CS5691: Pattern Recognition and Machine Learning

Assignment 1

Question 1

i)

Construct the Data matrix(X) from MNIST dataset by taking 100 images from each class 0 to 9 and converting each image into linear vector with 784(28*28) features.

X = d X N matrix (d=784, N=1000)

Now centering the data to perform PCA.

We subtract mean from each of the datapoint of X.

Now Computing the **Covariance Matrix = X*(X.T)/N**

Computing the Eigen vectors and Eigen values of the Covariance Matrix.

Rearrange the Eigen values in descending order and also rearrange the corresponding Eigen Vectors.

Each eigen value corresponds to the variance along that eigenvector(component).

Now each Eigen vector corresponds to a principal component.

Fig 1: Images of Principal Components

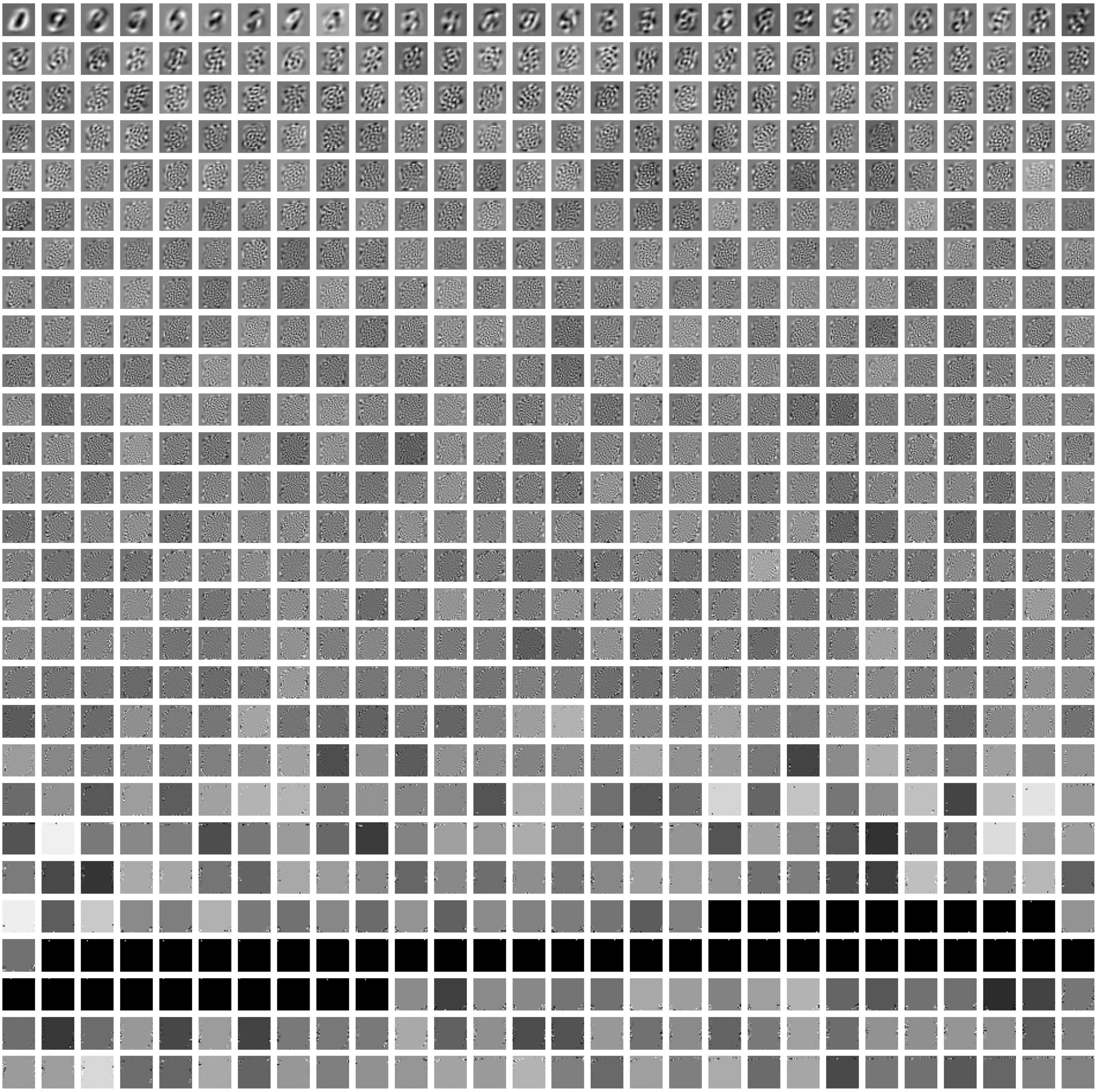


Fig 2: Percentage Variance explained by each principal component.

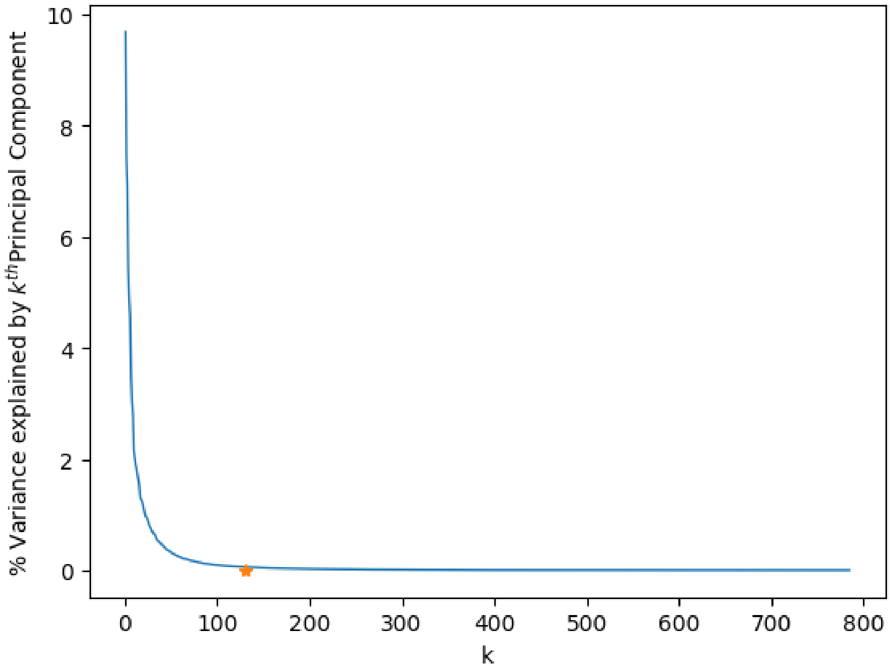
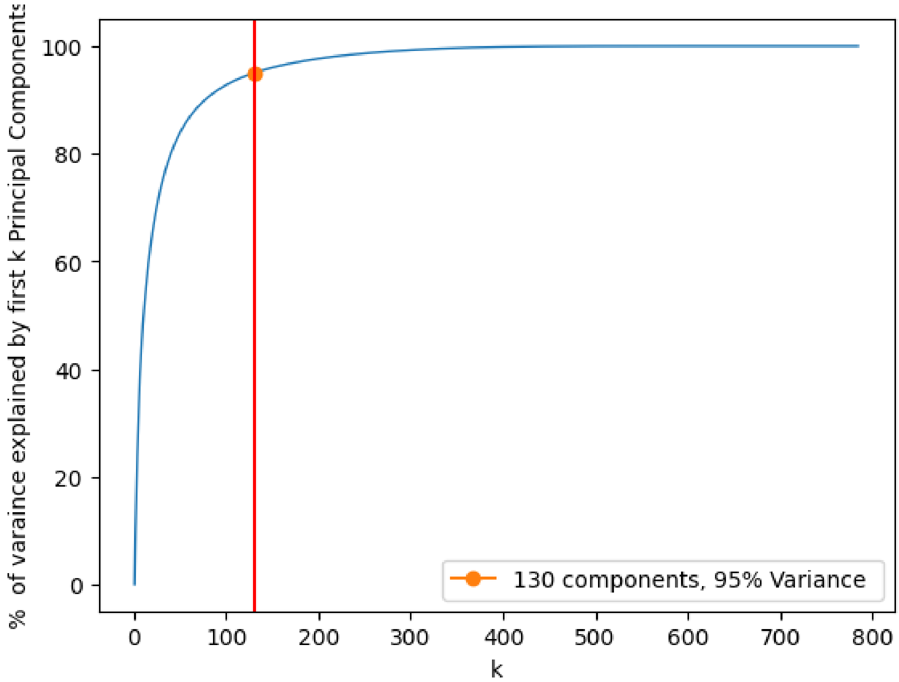


