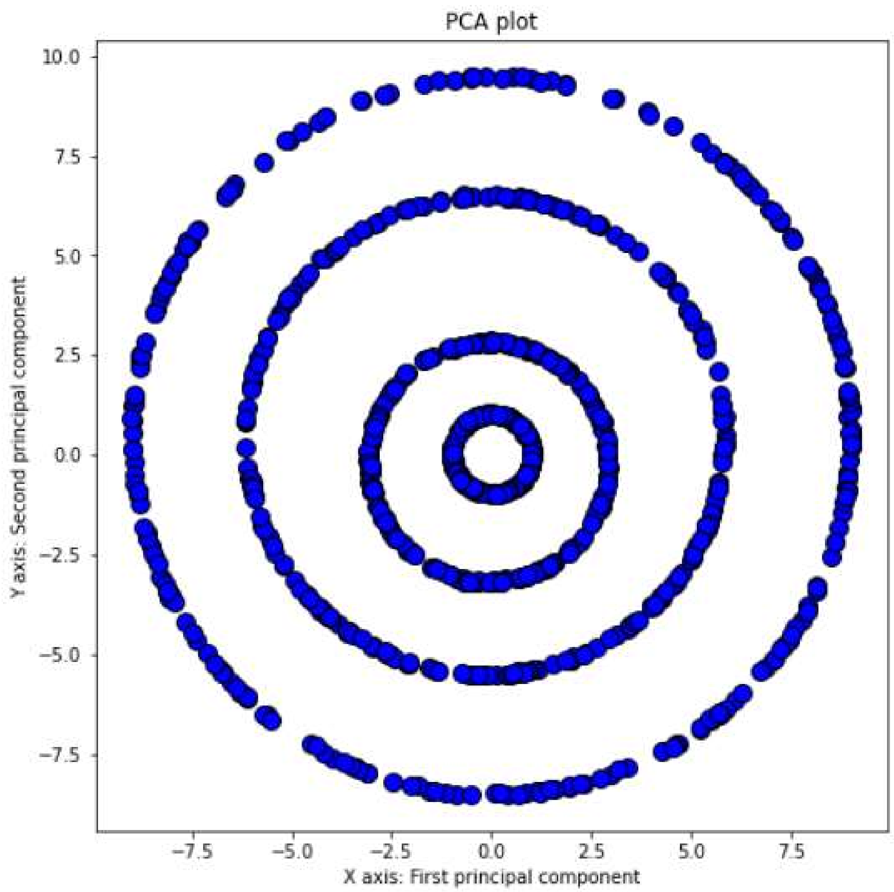
Report.pdf.

NAME : Sneha Mayanahalli Rajendranath

ROLL NO : CS21M522

Machine Learning Assignment - 01

1 i)
PCA plot


PCA plot
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Second principal component
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: First principal component

(1)

(i)

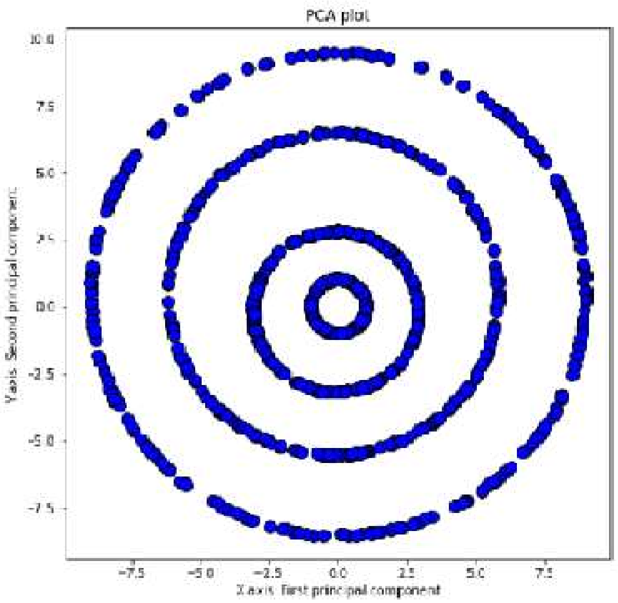
$\begin{bmatrix} \text{Variance of I principle} \\ \text{component} \end{bmatrix} = 0.54 = 54\%$

$\begin{bmatrix} \text{Variance of II principle} \\ \text{component} \end{bmatrix} = 0.46 = 46\%$

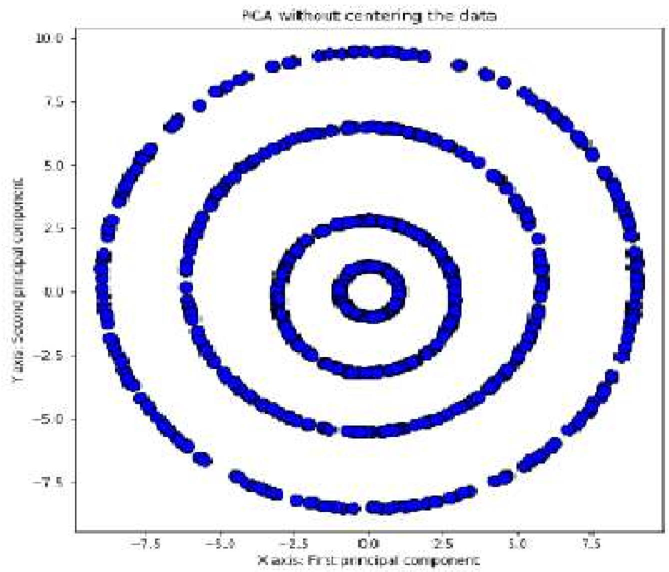
Variance tells about how-much information is contained by each of the principle components.

Higher the variance, more good it is because it contains more information about the data.

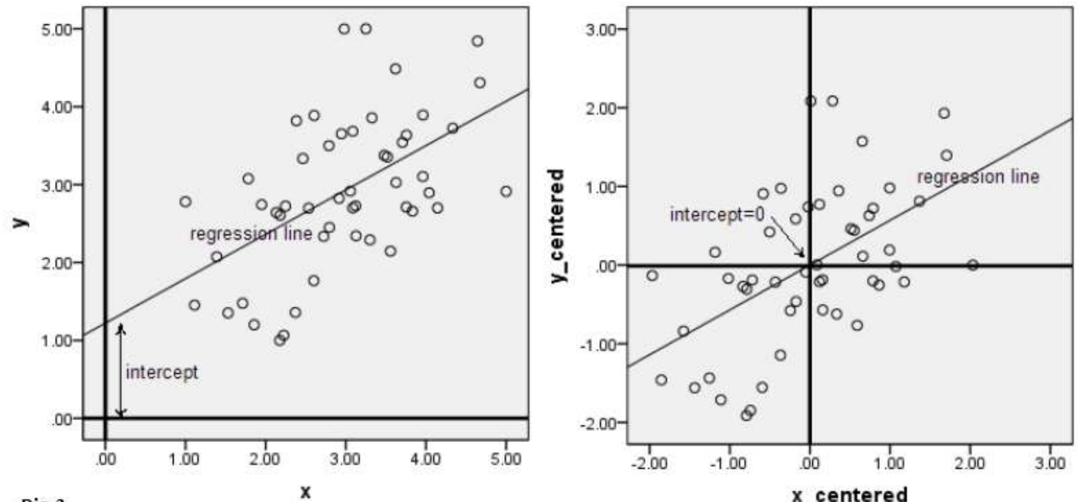
PCA with Centering the data



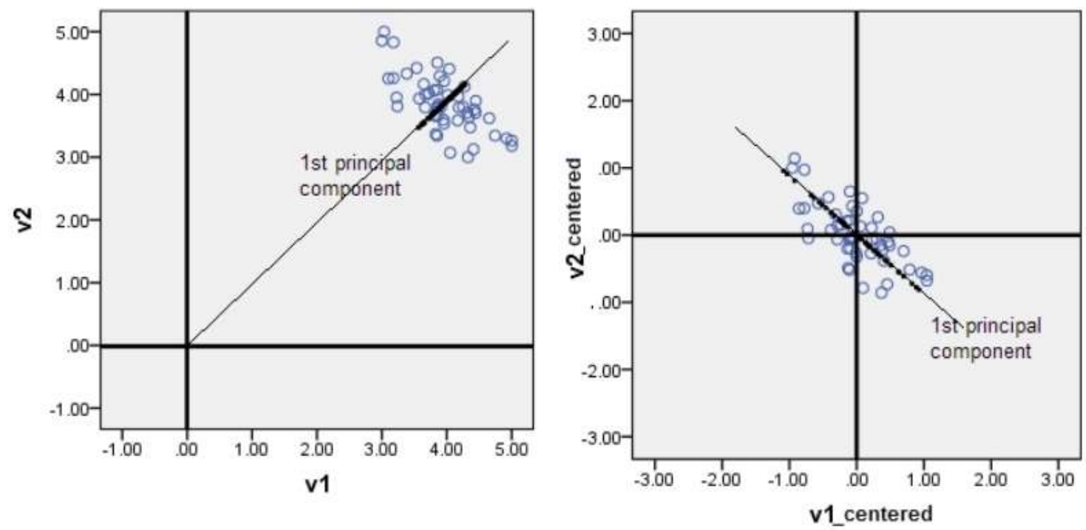
PCA without cenetring the data



Dig-2


y
x
regression line
intercept
y_centered
x_centered
regression line
intercept=0

Dig-3


v2
v1
1st principal component
v2_centered
v1_centered
1st principal component

(1)

(a)

Why centering the data in PCA is important?

(a) It is needed for statistical purposes

(b) without mean-centering, the first principle component found by PCA might correspond with the mean of the data instead of the direction of maximum variance.

