In God We Trust

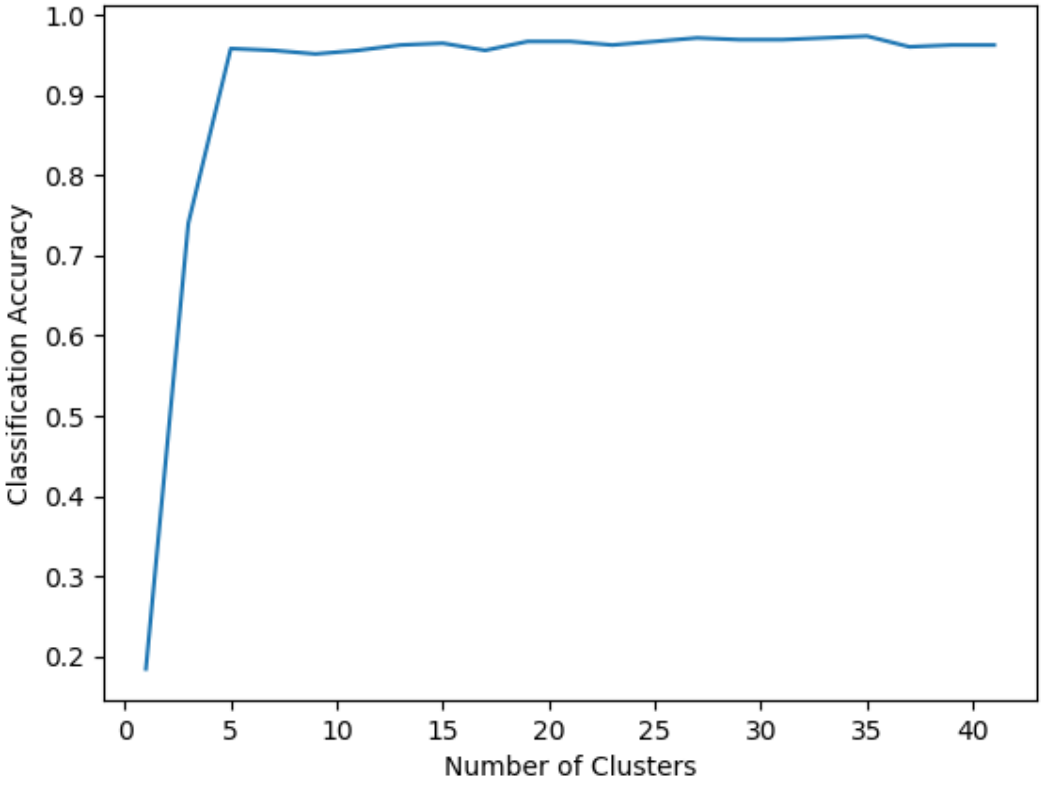
First of all, we have to calculate the right number of clusters in order to optimize the classification process for each dataset. In order to do that, we have drawn the figure of classification accuracy vs. number of clusters for each dataset. Then, the optimized number of clusters is calculated considering the following criteria:

1) It should give us an acceptable accuracy. Equivalently, an accuracy above a given threshold(97% or 98%). However, it isn't necessarily the maximum accuracy possible according to the second criterion.
2) Overfitting should be avoided. For example, we could prioritize c = 20 with an accuracy of 98% to one with c = 80 and accuracy of 98.3%; in spite of having a less accuracy. Here, time and resource consumption is a crucial factor in determining the best number of clusters for each dataset.

Having the criteria mentioned above, we have chosen the optimized number of clusters. Note that we haven't adopted a systematic approach for determining the right number of clusters and the number is extracted from the figures by means of vision, intuition and instinct.

More information about each figure is available in corresponding '.txt' files associated with each dataset.

For the '5clstrain1500.csv', the figure is as follows:

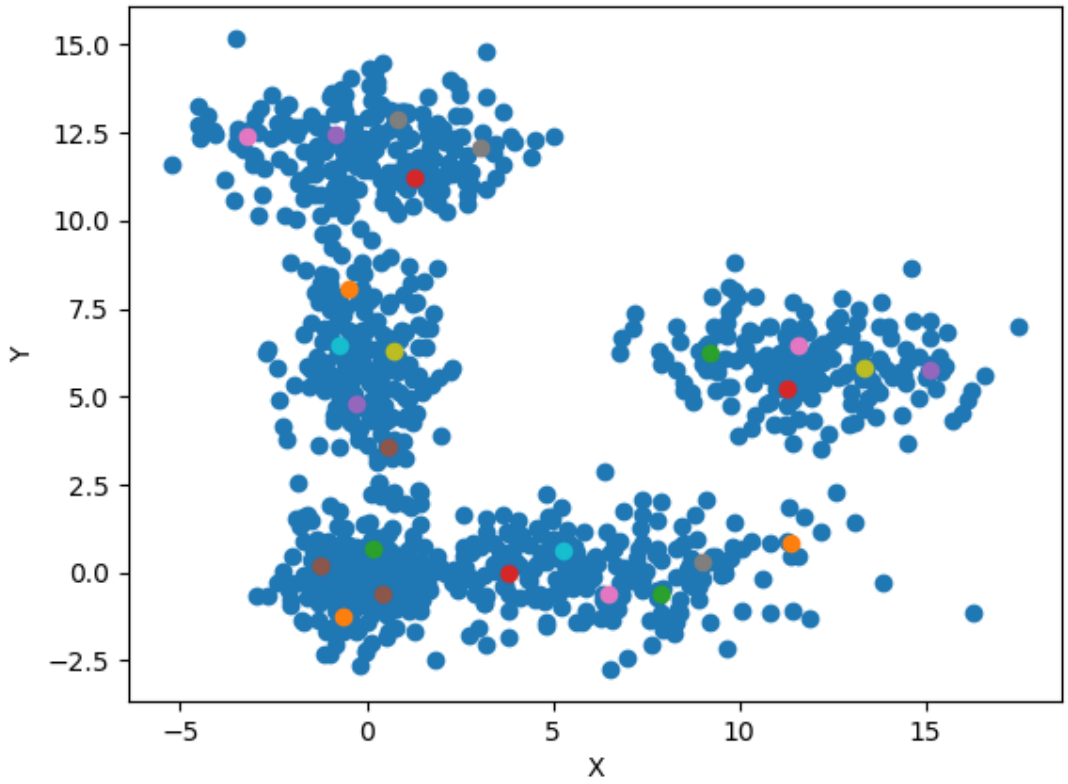


Hence, the optimized number of clusters which gives us the best accuracy is

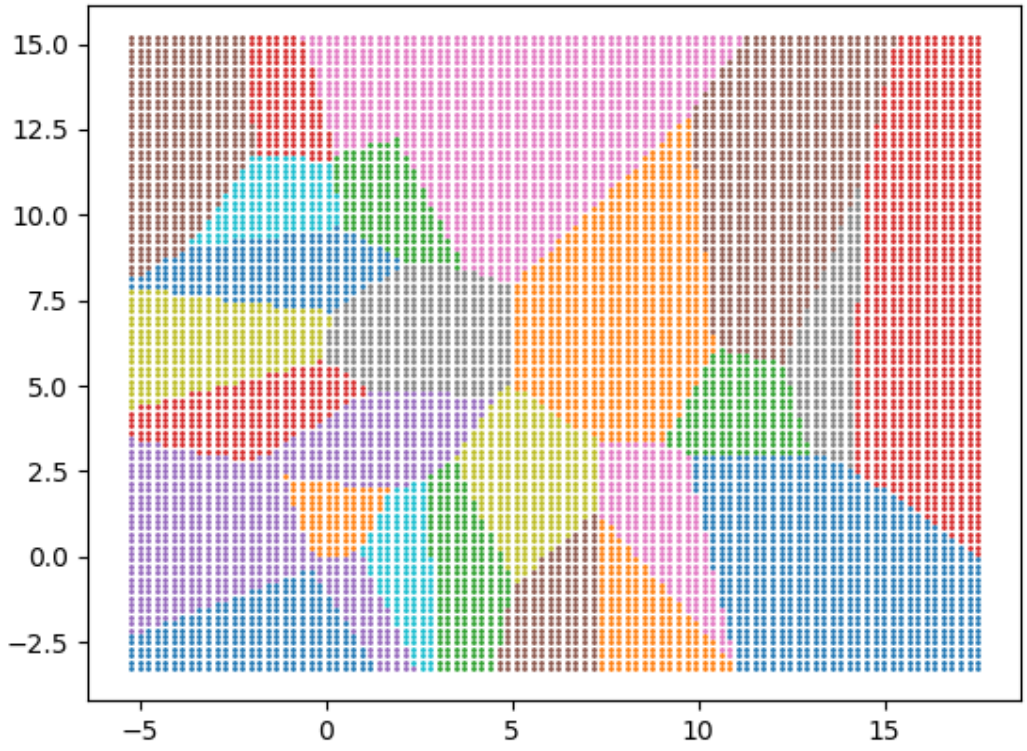
number of clusters = 27

RBF Train Accuracy = 0.9638095238095238