Fig 3: Cumulative Percentage Variance explained by “k” principal components.



from the above figure it is clear that 130 top components explain 95 percent (%) of variance .so these are the principal components after PCA.

ii)

Taking projections of each datapoint(image) on the top “l” principal components and multiplying it with the corresponding principal component and summing them up we get the reconstructed datapoint(image).

In general, we take ‘l’ such that it explains 95 % of variance.

If we take l=784 we precisely get the original data (but it takes lot of memory)

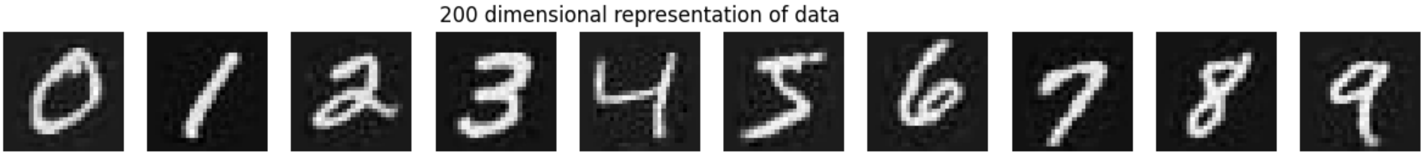
Fig 4: Different dimensional representation of Data

50 dimensional representation of data

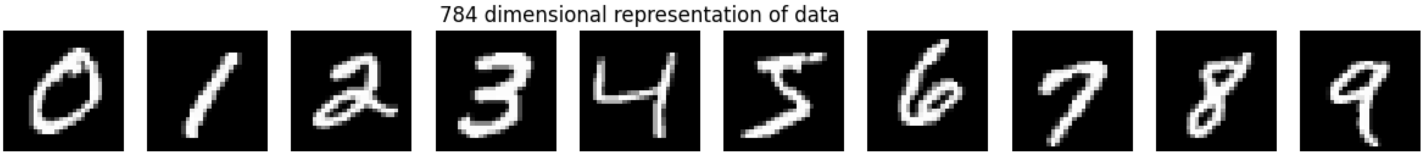
100 dimensional representation of data

130 dimensional representation of data

200 dimensional representation of data



784 dimensional representation of data



As we can clearly see in the above figures that we can clearly distinguish the digits even from 50-dimensional representation. (but there is some noise)

But for 130-dimensional representation there is relatively less noise compared to the case of 50- and 100-dimensional representation.

But in the Case of 130 the percentage variance explained is 95 percent which means almost all the information of the original data is captured.

Typically, for a downstream task, classifying digits, I would choose d=130-dimensional representation. (as it retains 95 percent information of the data)

iii)

polynomial kernel

A) $\kappa(x, y) = (1 + x^T y)^d$ for $d = \{2, 3, 4\}$

Here we map the data to higher dimension based on the kernel function.

Construct the polynomial kernel matrix given by the polynomial kernel function above

P_Kernel_matrix = (1+np.matmul(X.T, X)) **d

Centre the P_Kernel_matrix.

Compute the eigenvectors and eigen values for the P_Kernel_matrix.

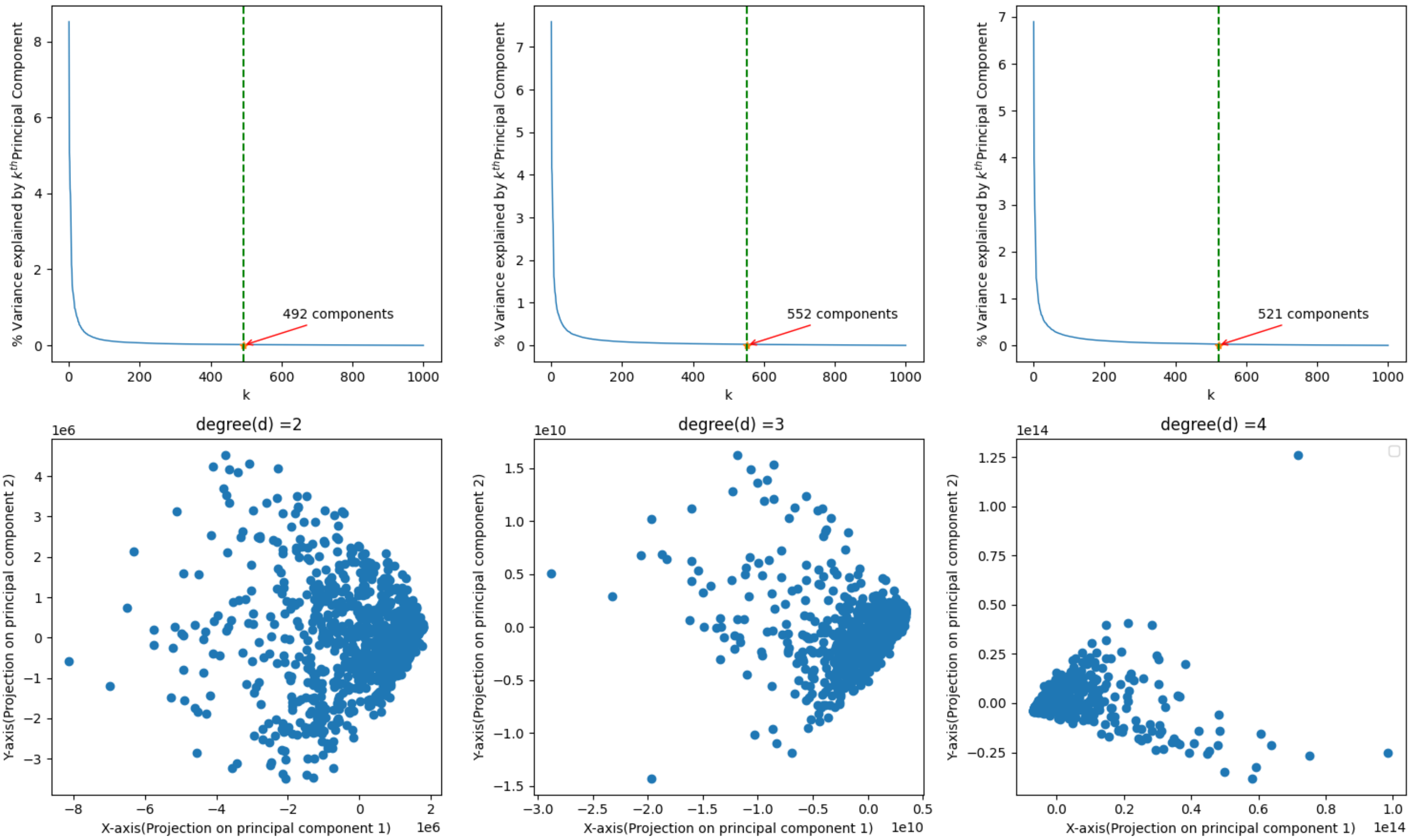
Now Compute the corresponding Alphas by dividing the eigen vector with root of eigenvalue

