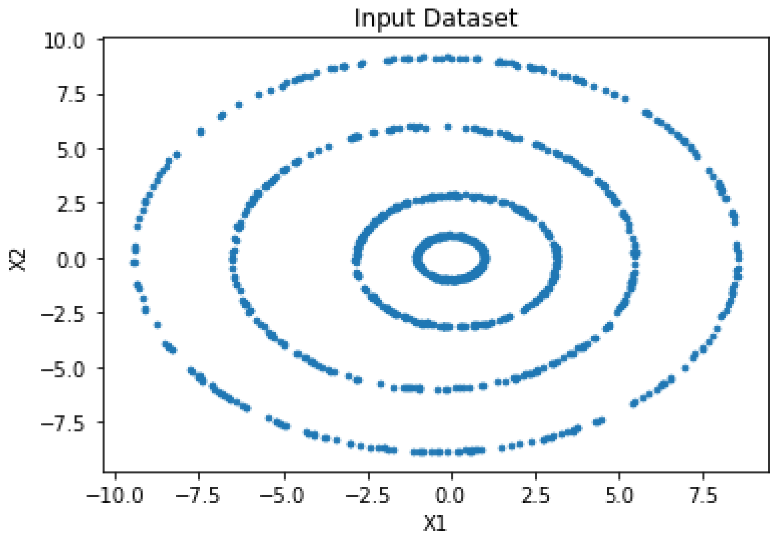
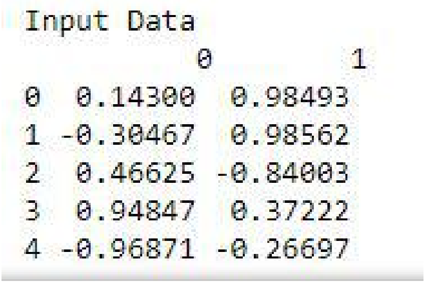
PRML ASSIGNMENT 1 REPORT

(1) You are given a data-set with 1000 data points each in $\mathbb{R}^2$.

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?

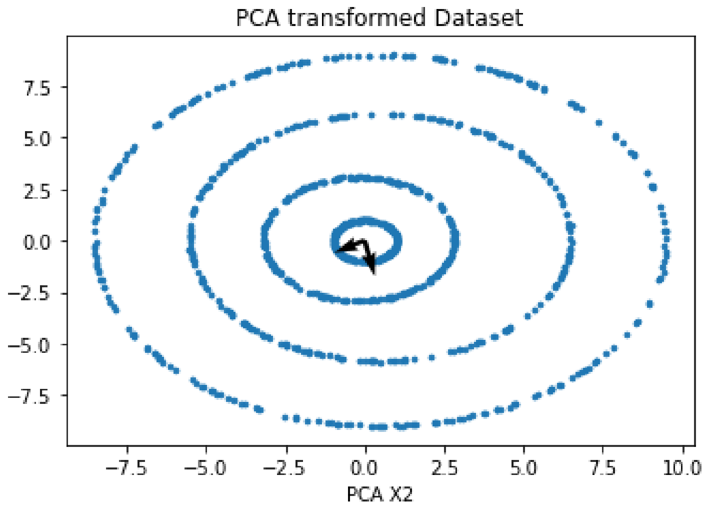
Input Dataset

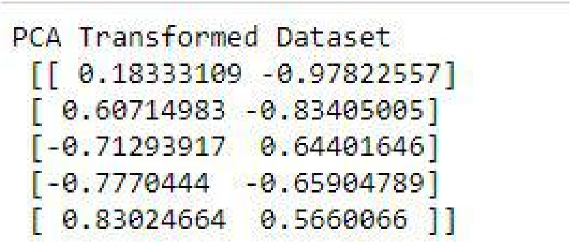




```
Input Data
           0        1
0  0.14300  0.98493
1 -0.30467  0.98562
2  0.46625 -0.84003
3  0.94847  0.37222
4 -0.96871 -0.26697
```

Steps Performed in PCA:
1.Calculate Mean; Perform Centering, (ie subtract mean)
2.Find Covarience Matrix
3.Find Eigen Value and Eigen Vector for Covarience Matrix
4.Choosing components and forming a Feature vector. Principal components==Eigen Vectors (Feature vector is a matrix that has eigenvectors of the components that we decide to keep; first sort wrt eigValues,then choose k highest. Here anyway d=2)
5.Transforming original dataset.(xTrans.omega)omega
6.Plot Result and feature vector





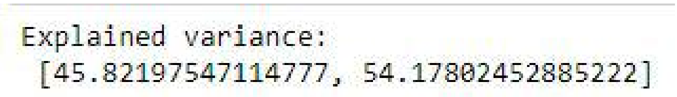
```
PCA Transformed Dataset
 [[ 0.18333109 -0.97822557]
 [ 0.60714983 -0.83405005]
 [-0.71293917  0.64401646]
 [-0.7770444  -0.65904789]
 [ 0.83024664  0.5660066 ]]
```

Explained variance is a statistical measure of how much variation in a dataset can be attributed to each of the principal components (eigenvectors) generated by the principal component analysis (PCA) method.
Refers to the amount of variability in a data set that can be attributed to each individual principal component.

