Kernel Methods for Pattern Analysis
Assignment Solution 3

Rachit Garg CS14B050 | Keerthana S CS13B041 | Sphoorti K CS13B042

April 28, 2017

1 ν SVR

1.1 Univariate dataset

Set of hundred points were randomly chosen from 0 to 1.

Parameter estimation

Appropriate kernel was chosen which was considerably white because all the points have to be close together for better regression fit. The kernel chosen was of $\sigma = 1$. Now parameters C and ν were chosen based on minimum validation error, which came out to be C = 500, and $\nu = 0.3$.

Plots

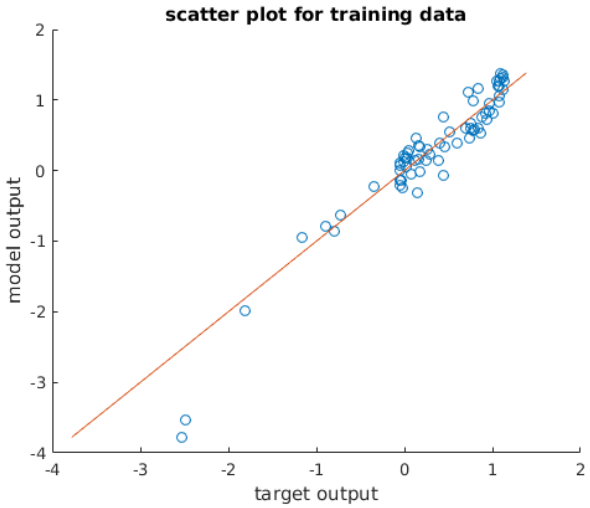

Figure 1: Target output vs model output for Train data

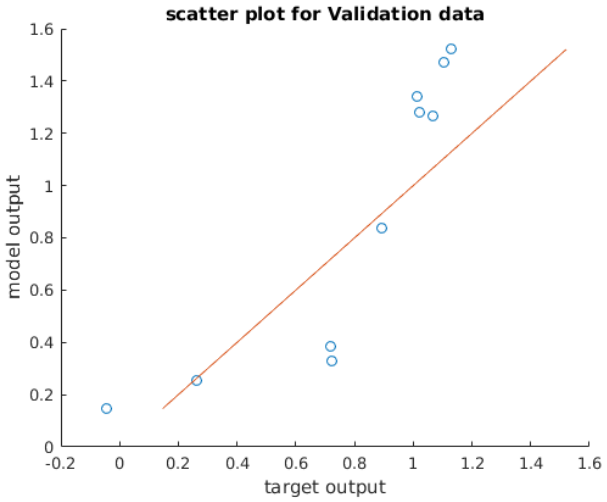

Figure 2: Target output vs model output for Validation data

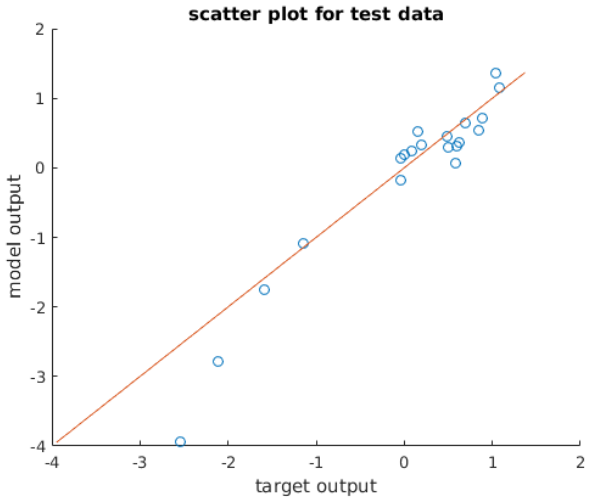

Figure 3: Target output vs model output for Test data

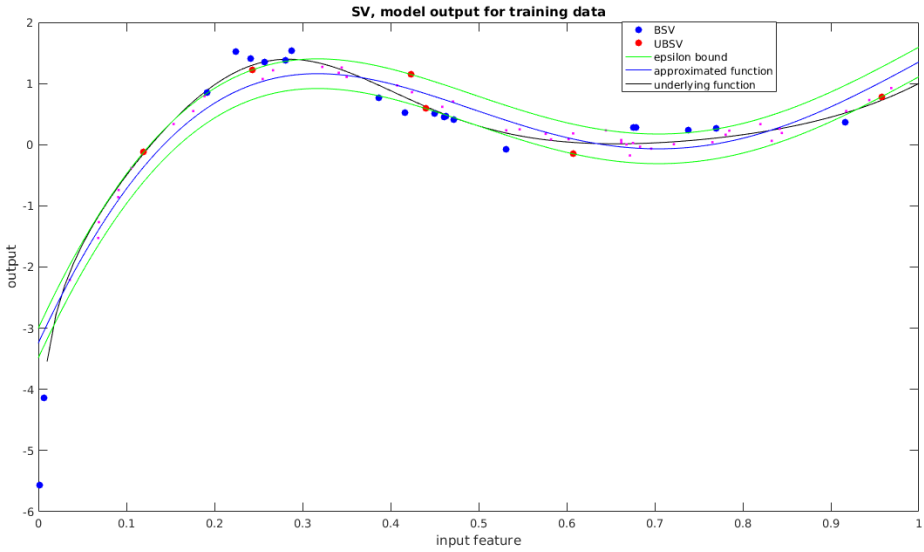

Figure 4: Underlying function, $\epsilon-$tube, target output and approximated function, bounded and unbounded support vectors for univariate dataset

Comparison with other models

For standard deviation in data of 0.15, the mean square error for SVR was around 0.05 which is slightly higher than polynomial fit. This could be because we are comparing mean square error, where linear model regression is built minimizing least square error itself while SVR is built with a different objective function where any error less than ϵ is considered a zero error.

Mean square error of SVR is better than MLFFNN. But the complexity of MLFFNN that achieves this error is too high, around 500 nodes. Which makes SVR much better model compared to MLFFNN.

1.2 Bivariate dataset

Parameter estimation

A suitable kernel for which the points are close enough(ie, the kernel gram matrix is considerably white) was chosen. For which trade off parameter and ν were chosen minimizing cross validation error. For which $C = 60$ and $\nu = 0.2$.

Plots

Figure 5: Target output vs model output for Train data

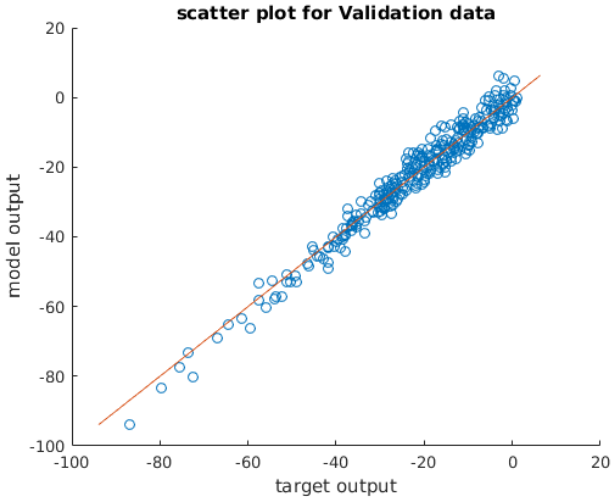

Figure 6: Target output vs model output for Validation data

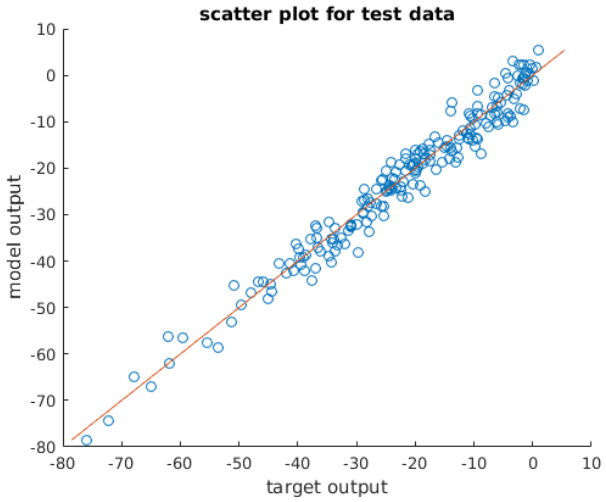

Figure 7: Target output vs model output for Test data

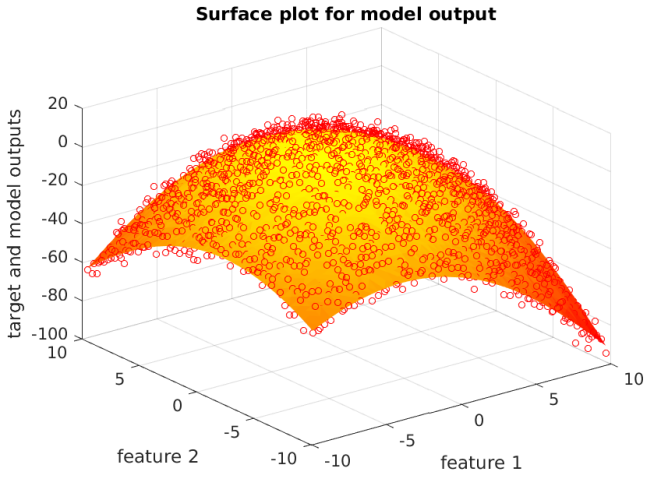

Figure 8: Target output and model output for Train data

Comparison with other models

SVR mean squared error is around 10, which is comparable to MLFFNN and RBF. All the three model have mean square error in the same range. But the error is high compared to linear model for regression.

2 ν -SVDD

2.1 Overlapping dataset

SVDD constructs a enclosing hypersphere for normal points in the kernel space.Since gaussian kernel is normalised kernel,we see that it becomes a hyperplane in gaussian kernel space.

Parameter estimation:

The hyperparameters are σ(width parameter of gaussian kernel) and $\nu.\sigma$ is chosen based on kernel gram matrix ,which is 33 in this case.Since we have only normal points in train data,we expect it to be white because the similarity of points within class should be high.

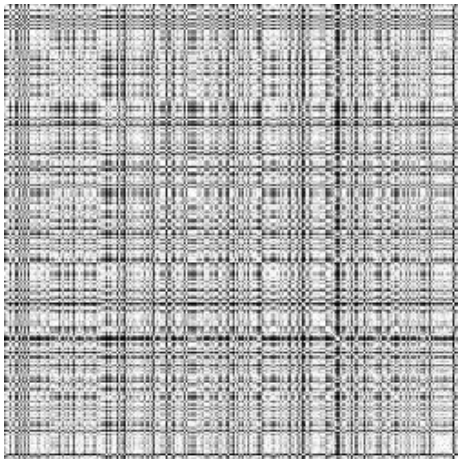

Figure 9: Figure showing kernel gram matrix for normal points

We choose ν by cross validation.The accuracies on validation set are :

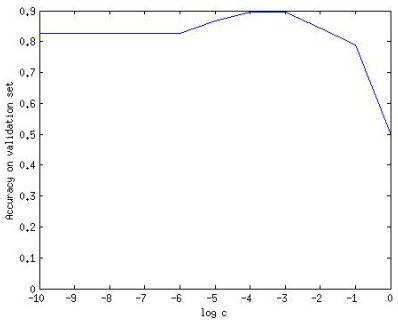

Figure 10: Plot for accuracies on validation set vs $\log(\nu)$

The best value of nu is obtained to be 0.0625.

Decision region plot:
The decision region plot for SVDD using gaussian kernel is:

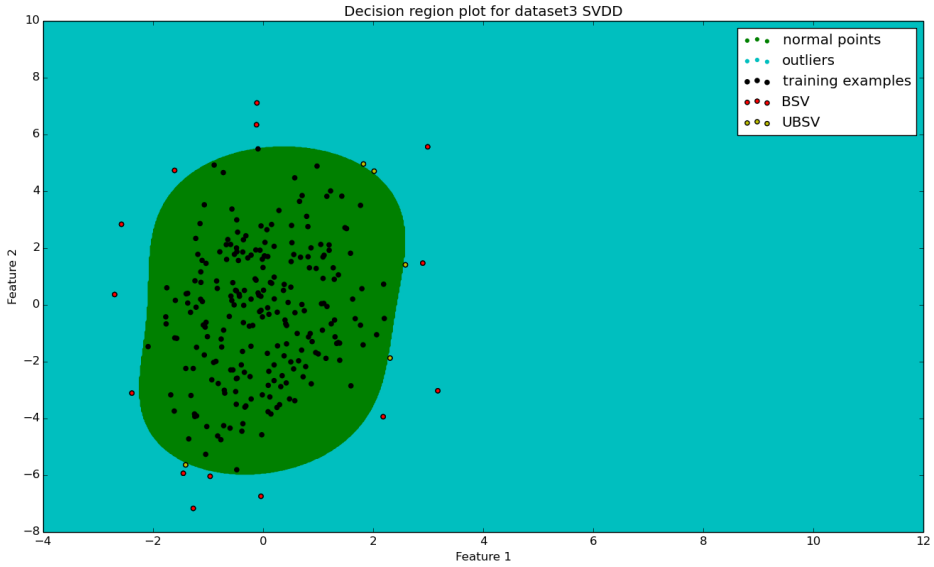

Figure 11: Figure showing decision region plot for overlapping dataset using SVDD

Since the boundary is hyperplane in gaussian kernel space ,we expect a non linear surface which separates normal and abnormal points in x space.The unbounded support vectors are those on the surface where as bounded support vectors are those outside the surface enclosing normal points.