Representatives here are simply the product of Kernel matrix and Alphas.

Consider the Representatives of the top 2components.

Now plot these Representatives of 1st and 2nd components as x and y respectively. (for d=2,3,4)

Fig 5: Plots for each degree of the polynomial kernel



B)

radial kernel

κ(x, y) = exp −(x−y).T* (x−y) /2σ2.

In this case we map the data to an infinite dimension.

Construct the radial kernel matrix given by the polynomial kernel function above.

Centre the R_Kernel_matrix obtained in the above step.

Compute the eigenvectors and eigen values for the R_Kernel_matrix.

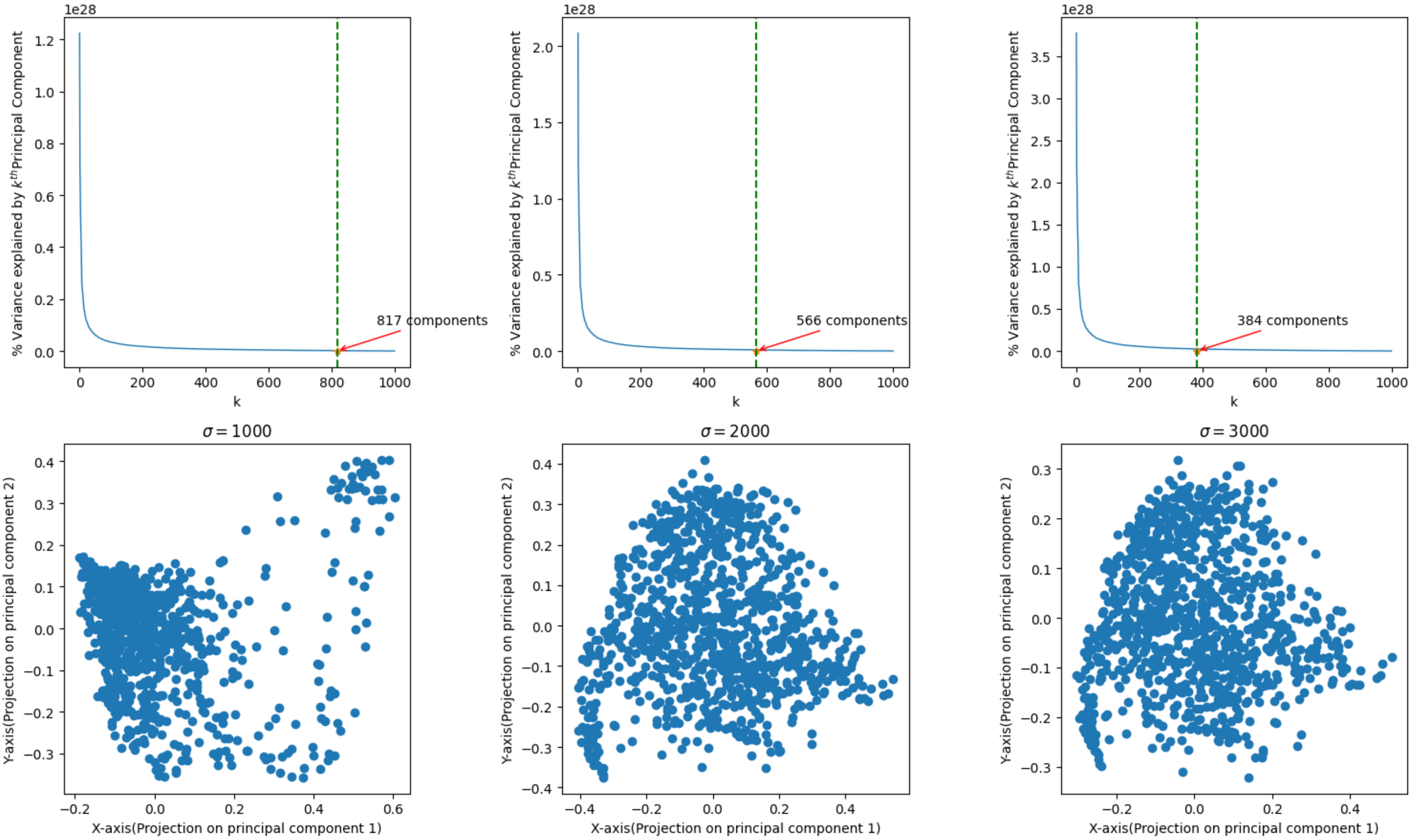
Now Compute the corresponding Alphas by dividing the eigen vector with root of eigenvalue.

Representatives here are simply the product of Kernel matrix and Alphas.

Consider the Representatives of the top 2components.

Now plot these Representatives of 1st and 2nd components as x and y respectively. (for σ=1000,2000,3000)

Fig 6: Plots for each σ of the polynomial kernel



iv)

Polynomial Kernel

d	% variance explained by top 2 principal components
2	13.58 %
3	11.75 %
4	10.86%

Radial Kernel

σ	% variance explained by top 2 principal components
1000	4.88 %
2000	12.17 %
3000	14.70 %

Polynomial Kernel with d=2 is better as the number of components required to explain is less compared to that of 3 and 4 and also percentage variance explained by top 2 components is also high.