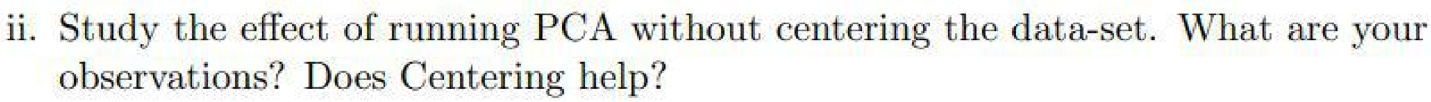
Explained variance can be represented as a function of ratio of related eigenvalue and sum of eigenvalues of all eigenvectors. Let’s say that there are N eigenvectors, then the explained variance for each eigenvector (principal component) can be expressed the ratio of eigenvalue of related eigenvalue λi and sum of all eigenvalues (λ1+λ2+…+λn)

(Refered:https://vitalflux.com/pca-explained-variance-concept-python-example/)

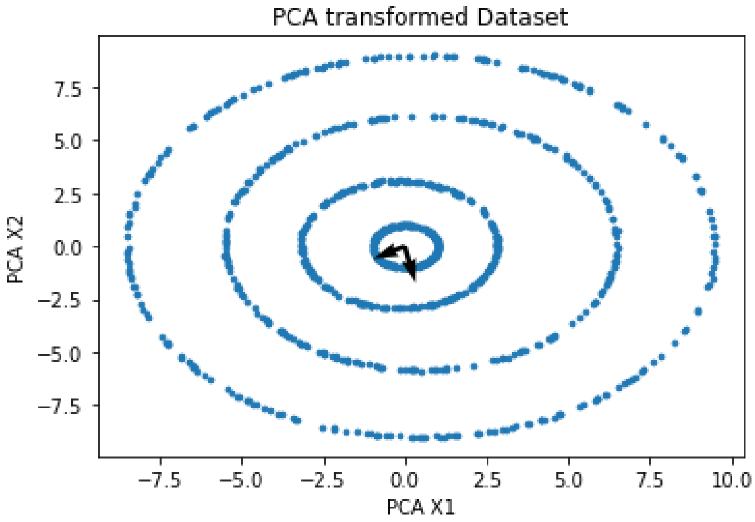
The total variance is the sum of variances of all individual principal components.
The fraction of variance explained by a principal component is the ratio between the variance of that principal component and the total variance

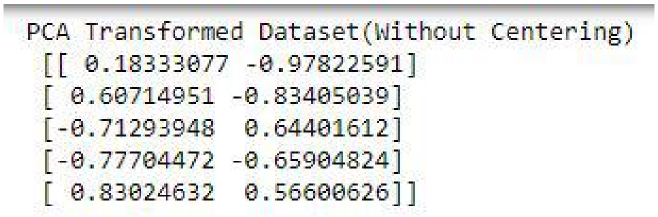


```
Explained variance:
 [45.82197547114777, 54.17802452885222]
```



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

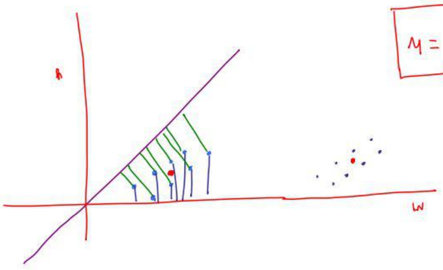




```
PCA Transformed Dataset(Without Centering)
 [[ 0.18333077 -0.97822591]
 [ 0.60714951 -0.83405039]
 [-0.71293948  0.64401612]
 [-0.77704472 -0.65904824]
 [ 0.83024632  0.56600626]]
```

Centering means Shifting origin to the ceentre of the given dataset. Centering always helps whenever mean is not zero(as the case here).Because in PCA algorithm,we only look for those lines which passes through origin as solution, therefore, when Centroid is non zero, we dont get correct solution without performing centering.

Reference:



$$\boxed{\mu = \frac{1}{n}\sum_{i=1}^{n} x_i}$$

$$x_i' \rightarrow \underline{x_i} - \underline{\mu} \quad \forall i$$

$$\frac{1}{n}\sum_{i=1}^{n} x_i' = 0$$

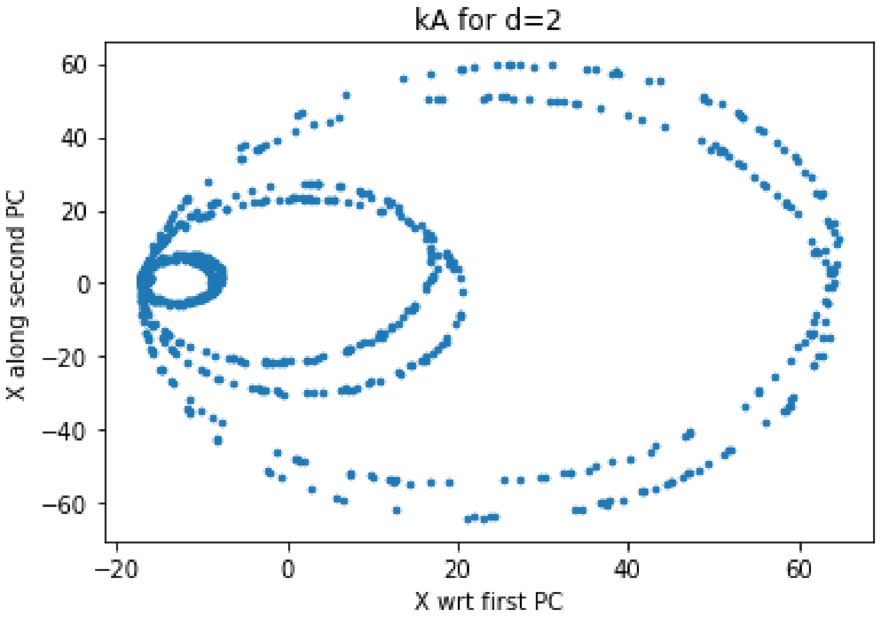
$$\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}x_i - \frac{1}{n}\sum_{i=1}^{n}\mu}_{\mu - \mu = 0}$$