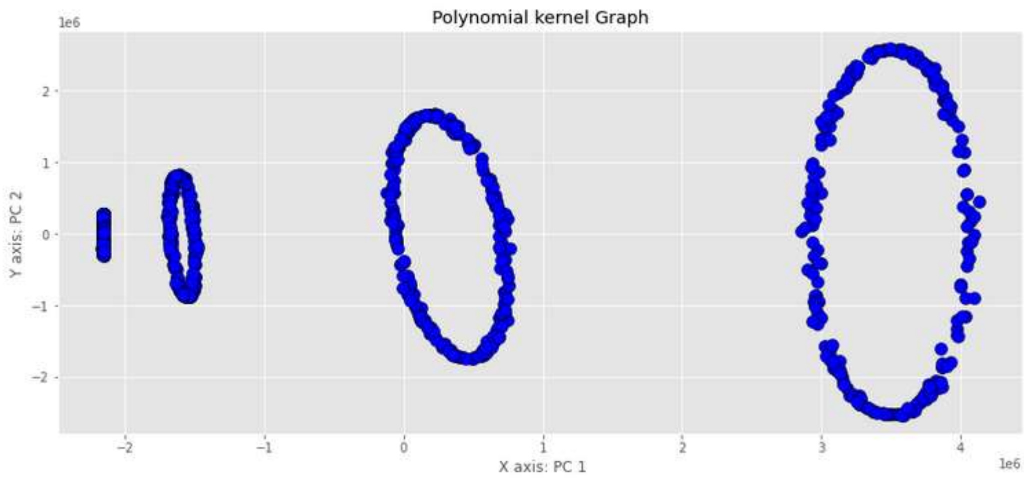
Hence centering is very important

In the Fig-1, we could-not see much difference between the plots before centering and after centering. This is because the Mean Values are very small and it is almost 0 (approximately equal to 0)

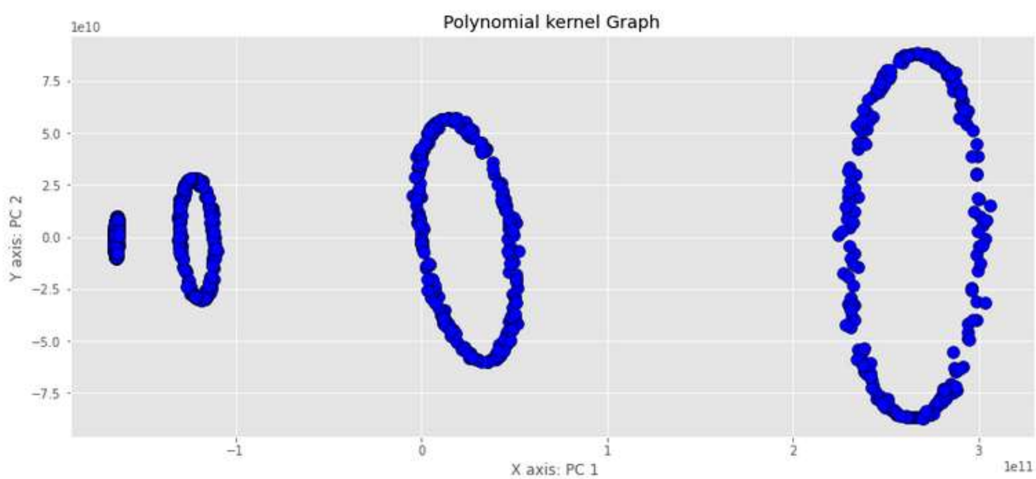
The Fig-2, Fig-3 shows the effect of centering the data on other samples. Here in Fig-2, we can see that the intercept is removed after centering the data. The same applies for Fig-3. The principle components after centering is not in the direction of the Mean of the data.

1) iii) A)

```
proj [-2162313.4172137    263121.14150229]
d is: 2
```

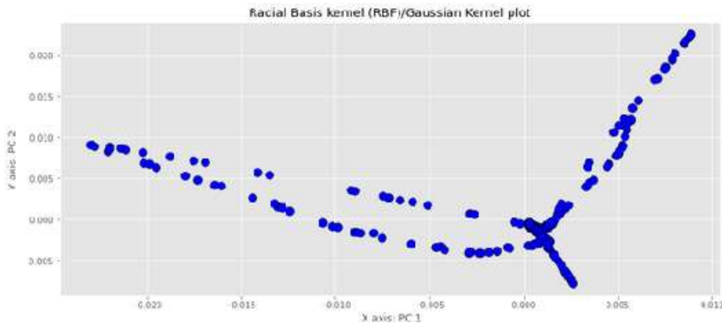



```
proj [-1.64320123e+11  9.16660678e+09]
d is: 3
```

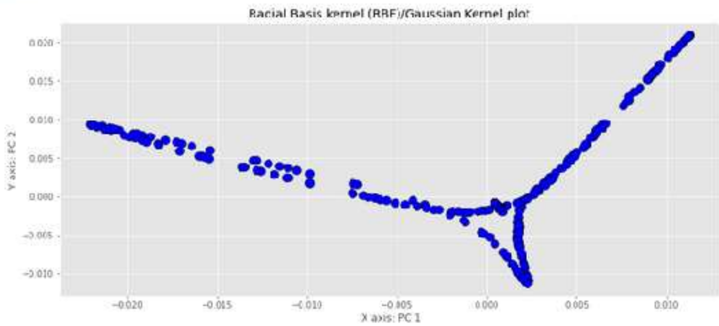


1) iii) B)