In case of radial kernel as the value of σ increases the percentage variance on top 2 components increases and also the number of components to explain the 95 % variance decreases.

polynomial kernel d=2 ,492 components required for 95%varinace

polynomial kernel d=3 ,552 components required for 95%varinace

polynomial kernel d=4 ,521 components required for 95%varinace

radial kernel σ =3000 ,384 components required for 95%varinace

As σ increases no of components decreases for 95 % variance.

I would choose the radial kernel with σ in some appropriate range(σ>3000). because it explains more percentage variance compared to that of polynomial kernel

Also, radial kernel requires less number of components to explain 95 % of variance.

Question 2

i)

Construct the dataset by converting the given (cm dataset 2.csv) into NumPy array.

Number of clusters =2

Cluster indices “0”, “1”.

Assign Each Datapoint with Random number from “0” or “1”.

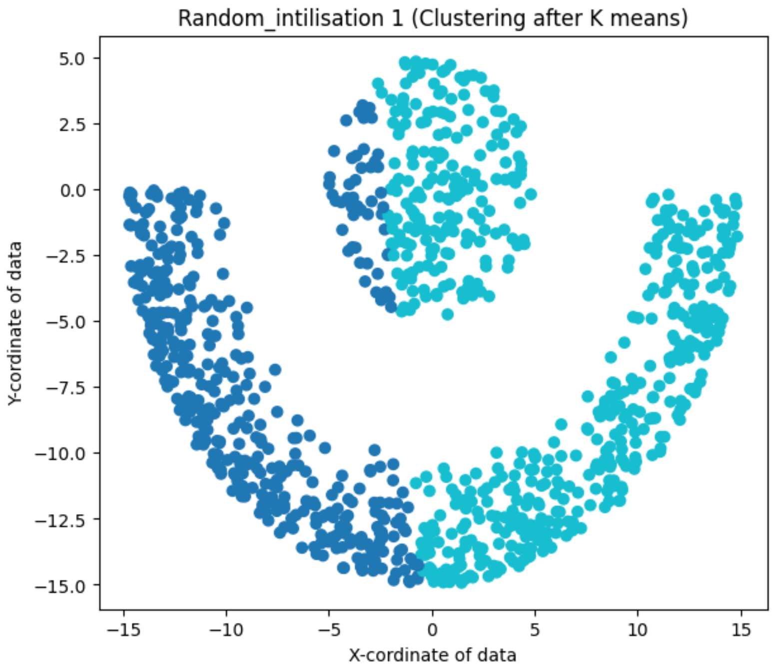
Calculate the Means of the K clusters Obtained.

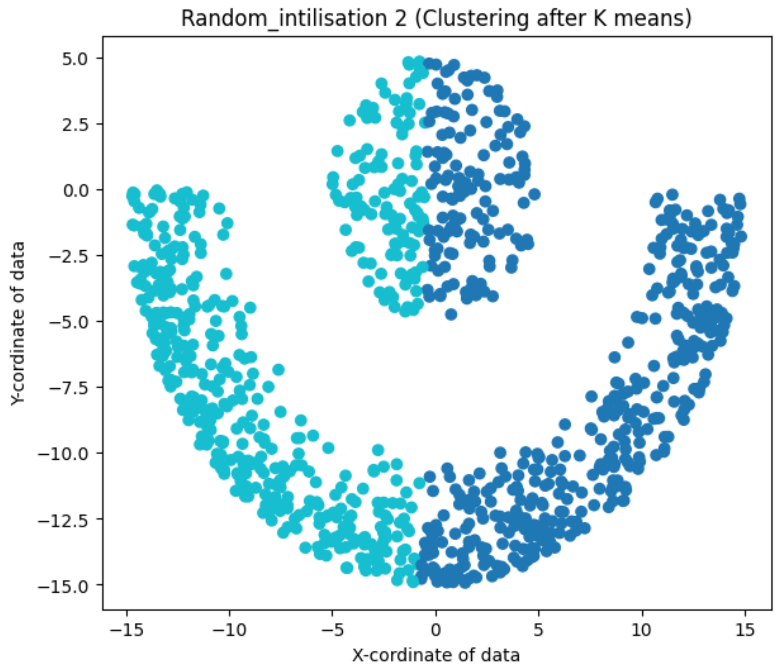
Perform Lloyds algorithm based on this random assignment.

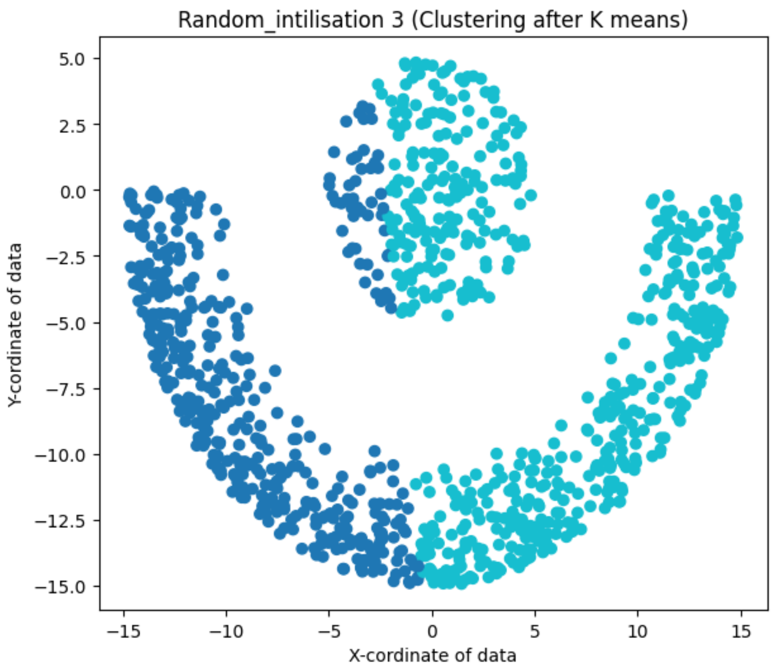
Find the Error (Objective function) in each step(iteration).

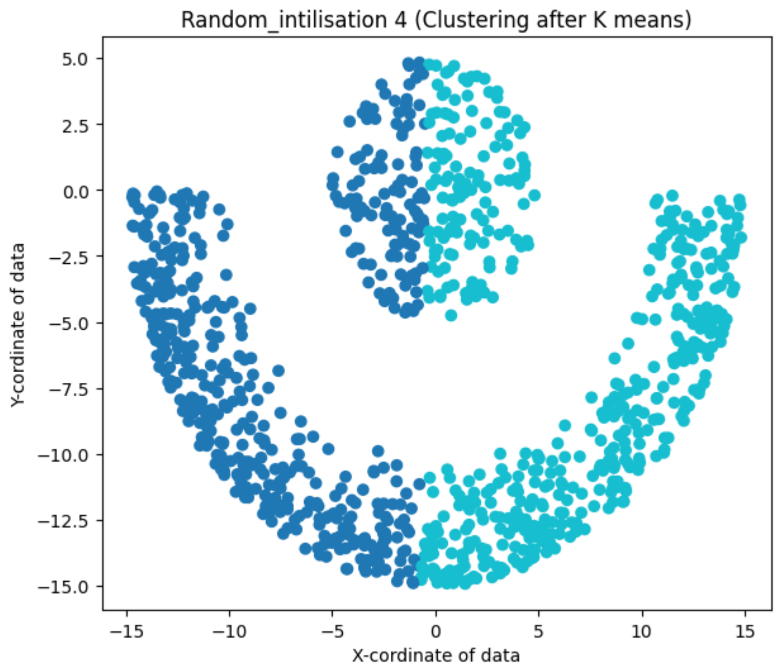
Now Plot the Points in different Clusters with different colors and also plot the error vs iterations.