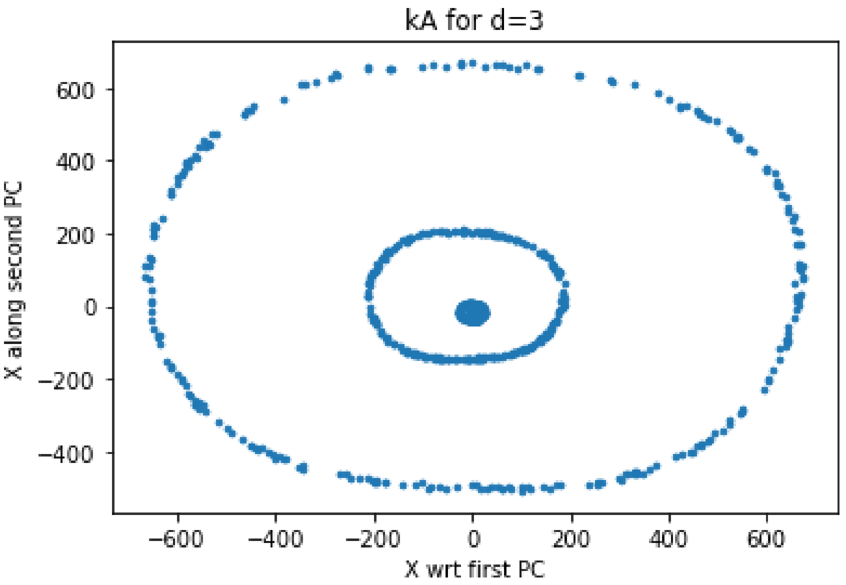
iii. Write a piece of code to implement the Kernel PCA algorithm on this dataset. Use the following kernels :

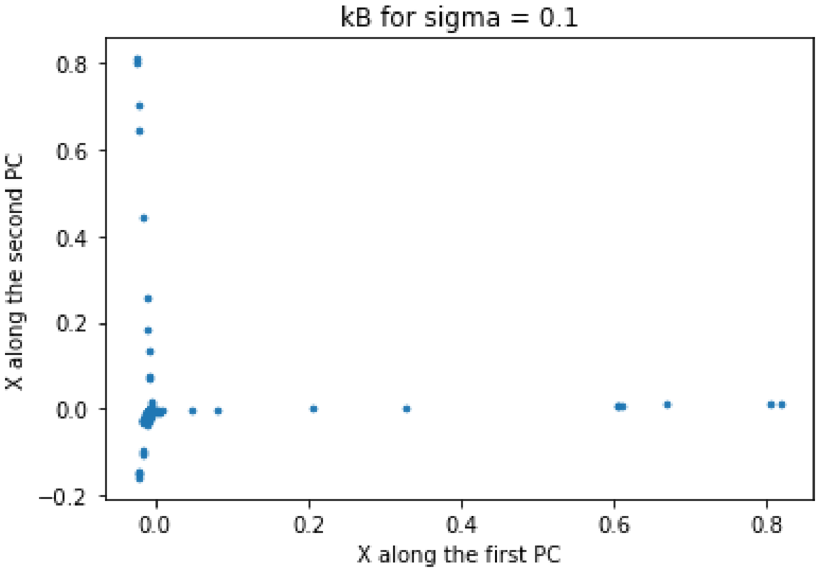
A. $\kappa(x, y) = (1 + x^T y)^d$ for $d = \{2, 3\}$

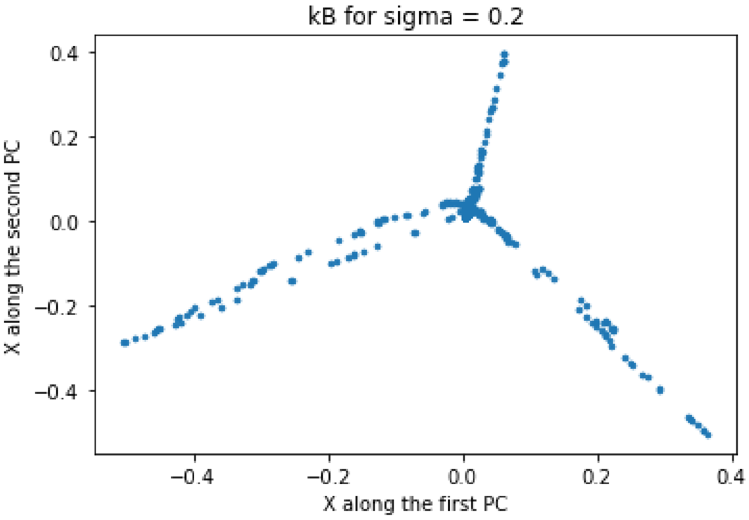
B. $\kappa(x, y) = \exp \frac{-(x-y)^T(x-y)}{2\sigma^2}$ for $\sigma = \{0.1, 0.2, \ldots, 1\}$

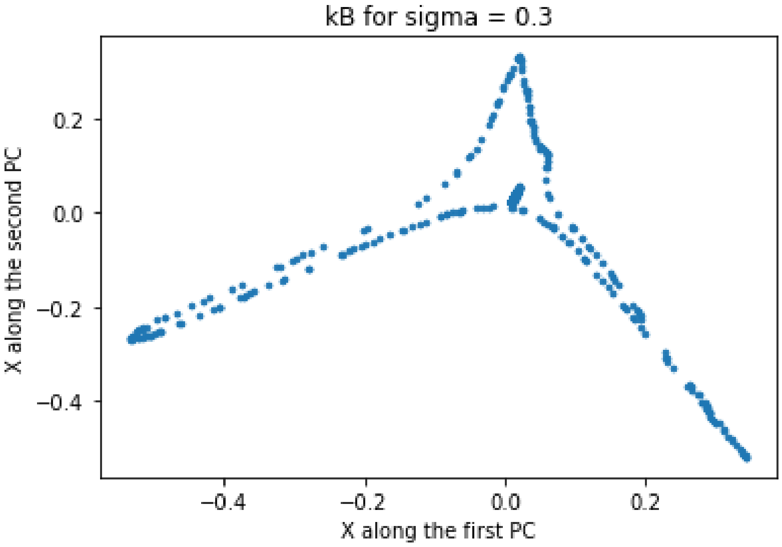
Plot the projection of each point in the dataset onto the top-2 components for each kernel. Use one plot for each kernel and in the case of (B), use a different plot for each value of σ.

