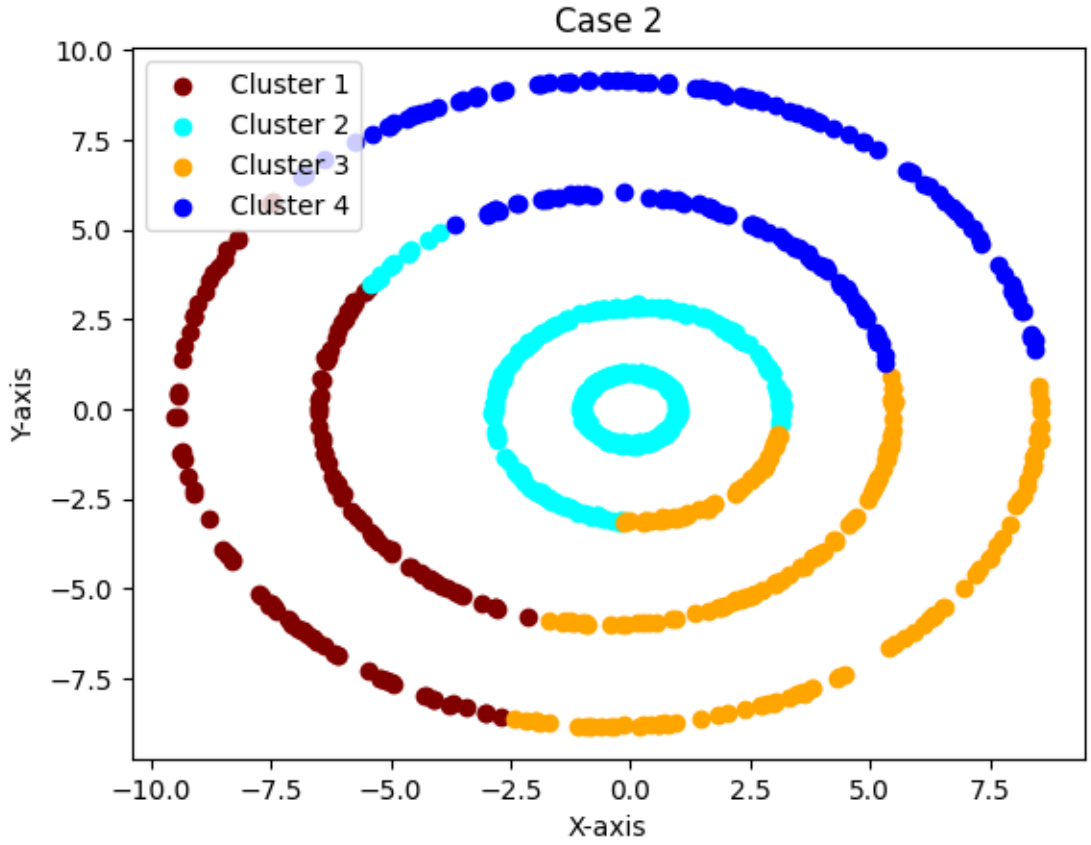
Step 2 :- Run K-means algorithm until convergence. That is until no point jumps from one cluster to other cluster. Once the condition is reached stop the algorithm.

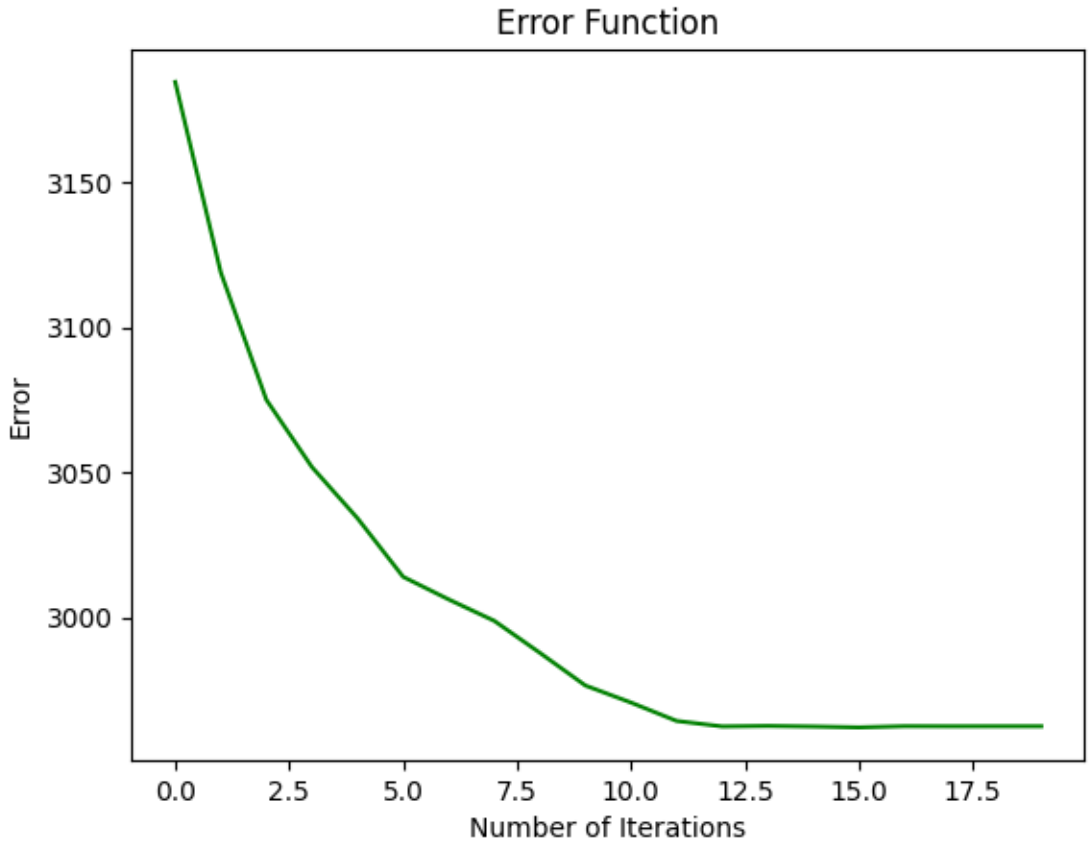
Step 3 :- Plot the error function with respect to number of iterations. Error function is just sum of distances of points from the assigned clusters mean.