Confusion matrix

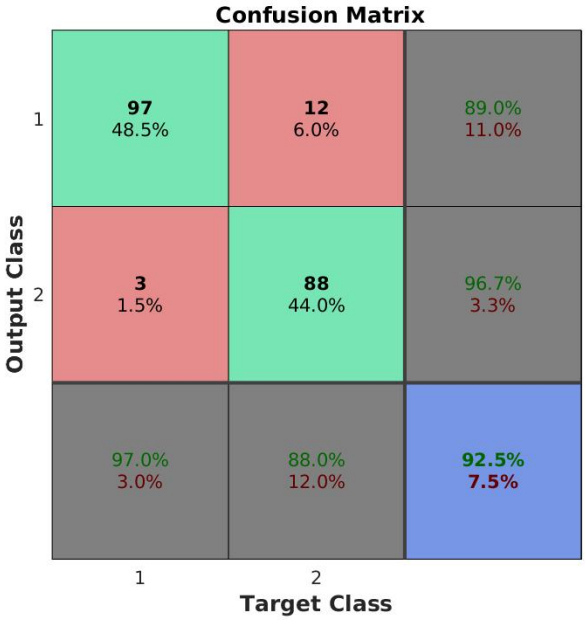

Figure 12: Figure showing confusion matrix for test set for overlapping dataset ν -SVDD ,the first column corresponds to normal points and the 2nd to abnormal points

The accuracy is observed to be 92.5%.The reason for false positives and false negatives is that it is linearly non-separable.
The percentage of true positives and false positives is 97% and 3% respectively.

2.2 Multivariate dataset

The data set is housing dataset which has 13 attributes.
Parameter estimation:
The best σ obtained is 200.

The plot for validation accuracy is:

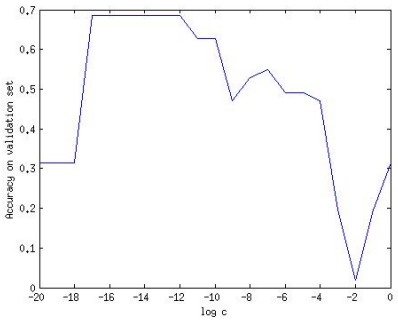

Figure 13: Plot for accuracies on validation set vs log(ν)

The best ν is observed to be 7.6294e-06.
ν is a tradeoff parameter between radius of hypersphere and number of support vectors.It is chosen such that fraction of false positives and false negatives is less.

Confusion matrix The confusion matrix for multivariate data set using ν SVDD is:

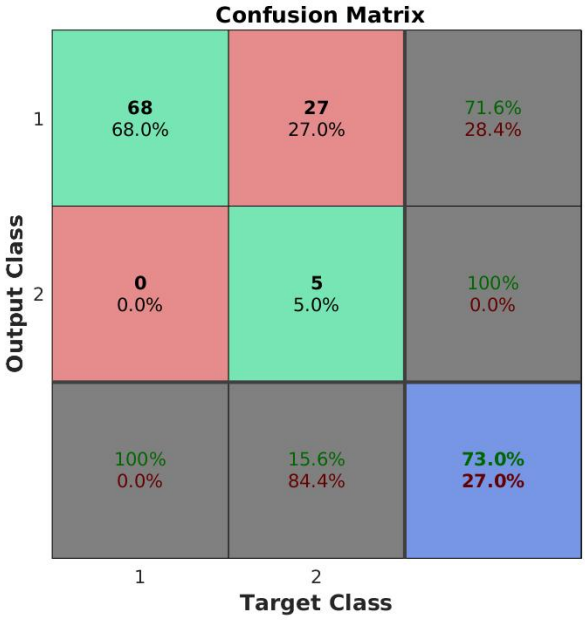

Figure 14: Figure showing confusion matrix for test set for overlapping dataset ν -SVDD ,the first row and column corresponds to normal points and the 2nd to abnormal points

The accuracy for test data is observed to be 72%.
The percentage of true positives and false positives are 66.2% and 33.8% respectively.

3 Clustering

3.1 Non - Linearly Separable data set

Normal K means

Parameter estimation: The parameter varied here was the number of clusters. Cluster purity was used as an evaluation measure. Cluster purity is the percent of the total number of data points that were classified correctly, in the unit range [0..1].

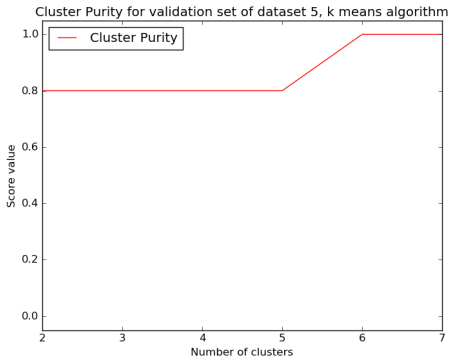

Figure 15: Plot for validation set cluster purity vs number of clusters

We observe that the best cluster purity for this algorithm is observed when number of clusters are any number greater than equal to 6. But cluster purity is only an evaluation measure for number of points that are classified correctly. It can be very high if we have a lot of clusters.
Hence we showcase two results here. One for when the number of clusters is same as the number of classes and second when the cluster purity is the best. We show that if we pick the number of clusters as same as the number of classes then the k means algorithm does not cluster well as compared to the kernel k means which does better.
Decision region plot:
The decision region plot after different iterations looks like:

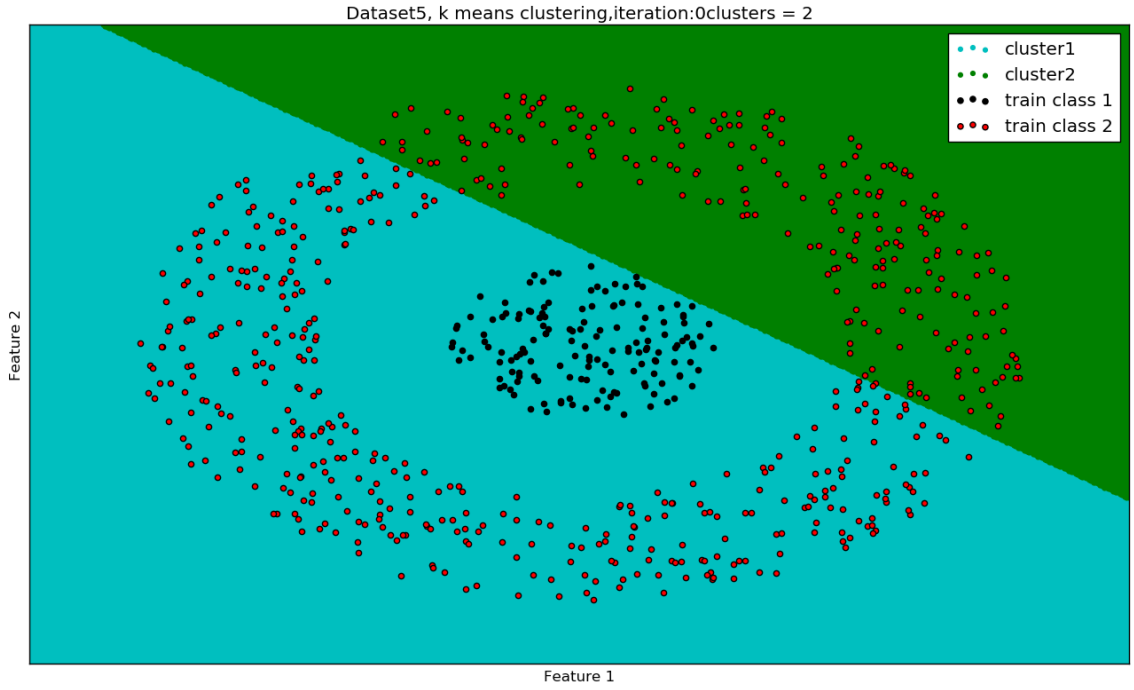

Figure 16: Figure showing decision region plot for dataset 5 using k means after initialization,cluster number = 2

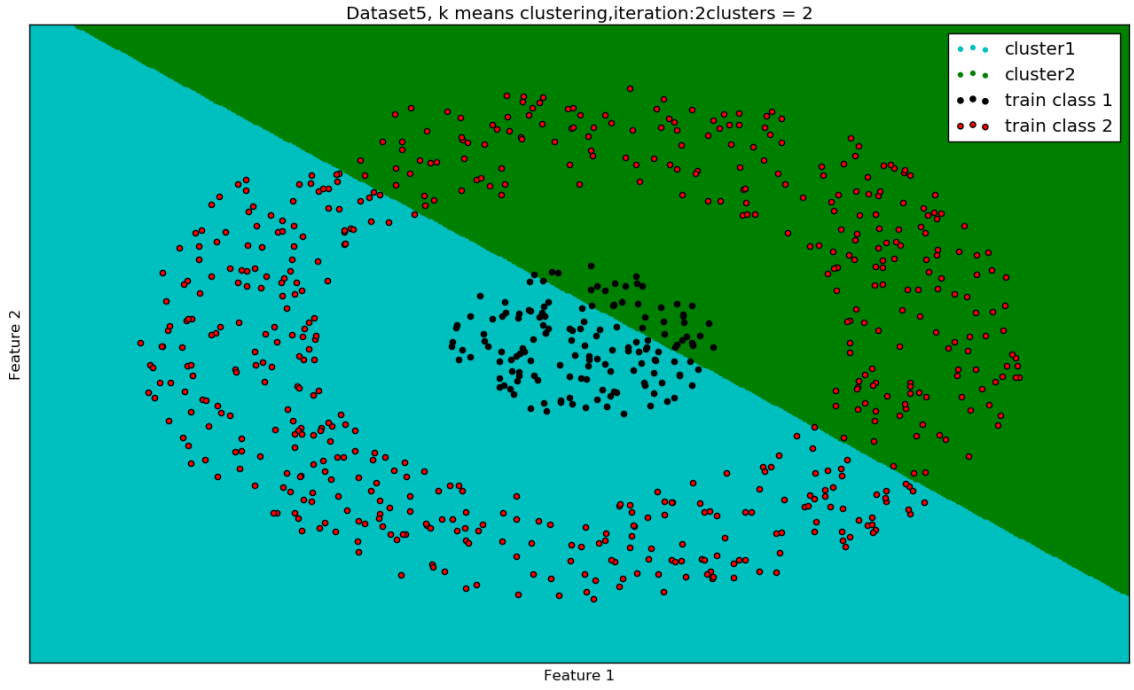

Figure 17: Figure showing decision region plot for dataset 5 using k means after second iteration,cluster number = 2

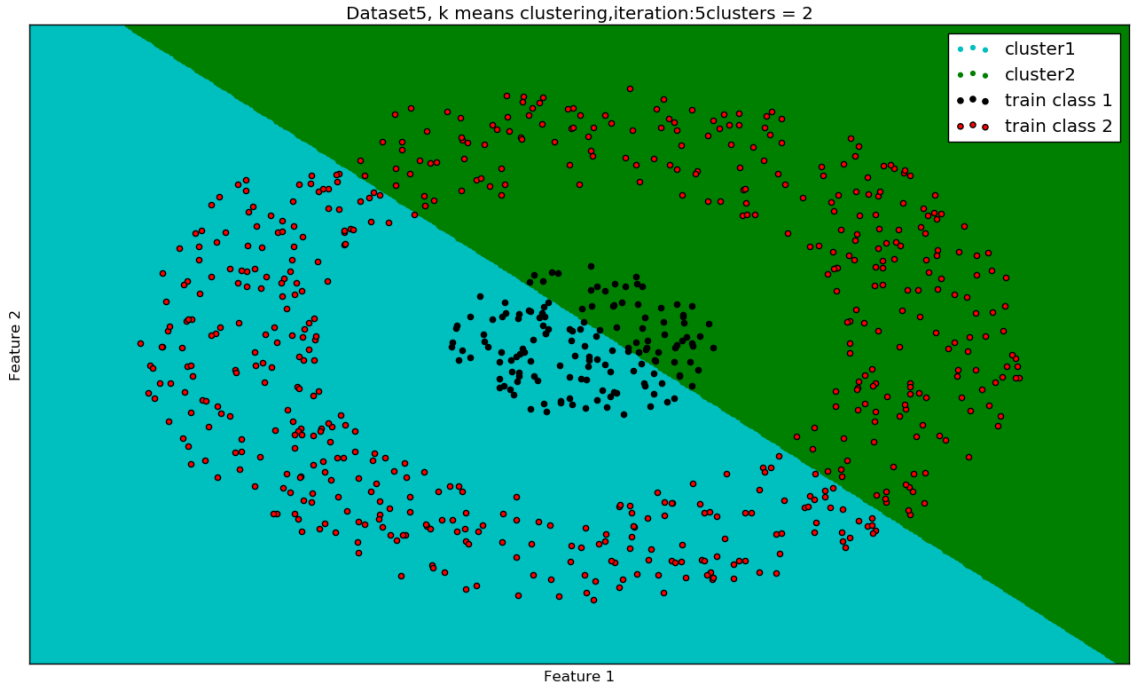

Figure 18: Figure showing decision region plot for dataset 5 using k means after fifth,cluster number = 2

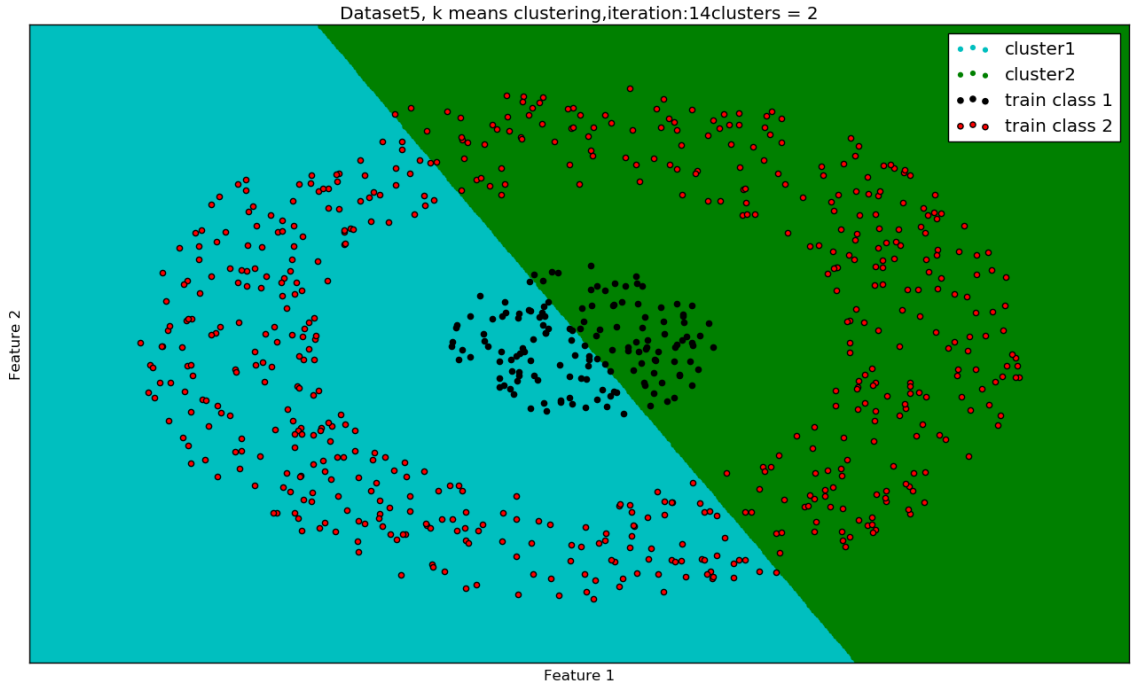

Figure 19: Figure showing decision region plot for dataset 5 using k means after convergence,cluster number = 2

We note that the decision region boundaries are linear and that the datasets are converging from the initial iteration to a point where the classes are split in the two clusters.