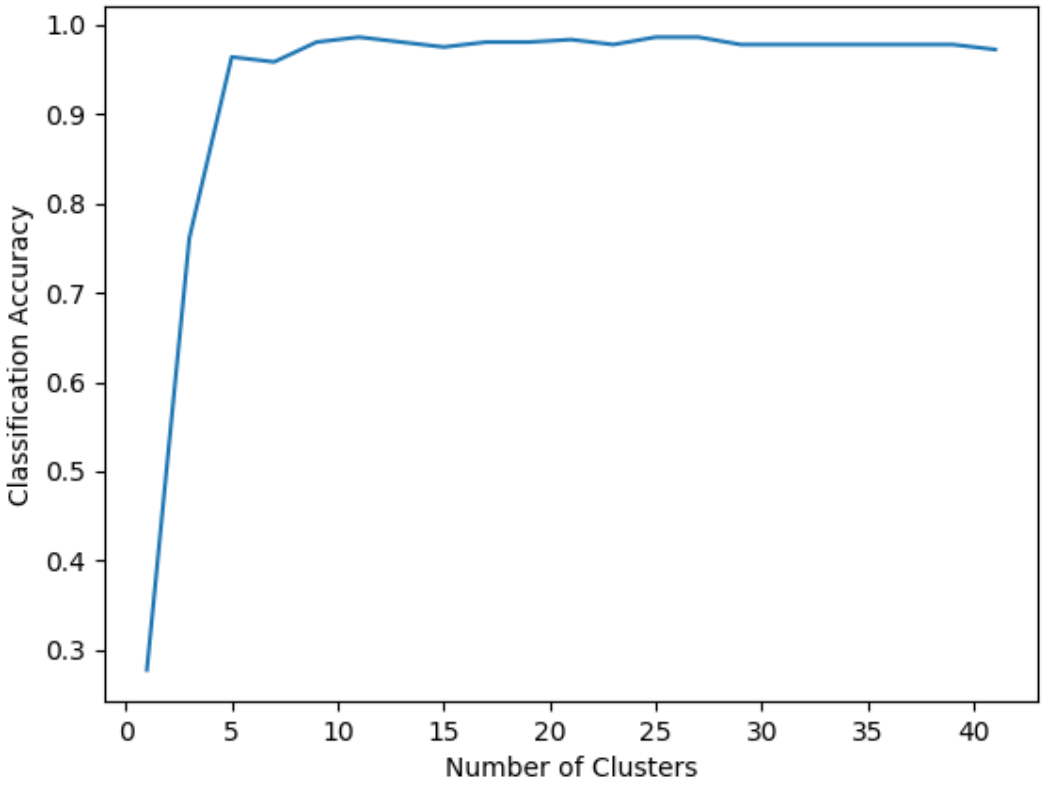
RBF Test Accuracy = 0.9711111111111111



In that case, the FCM cluster boundaries will be as follows:



For the ‘4clstrain1200.csv’, the figure is as follows:

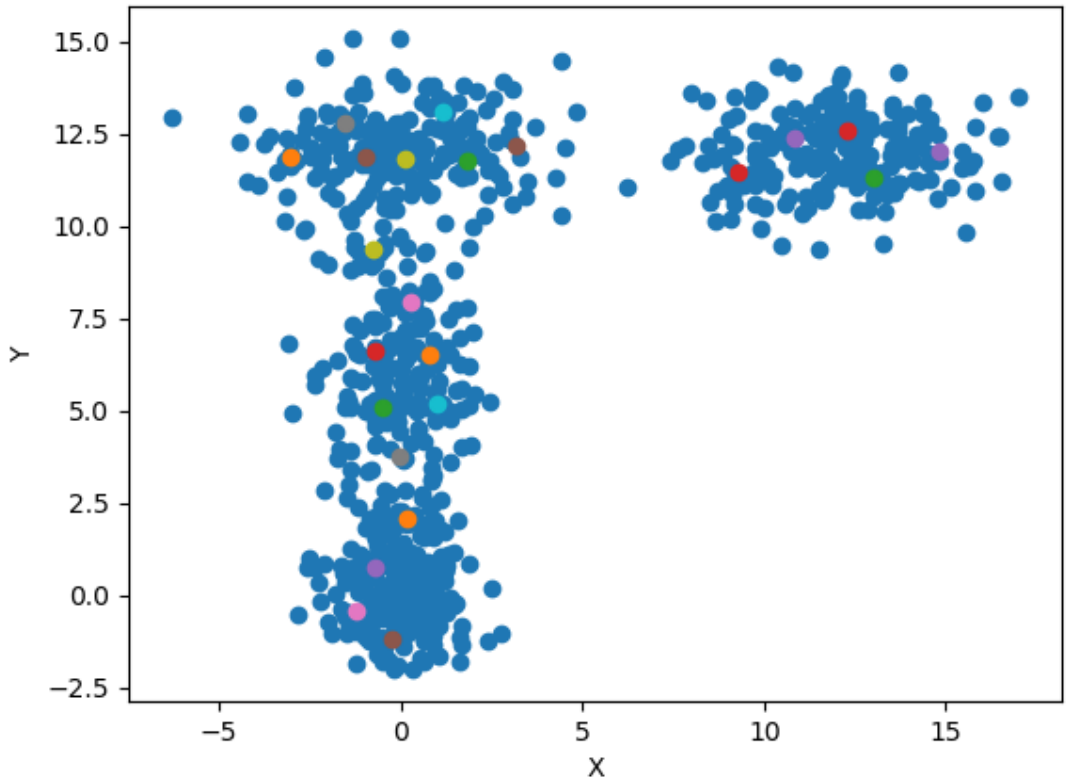


Hence, the optimized number of clusters which gives us the best accuracy is

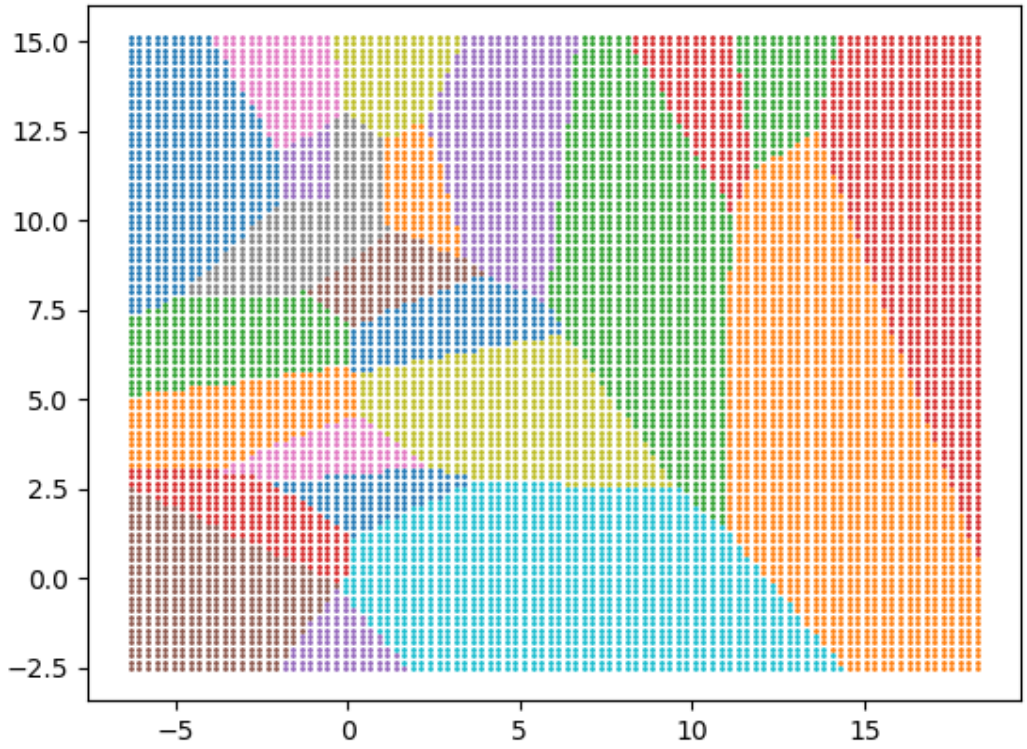
number of clusters = 25

RBF Train Accuracy = 0.975