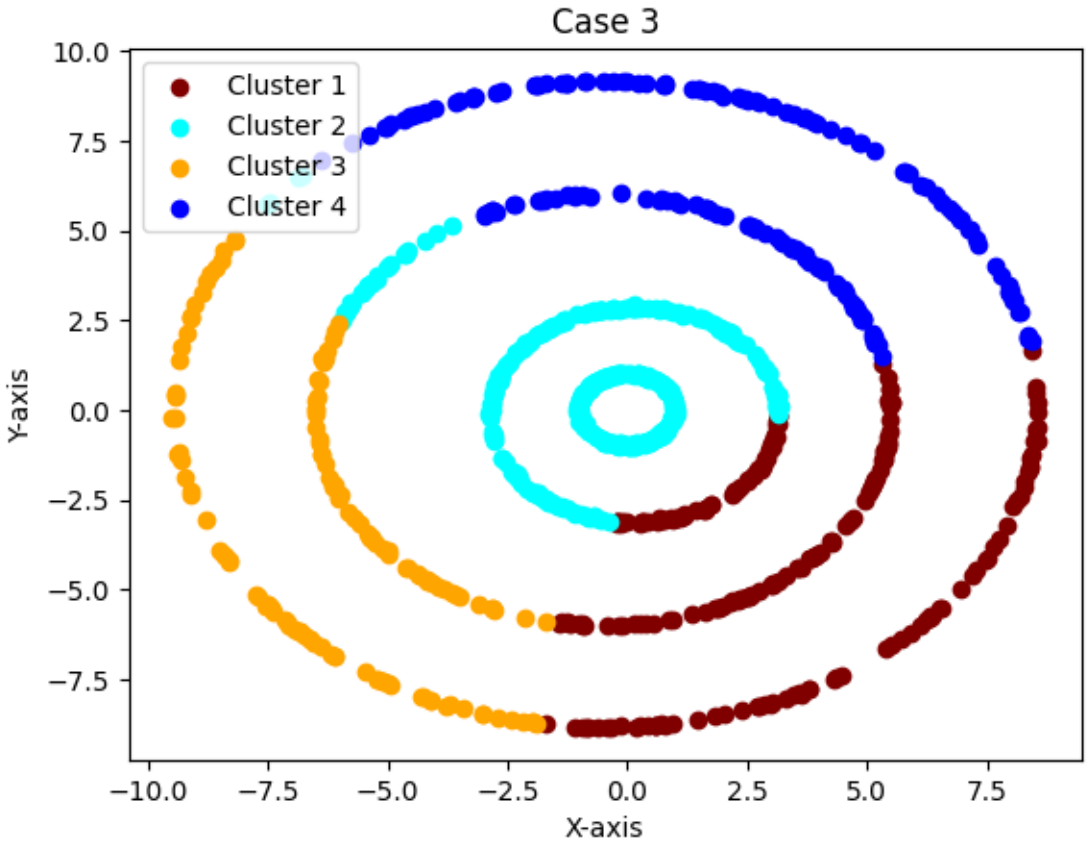
Following are the results obtained.