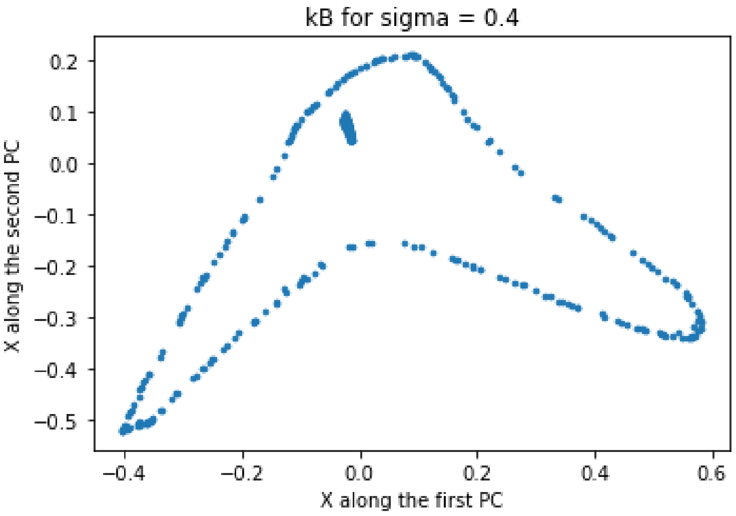




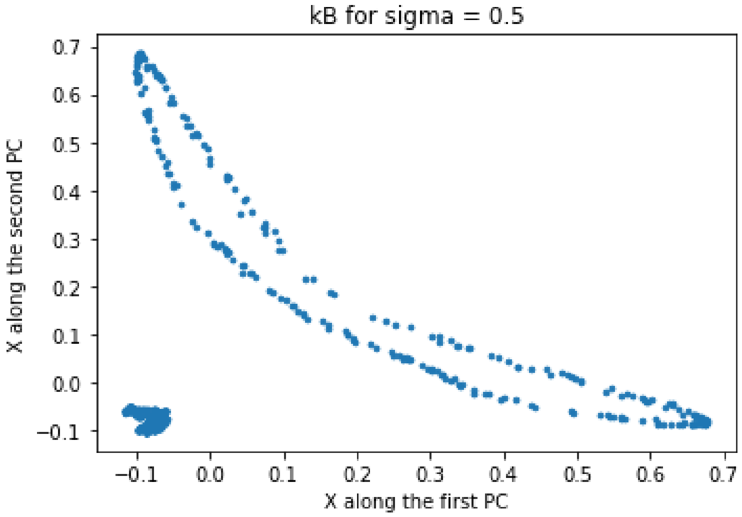

kB for sigma = 0.1
X along the second PC
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
X along the first PC


kB for sigma = 0.2
X along the second PC
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
X along the first PC

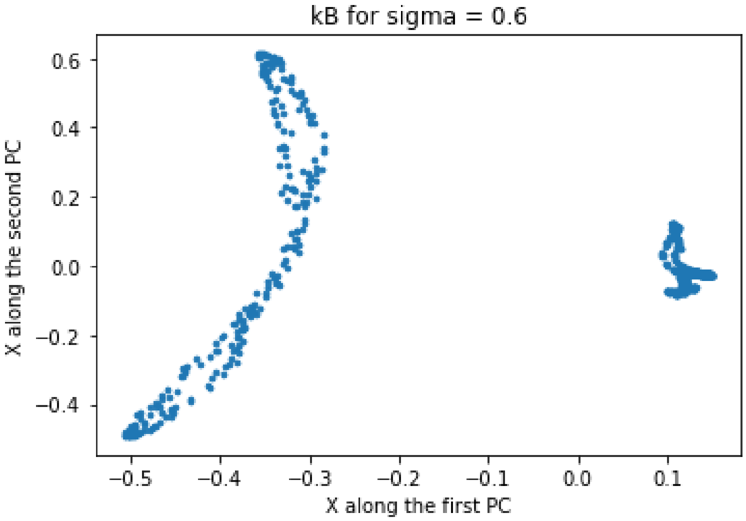

kB for sigma = 0.3
X along the second PC
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
X along the first PC