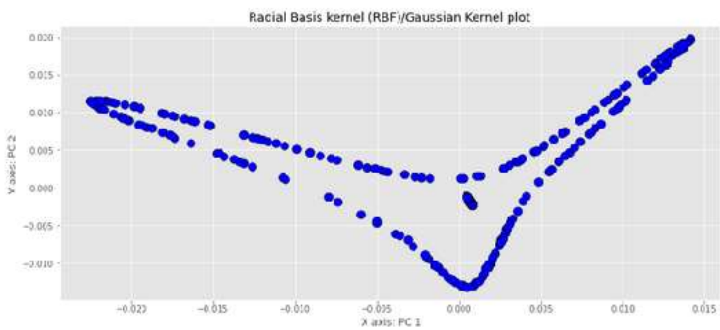
```
Sigma is: 0.1
```



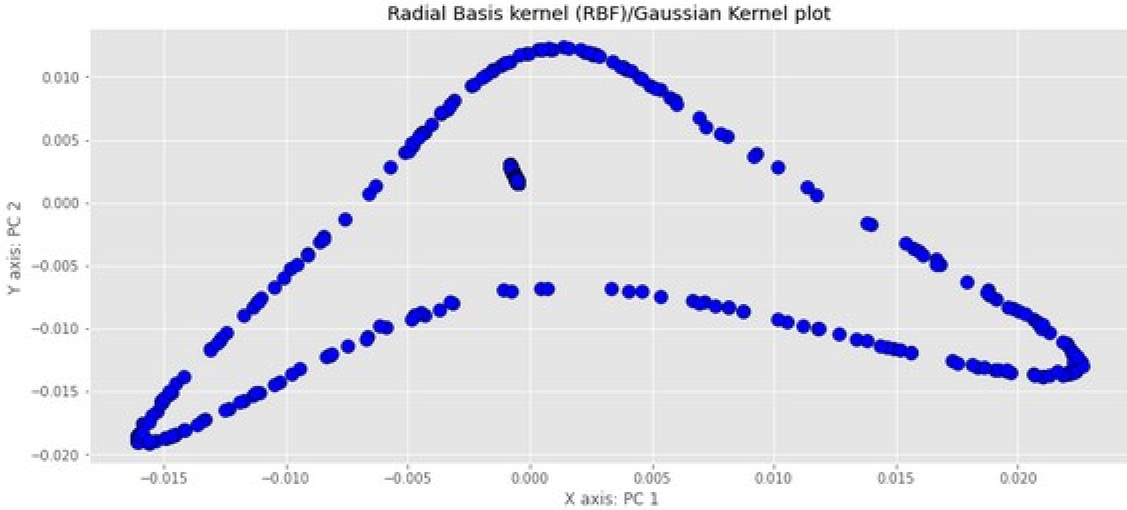

```
Sigma is: 0.2
```



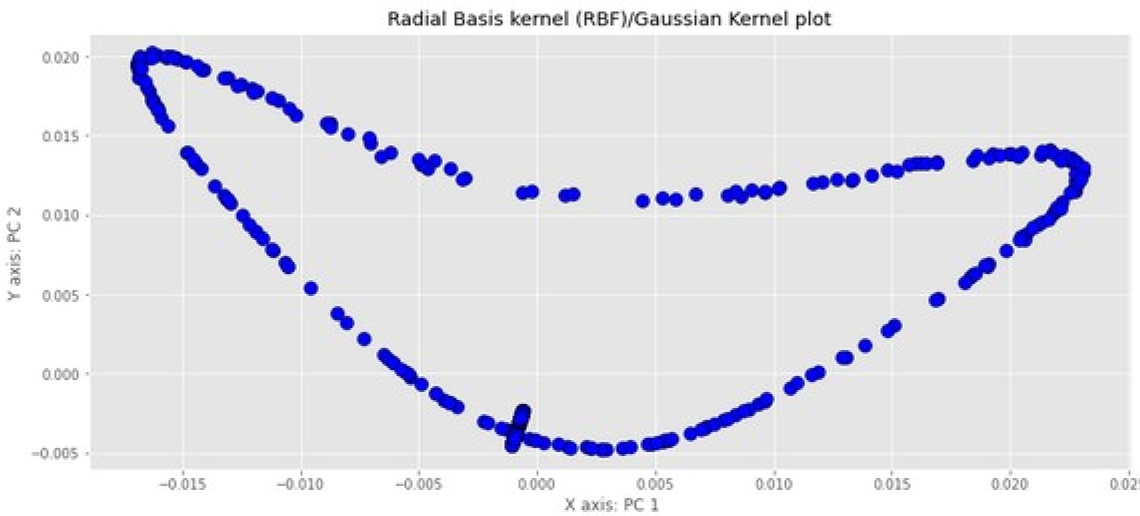

```
Sigma is: 0.3
```



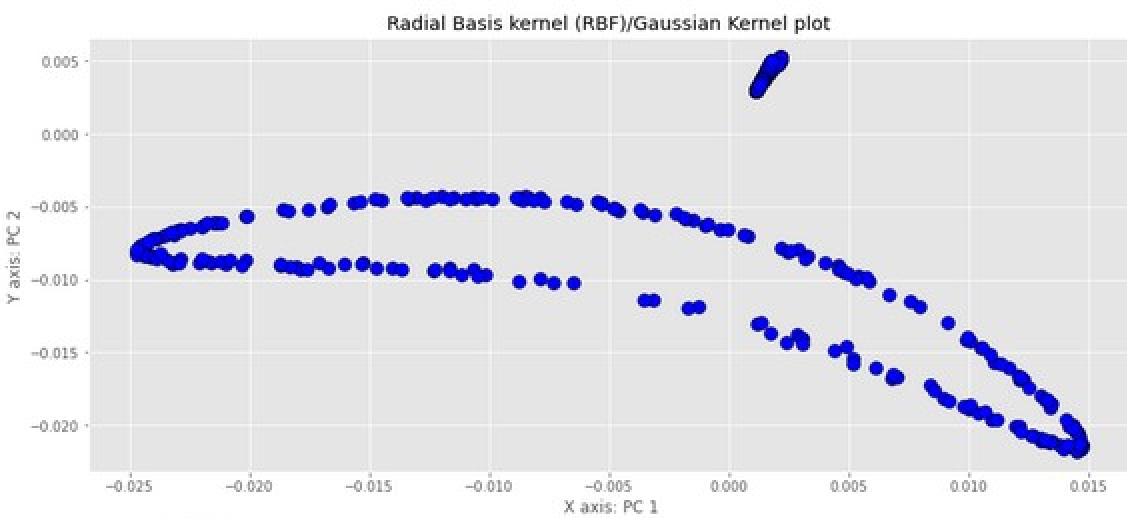
Sigma is: 0.4



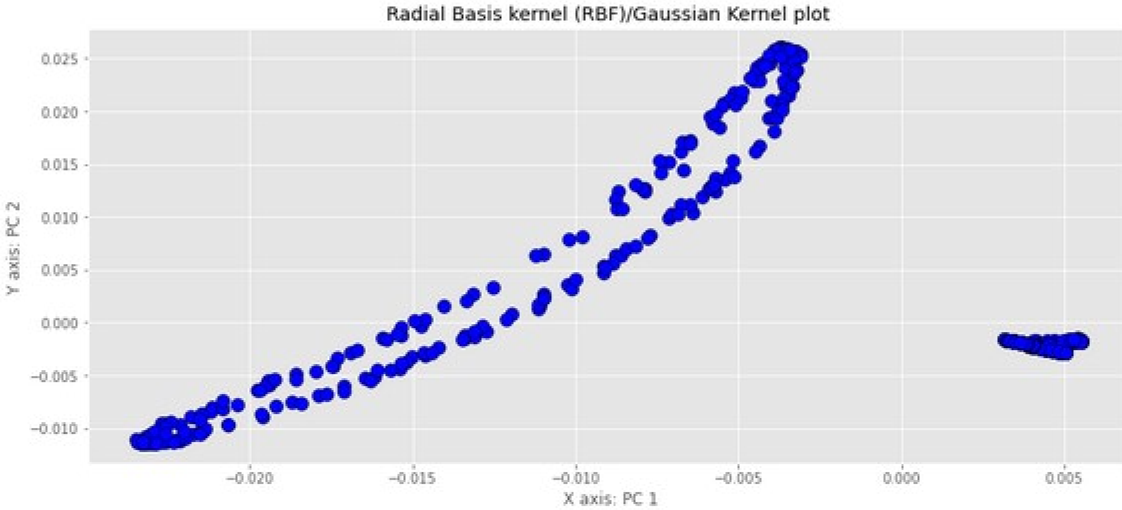
Sigma is: 0.5



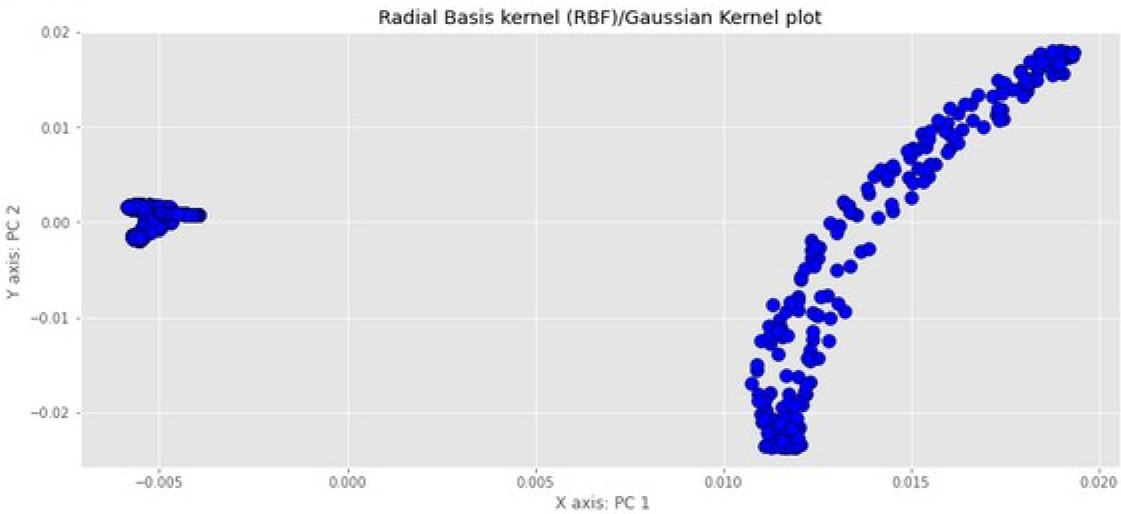
Sigma is: 0.6



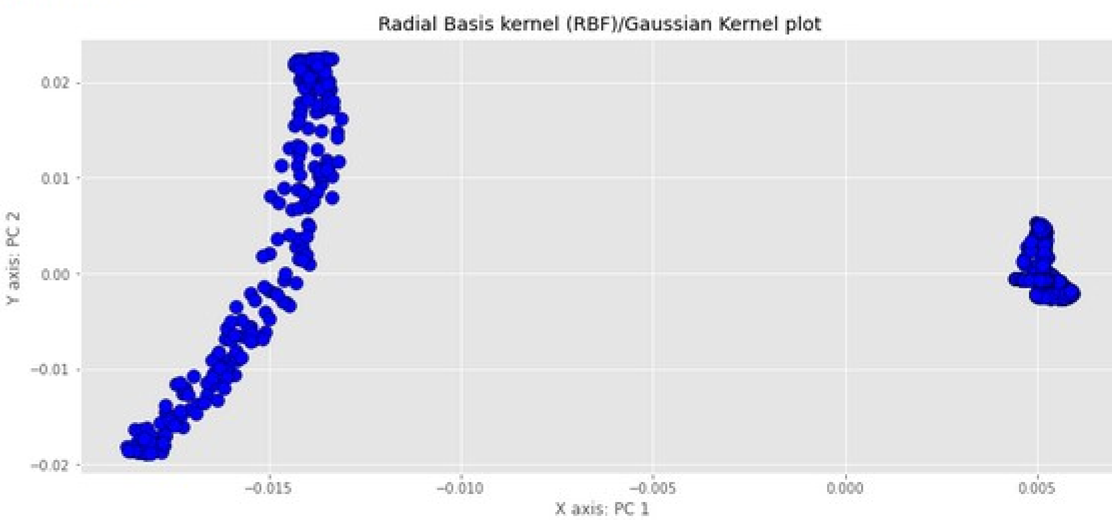
Sigma is: 0.7



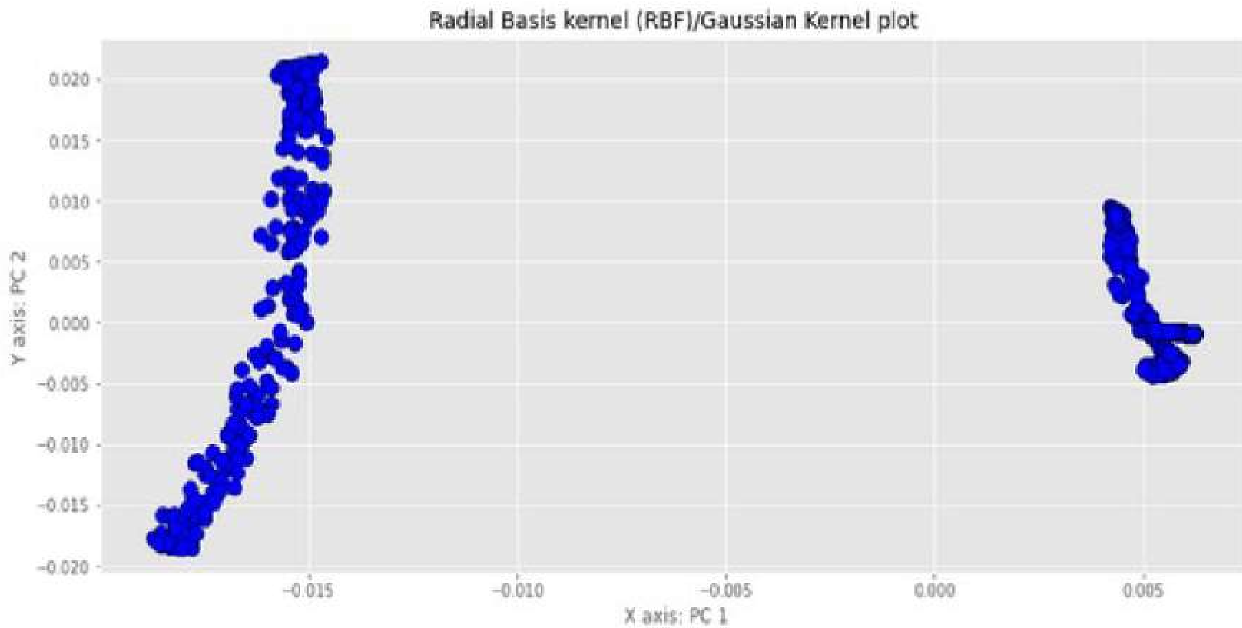
Sigma is: 0.8



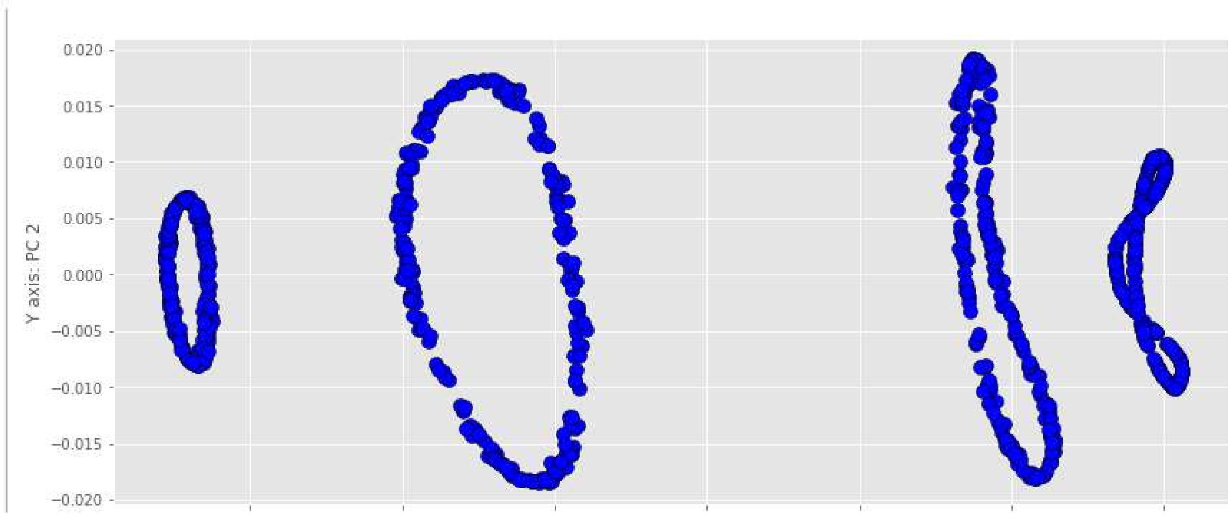
Sigma is: 0.9



Sigma is: 1



Sigma is : 3.16



(1) (iii) (A)

By using the polynomial kernel of degree 2 and degree 3, we can see that the plots are linearly separable