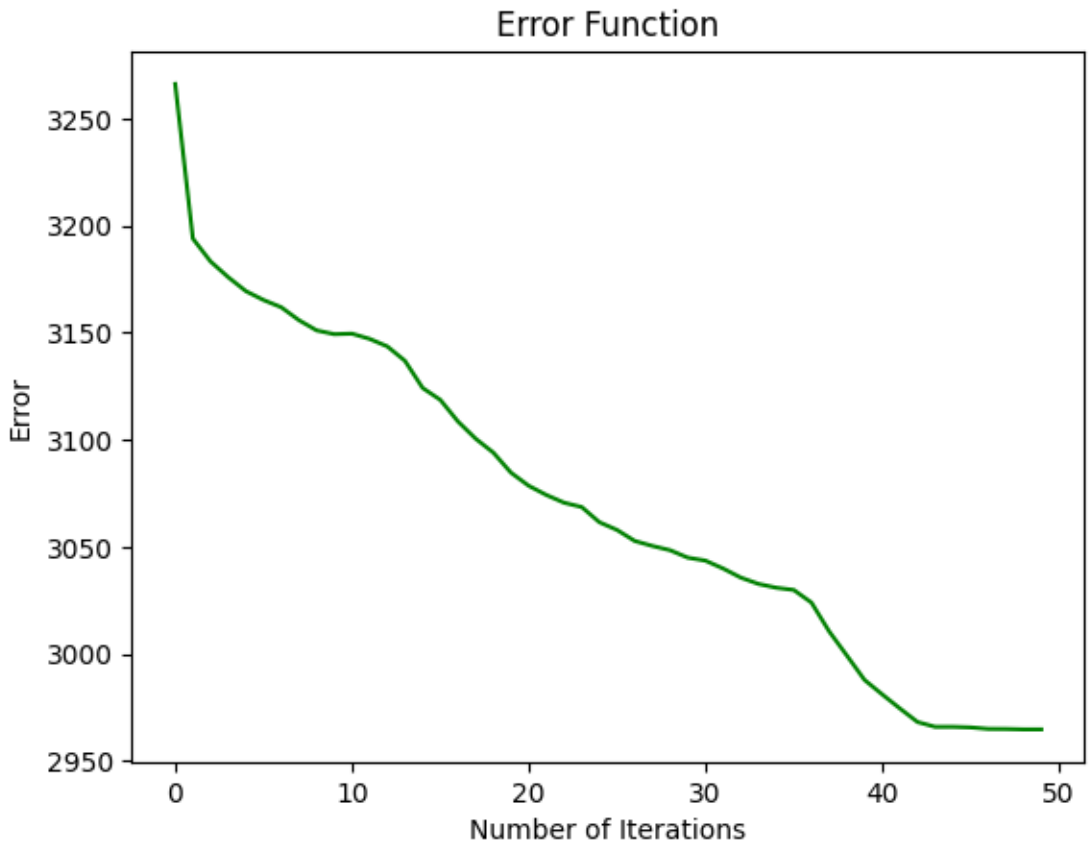


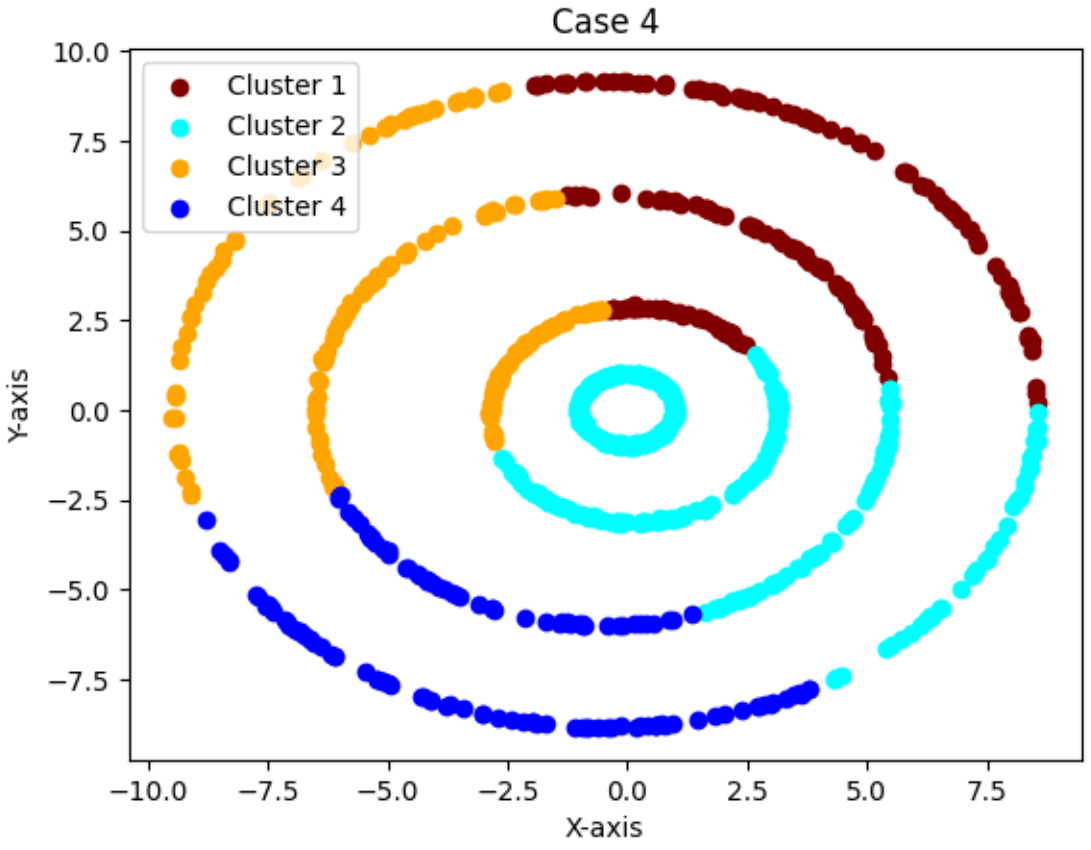

Error Function
Error
Number of Iterations
3250
3200
3150
3100
3050
3000
0
5
10
15
20
25
30

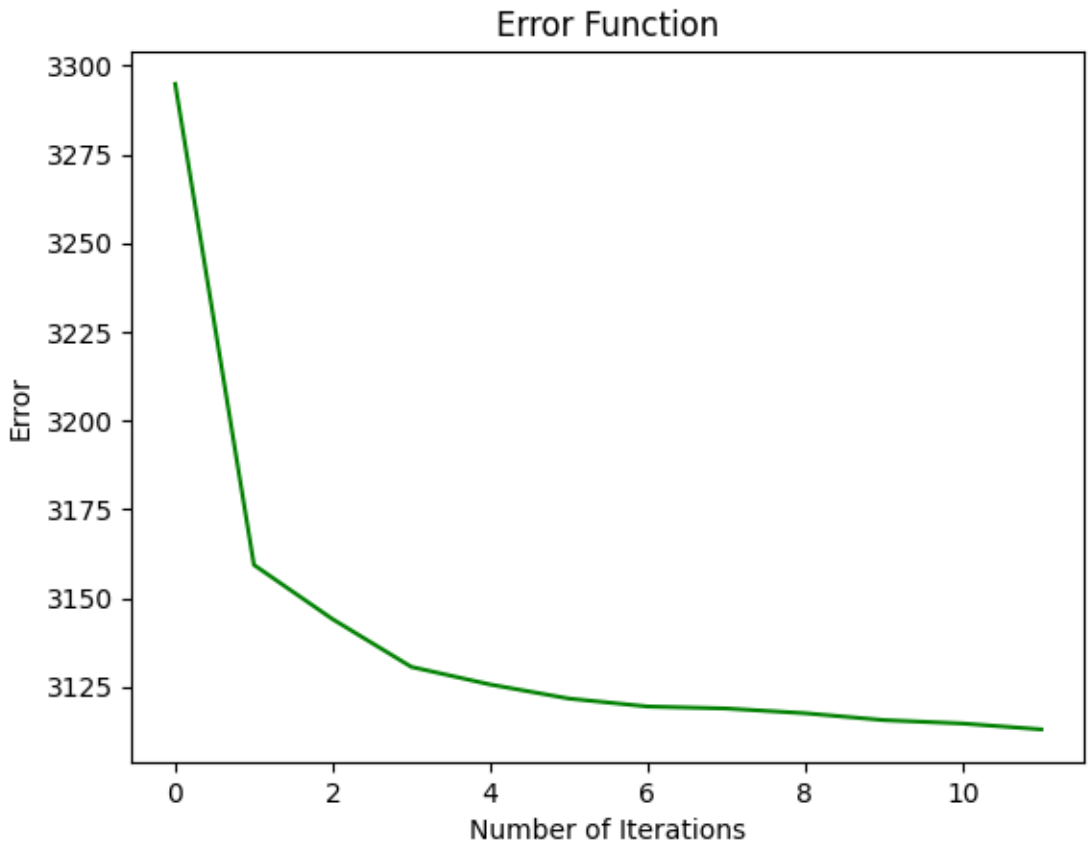

Case 2
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Y-axis
X-axis
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