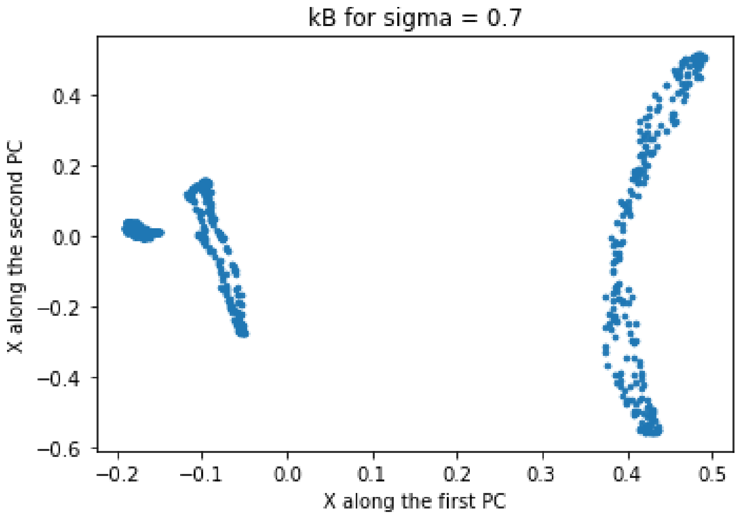
kB for sigma = 0.4

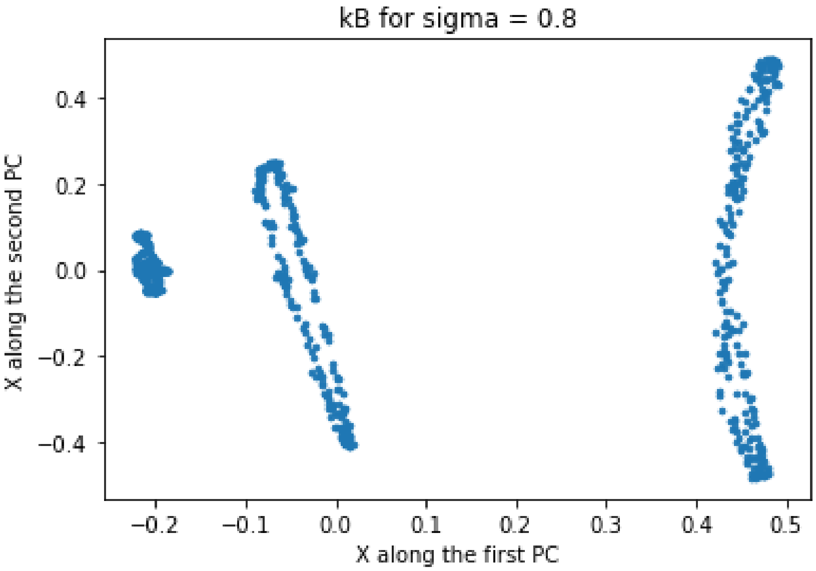


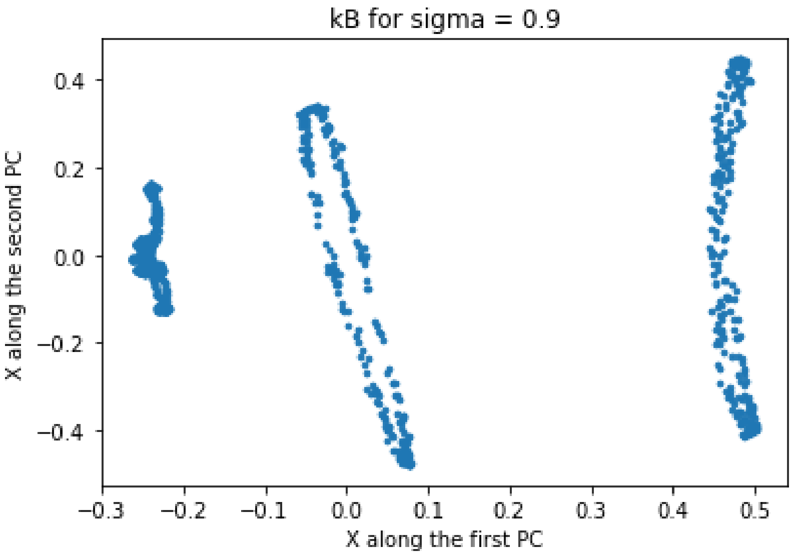
kB for sigma = 0.5

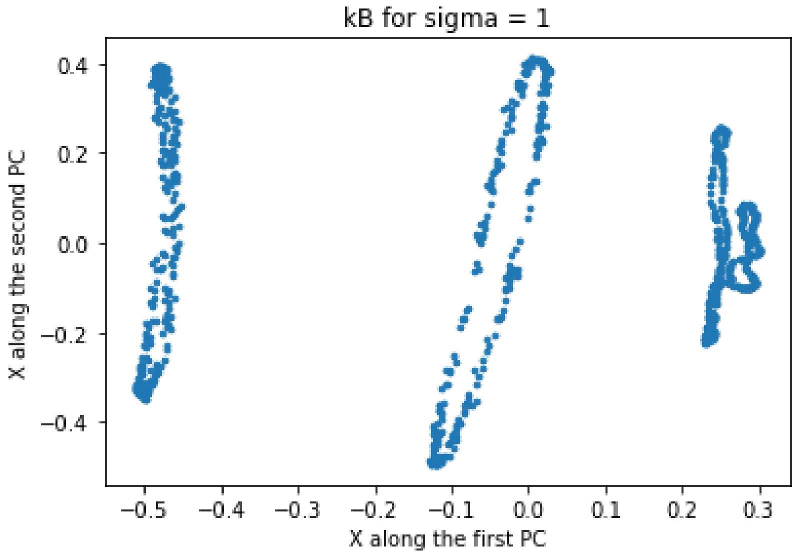


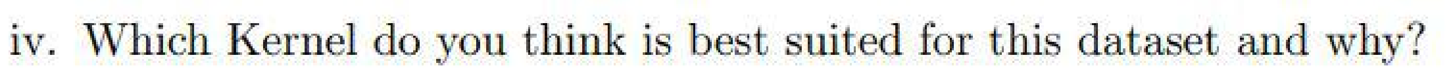
kB for sigma = 0.6


kB for sigma = 0.7
X along the second PC
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
X along the first PC


kB for sigma = 0.8
X along the second PC
0.4
0.2
0.0
−0.2
−0.4
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
X along the first PC


kB for sigma = 0.9
X along the second PC
0.4
0.2
0.0
−0.2
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
X along the first PC