(1) (iii) (B)

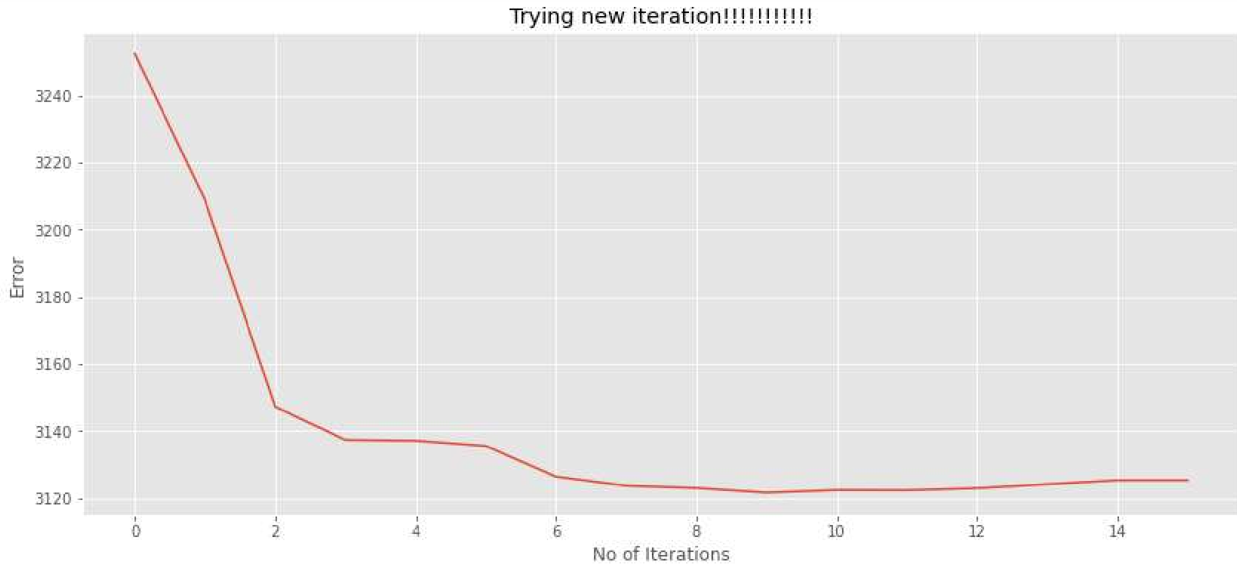
By using Gaussian kernel, with the higher sigma values, the graph becomes more linearly separable. You can observe the diagram when sigma = 3.16. The graph is completly linearly separable.
∴ " As sigma-values increases, more linear separable the graph is ".

(1) (iv) (A)

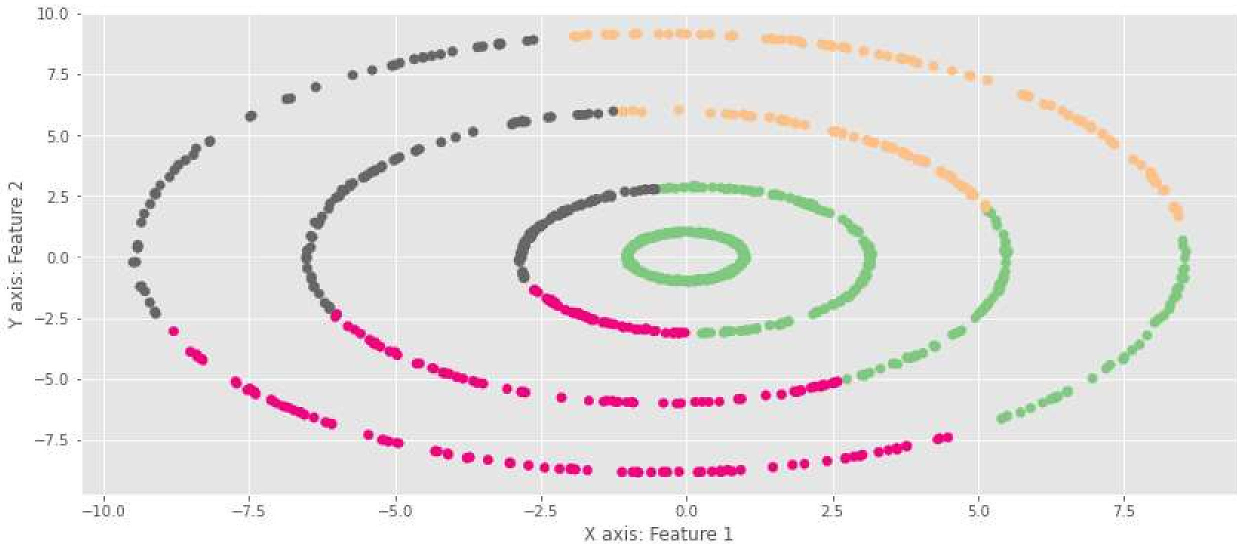
Refer diagrams of 1 (iii)

Ofcourse by looking at the graph, we say polynomial kernel is best suited for this data-set. Because by using the polynomial kernel the graph is linearly separable.
Gaussian kernel with higher sigma value (sigma = 3.16) also makes the graph linearly separable.