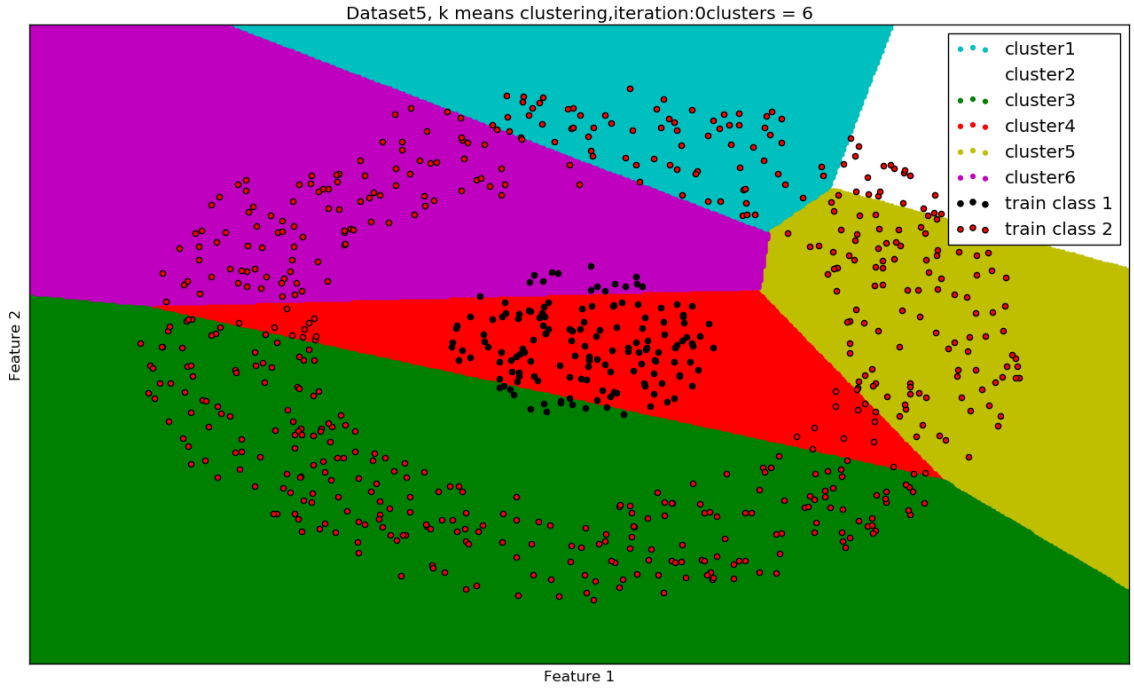

Figure 20: Figure showing decision region plot for dataset 5 using k means after initialization,cluster number = 6

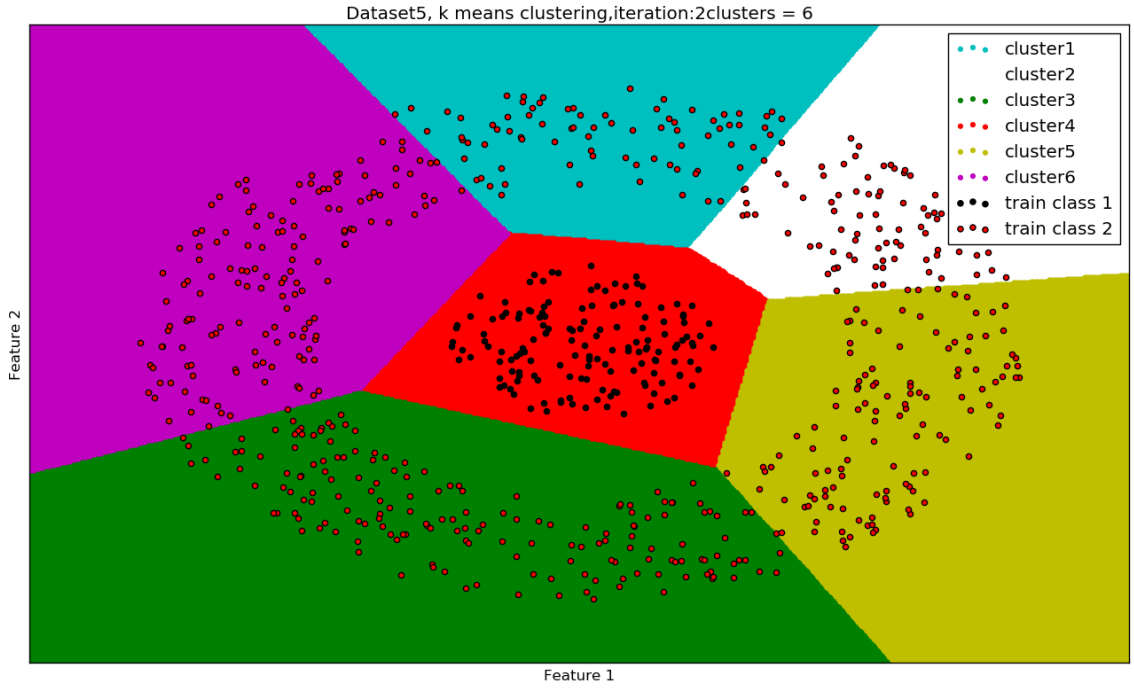

Figure 21: Figure showing decision region plot for dataset 5 using k means after second iteration,cluster number = 6

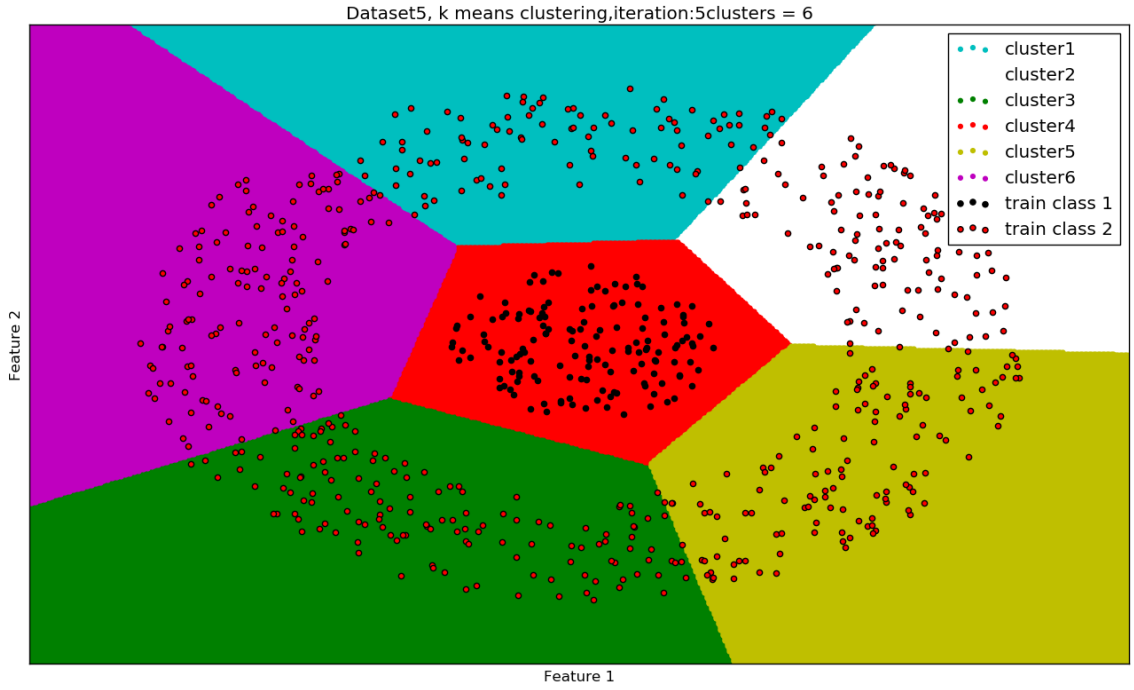

Figure 22: Figure showing decision region plot for dataset 5 using k means after fifth,cluster number = 6

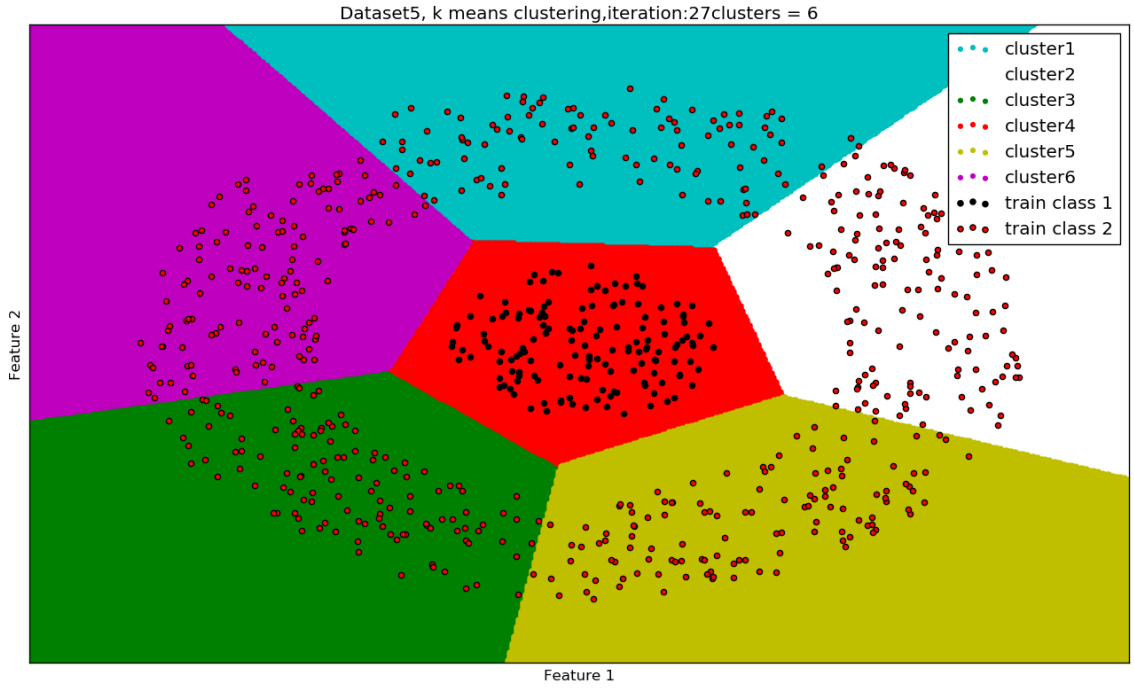

Figure 23: Figure showing decision region plot for dataset 5 using k means after convergence,cluster number = 6

We note that the decision region boundaries are linear, and that with the increase in iterations, more and more points in the outside circle are being equally divided among the 5 clusters.
From the figures we can see the difference in clustering when the number of clusters is more and when the number of clusters is less. When number of clusters is less we get a very poor clustering where the classes are split in both the clusters. When the number of clusters is more we can eventually cluster the points in the middle as separate and the dataset in middle is clustered

correctly. The dataset surrounding it is divided into multiple clusters because the points in the bigger radius circle are too far away for all to come under one cluster.
We note that normal k means clustering cannot separate this dataset as the clustering for the bigger circle has points very far away in x-space. Hence these points won't come under one cluster using normal k-means. Thus we try to look at kernel k - means for the same dataset.

Kernel K means: Gaussian Kernel

Parameter estimation:
The kernel parameter to be estimated for gaussian kernel is σ where kernel function in gaussian kernel is given by:
$K(x_m, x_n) = e^{-||x_m - x_n||^2/\sigma}$ where σ is width parameter.
The σ value is chosen based on kernel gram matrix and the value obtained is σ=0.18.