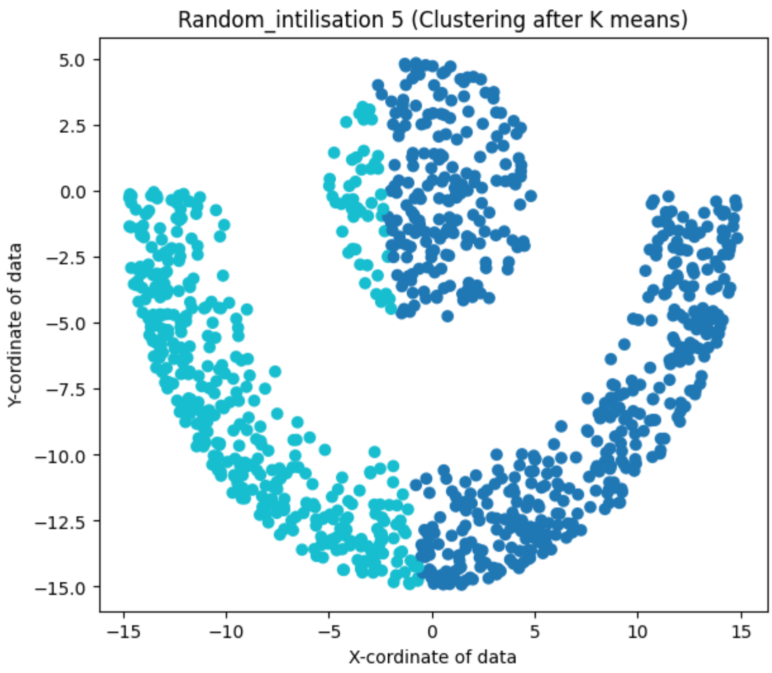
Fig 7: Clustering for K=2 for 5 Random Initializations

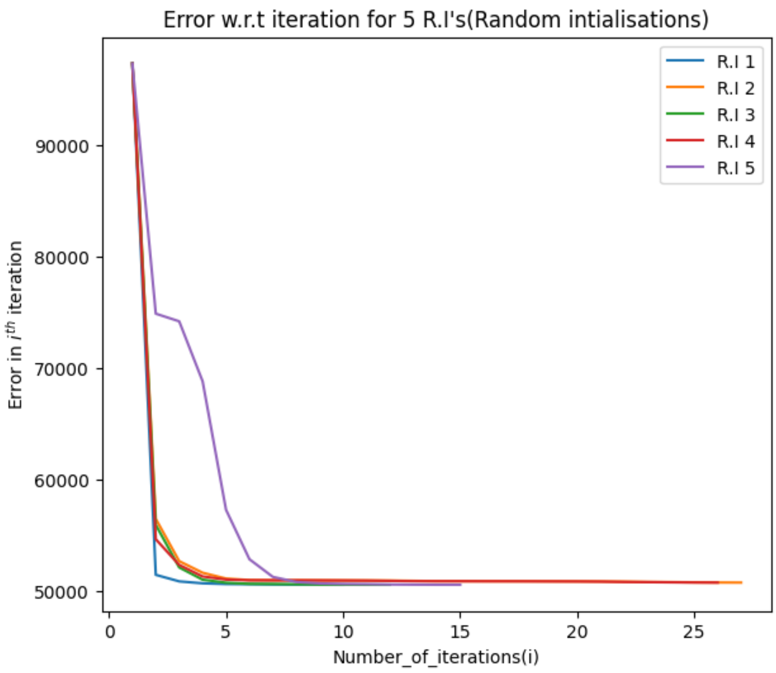

Random_intilisation 1 (Clustering after K means)
Y-cordinate of data
X-cordinate of data


Random_intilisation 2 (Clustering after K means)
Y-cordinate of data
X-cordinate of data


Random_intilisation 3 (Clustering after K means)
Y-cordinate of data
X-cordinate of data


Random_intilisation 4 (Clustering after K means)
Y-cordinate of data
X-cordinate of data


Random_intilisation 5 (Clustering after K means)
Y-cordinate of data
X-cordinate of data


Error w.r.t iteration for 5 R.I's(Random intialisations)
R.I 1
R.I 2
R.I 3
R.I 4
R.I 5
Error in i^{th} iteration
Number_of_iterations(i)

ii)

Construct the dataset by converting the given (cm dataset 2.csv) into NumPy array.

Number of clusters =K

Cluster indices “0” to “K”.

Assign Each Datapoint with Random number from “0” to “1”.