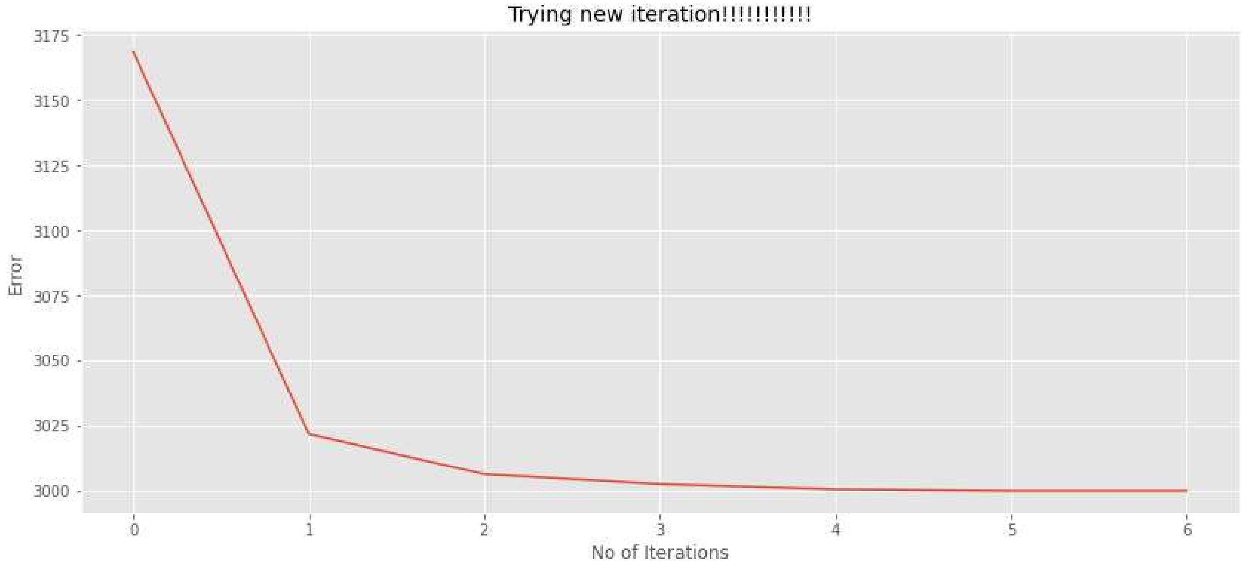
2 i)



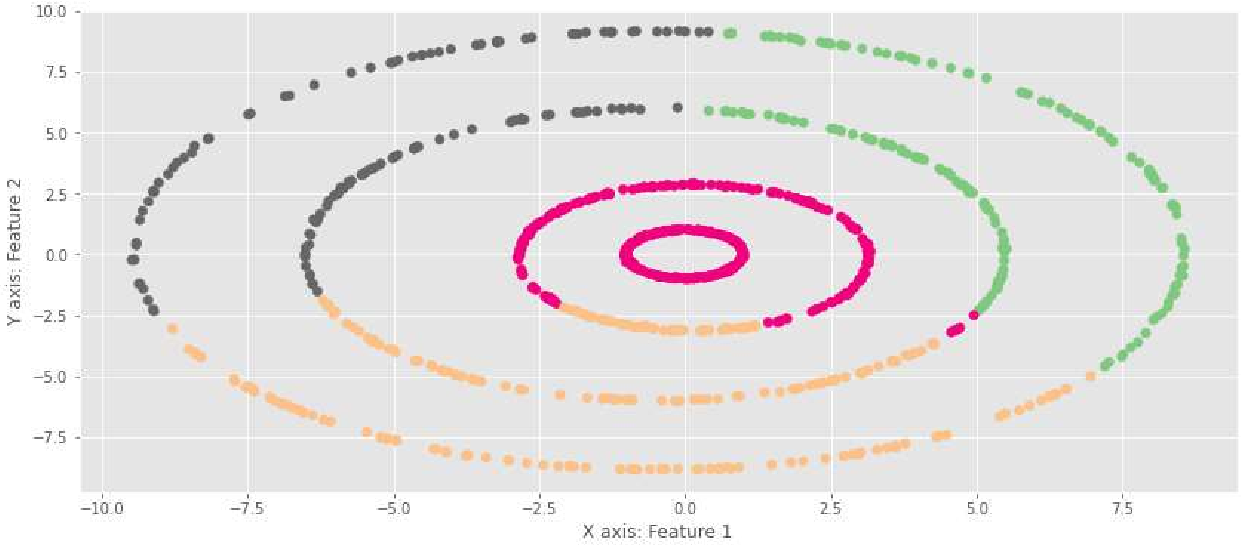
Trying new iteration!!!!!!!!!!!
3240
3220
3200
3180
3160
3140
3120
Error
0
2
4
6
8
10
12
14
No of Iterations



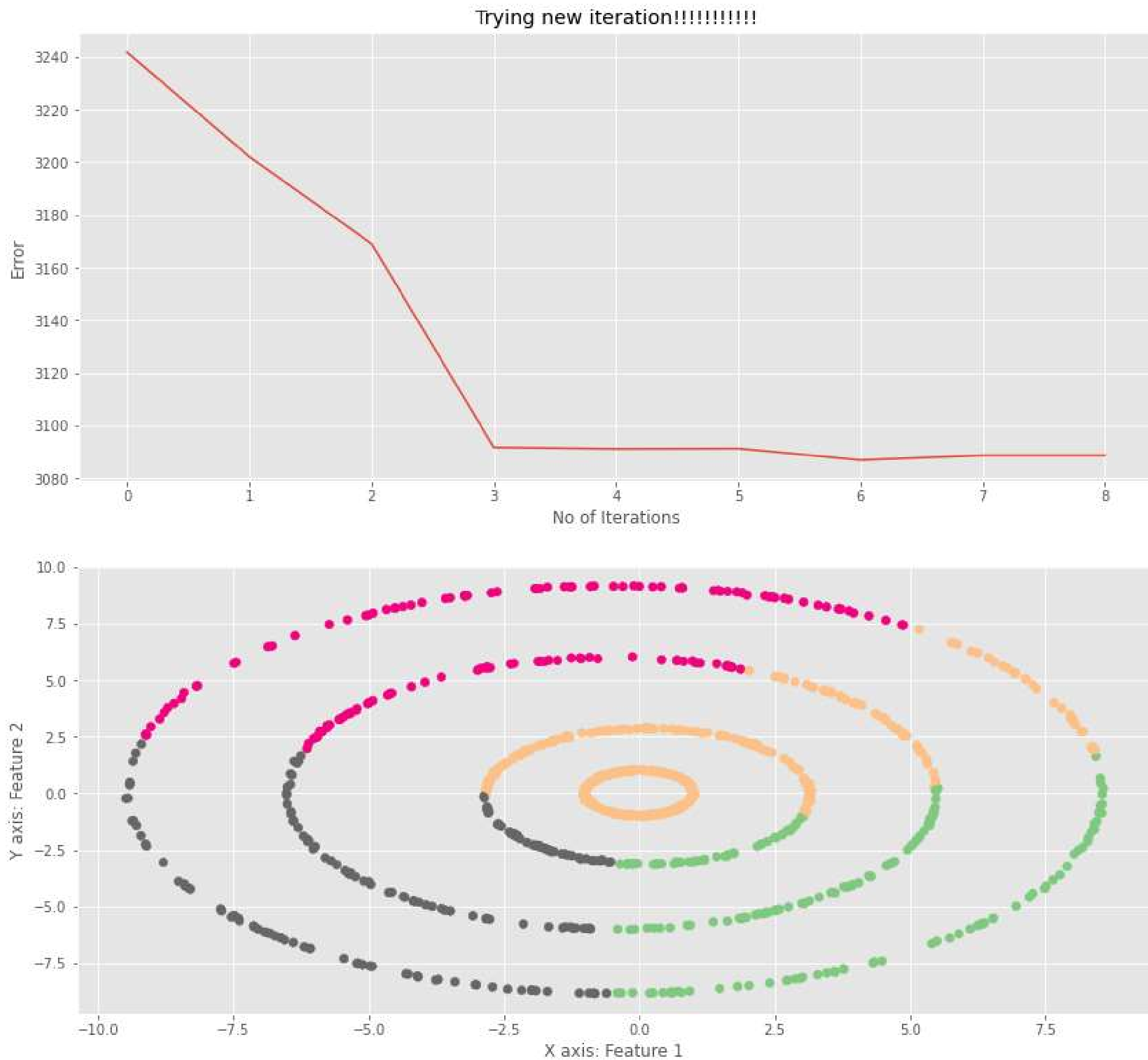
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1



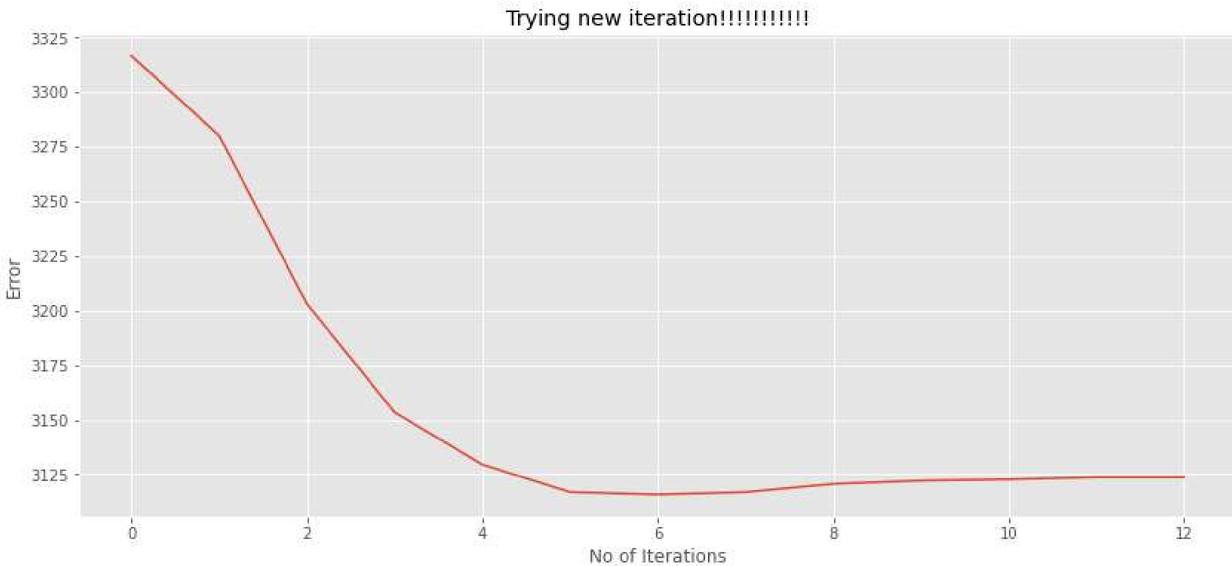
Trying new iteration!!!!!!!!!!!
Error
No of Iterations
3175
3150
3125
3100
3075
3050
3025
3000
0
1
2
3
4
5
6