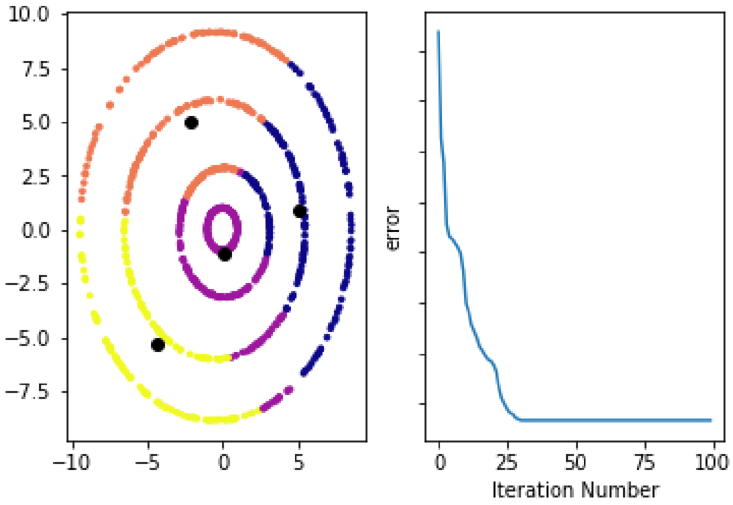


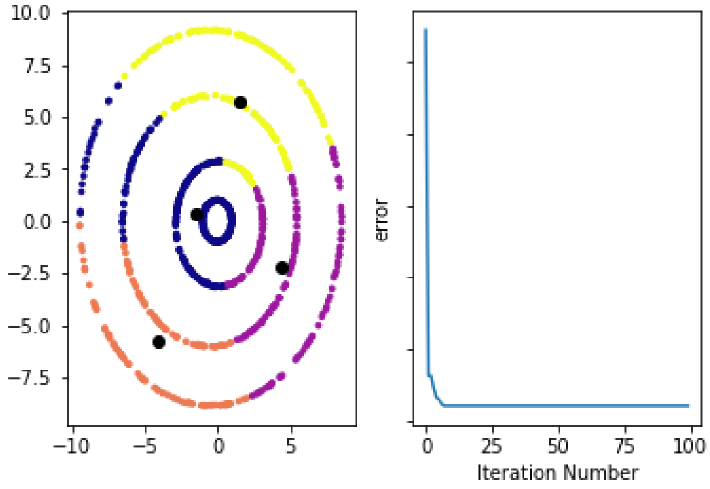

iv. Which Kernel do you think is best suited for this dataset and why?

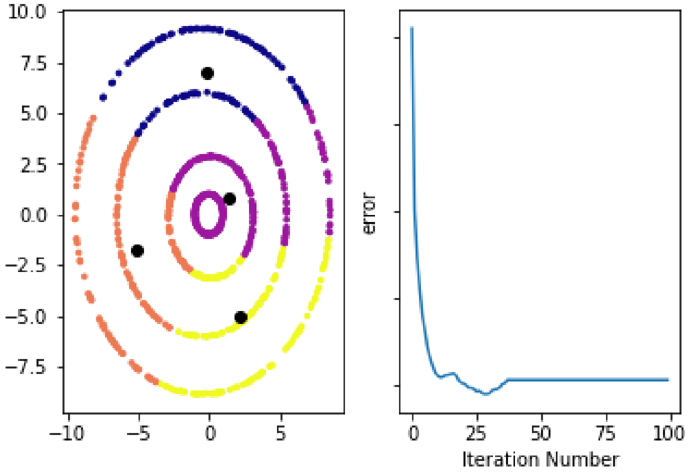
Kernal given as B (which is Radial Basis Function) is best suited here as it seperates out the dataset more precisely. We can observe that the datapoints forming circular structure in input which we can think as a group are seperated out better at Radial basis function especially for higher value of sigma.

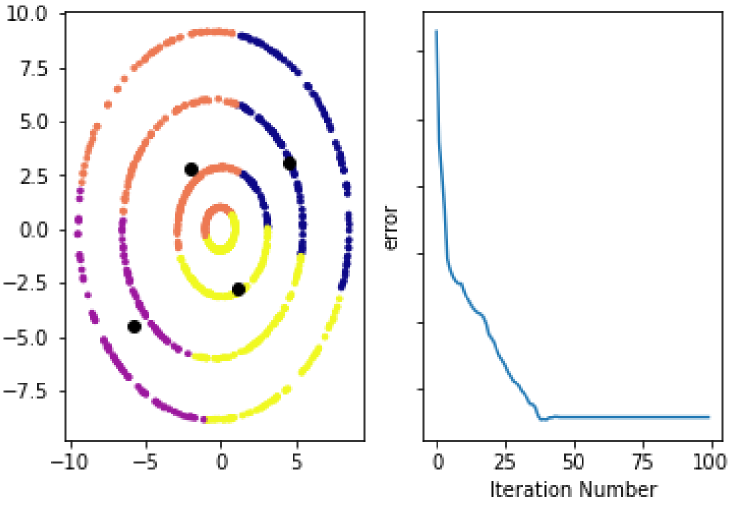
(2) You are given a data-set with 1000 data points each in $\mathbb{R}^2$.

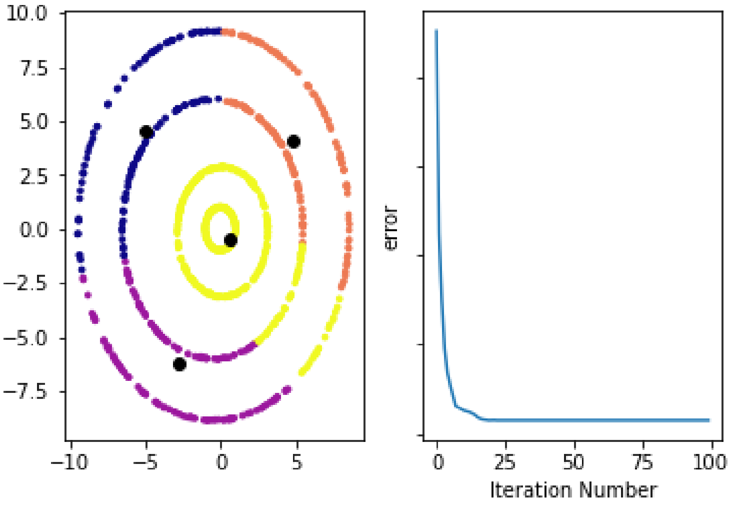
i. Write a piece of code to run the algorithm studied in class for the K-means problem with $k = 4$. Try 5 different random initialization and plot the error function w.r.t iterations in each case. In each case, plot the clusters obtained in different colors.