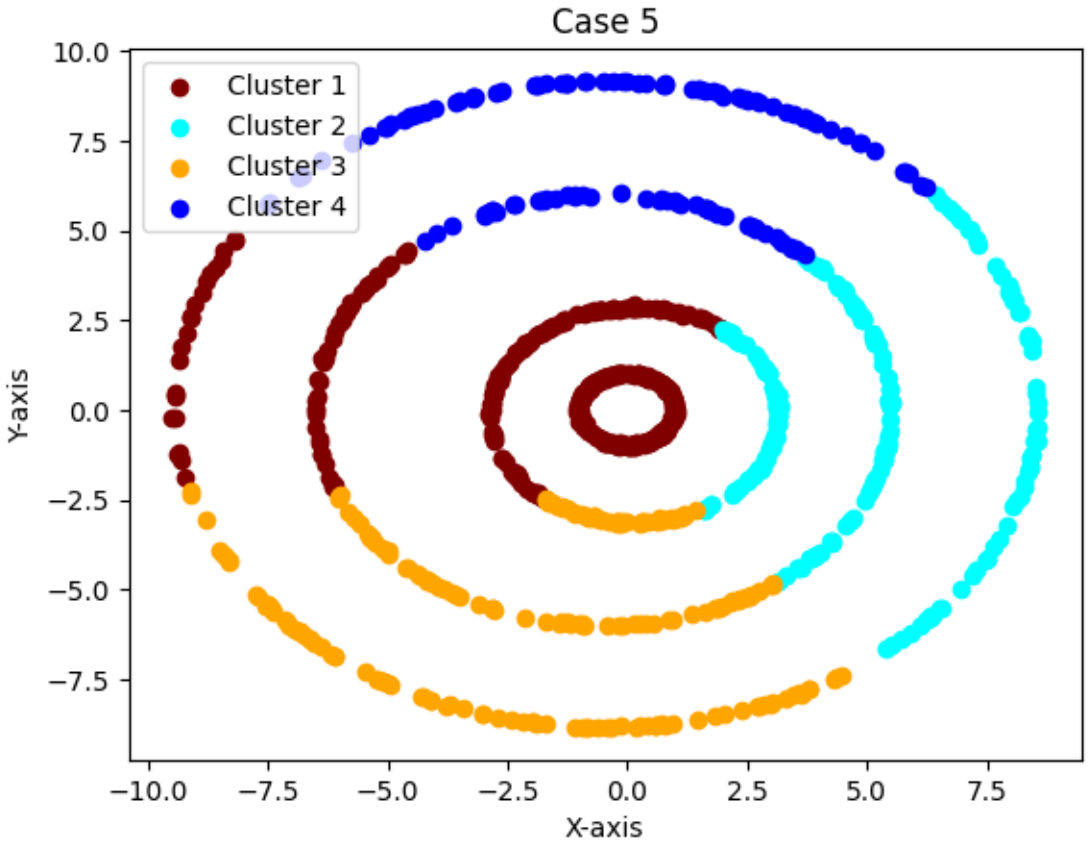

Error Function
Error
3150
3100
3050
3000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Number of Iterations

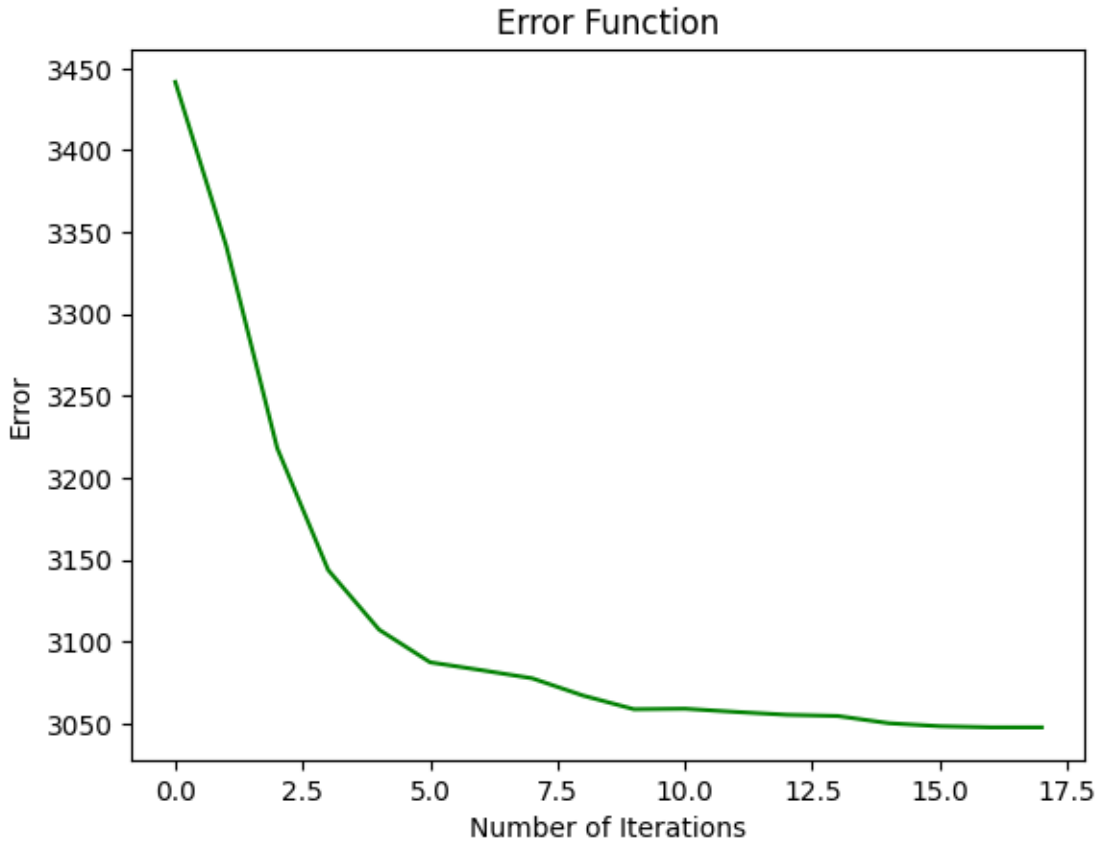

Case 3
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Y-axis
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


Error Function
Error
Number of Iterations
3250
3200
3150
3100
3050
3000
2950
0
10
20
30
40
50


Case 4
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Y-axis
X-axis
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


Error Function
Error
Number of Iterations
3300
3275
3250
3225
3200
3175
3150
3125
0
2
4
6
8
10


Case 5
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Y-axis
X-axis
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