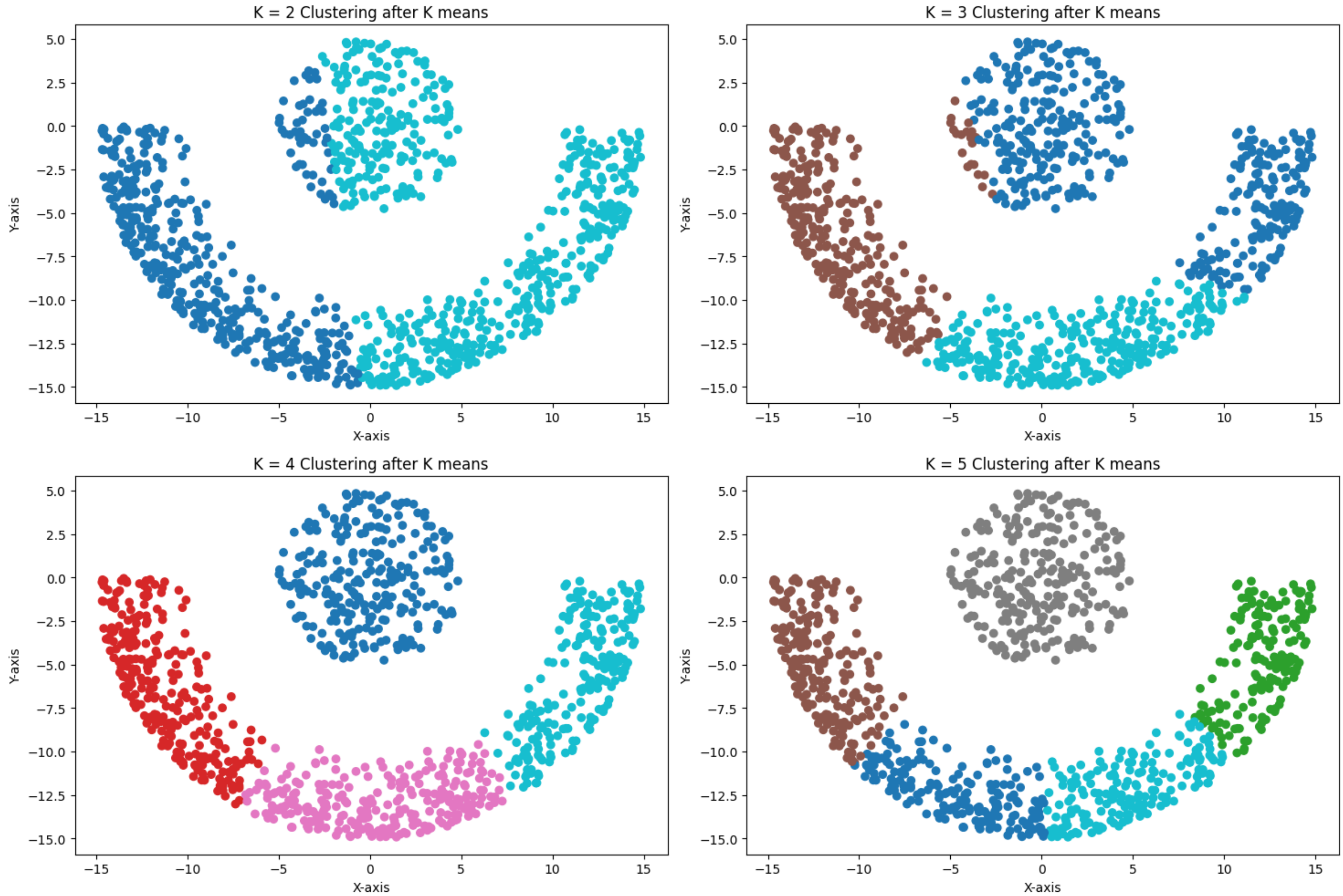
Calculate the Means of the K clusters Obtained.

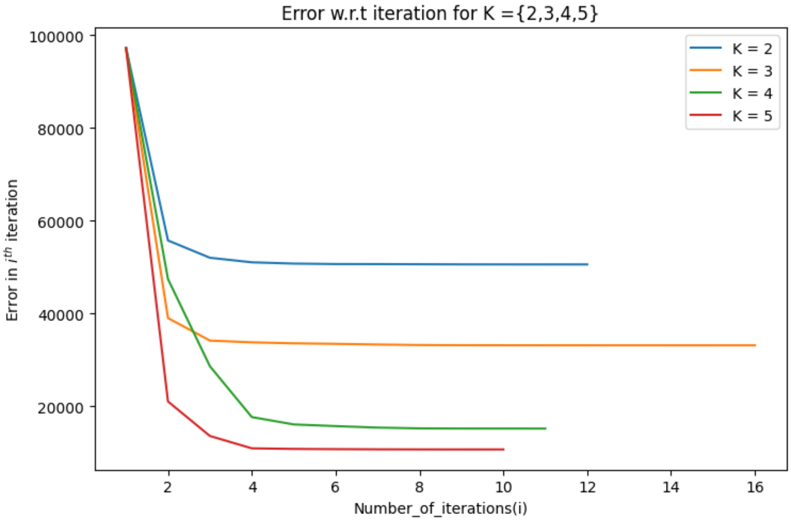
Perform Lloyds algorithm based on this random assignment.

Find the Error (Objective function) in each step(iteration).

Now Plot the Points in different Clusters with different colors and also plot the error vs iterations. (for K=2,3,4,5)

Fig 8: Clustering for K=2,3,4,5 for a Random Initialization.





iii)

consider a kernel and then find the kernel matrix for the given dataset

compute eigenvalues and vectors.

Now Consider the Matrix(H) that contains top K (here k=2) eigen vectors.

Normalize each row of the above matrix and then now treat each row as a datapoint and apply Lloyd's algorithm.

Plot the Obtained clustering for the original dataset.

Fig 9: Spectral Clustering for Radial kernel(σ=1,100,1000,10000)