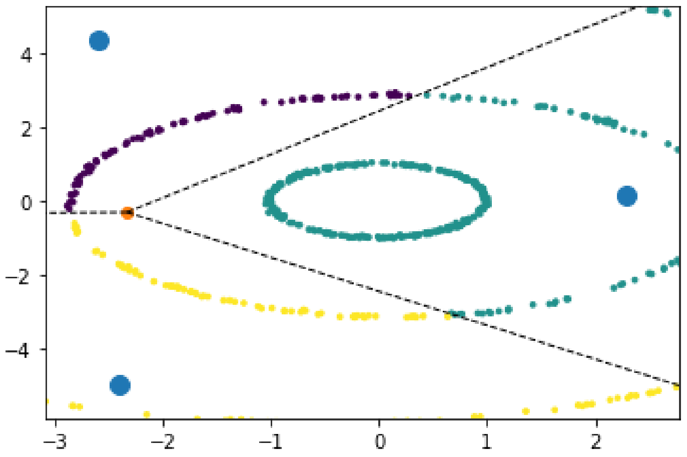



10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10
−5
0
5
error
0
25
50
75
100
Iteration Number

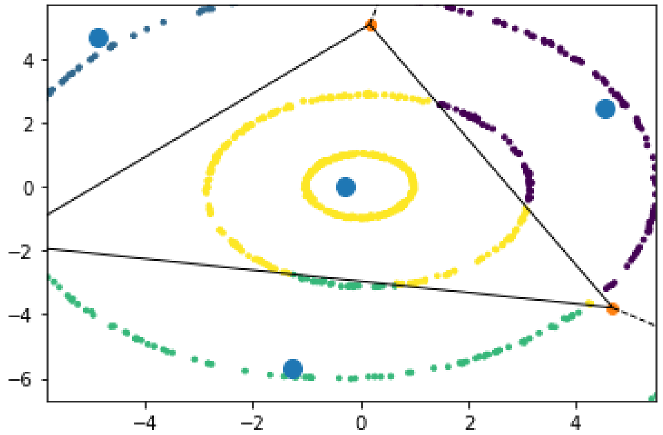

10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10
−5
0
5
error
0
25
50
75
100
Iteration Number

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 $\mathbb{R}^2$).