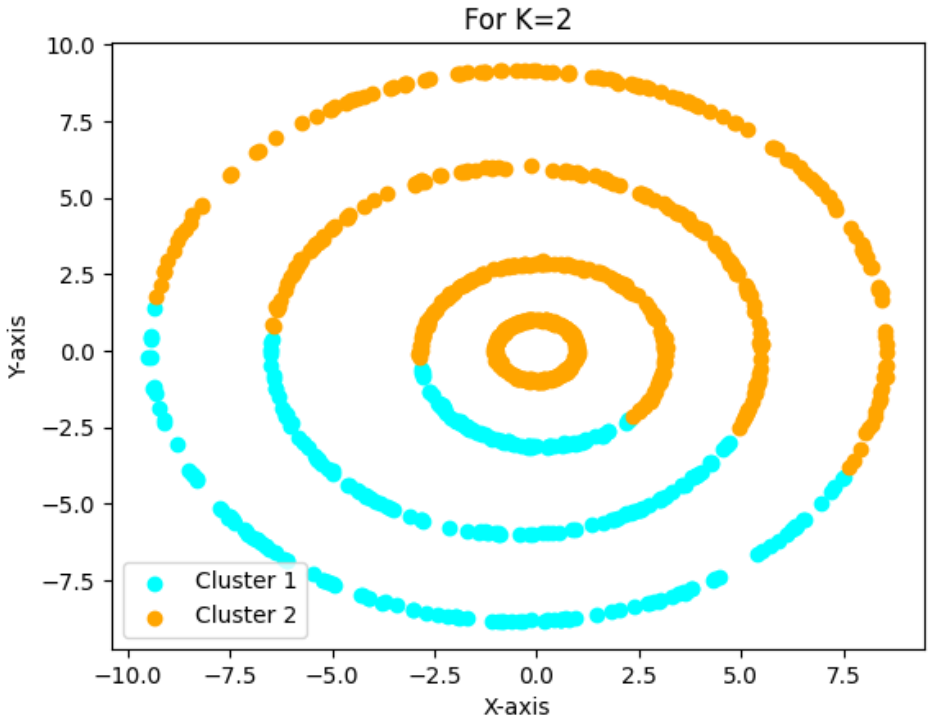


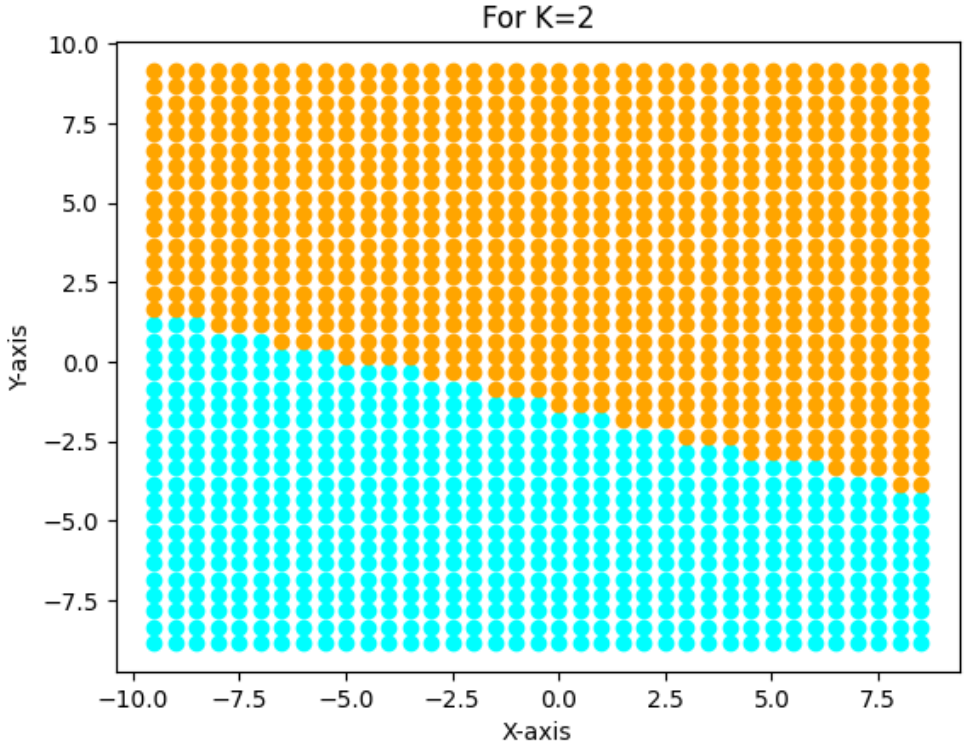
Q 2 ii)

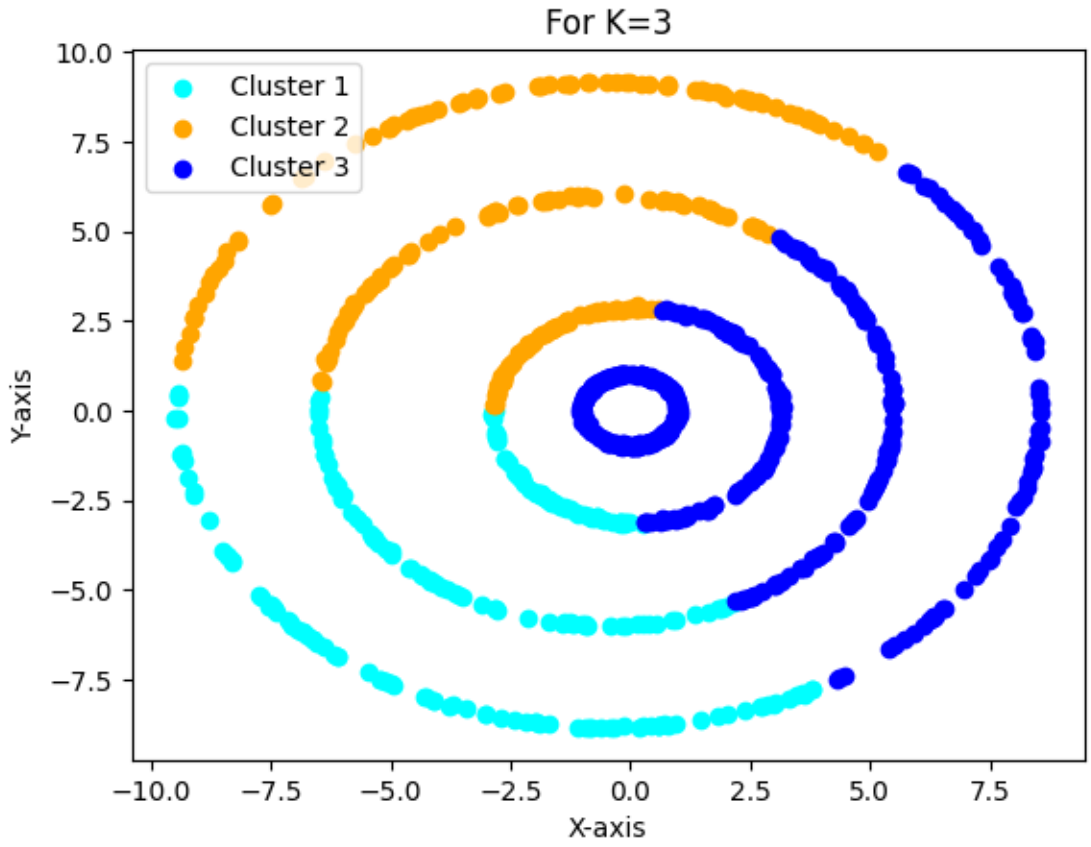
Step 1 :- Take random points as mean for each of four clusters and are same for k=2 to k=5.