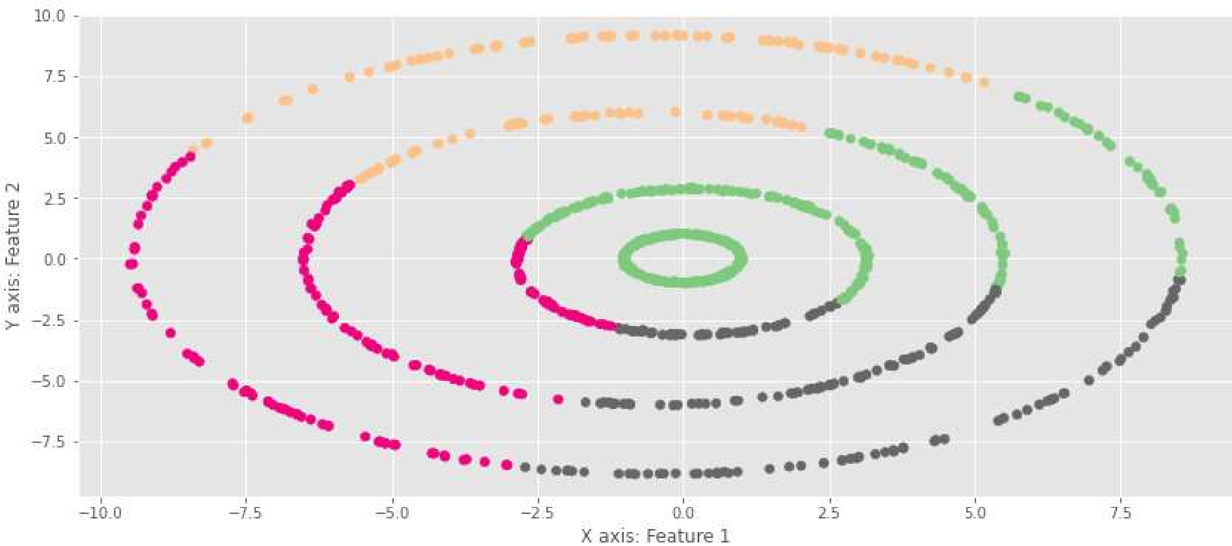


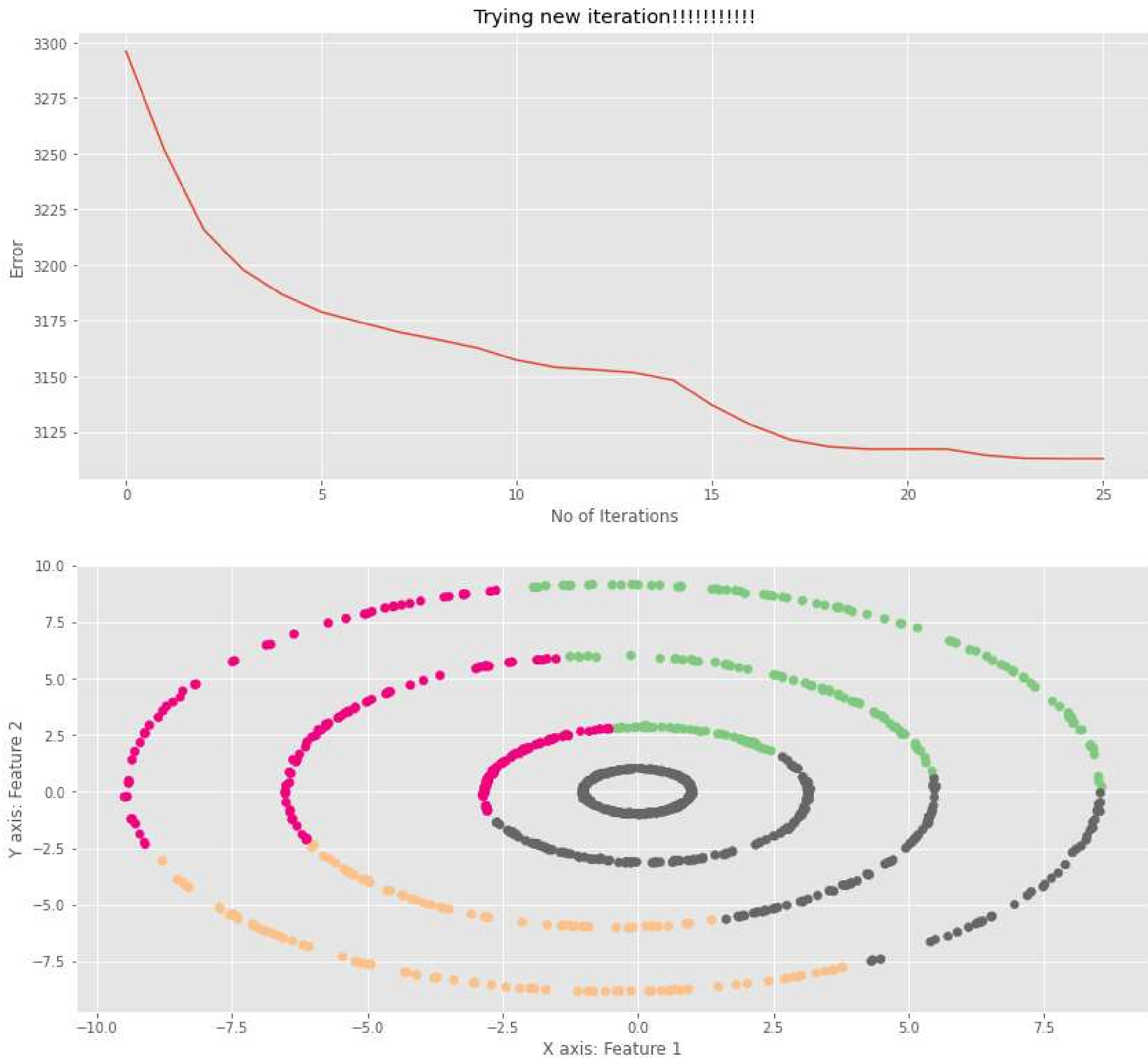
Y axis: Feature 2
X axis: Feature 1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5



Trying new iteration!!!!!!!!!!!
Error
No of Iterations
Y axis: Feature 2
X axis: Feature 1


Trying new iteration!!!!!!!!!!!
3325
3300
3275
3250
3225
3200
3175
3150
3125
Error
0
2
4
6
8
10
12
No of Iterations


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1



Trying new iteration!!!!!!!!!!!
3300
3275
3250
3225
3200
3175
3150
3125
Error
0
5
10
15
20
25
No of Iterations
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1

(2)

(i)

By looking at the graph of 5 different random initialization we can say that,

i) Every time different centroid will be choosen for the clusters

iii) Every time the algorithm converges and the best clusters are formed.

Error function :

X axis : No of iterations.

we try to run for maximum iteration and see that the algorithm converges.

Y axis : Error.

Error is calculated as below

$$\left[\text{Error}\right] = \left\|\left[\text{data point}\right] - \left[\text{its centroid}\right]\right\|^2 \quad \text{... for each point.}$$

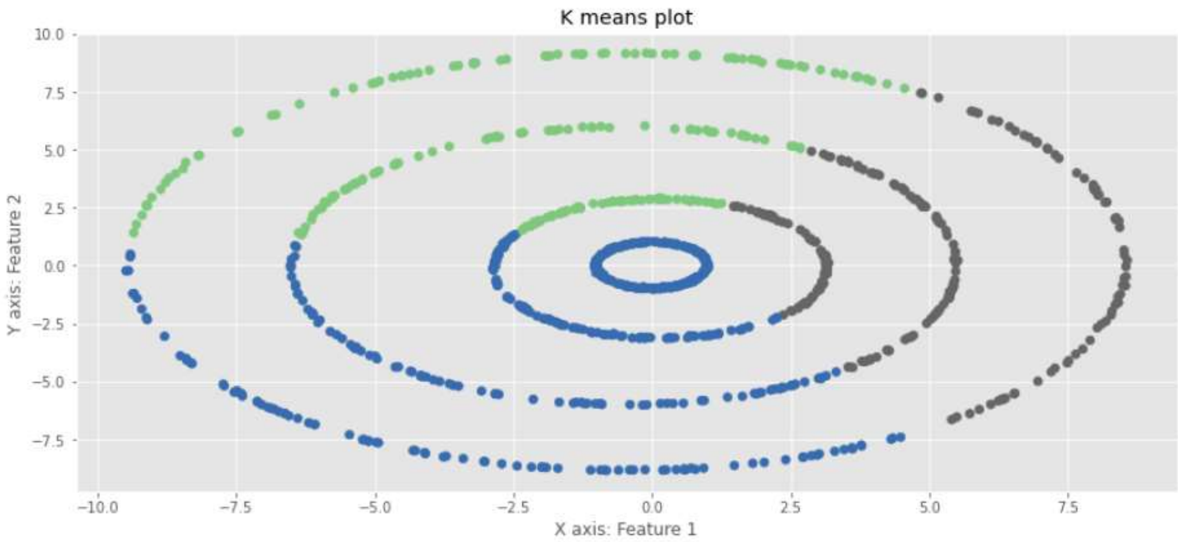
$$\begin{bmatrix}\text{total}\\\text{error}\end{bmatrix} = \text{error}_1 + \text{error}_2 + \cdots + \text{error}_n \quad \text{... for all the data points.}$$

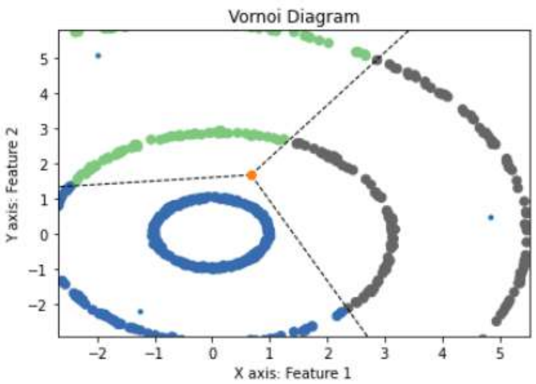
The error can also be said as the "sum of the euclidean distance between the data-point and its centroid".