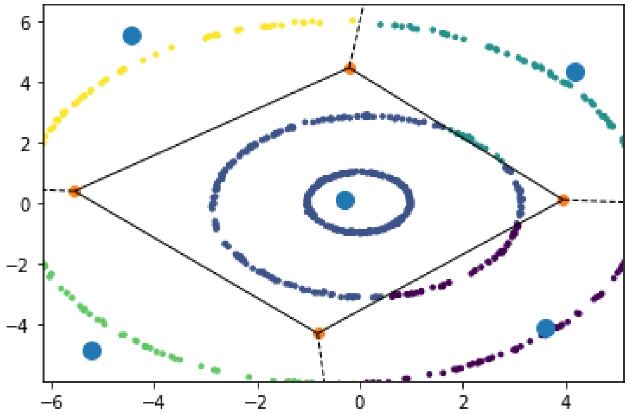
Using scipy



Voronoi regions for K =3



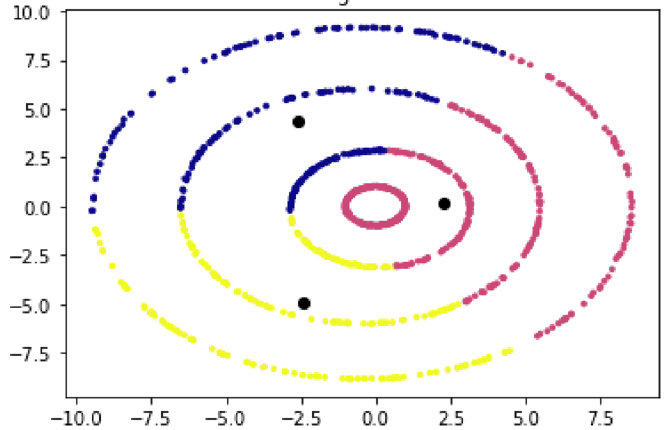
Voronoi regions for K =4



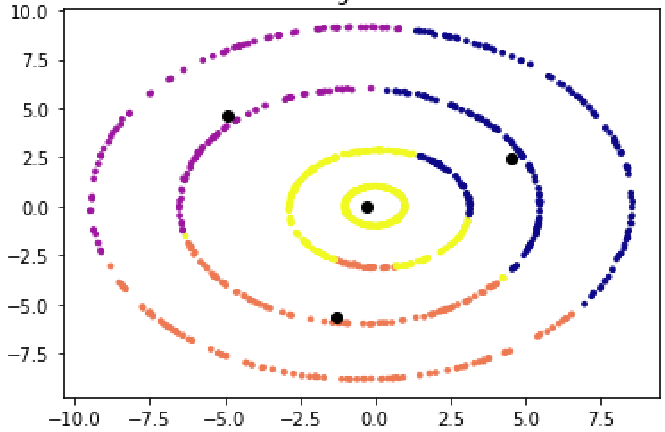
Voronoi regions for K =5

Without using scipy for same output

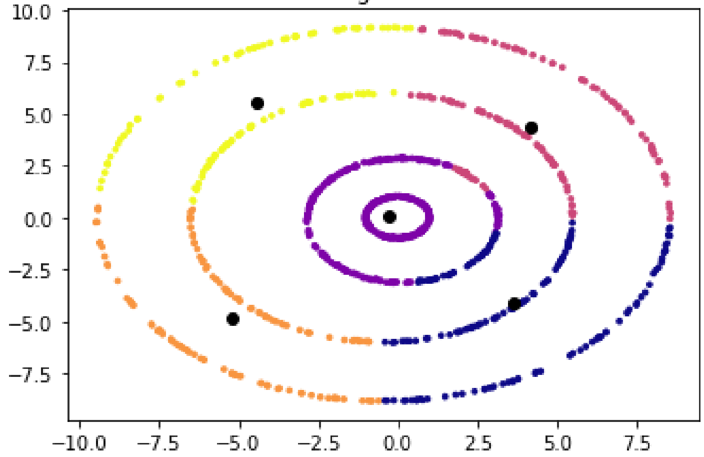
Voronoi regions for K =3



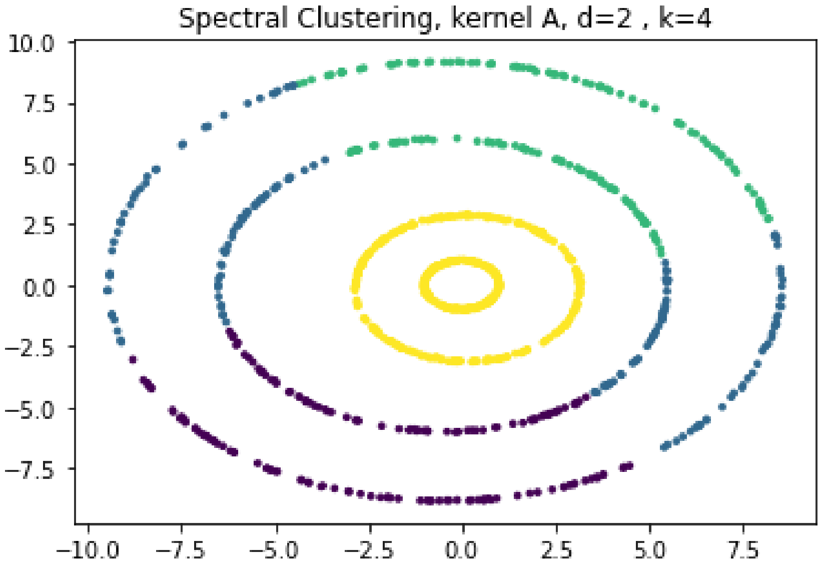
Voronoi regions for K =4

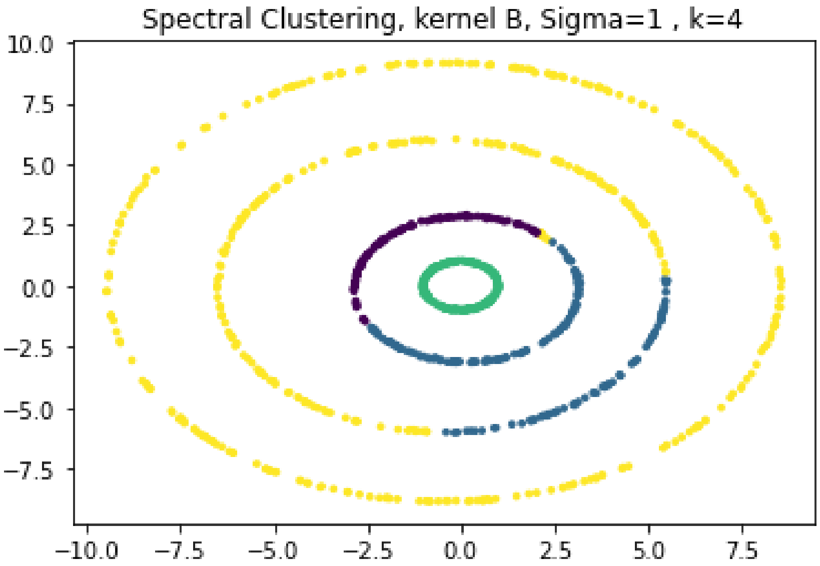


Voronoi regions for K =5

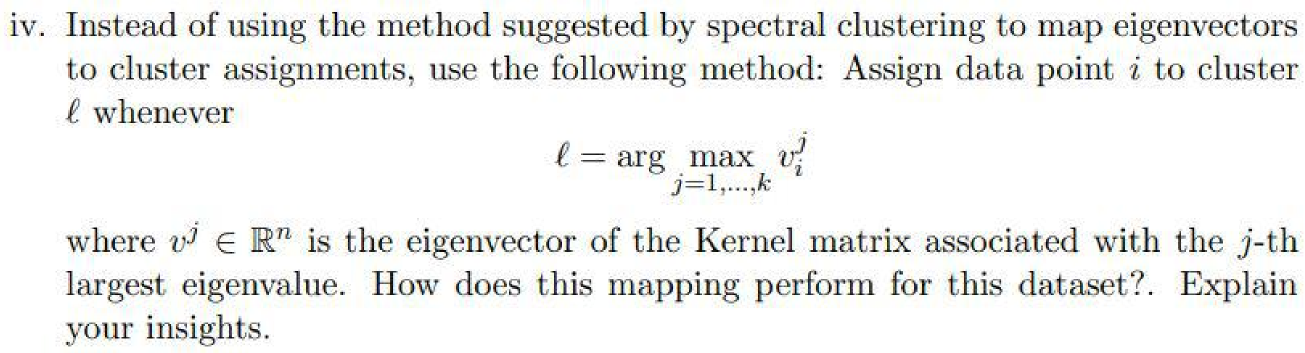


iii. Run the spectral clustering algorithm (spectral relaxation of K-means using Kernel-PCA) $k = 4$. Choose an appropriate kernel for this data-set and plot the clusters obtained in different colors. Explain your choice of kernel based on the output you obtain.





We can observe that both polynomial and radial basis kernels hasn't clustered ideally.
As input datapoints are in form similar to concentric circle, we would like to cluster the data points such that points forming circle has to be grouped together as the cluster. But here for both kernels and for chosen parameter values, we hadn't achieved this objective.
But for this particular instance output, we can observe that the radial basis function seperated out inner and outer most circle to same cluster and hence it can be assumed to be better for this particular instance.



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 ℓ whenever

$$\ell = \arg\max_{j=1,\ldots,k} v_i^j$$

where $v^j \in \mathbb{R}^n$ 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.

Just by changing the way we assign at zMat in previous code , we can achieve this objective.