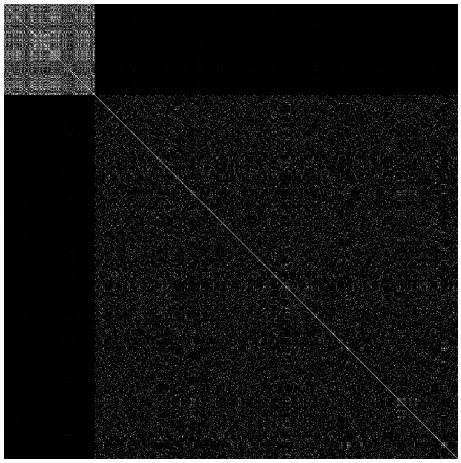

Figure 24: Grey level image of kernel gram matrix for dataset 5 training set using Gaussian kernel

Values in kernel gram matrices are values of normalised kernel function for all pair of training examples which is measure of similarity of points.In the above figure we see that the region near diagonal has higher values since we expect examples of same class to be similar,on the other hand,region away from diagonal is grey/black since they belong to different class.
We observe that the clustering is perfect in the case for number of clusters as two. It is able to identify each class as separate, hence we don't continue with choosing number of clusters as the hyper parameter.

Decision region plot:
The decision region plot for different iterations looks like:

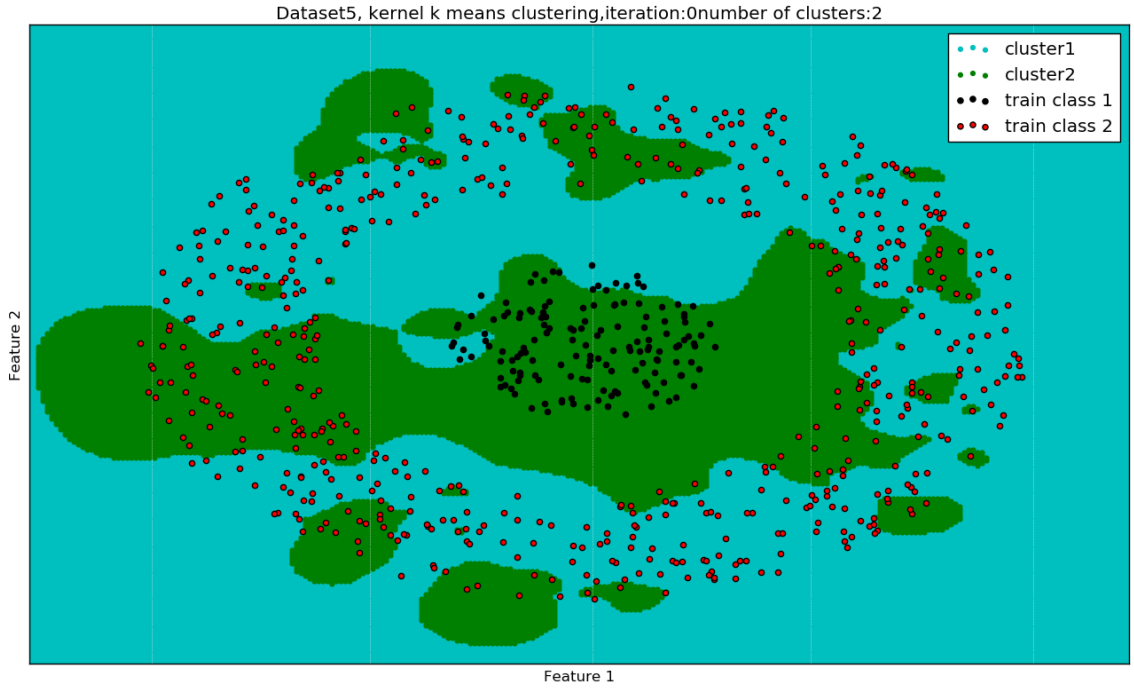

Figure 25: Figure showing decision region plot for dataset 5 using kernel k means after initialization,cluster number = 2

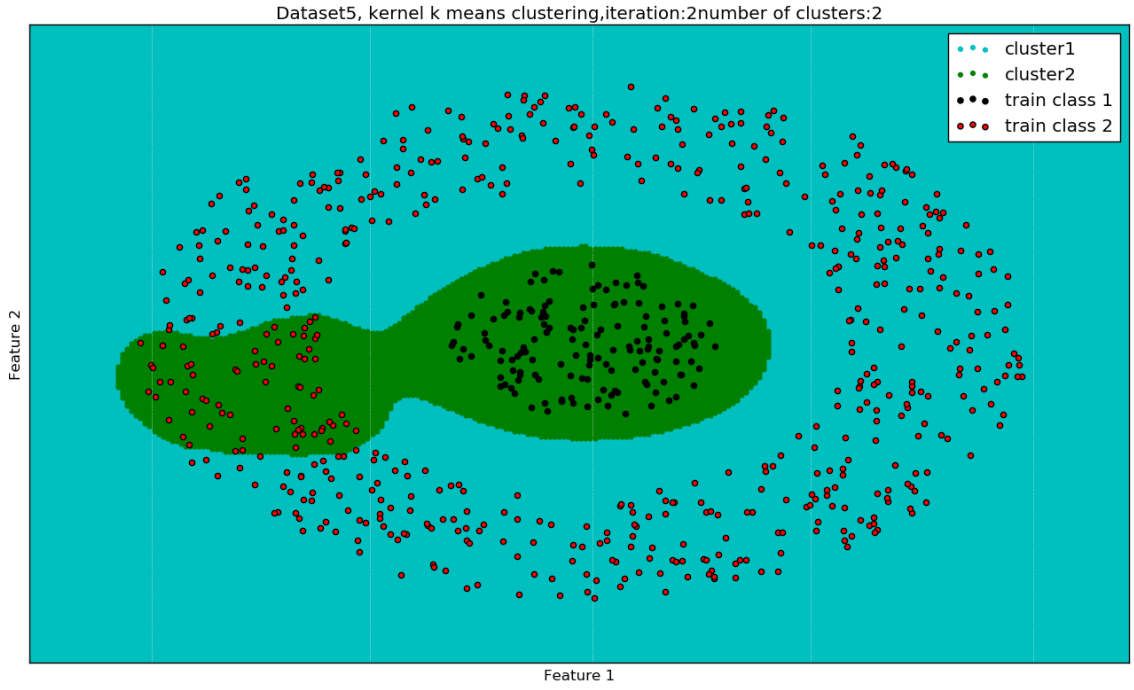

Figure 26: Figure showing decision region plot for dataset 5 using kernel k means after second iteration,cluster number = 2

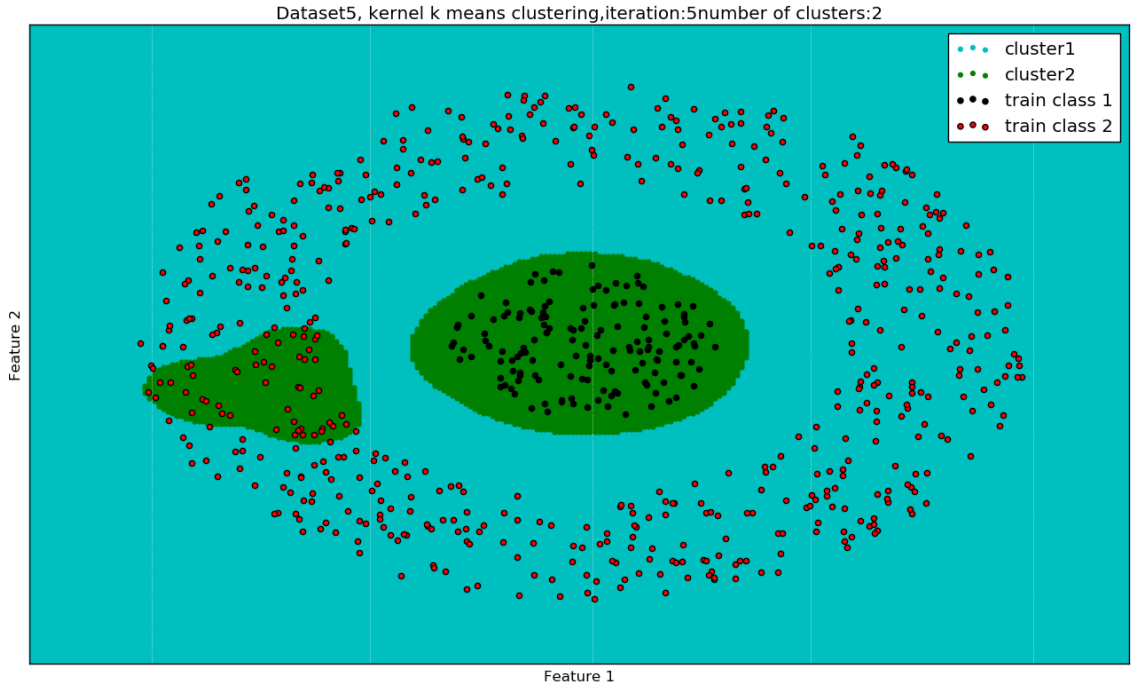

Figure 27: Figure showing decision region plot for dataset 5 using kernel k means after fifth,cluster number = 2

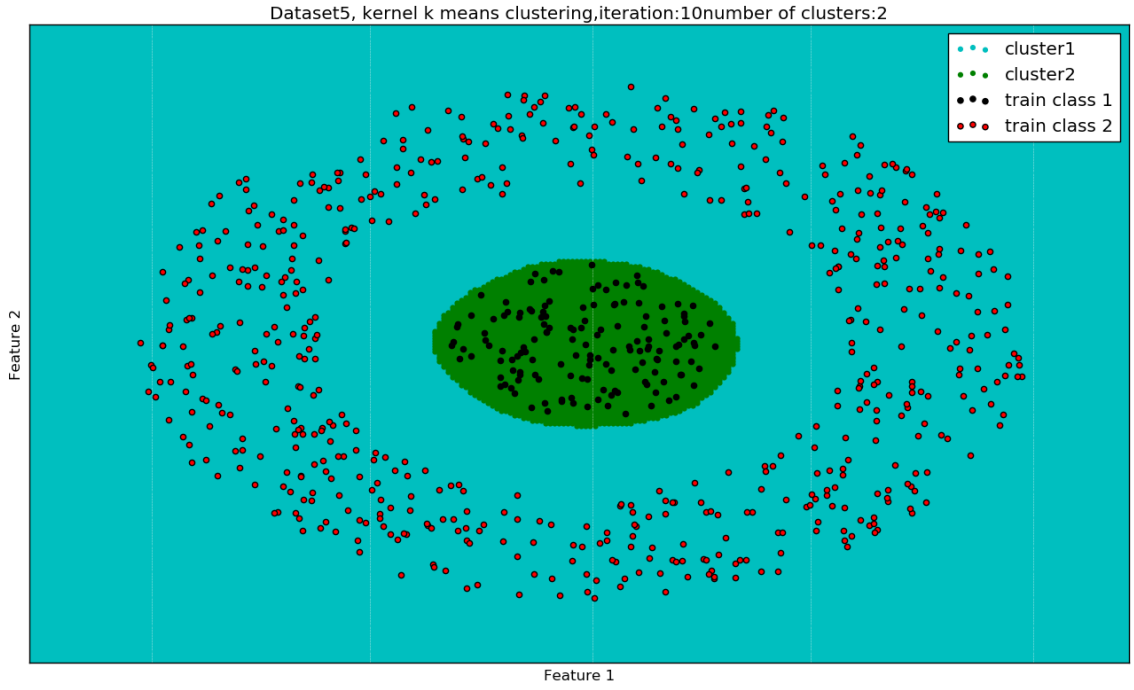

Figure 28: Figure showing decision region plot for dataset 5 using kernel k means after convergence,cluster number = 2

Kernel based soft clustering: Gaussian Kernel

Parameter estimation: Kernel gram matrix is used to choose kernel parameters. The same kernel is used as in previous part. We show the decision region plots now.
Decision region plot:
The decision region plot for various iterations looks like:

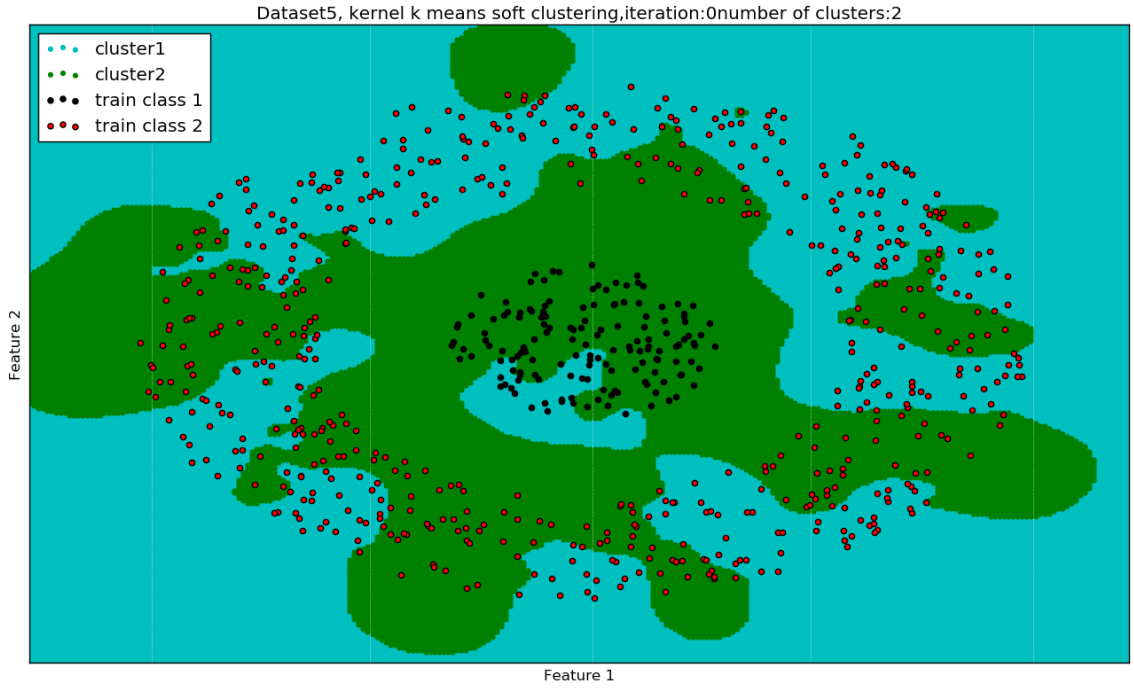

Figure 29: Figure showing decision region plot for dataset 5 using soft kernel k means after initialization,cluster number = 2

