# EECE5644 Fall 2019 – Homework 4

Alan Luke Jacob

NU ID: 001056590

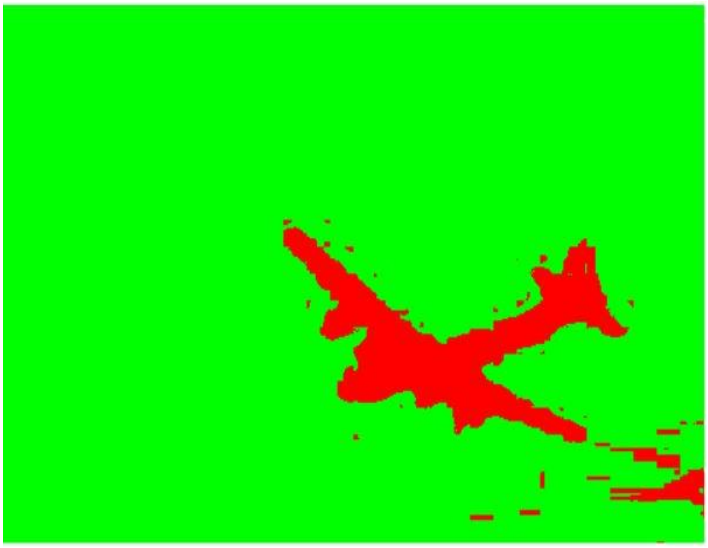
## Question 1:

### Plots and Figures:

Implementing K means Clustering algorithm and GMM based Clustering algorithm for the given test images.

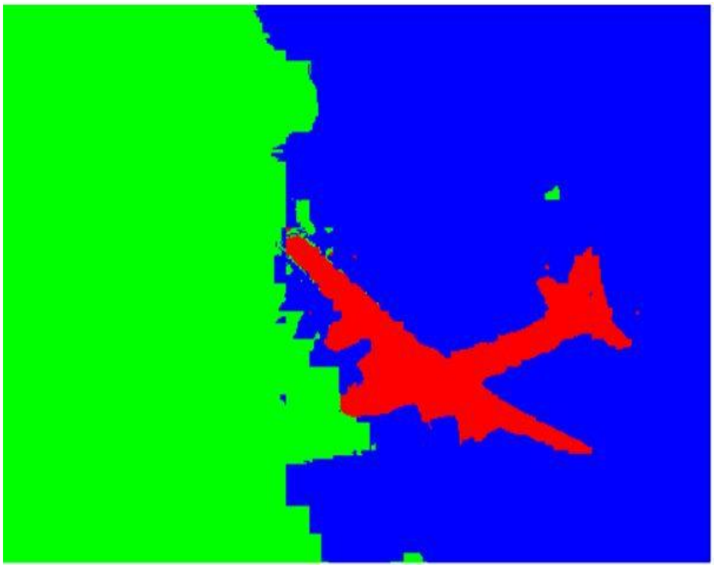
K means K=2:

GMM K=2:



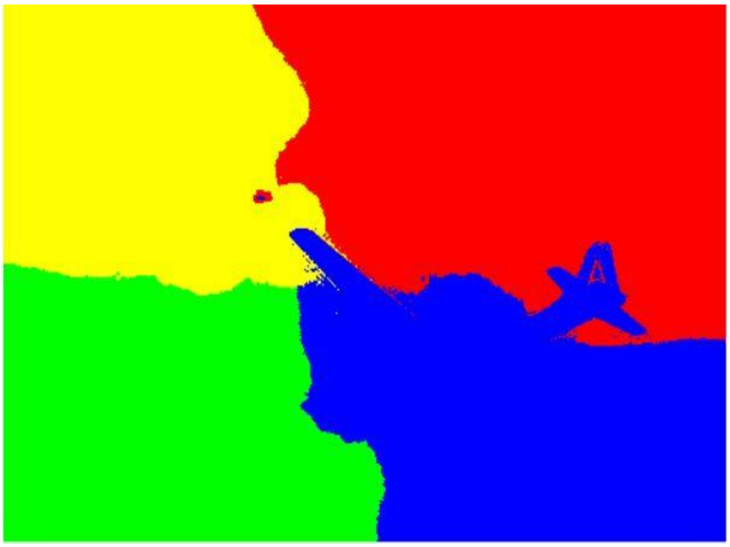
K means K=3

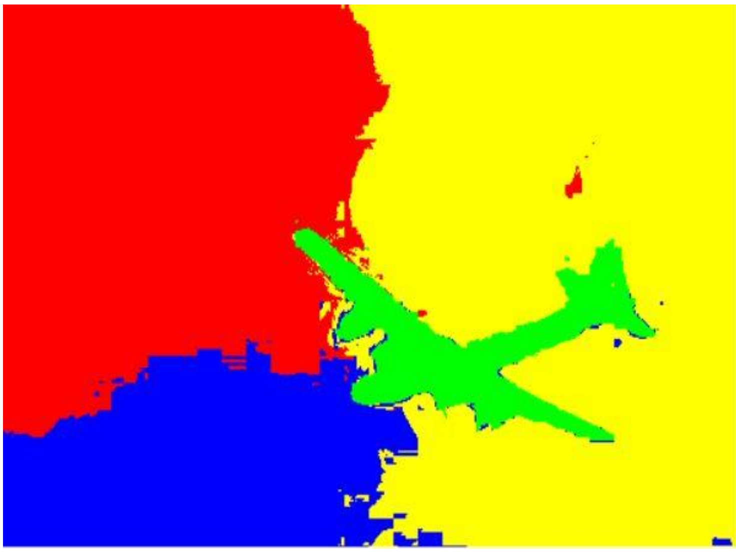
GMM K=3:



K means K=4:

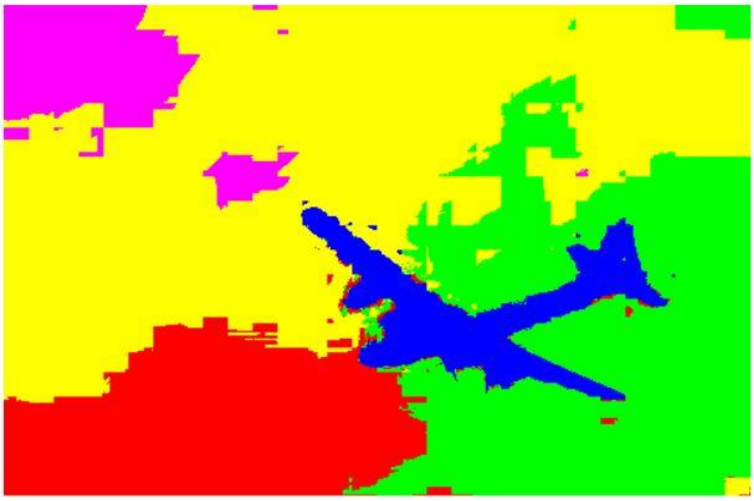
GMM K=4:

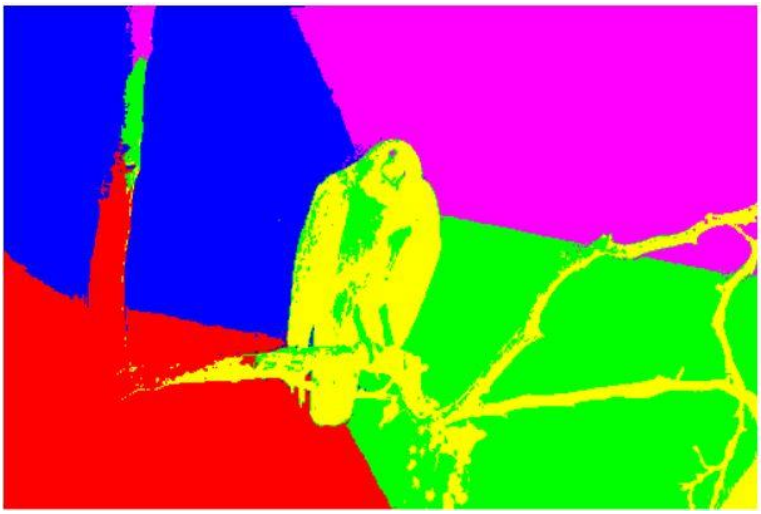




K means K=5:

GMM K=5:





Theory:

**Kmeans:**

The following library is used to implement Kmeans clustering

from sklearn. cluster import KMeans

kmeans = KMeans (n_clusters=k, init='k-means++’, max_iter=300, tol=0.0001)

kmeans = kmeans.fit(Y_norm)

n_clusters=k, defines the number of clusters init='k-means++': selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

max_iter=300 defines the Maximum number of iterations of the k-means algorithm for a single run. In our case it is set to 300.

tol=0.0001, defines the Relative tolerance with regards to inertia to declare convergence.