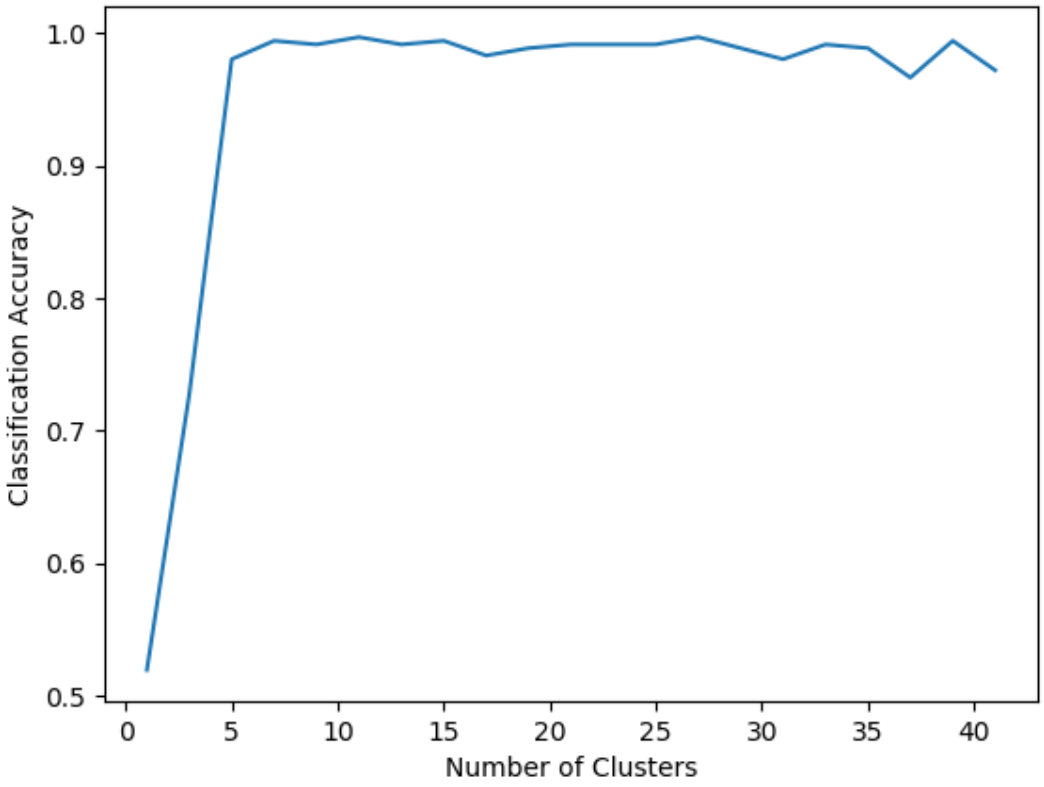
RBF Test Accuracy = 0.9861111111111112



In that case, the FCM cluster boundaries will be as follows:



For the ‘2clstrain1200.csv’, the figure is as follows:

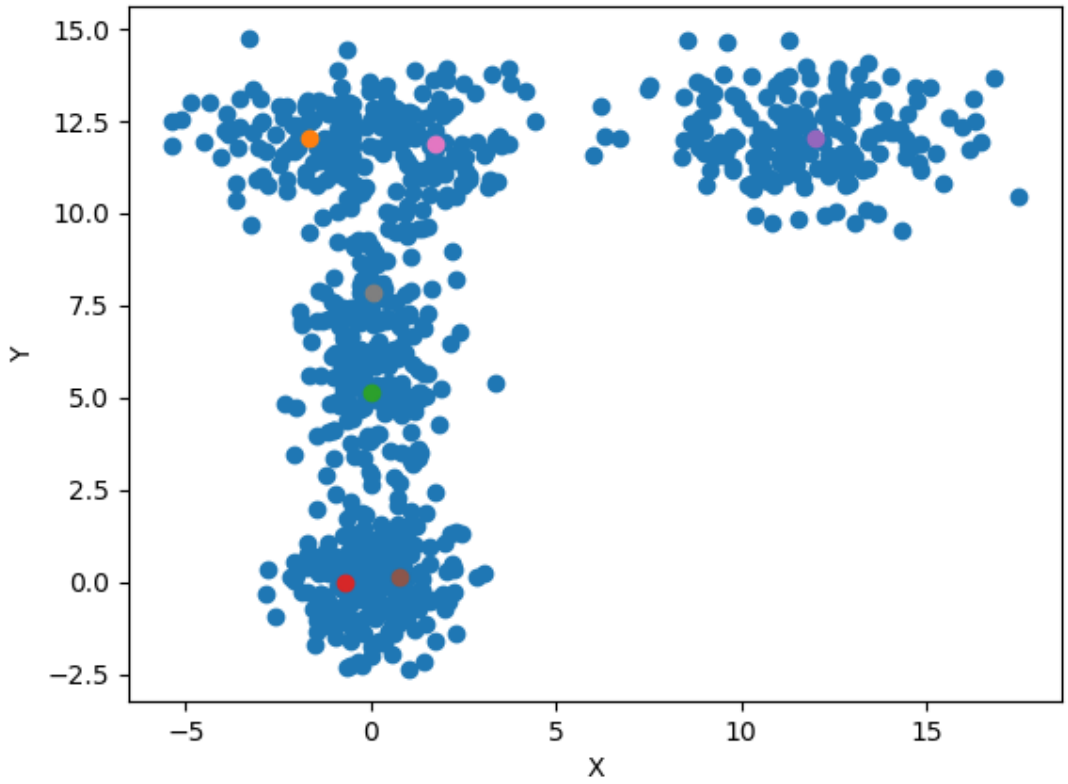


Hence, the optimized number of clusters which gives us the best accuracy is

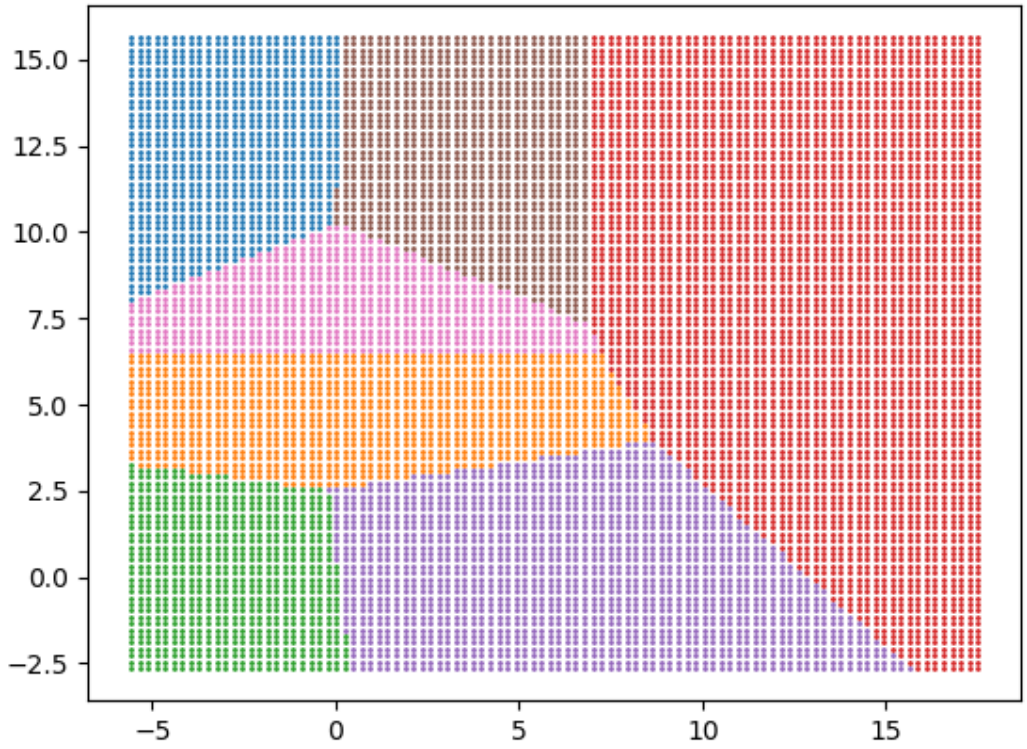
number of clusters = 7

RBF Train Accuracy = 0.9773809523809524

RBF Test Accuracy = 0.9944444444444445



In that case, the FCM cluster boundaries will be as follows:



We have calculated the accuracy of these special cases because they were asked in the project definition:

1- Gamma = 0.1, c = 40

2- Gamma = 1, c = 3

RBF on these special cases is run on different datasets and the result is illustrated in the table below:

gamma	c	Dataset	Training accuracy	Test accuracy
0.1	40	5clstrain1500	0.9619047619047619	0.9533333333333334
0.1	40	4clstrain1200	0.9785714285714285	0.975
0.1	40	2clstrain1200	0.9821428571428571	0.9777777777777777
1	3	5clstrain1500	0.599047619047619	0.6022222222222222
1	3	4clstrain1200	0.7714285714285715	0.7
1	3	2clstrain1200	0.7488095238095238	0.75

Analysis:

1) As we see, we can approximately say in a fixed number of clusters, accuracy decreases with respect to an increase in the number of classes.
2) gamma is a parameter of the RBF kernel and can be thought of as the 'spread' of the kernel and therefore the decision region. When gamma is low, the 'curve' of the decision boundary is very low and thus the decision region is very broad. When gamma is high, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points.