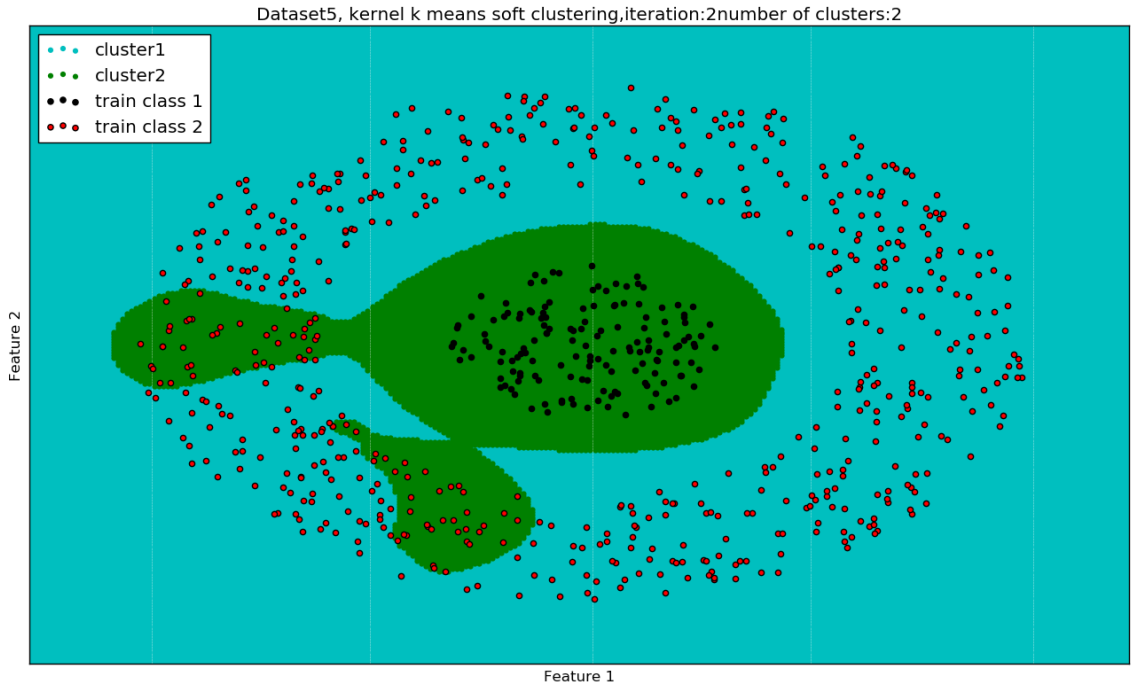

Figure 30: Figure showing decision region plot for dataset 5 using soft kernel k means after second iteration,cluster number = 2

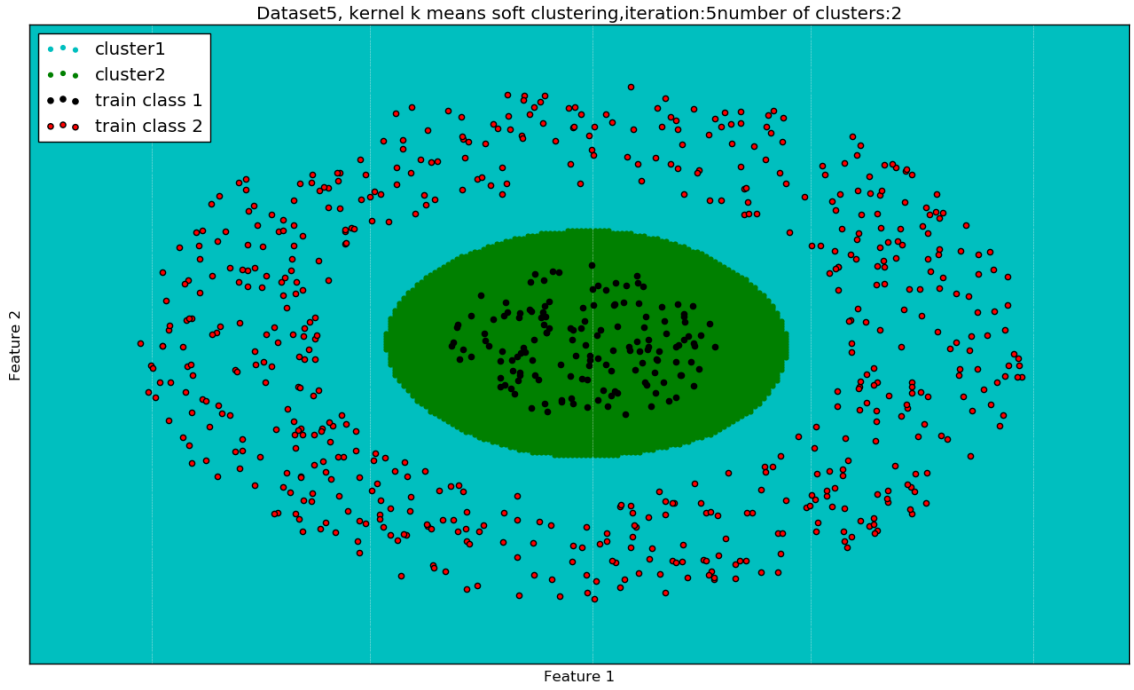

Figure 31: Figure showing decision region plot for dataset 5 using soft kernel k means after fifth,cluster number = 2

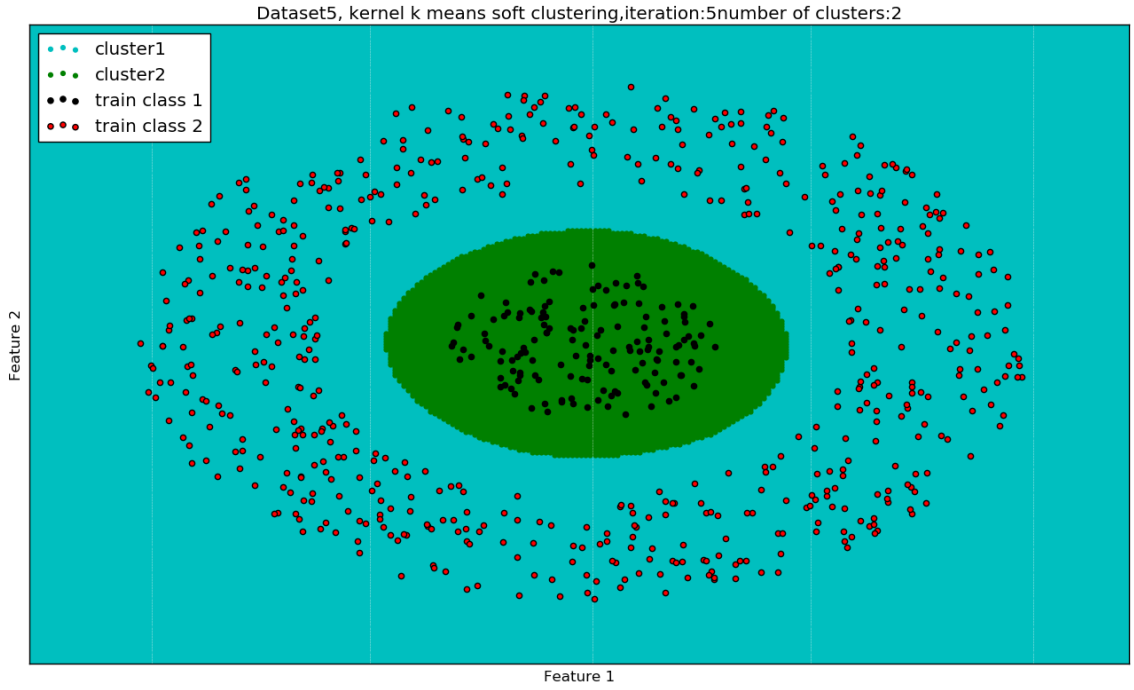

Figure 32: Figure showing decision region plot for dataset 5 using soft kernel k means after convergence,cluster number = 2

3.2 Overlapping data set

Normal K means

Parameter estimation: The number of clusters chosen here were on the basis of the class labels. As this dataset contains three classes, that was chosen as the number of clusters.
All the results show that this dataset is not ideal for clustering as the clustering algorithm doesn't

know about the labels of the class, hence it treats the overlapped data in one cluster and doesn't penalize. Here we show the plots for one initialization of the means.

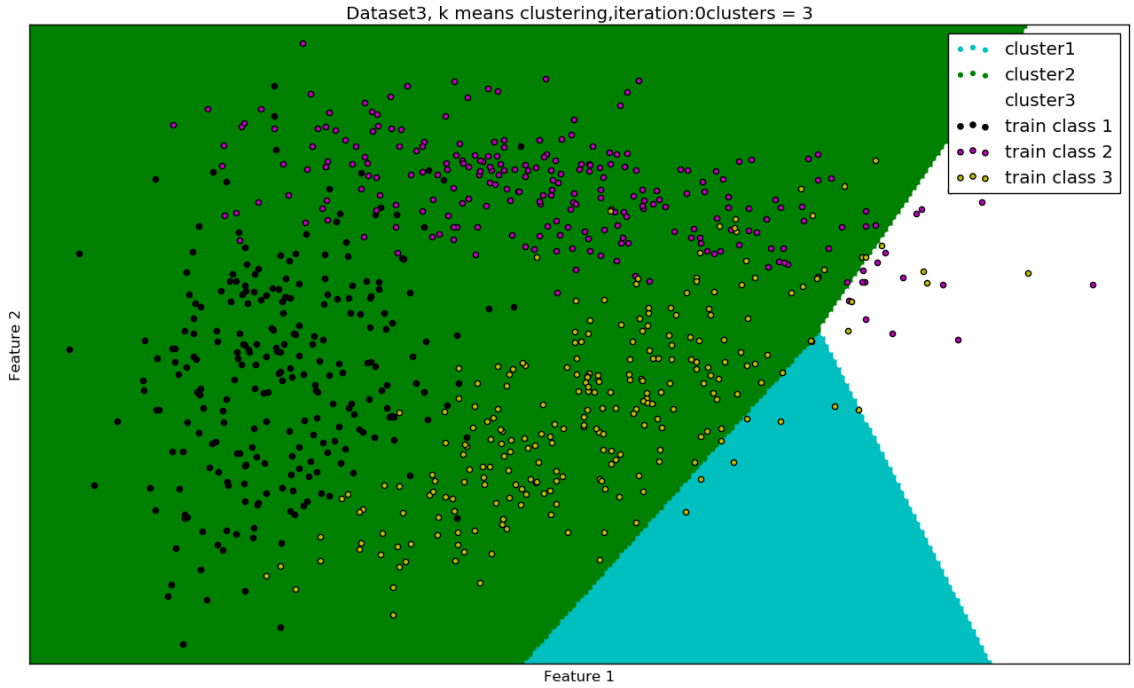

Figure 33: Figure showing decision region plot for dataset 3 using k means after initialization,cluster number = 3

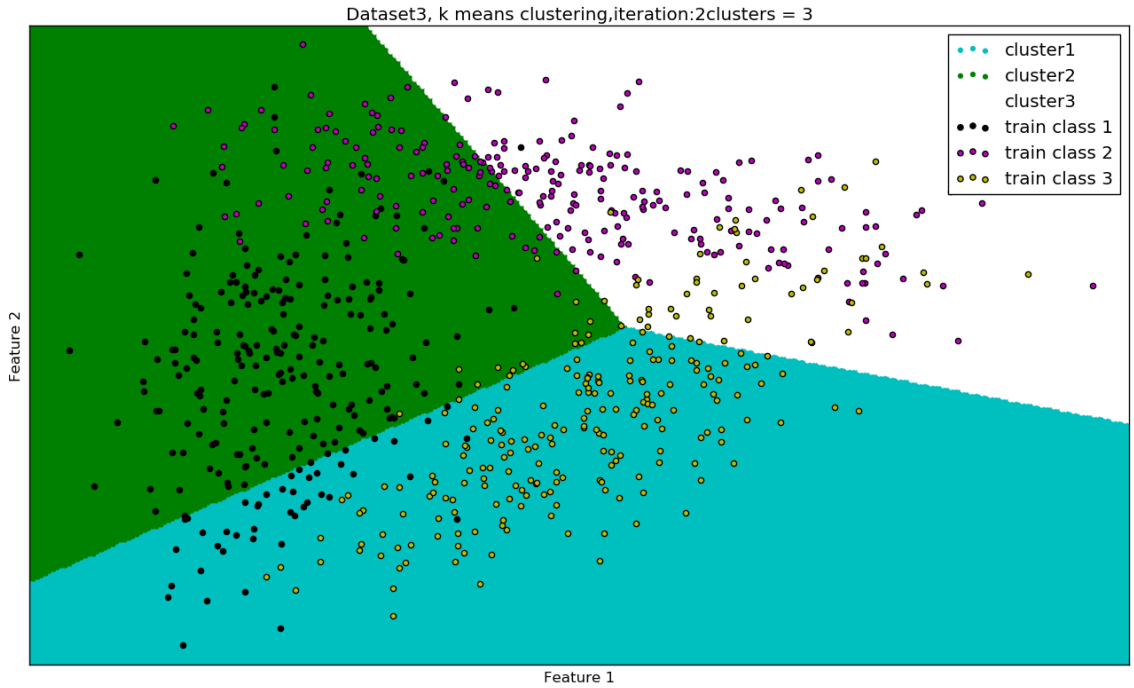

Figure 34: Figure showing decision region plot for dataset 3 using k means after second iteration,cluster number = 3