**GMM Clustering:**

The following library is used to implement GMM clustering

from sklearn.mixture import GaussianMixture

gmm = GaussianMixture(n_components=k, tol=0.001,max_iter=100,init_params='kmeans')
gmm.fit(Y_norm)

n_components=k, defines the number of mixture components which is K in our case

max_iter=100 , defines the number of EM iterations to perform.

`init_params='kmeans',` The method used to initialize the weights, the means and the precisions. It can be one of 'kmeans' or 'random'. In our case it is 'kmeans' `responsibilities are initialized using kmeans.`

tol=0.001, defines the convergence threshold. EM iterations will stop when the lower bound average gain is below this threshold.

Results:

It can be observed for lower values of K such as 2,3,4 the GMM clustering algorithm is better able to cluster pixels in the case of the aeroplane image. However, the performance on K=3,4 Kmeans produces better segmentation for the bird image.

For higher values of K(clusters)K =5,6 the Kmeans algorithm outputs a better result than GMM based clustering in cases of the aeroplane and the bird. It is better able to notice intricate details in the images such as the A on the tail of the aeroplane and the intensity of the pixels in the original Image as compared to GMM clustering which is unable to produce these results.

## Question 2:

## Plots and Figures:

1. Gaussian SVM

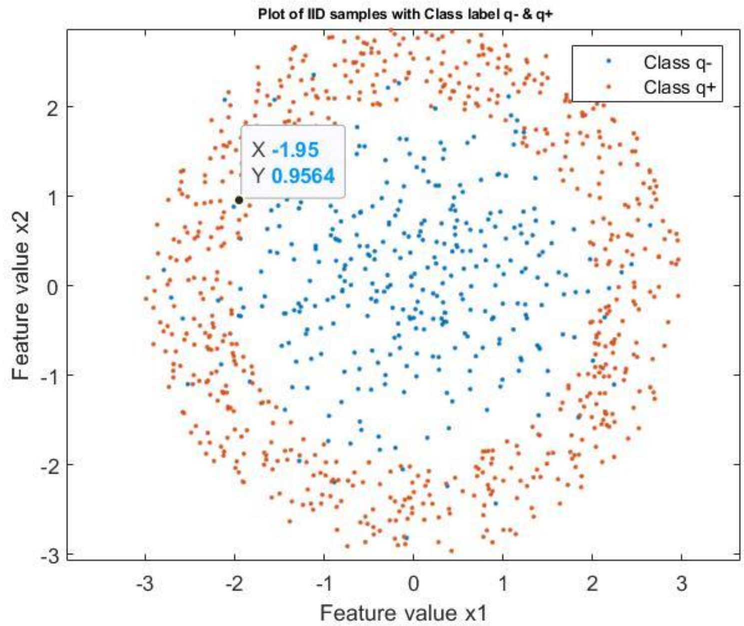
### Training Data set plot

Number of samples generated for Class 1:

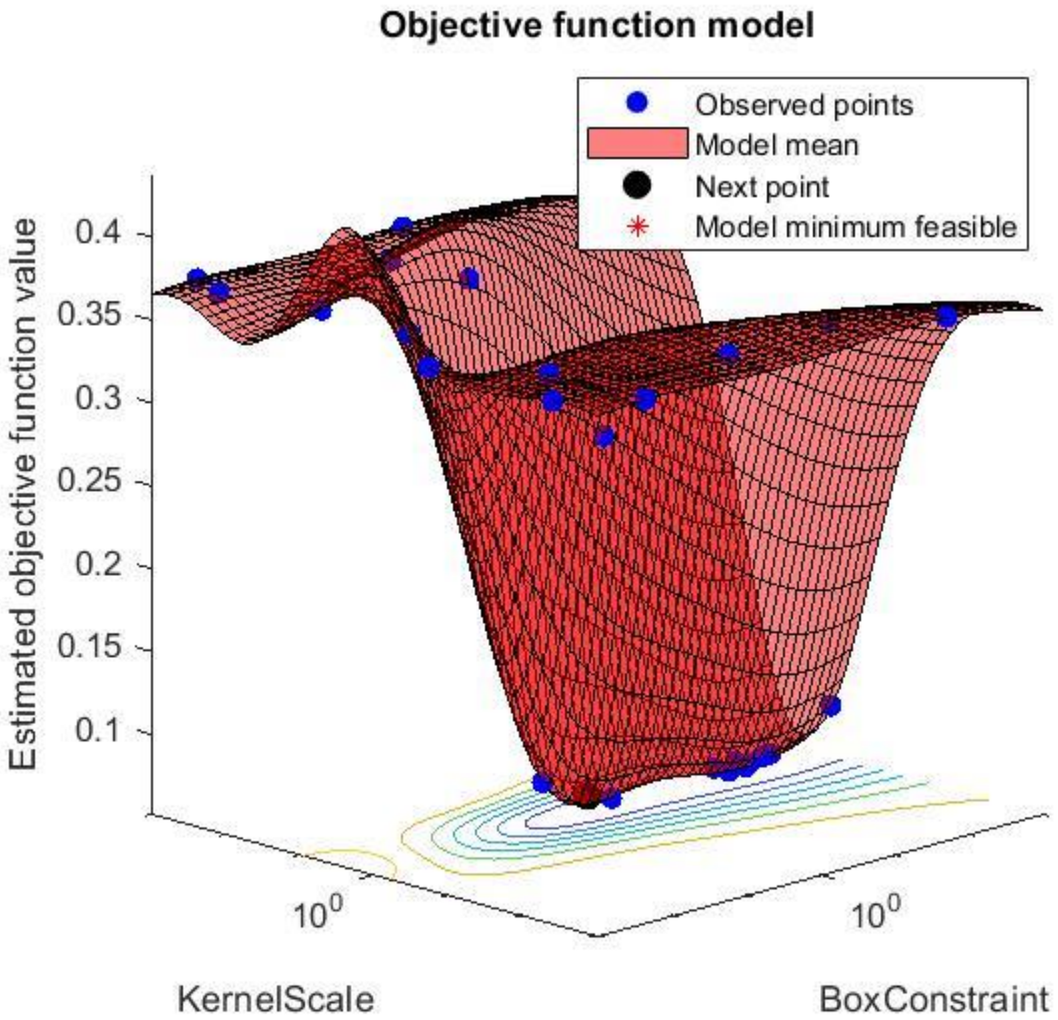
334

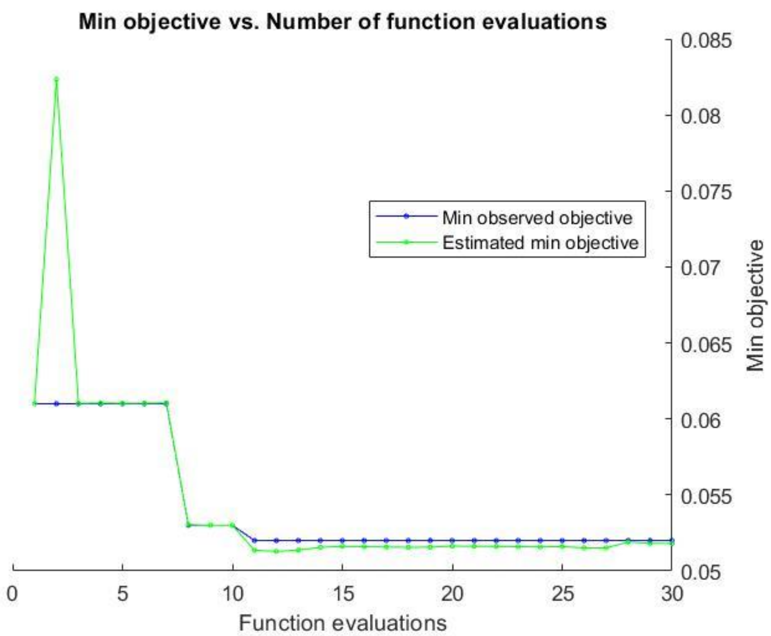
Number of samples generated for Class 2:

666



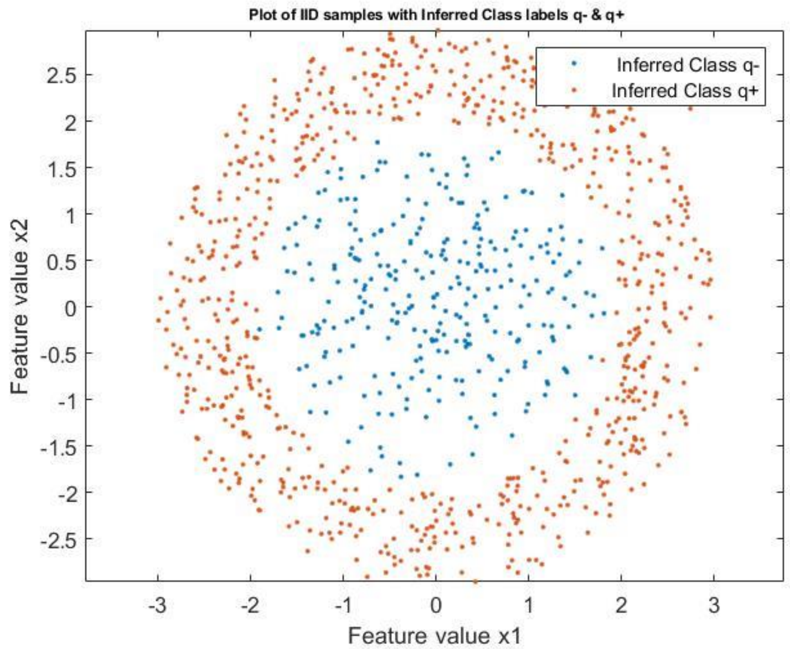

```
model2=fitcsvm(x',L','KernelFunction','gaussian','Optimiz
eHyperparameters','auto');
mdl1=crossval(model2,'KFold',10);
min_loss1=kfoldLoss(mdl1);
```


Objective function model
Observed points
Model mean
Next point
Model minimum feasible
Estimated objective function value
0.4
0.35
0.3
0.25
0.2
0.15
0.1
$10^0$
$10^0$
KernelScale
BoxConstraint


Min objective vs. Number of function evaluations
Min observed objective
Estimated min objective
Min objective
0.085
0.08
0.075
0.07
0.065
0.06
0.055
0.05
0
5
10
15
20
25
30
Function evaluations

Optimizing BoxConstraint(C) and the KernelScale for minimum value of objective function and then using those optimized parameters to train over entire training set

```
model3=fitcsvm(x',L','KernelFunction','gaussian','BoxCons
traint',model2.BoxConstraints(1),'KernelScale',model2.Ker
nelParameters.Scale);
```




Number of samples generated for Class 1
367

Number of samples generated for Class 2
633

Best observed feasible point (according to models):

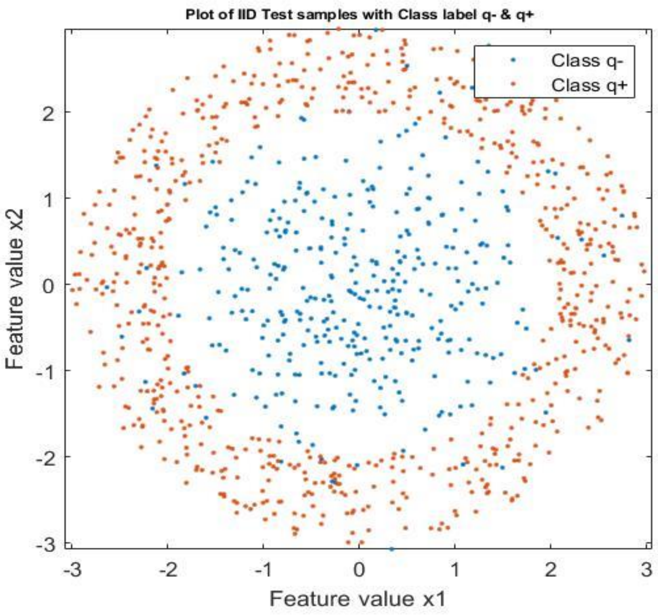
| BoxConstraint | KernelScale |
|---|---|
| 147.09 | 1.6275 |