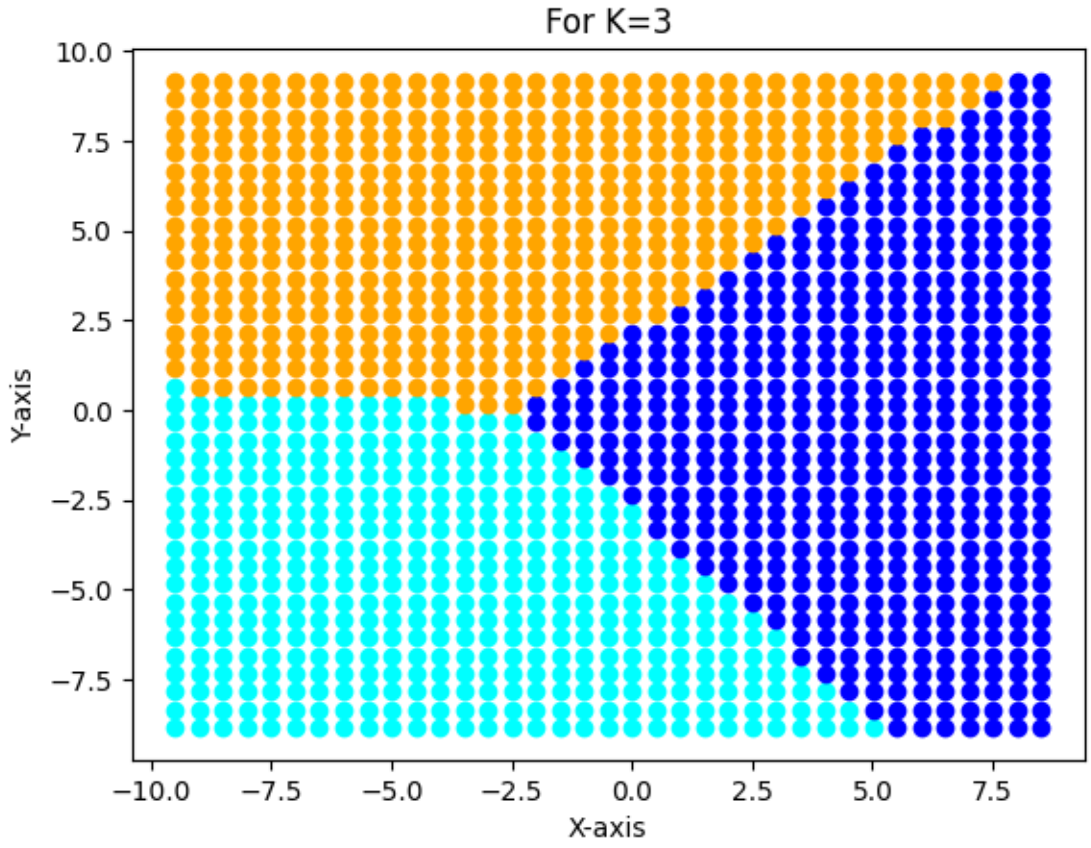
Step 2 :- Run K-means algorithm until convergence. That is until no point jumps from one cluster to other cluster. Once the condition is reached stop the algorithm.

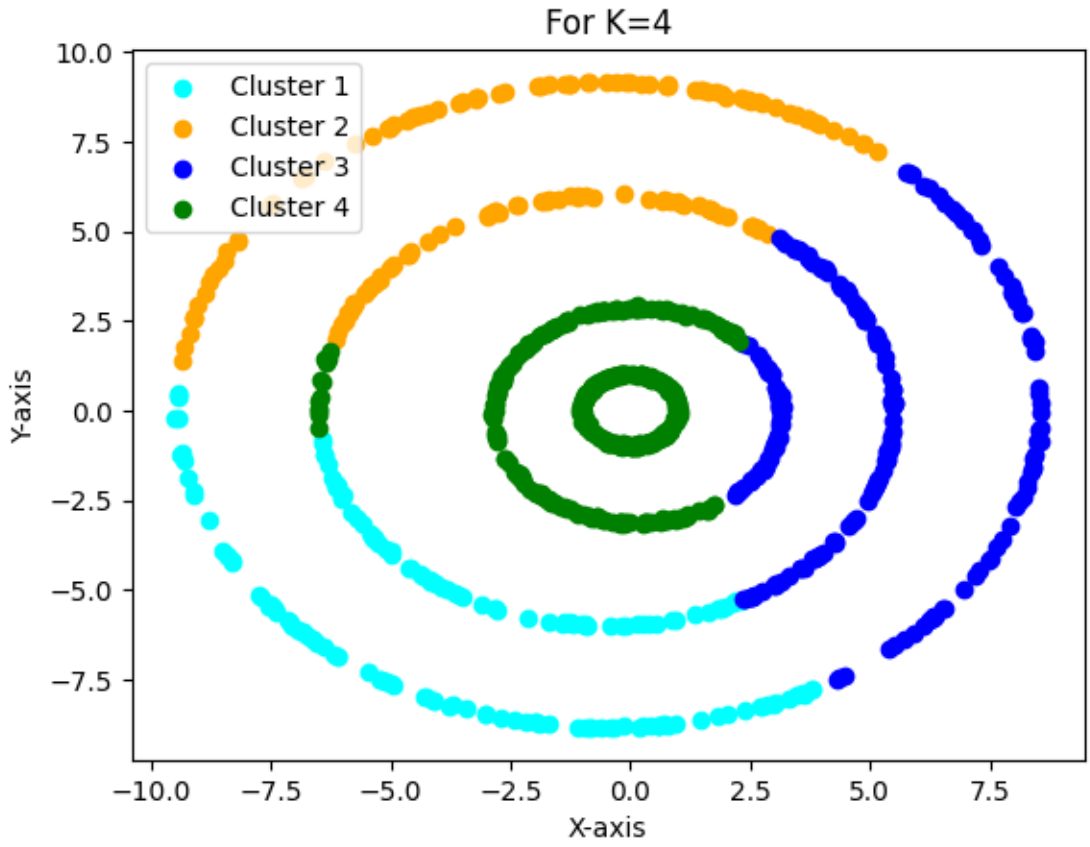
Step 3 :- Plot the Voronoi regions for k=2,3,4,5.