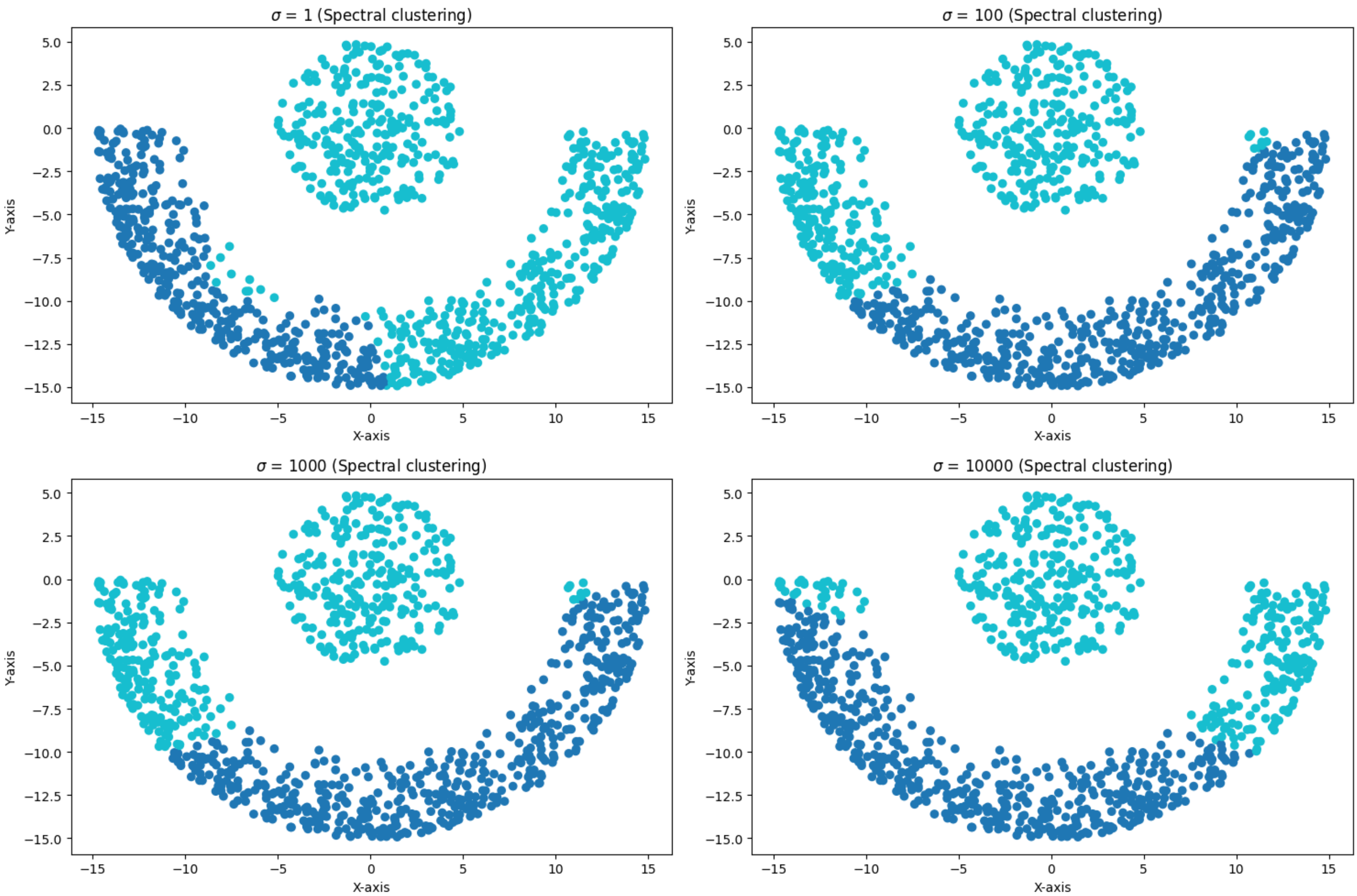
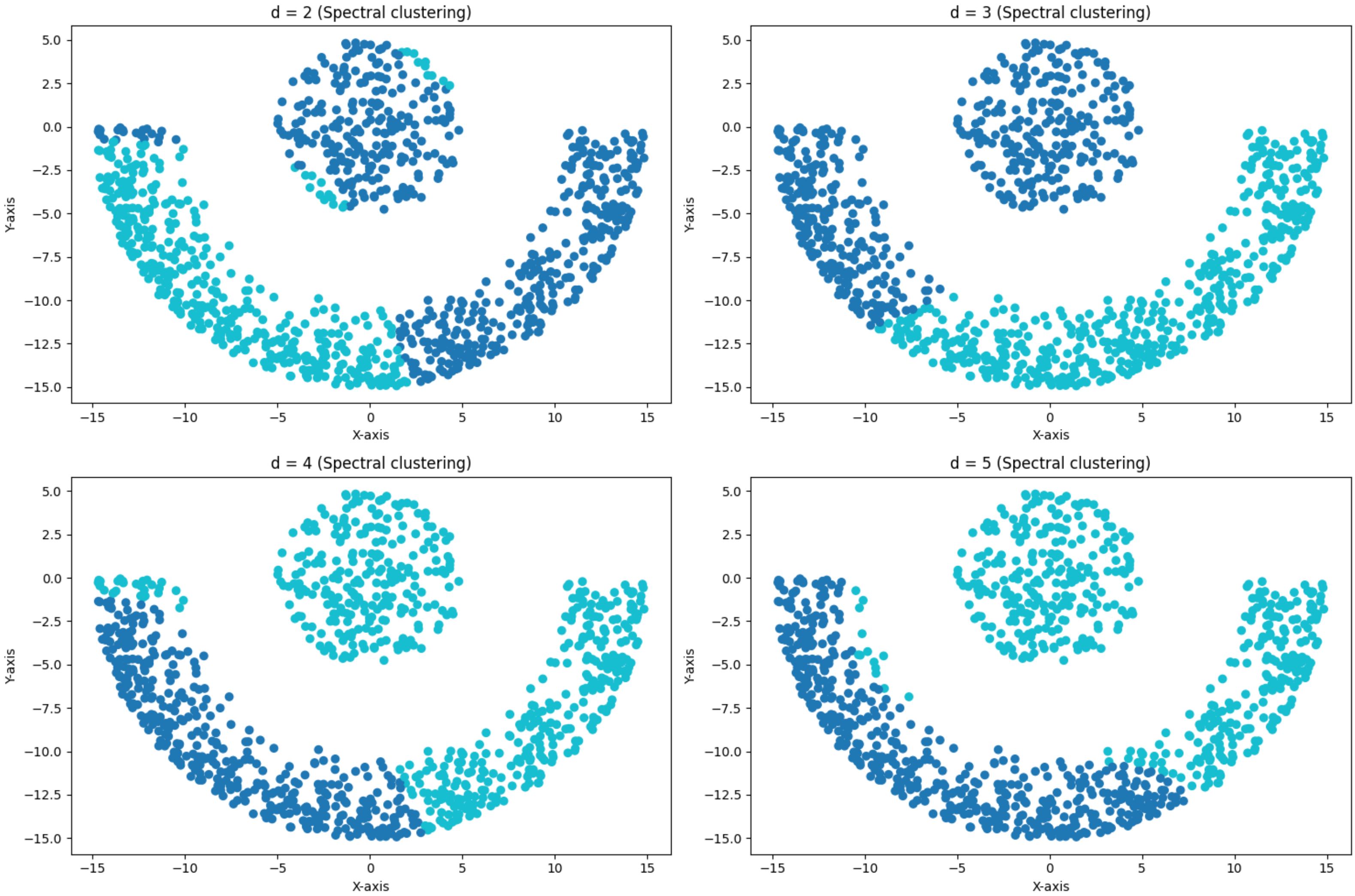


Fig 10: Spectral Clustering for Polynomial kernel(d=2,3,4,5)



The clustering are sometimes heavily dependent on the initialization.

In this case the clusters obtained from both the polynomial and radial kernels not the clustering what we expect.

In this case radial will be preferred out of the two as it gets less error. (for small value of σ)

iv)

consider a kernel and then find the kernel matrix for the given dataset.