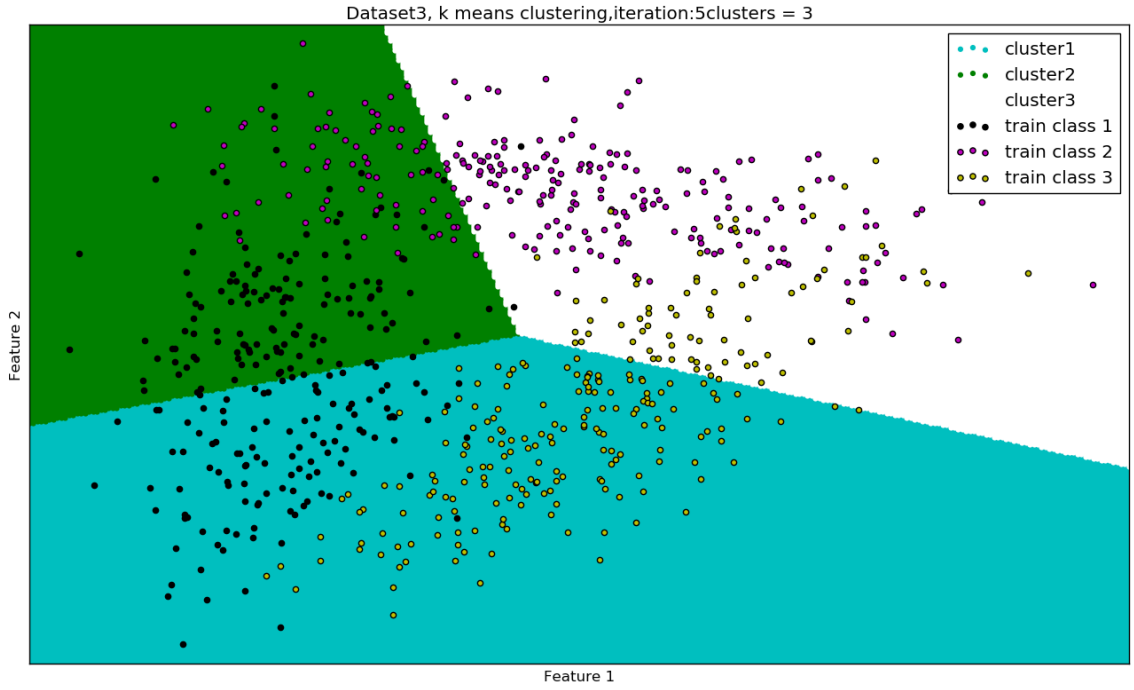

Figure 35: Figure showing decision region plot for dataset 3 using k means after fifth iteration,cluster number = 3

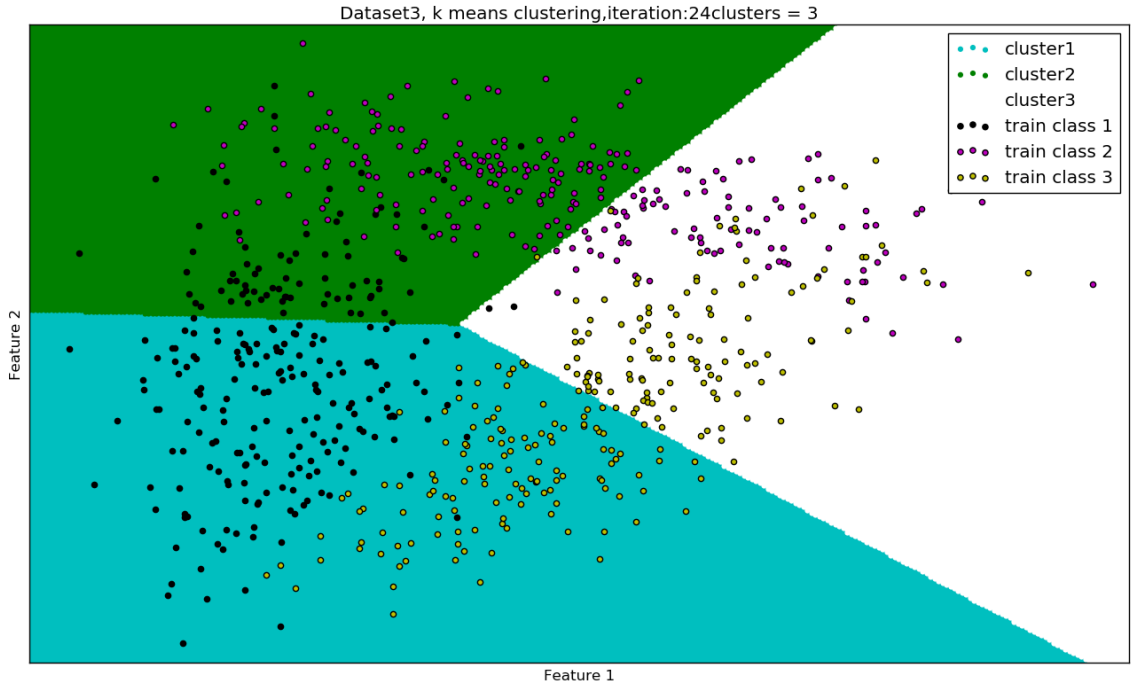

Figure 36: Figure showing decision region plot for dataset 3 using k means after convergence,cluster number = 3

We note that the decision region boundaries are linear and that the datasets are converging from the initial iteration to a point where the clusters seem to have an even amount of points. The clusters don't agree with the class labels as the classes are very close together and the clustering algorithm doesn't know the class labels. The output plots on this dataset depended a lot on the initialization of the centers. We just showcased here one of the plots.

Kernel K means: Gaussian Kernel

Parameter estimation:
The kernel parameter to be estimated for gaussian kernel is σ where kernel function in gaussian kernel is given by:
$K(x_m, x_n) = e^{-||x_m - x_n||^2/\sigma}$ where σ is width parameter.
The σ value is chosen based on kernel gram matrix and the value obtained is σ=5.

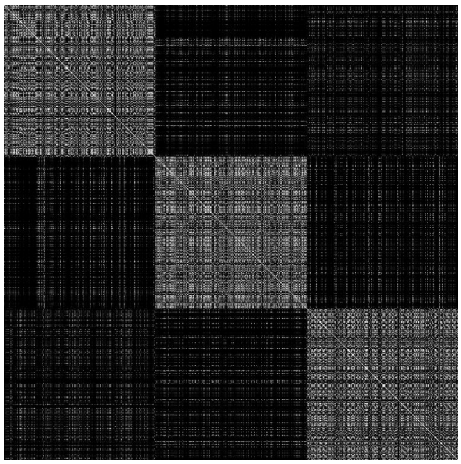

Figure 37: Grey level image of kernel gram matrix for dataset 3 training set using Gaussian kernel

Values in kernel gram matrices are values of normalised kernel function for all pair of training examples which is measure of similarity of points.In the above figure we see that the region near diagonal has higher values since we expect examples of same class to be similar,on the other hand,region away from diagonal is grey/black since they belong to different class.
We observe that the clustering even though is able to separate the classes as different clusters, it is not really able to notice the overlap in the classes. That seems a natural consequence of an unsupervised algorithm.
Decision region plot:
The decision region plot for different iterations looks like:

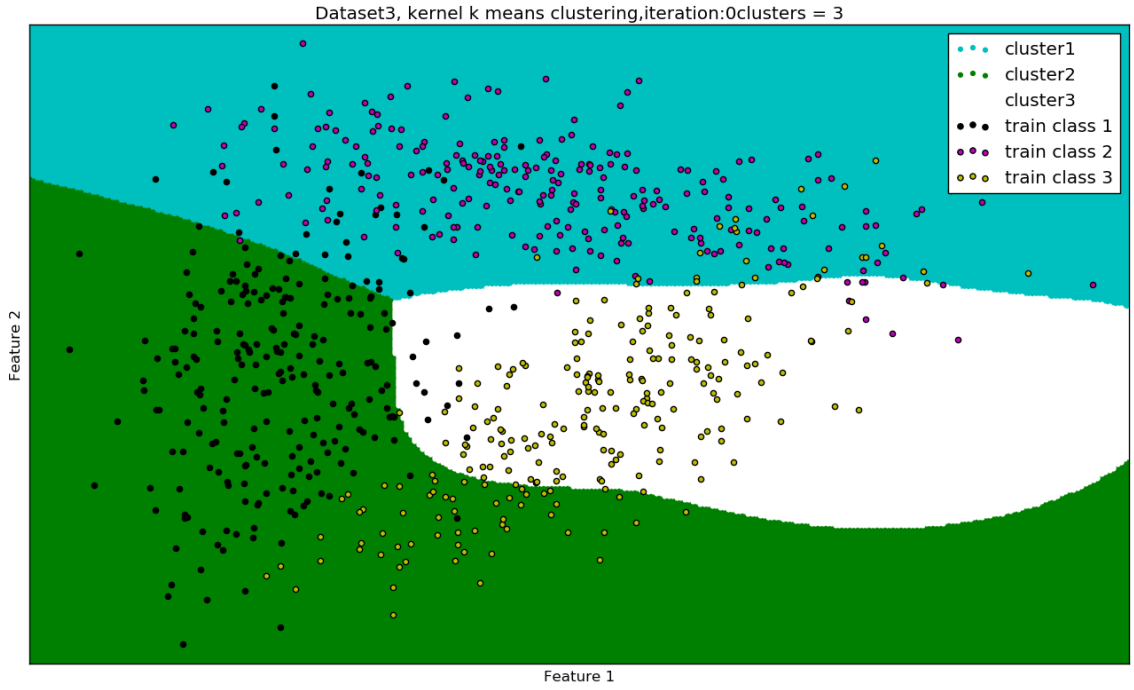

Figure 38: Figure showing decision region plot for dataset 3 using kernel k means after initialization,cluster number = 3

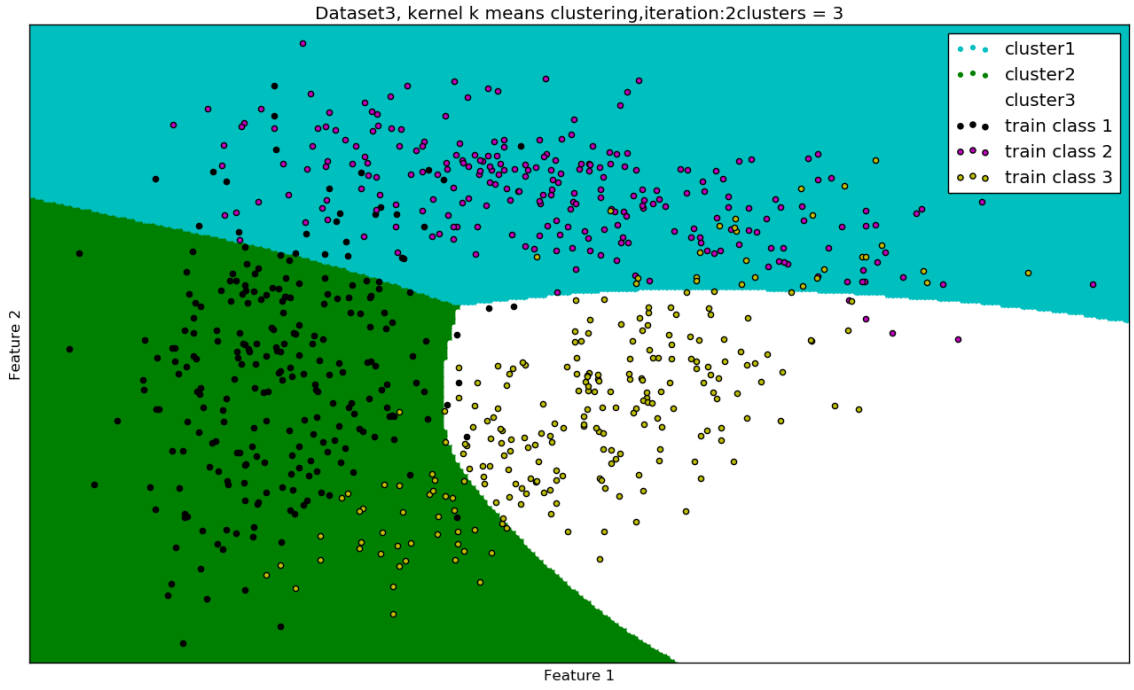

Figure 39: Figure showing decision region plot for dataset 3 using kernel k means after second iteration,cluster number = 3