Observation :

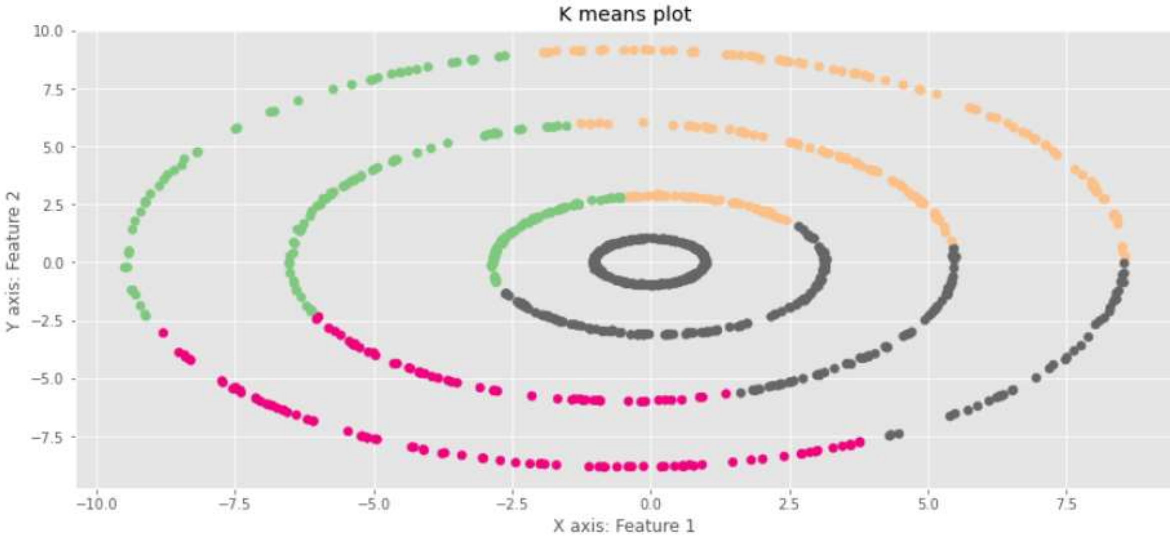
We see that the error is high initially when the algorithm chooses random centroids. The error goes on reducing for each iteration of the algorithm. This is because the datapoints move to the right clusters based on its position from the centroid. Finally the algorithm converges when the error becomes 0 (zero).

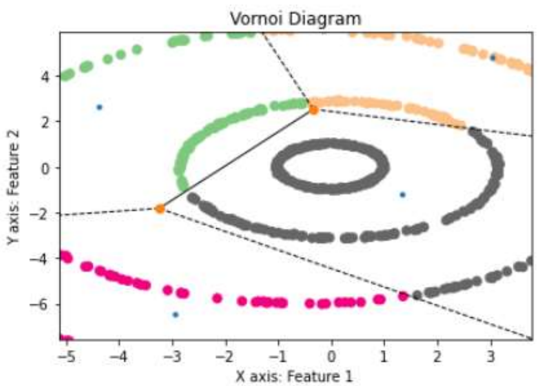
2) ii)
K = 3


K means plot
Y axis: Feature 2
X axis: Feature 1

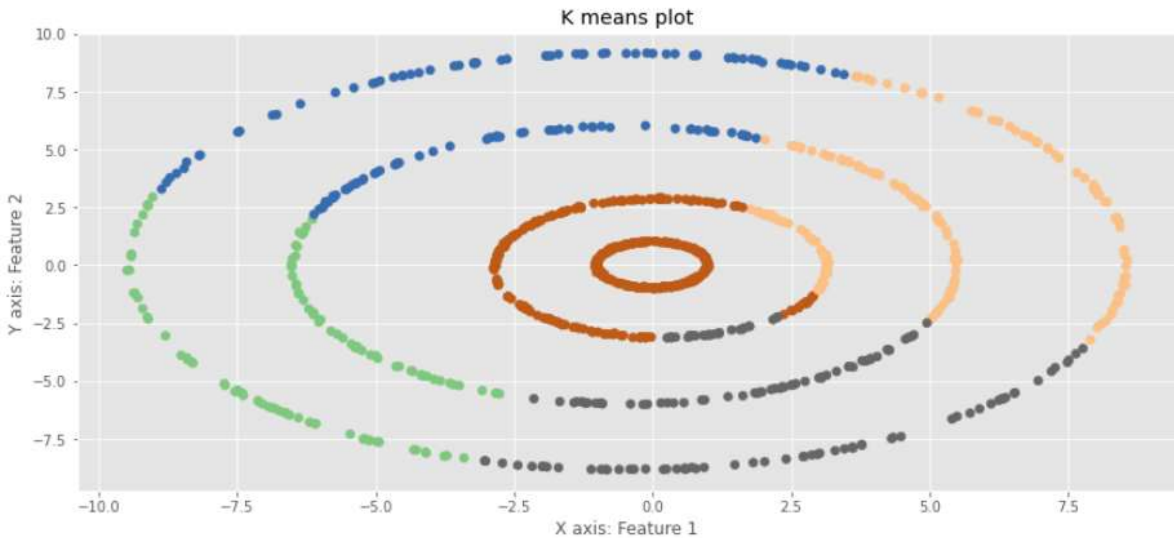

Vornoi Diagram
Y axis: Feature 2
X axis: Feature 1

K = 4

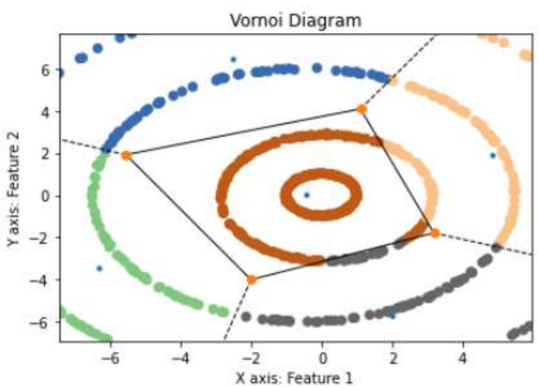

K means plot
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1


Vornoi Diagram
4
2
0
−2
−4
−6
Y axis: Feature 2
−5
−4
−3
−2
−1
0
1
2
3
X axis: Feature 1

K = 5

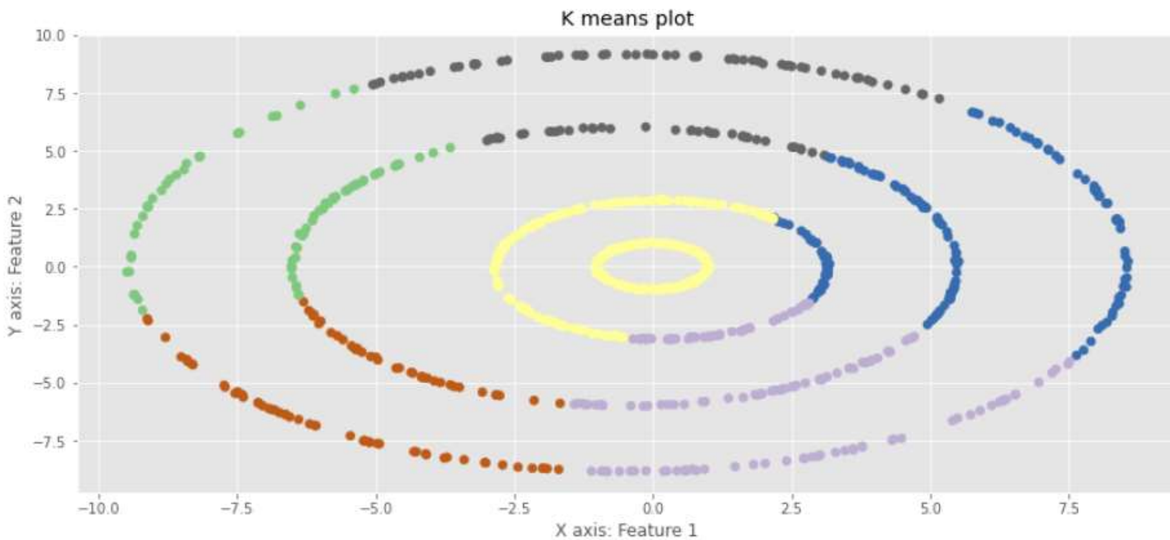


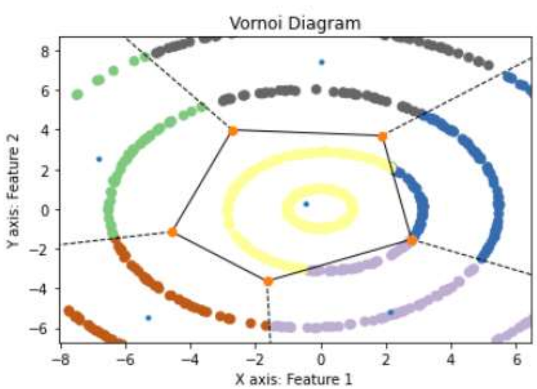
K means plot
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1



Vornoi Diagram
6
4
2
0
−2
−4
−6
Y axis: Feature 2
−6
−4
−2
0
2
4
X axis: Feature 1

K = 6


K means plot
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
Y axis: Feature 2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
X axis: Feature 1


Vornoi Diagram
8
6
4
2
0
−2
−4
−6
Y axis: Feature 2
−8
−6
−4
−2
0
2
4
6
X axis: Feature 1