For examples of the second tip, you can visit the following link to see more examples about the influence of gamma on the classifier's performance and accuracy:

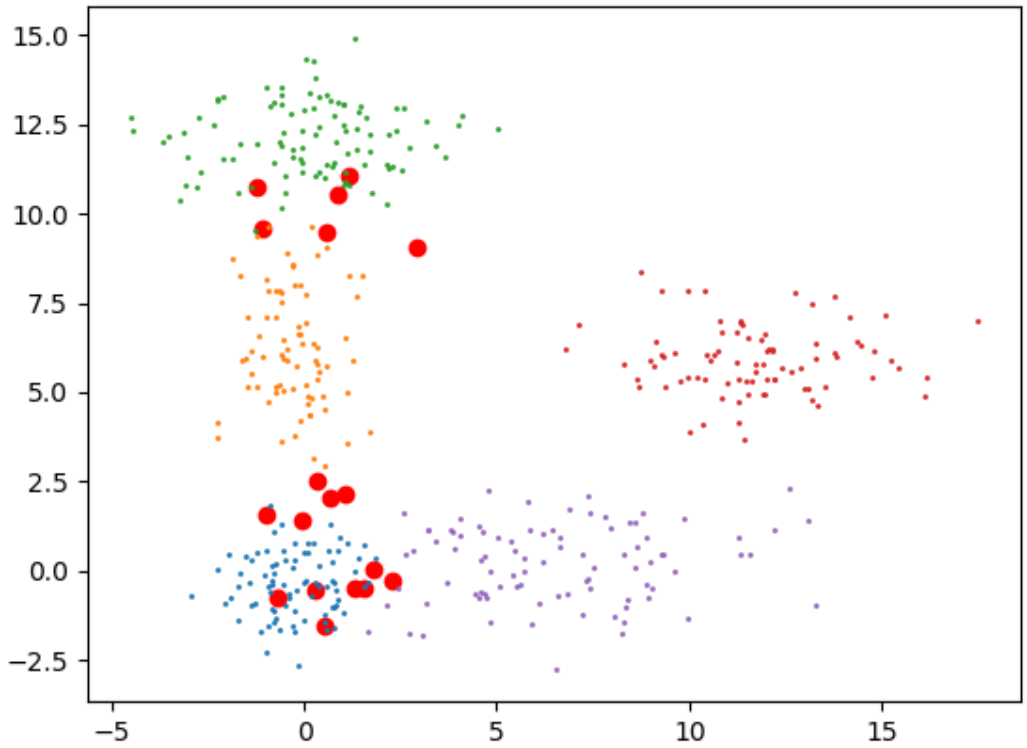
https://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/

RBF Neural Network:

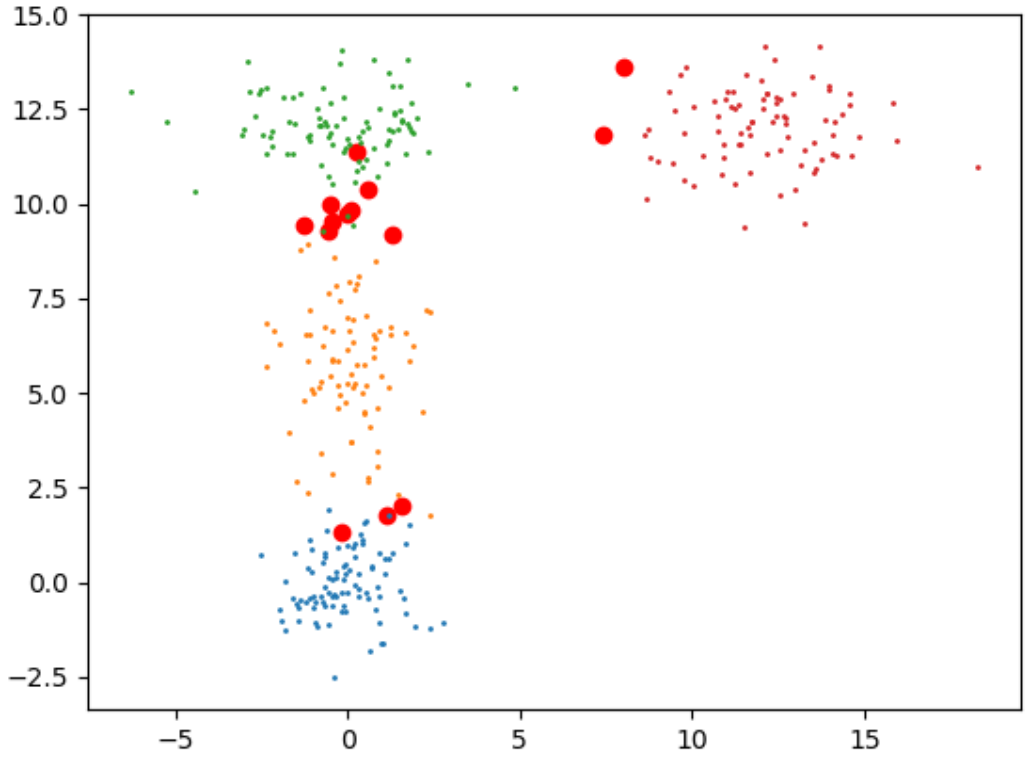
We have run an RBF with optimum number of clusters again, this time in order to draw a scatter plot in which wrong classified points are demonstrated by a distinctive red circle of a greater radius and every class is assigned a unique distinctive color other than red (or it's simply red having a lesser radius).

We have drawn the figures with the optimum number of clusters as mentioned before:

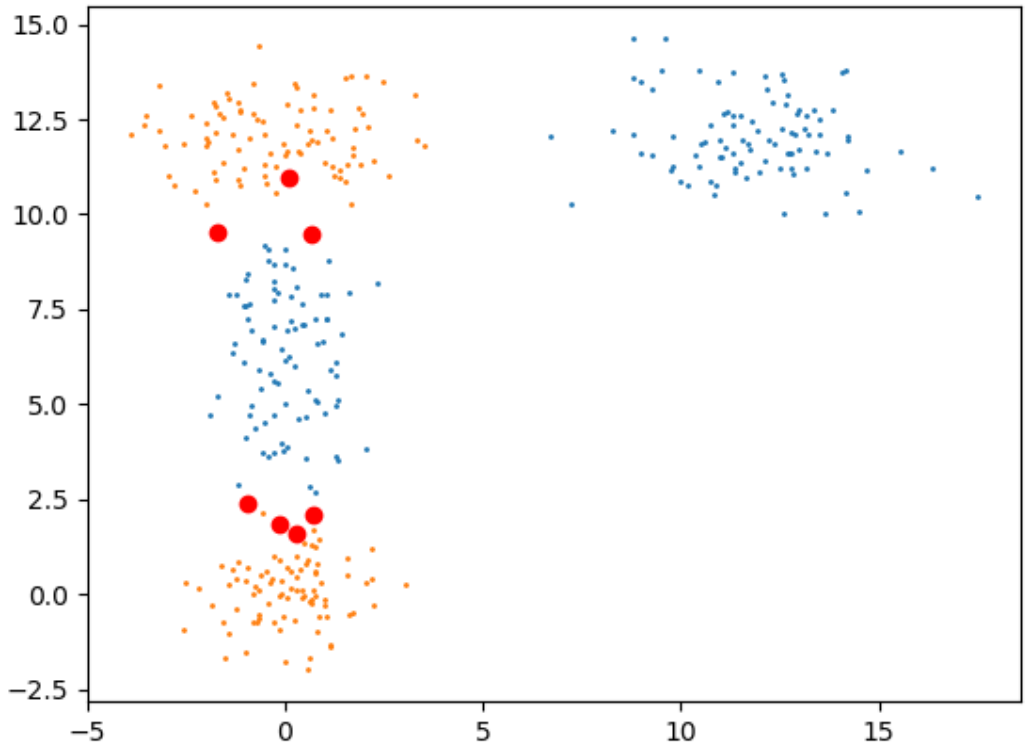
For the '5clstrain1500.csv' with number of clusters = 27:



For the ‘4clstrain1200.csv’ with number of clusters = 25:



For the ‘2clstrain1200.csv’ with number of clusters = 7:


15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5
0
5
10
15