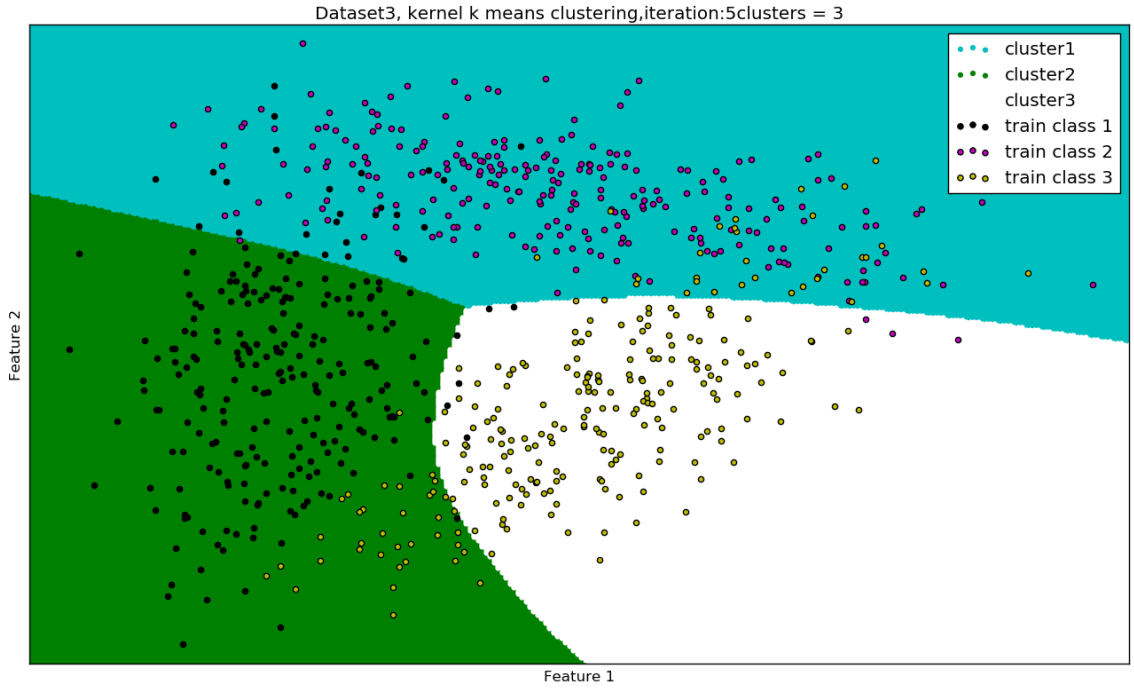

Figure 40: Figure showing decision region plot for dataset 3 using kernel k means after fifth,cluster number = 3

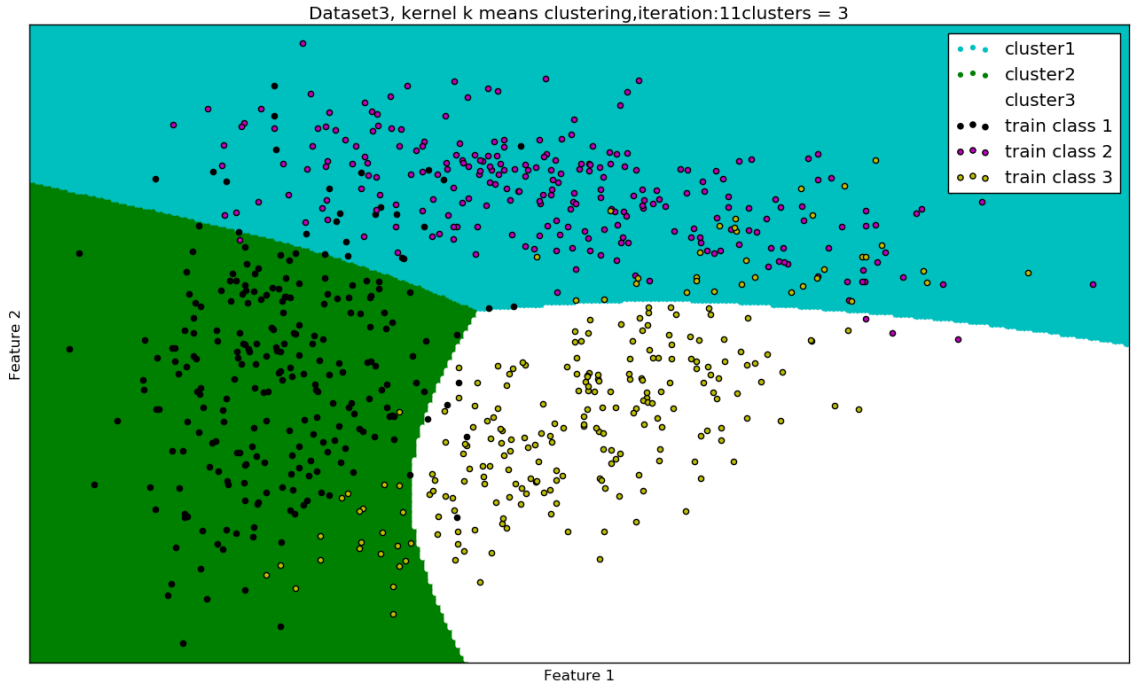

Figure 41: Figure showing decision region plot for dataset 3 using kernel k means after convergence,cluster number = 3

Kernel based soft clustering: Gaussian Kernel

Parameter estimation: Kernel gram matrix is used to choose kernel parameters. The same kernel is used as in previous part. We show the decision region plots now.
Decision region plot:
The decision region plot for various iterations looks like:

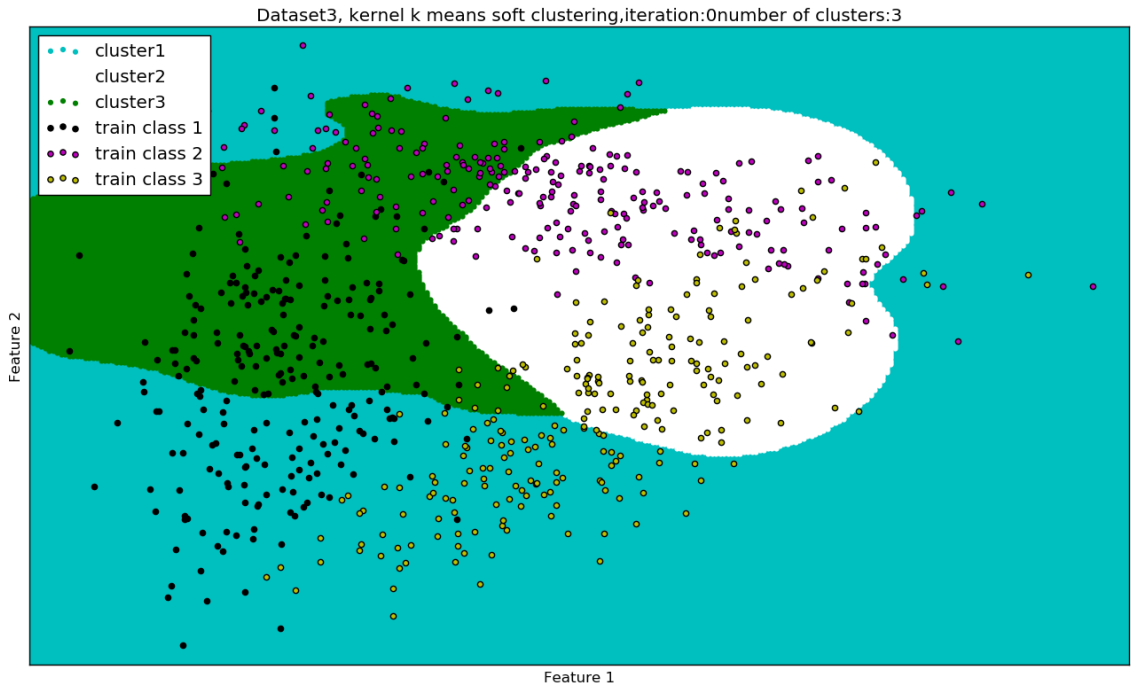

Figure 42: Figure showing decision region plot for dataset 3 using soft kernel k means after initialization,cluster number = 3

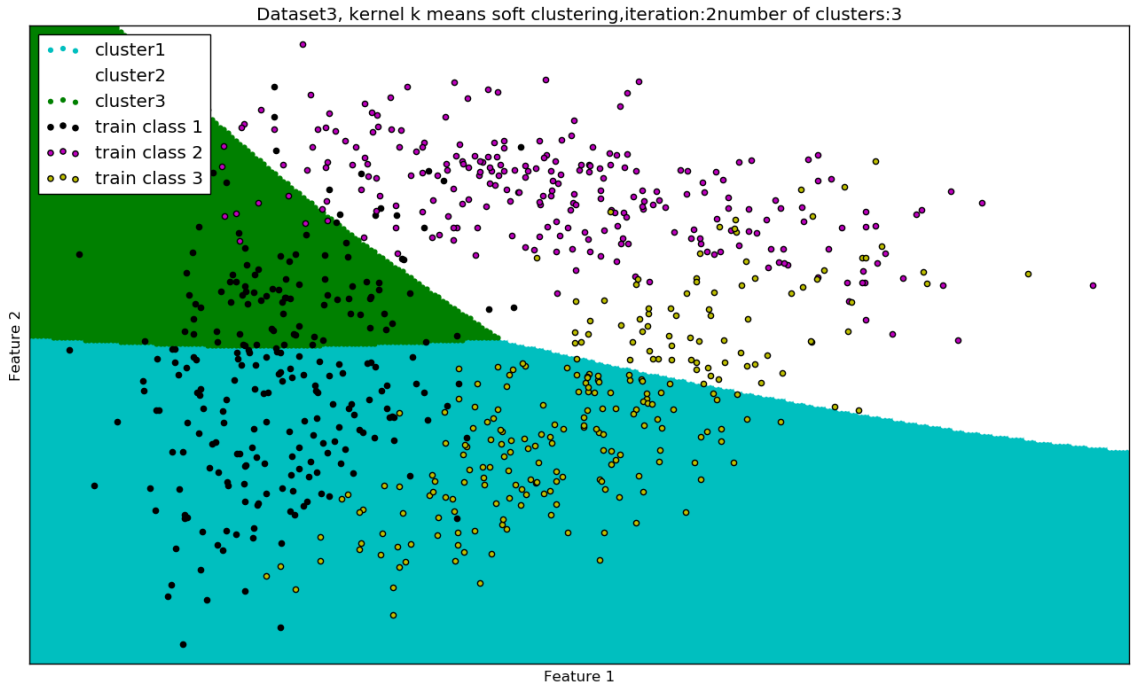

Figure 43: Figure showing decision region plot for dataset 3 using soft kernel k means after second iteration,cluster number = 3

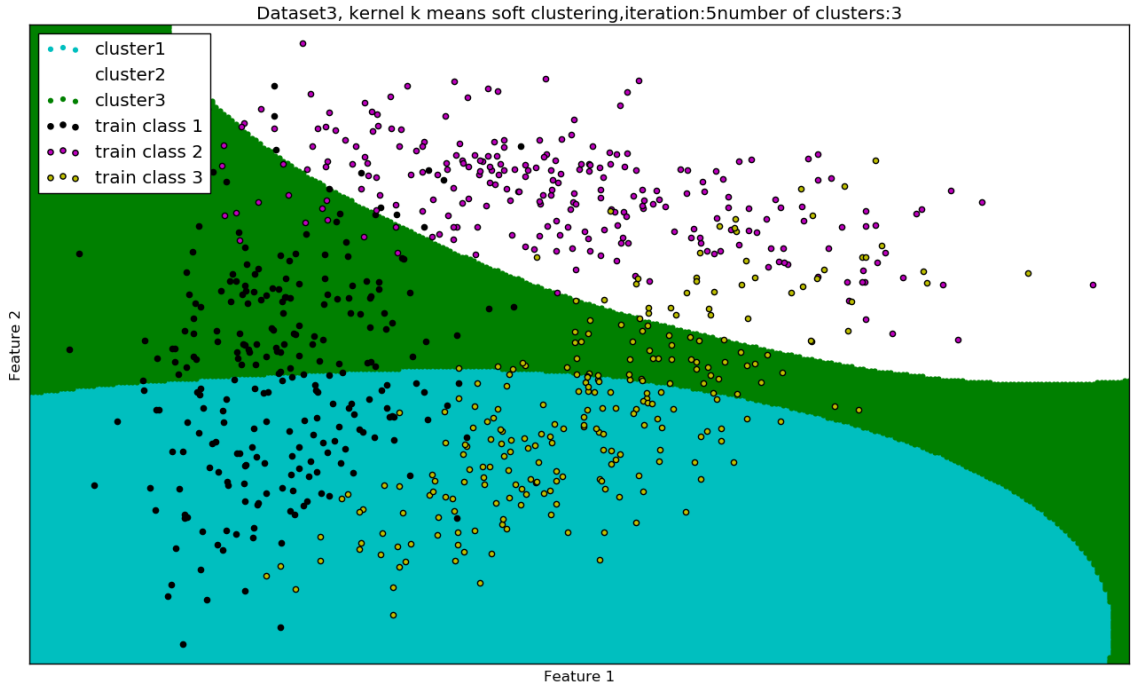

Figure 44: Figure showing decision region plot for dataset 3 using soft kernel k means after fifth,cluster number = 3

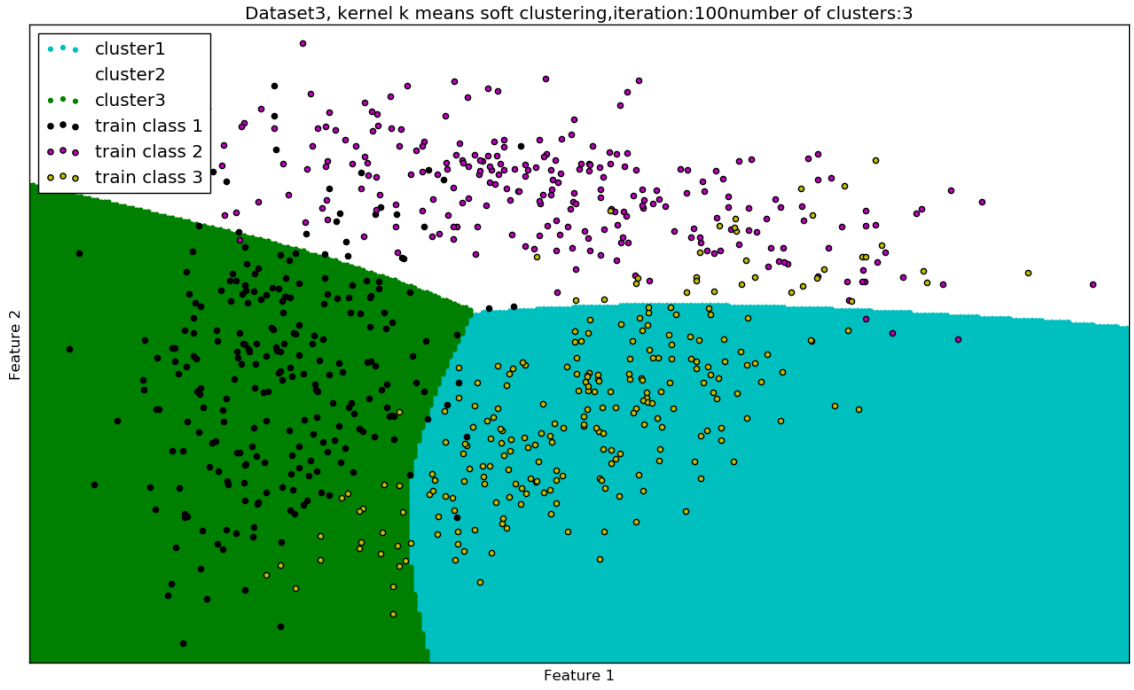

Figure 45: Figure showing decision region plot for dataset 3 using soft kernel k means after convergence,cluster number = 3

4 Dimension Reduction

4.1 PCA

Parameter selection

PCA was performed to get reduced dimension data. Using the projected data, Kernel gram matrix was plotted and best kernel was chosen which turned to be $\sigma = 0.5$. Classification using Gaussian kernel was performed using that kernel.