2

ii

The plots shows the vornoi diagrams for different k values

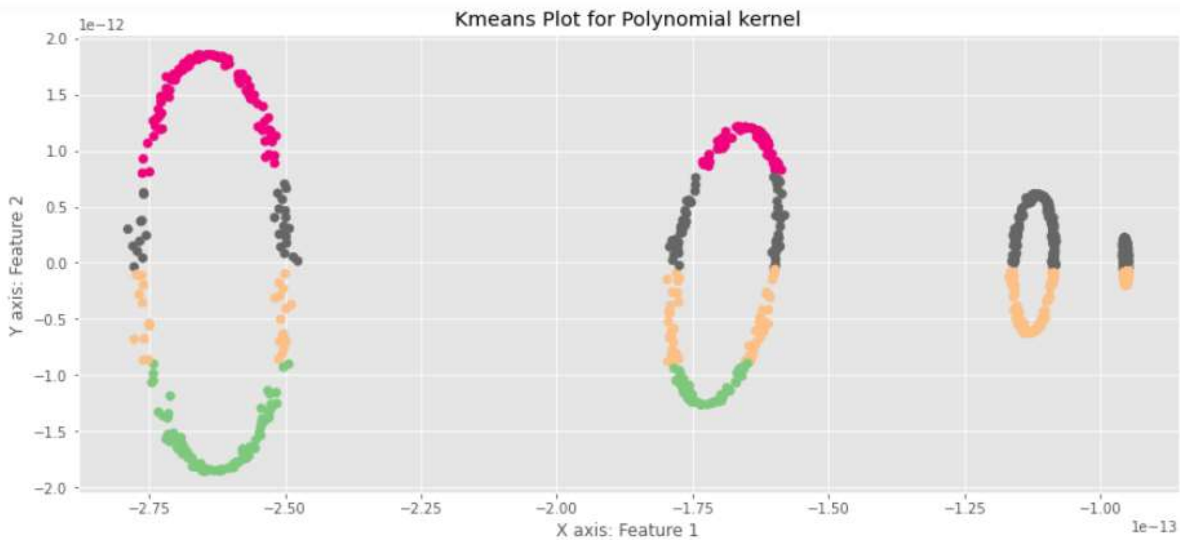
(Blue point) → It is the cluster centroid
in vornoi dig

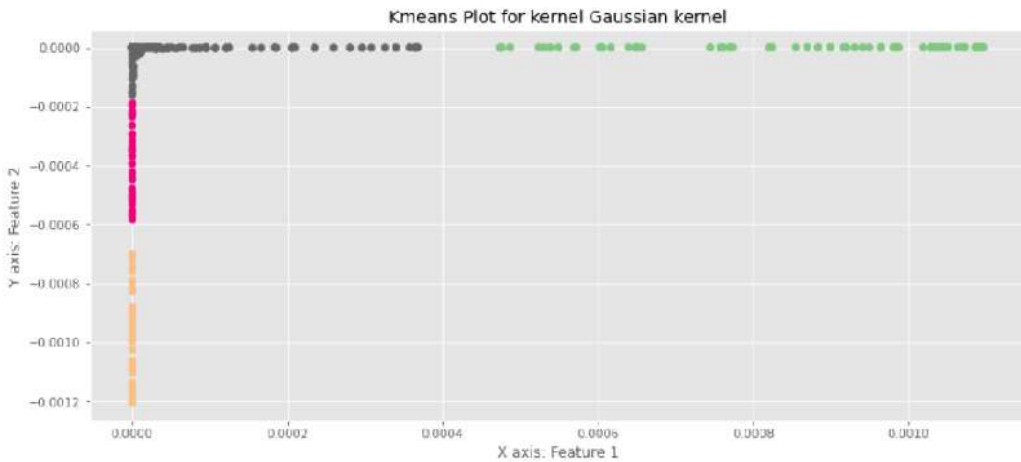
What does vornoi diagram says?

a) It shows the cluster centroid.

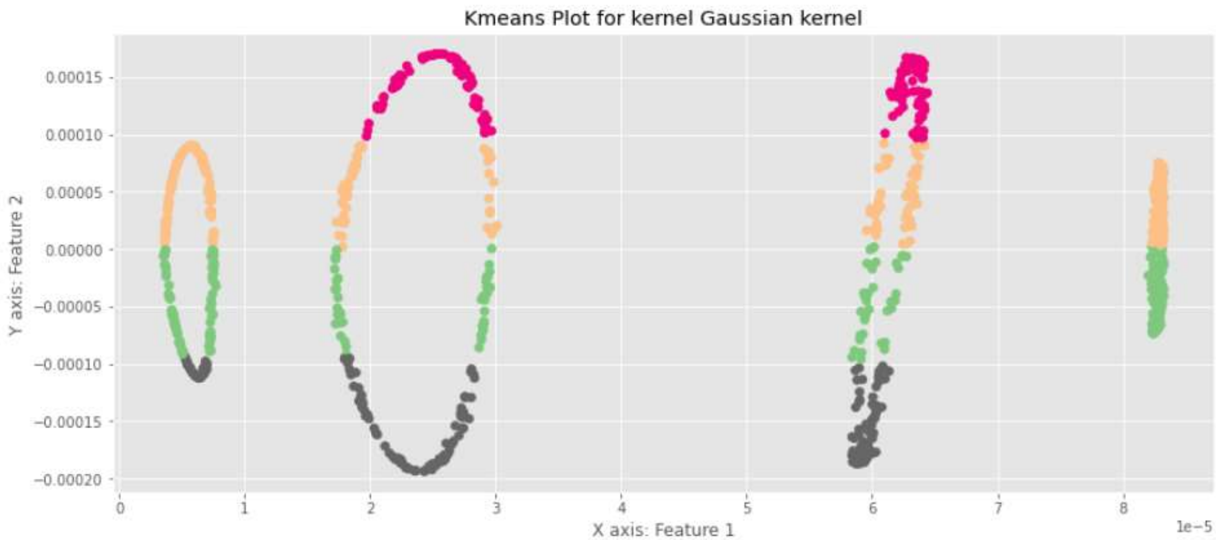
b) It shows the partition that each cluster has for its data points.

2 iii)


Kmeans Plot for Polynomial kernel
1e−12
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Y axis: Feature 2
−2.75
−2.50
−2.25
−2.00
−1.75
−1.50
−1.25
−1.00
1e−13
X axis: Feature 1


Kmeans Plot for kernel Gaussian kernel
0.0000
−0.0002
−0.0004
−0.0006
−0.0008
−0.0010
−0.0012
Y axis: Feature 2
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
X axis: Feature 1

When gamma is 0.05
(Smaller value of gamma, the gaussian kernel gives linearly separable clusters)



(2)

(iii)

Spectral clustering algorithm
(Spectral relaxation of k-means using kernel PCA)

Observation

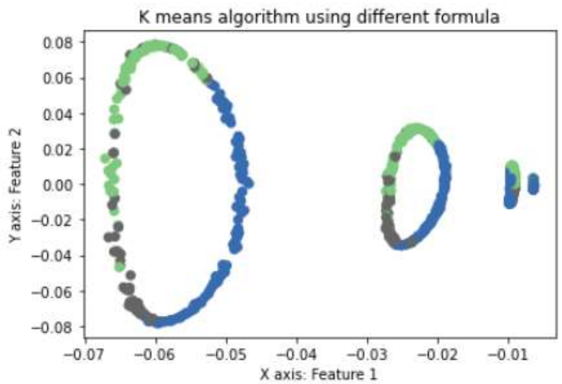
Polynomial kernel	Gaussian kernel
i) converges little late (for iteration no = 12 or 13)	converges early (for iteration no = 4, 5)
ii) Gives linearly separable clusters.	It will provide linearly separable cluster only for the ~~higher~~ lower gamma value. [eg gamma = 0.05]
iii) cluster centroids are choosen according to the ~~linear~~ different planes so the the clusters are linearly separable	iii) The two or more centroids may lie on the same plane for higher gamma value. Therefore, clusters may not be linearly separable.

∴ By above observation, eventhough the polynomial converges little late, it is best suited for the current data-set.

Steps used in the algorithm:

i) Calculate kernel matrix
ii) Do Eigen decomposition.
iii) Do normalization.
iv) Consider this as the input and apply k-means algorithm over it.

2) iv)



K means algorithm using different formula
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
Y axis: Feature 2
−0.07
−0.06
−0.05
−0.04
−0.03
−0.02
−0.01
X axis: Feature 1

(2)

(iv)

By using the above formula, the kmeans-algorithm could-not converge even after 1000 iterations.

This is because the data-points keep moving between the clusters. The reason for this is the data-point just chooses the cluster according to the formula and ~~&~~ not by the distance from its centroid.

By using this formula, the distance between the data-point and its centroid will never reduce. It just keeps on oscillating to different random values.

Therefore the error function never becomes zero.

Therefore the algorithm never converges.

And ~~so~~ from the plot we can observe the below things.

(i) The clusters are not linearly separable

(ii) two or more clusters overlap at some points

(iii) Therefore, cluster centroids are not defined properly since two or more data-points of one cluster goes to the other cluster [i.e; not linearly separable]

Steps used in the algorithm :

i) ~~&~~ Calculate Kernel Matrix

ii) ~~Do~~ eigen decomposition to get the eigen vectors

iii) Sort the eigen vectors based on the eigen values. Keep the top 4 eigen vector [top 4 columns]

(iv) apply the formulla.

Eg

$$\begin{array}{ll} \text{pos:} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ & \begin{bmatrix} \boxed{0 \quad 1 \quad 4 \quad 2} \\ 5 \quad \boxed{100} \quad 1 \quad 2 \end{bmatrix} \end{array}$$

4 is highest.
Therefore this ~~dxp~~ data point will go to the III cluster (given by pos)

100 is highest.
Therefore this data point will go to the II cluster [given by pos]