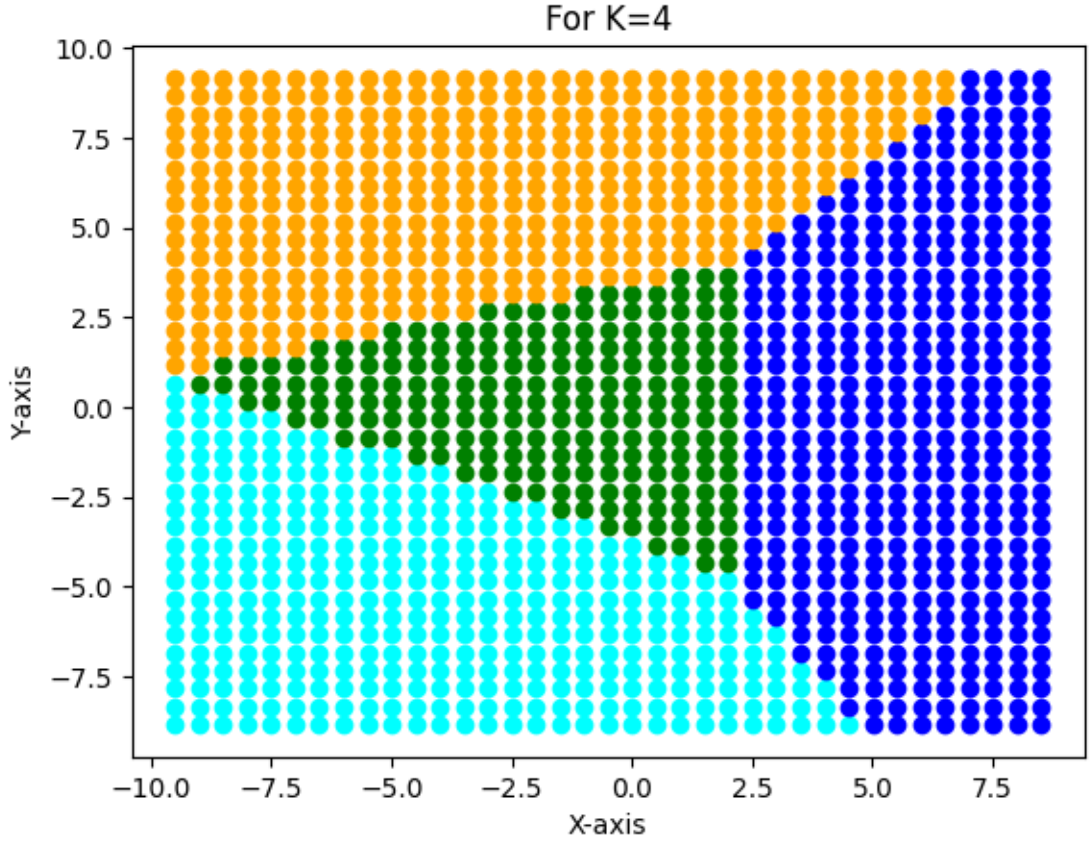

For K=2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
Cluster 1
Cluster 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis

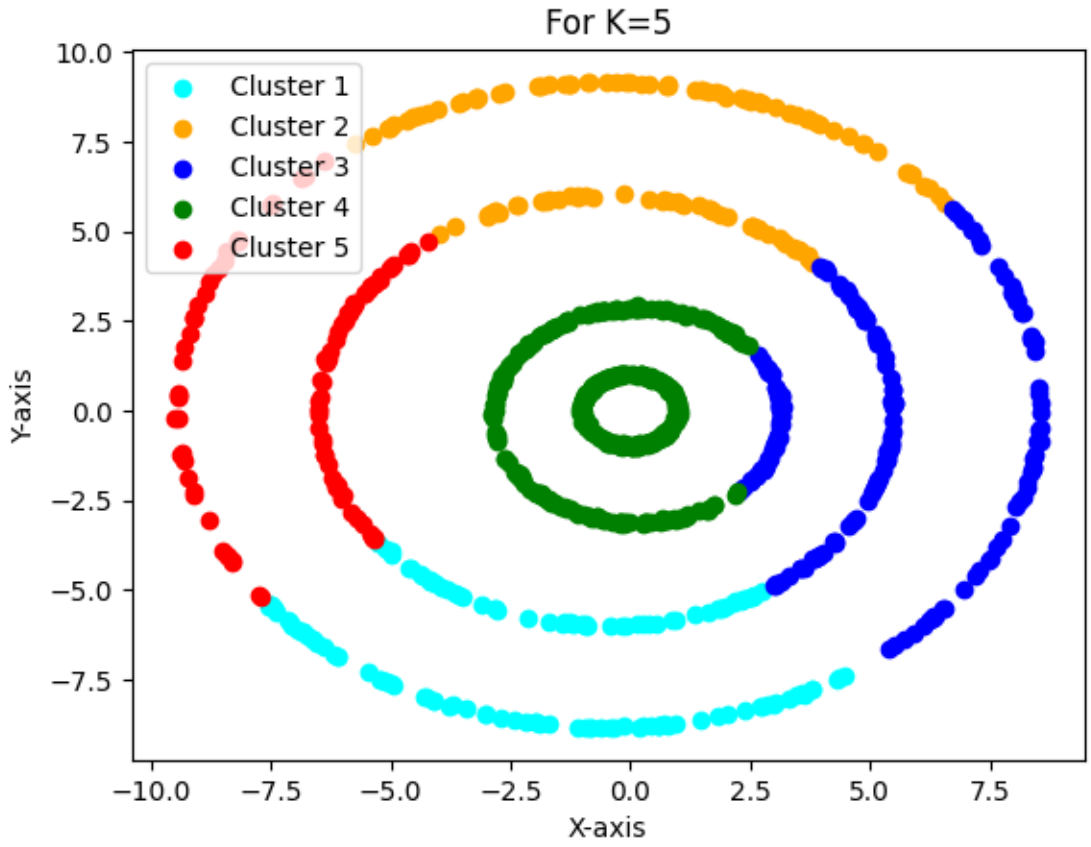

For K=2
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis

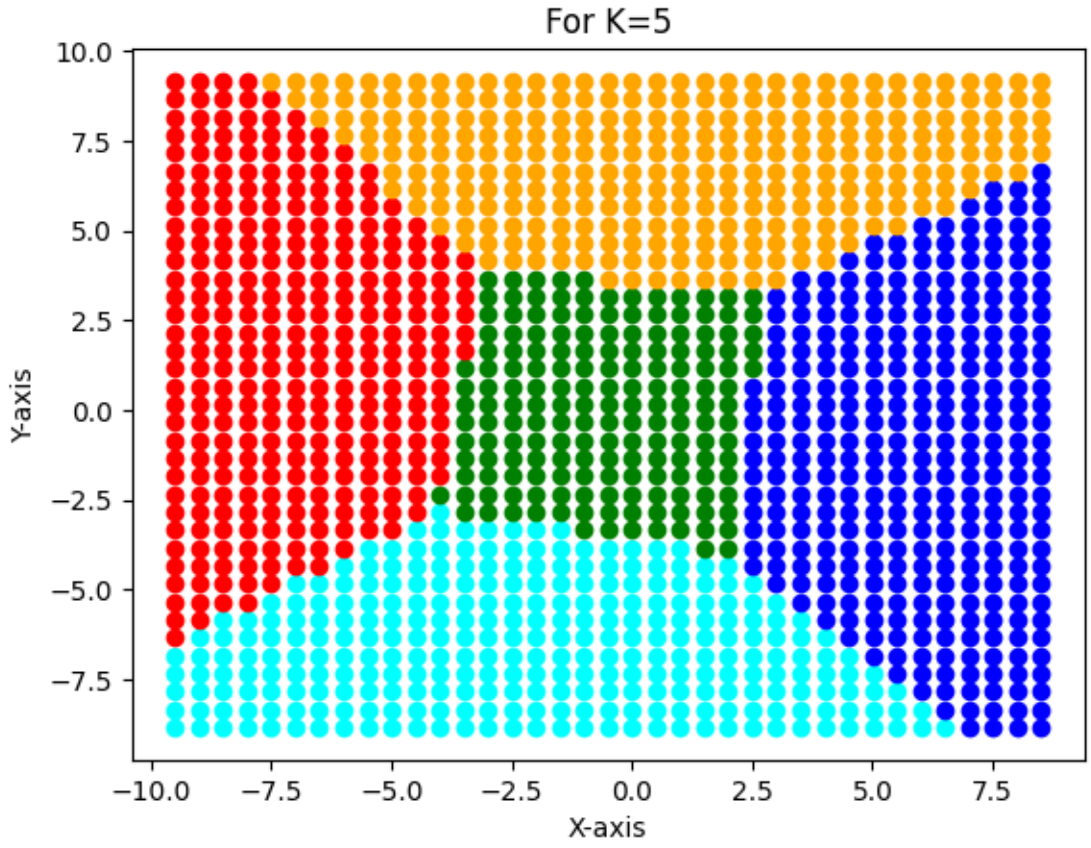

For K=3
Cluster 1
Cluster 2
Cluster 3
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


For K=3
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


For K=4
Cluster 1
Cluster 2
Cluster 3
Cluster 4
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


For K=4
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


For K=5
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis


For K=5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y-axis
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X-axis

Q 2 iii)

Step 1 :- Choose kernel map and compute kernel matrix.

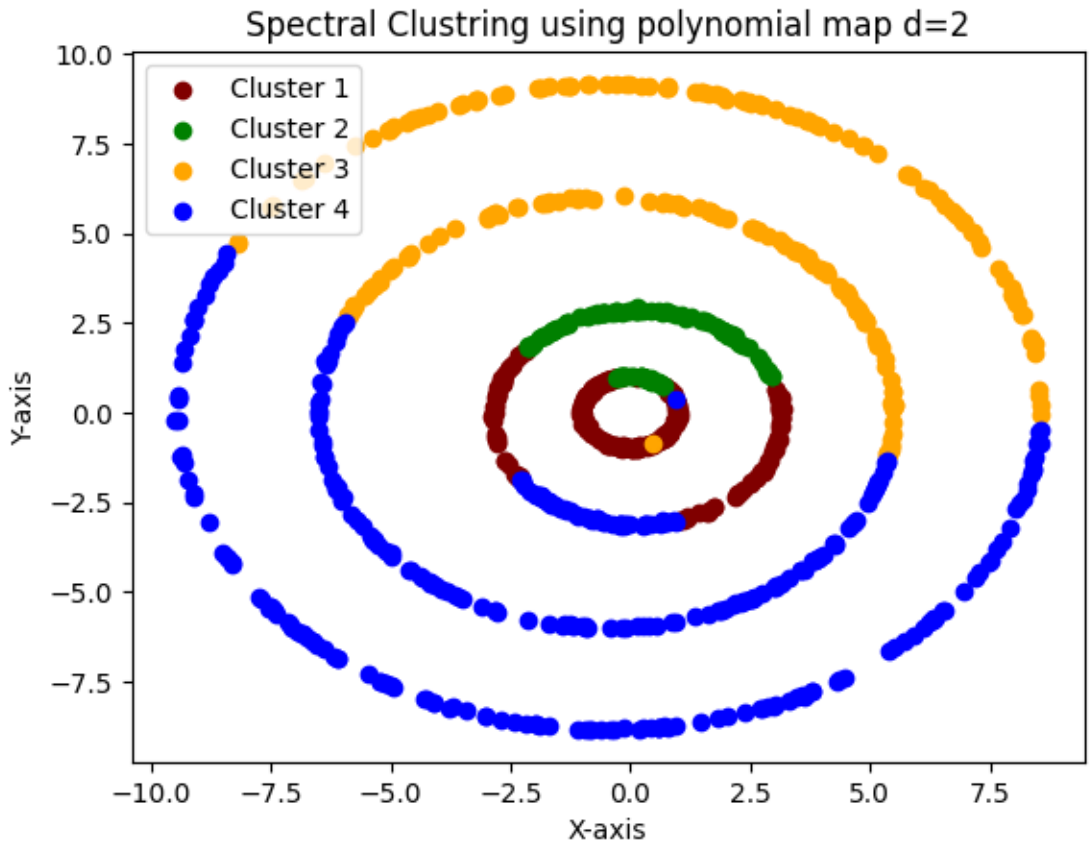
Polynomial map K(x, y) = (1 + x ^T y) d for d = {2, 3}

Step 2 :- Calculate eigen value and eigen vector for kernel matrix. Sort them in decreasing order and remove imaginary part.

Step 3 :- Take top 4 eigen vectors that is principal components for further calculation.

Step 4:- Run the K-means algorithm on this higher dimensional data which will form cluster in higher dimension. Map these to original points in given data and plot on graph.

Following are results of spectral clustering.



For polynomial map of d=2 , we get clearly distinct clusters than the others map plotting's hence the polynomial map of d=2 is better choice.

Q 2 iv)

In this case, the green cluster is located at opposite directions also the blue is also scattered where as in polynomial map we get better clusters.