Eigen values Selection

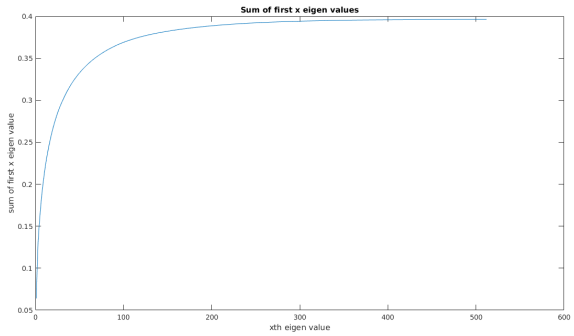

Figure 46: Cumulative eigen values

As we can see from the cumulative eigen values plot, magnitude of consecutive eigen values drop very quick. After 20% of eigen values there is almost no visible change in cumulative eigen values.

First 20% of the eigen values were chosen, and confusion matrix for test data was plotted.

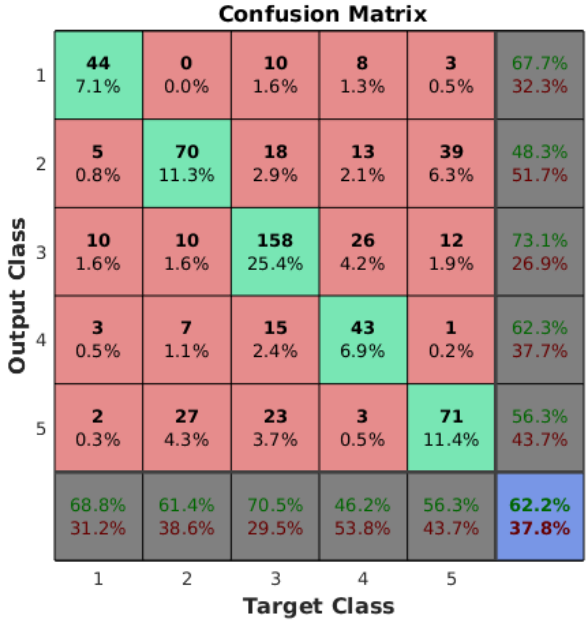

Figure 47: Confusion matrix for Test data

4.2 AANN

Parameter selection

Similar to other dimension reduction methods, for the projected data Kernel gram matrix was plotted which had best $\sigma = 0.03$. For which classification was performed.

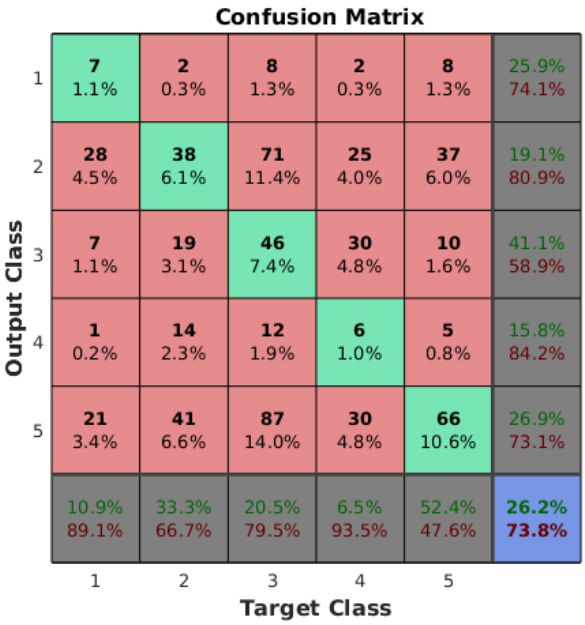

Figure 48: Confusion matrix for Test data

Only one auto-encoder layer was used with no of layers, with hidden layer nodes 450. We can see the performance is very low compared to both PCA and Kernel PCA.

4.3 KPCA

Parameter selection

From kernel gram matrix, a suitable kernel was chosen which turned out to be $\sigma = 0.5$. Using this sigma Kernel PCA was performed. Now after dimensionality reduction, kernel gram matrix was again plotted for which $\sigma = 2000$ turned out be the best. Using which classification was performed. All the analysis was done using Gaussian kernel.

Eigen values Selection

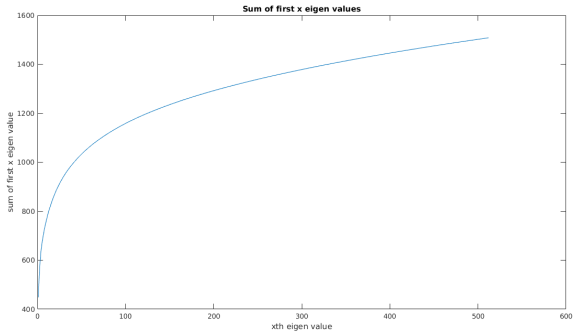

Figure 49: Cumulative eigen values

From the cumulative plot, the convergence is not as quick compared to PCA.

Confusion matrix was plotted taking all the eigen values after taking to kernel space.

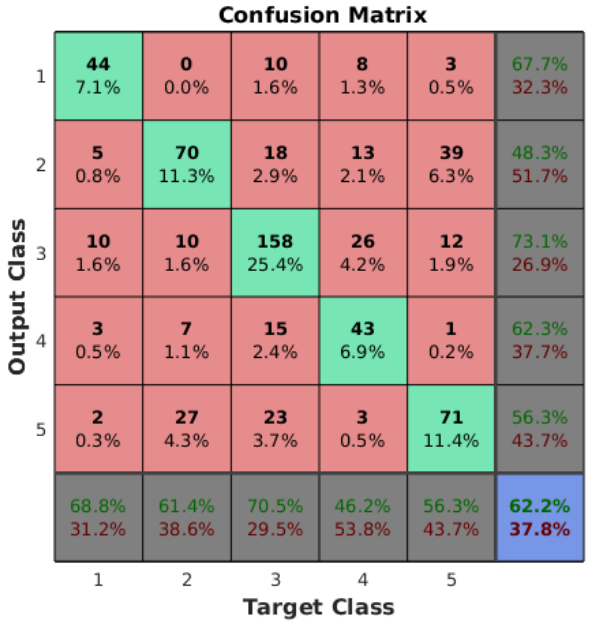

Figure 50: Confusion matrix for Test data

Confusion matrix was plotted taking 20% of eigen values after taking to kernel space.

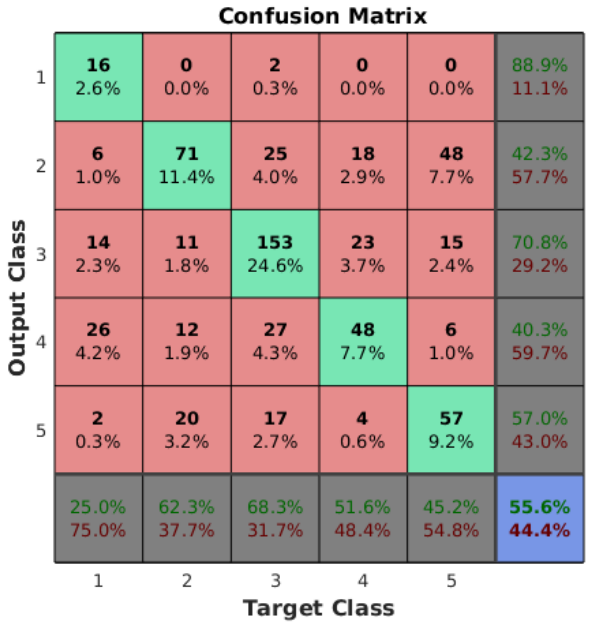

Figure 51: Confusion matrix for Test data

Comparison between different feature selection methods

Among all the feature selection methods, PCA performs the best. The cumulative eigen value plot converges faster compared to Kernel PCA. In case of Kernel PCA, the accuracy is 55% when all

the eigen values are chosen after projecting in kernel space. Choosing 20% of eigen values shows a decrease of only 1% in accuracy. So in both cases PCA and Kernel PCA, the accuracy drop because of feature selection is very less when compared to selecting all eigen values.
AANN performance is bad because, feature selection using Neural network is not compatible with classification method (SVM). The non linearity involved in deriving the projection of original data is complex. And the non linearity in SVM is Gaussian.

5 Graph kernel

Type of kernel used:Random walk kernel until p,where 'p' is the maximum lengths until which walks are considered.

5.1 ν SVM classifier

Parameter estimation
Parameters to be estimated are:"p" and ν
Dataset:NCI109
No. of classes:2
The number of nodes in the dataset are in the range of 10 to 49.Various "p",i.e till 20 are tried and kernel gram matrices are constructed.

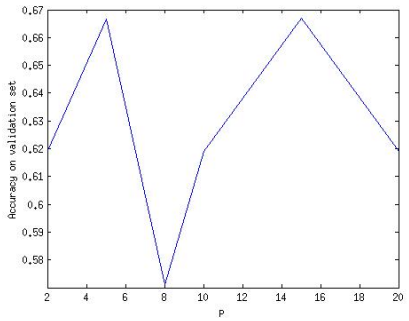

Figure 52: Plot for accuracy on validation set vs p

Value of "p" is chosen based on validation accuracies which is obtained to be 5 in this case.

Confusion matrix
The confusion matrix for p=5 is:

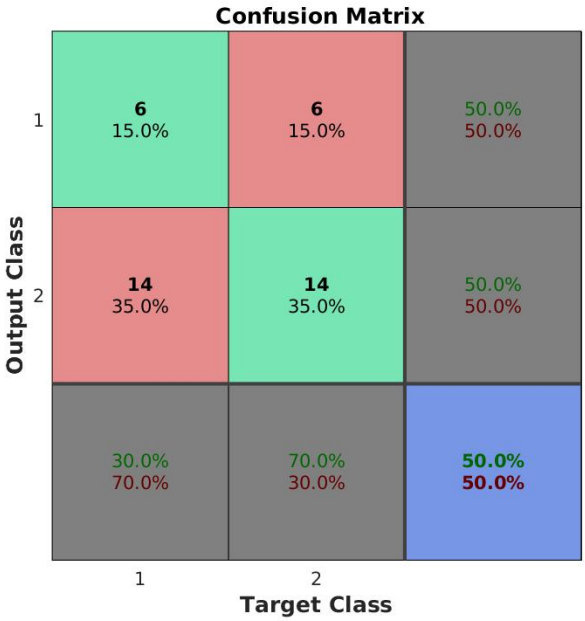

Figure 53: Figure showing confusion matrix for test set for NCI109 dataset ν -SVM

The accuracy is observed to be 50%.

5.2 Kernel kmeans clustering

Parameter estimation Dataset:ENZYMES
The values of "p" are varied from 1 to 20.

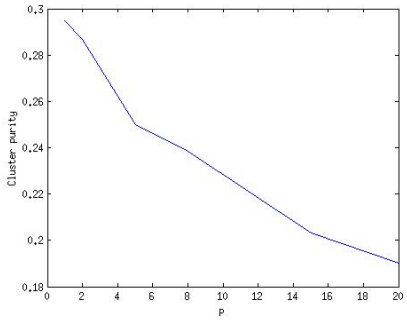

Figure 54: Plot for cluster purity vs p

Best value of p is observed to be "1",where the cluster purity is maximum.The cluster purity value for p=1 is 0.2949
The number of points belonging to each class in each cluster is given by:

class	class1	class2	class3	class4	class5	class6
cluster1	19	18	4	11	3	7
cluster2	21	40	21	16	15	25
cluster3	15	15	26	8	38	26
cluster4	18	4	3	17	8	4
cluster5	18	13	33	23	7	11
cluster6	9	10	13	25	29	27

Table 1: A table consisting of number of points of each class for each cluster for p=1 .

We observe that cluster purity decreases with increase in "p",because more no.of points seem to fall in single cluster and other clusters have less no. of points.This may be because of "tottering",i.e high similarity scores because of repeated nodes in walk,so many points fall into single cluster.