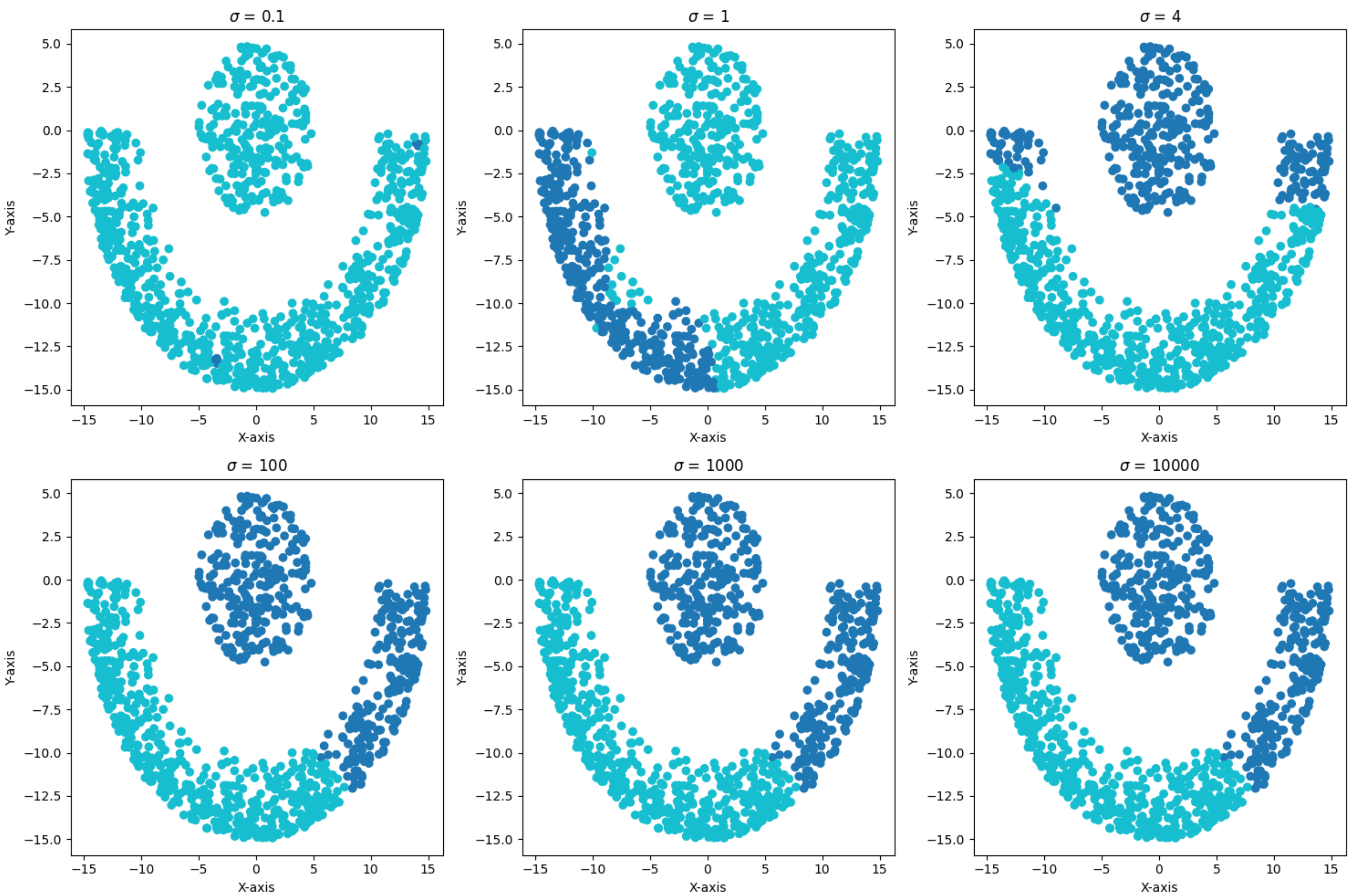
compute eigenvalues and vectors.

Now Consider the Matrix(H) that contains top K (here k=2) eigen vectors.

Now instead of using the method suggested by spectral clustering to map eigenvectors to cluster assignments, we use the following method: Assign data point i to cluster l whenever.

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

Fig 11: Radial kernel (clustering according to above assignment)

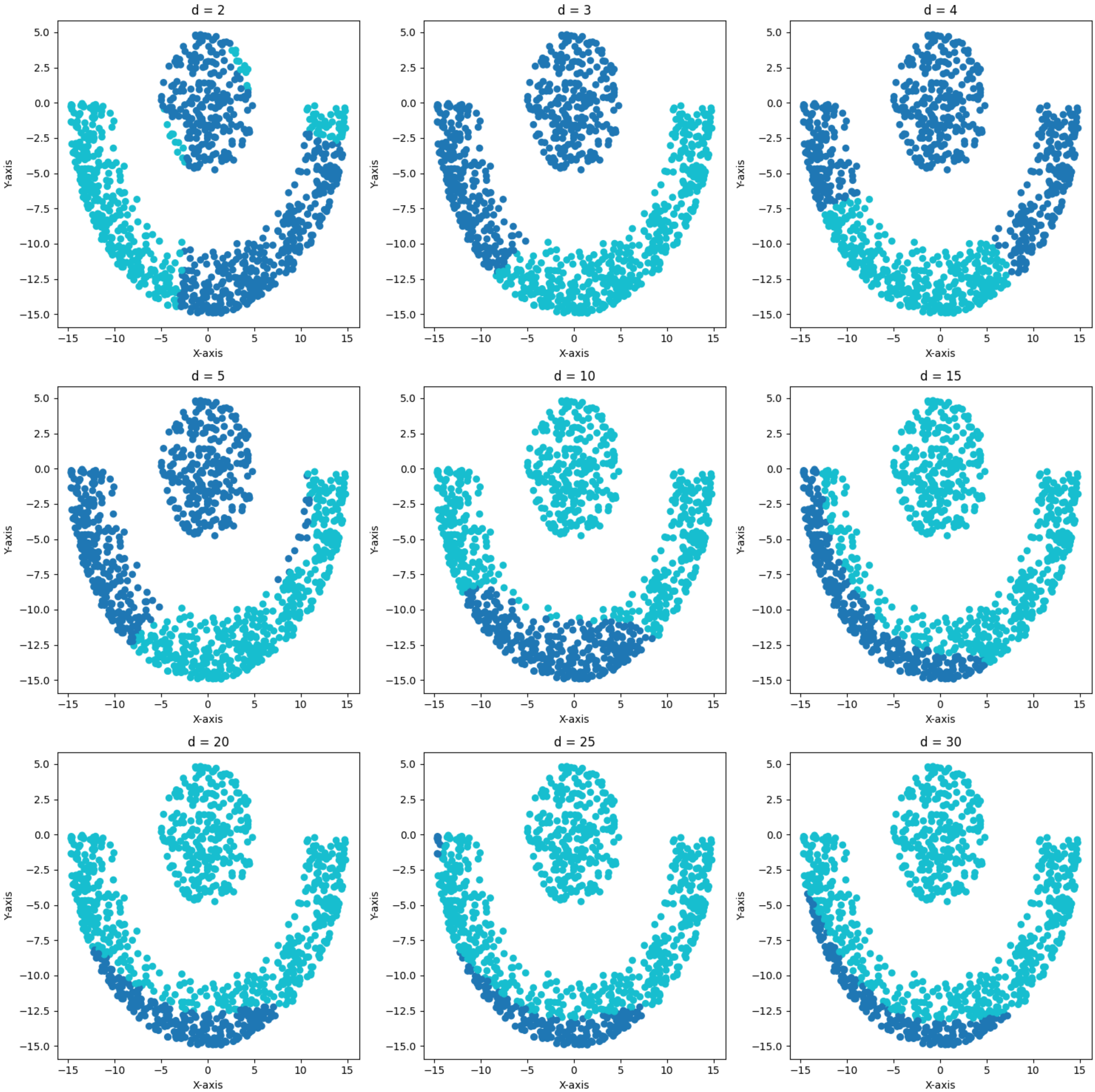


for Higher values of σ there is not much difference in the clustering.

for very small values of σ almost all the points are getting into a single cluster.

for σ =4 the clustering is good in this case.

Fig 12: Polynomial kernel (clustering according to above assignment)



for high degrees the one of the cluster gets assigned with most of the points.

for low degrees the clustring is not proper.