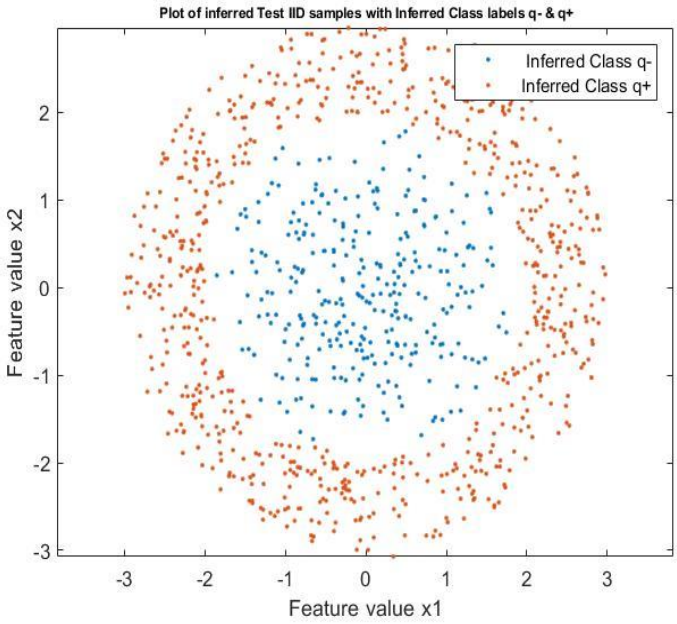
The number of Misclassification errors in training set: 48

Probability of error in training set:
0.0480

Number of samples inferred as Class1:
319

Number of samples inferred as Class:2
681




Number of Test samples generated for Class 1
364

Number of Test samples generated for Class 2
636

The number of Misclassification errors in test set: 48

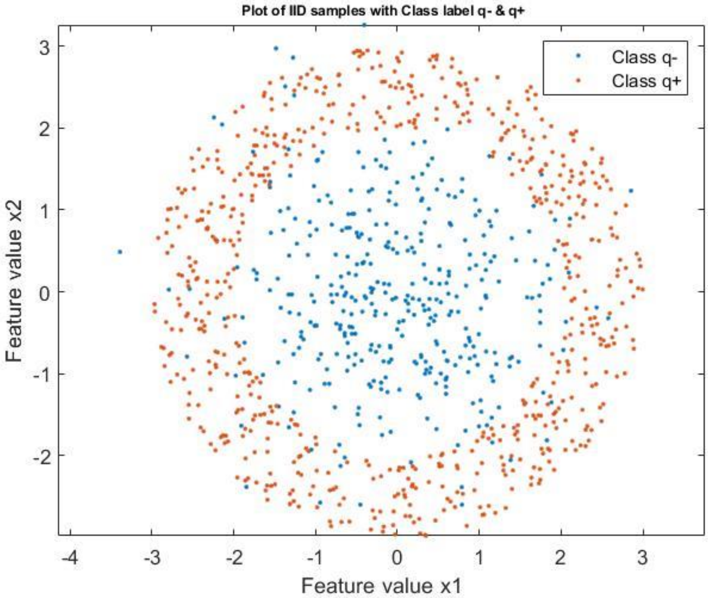
Probability of error in training set:
0.0480

Number of Test samples inferred as Class1:
316

Number of Test samples inferred as Class:2
684

# 2.Linear SVM

Training Data set plot




Number of samples generated for Class 1:

368

Number of samples generated for Class 2:

632

Best observed feasible point:

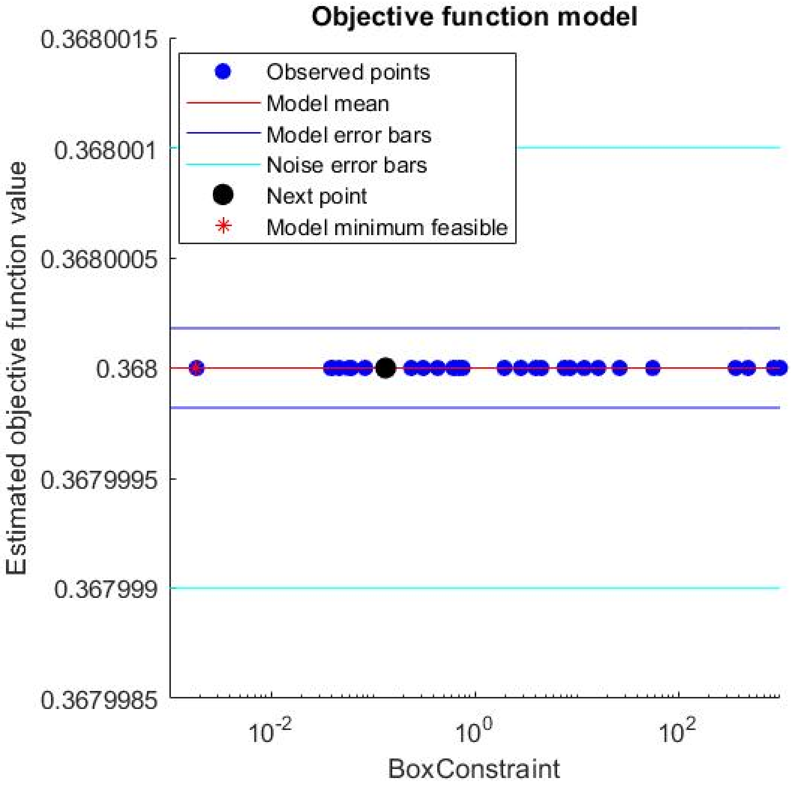
BoxConstraint

_____________

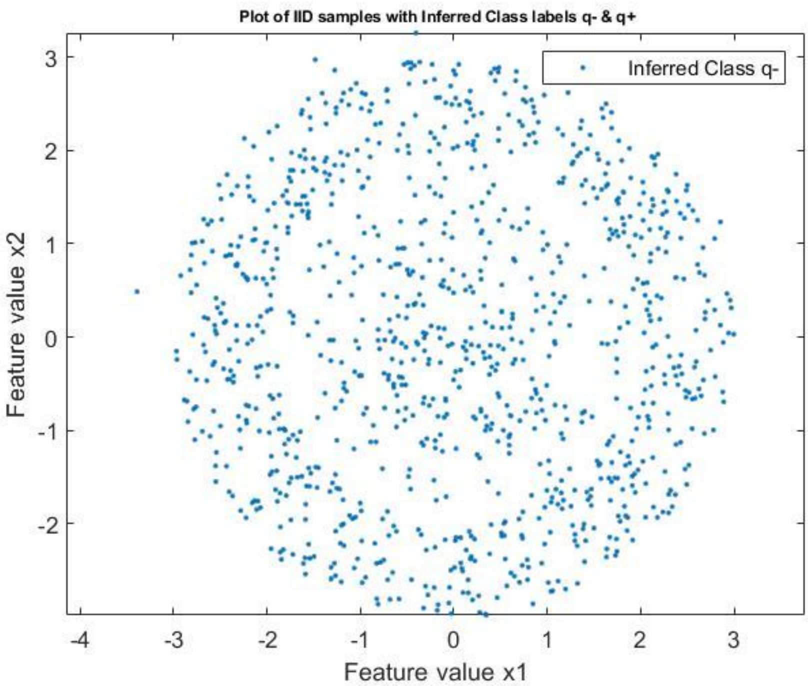
0.046711

```
model2=fitcsvm(x',L','KernelFunction','linear','OptimizeH
yperparameters','BoxConstraint');
mdl1=crossval(model2,'KFold',10);
```

Optimizing BoxConstraint(C) for minimum value of objective function and then using those optimized parameters to train over entire training set




Model Minimum feasible point: C= 0.046711

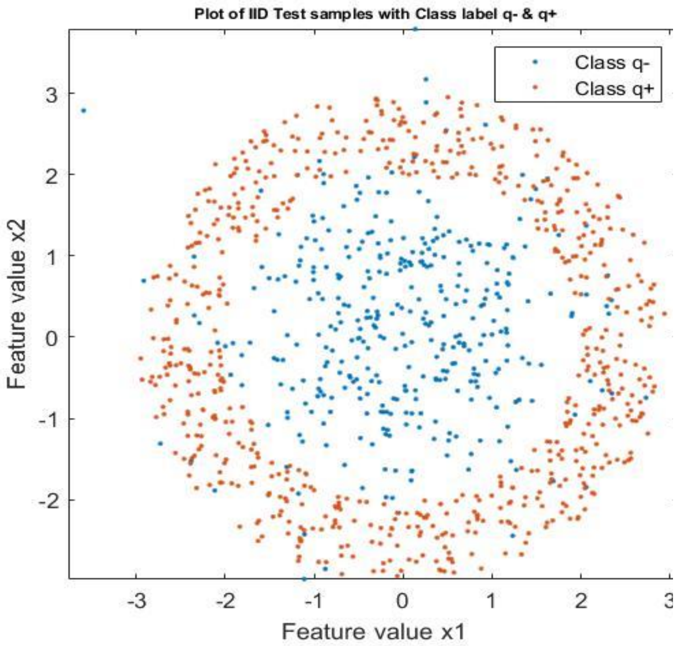


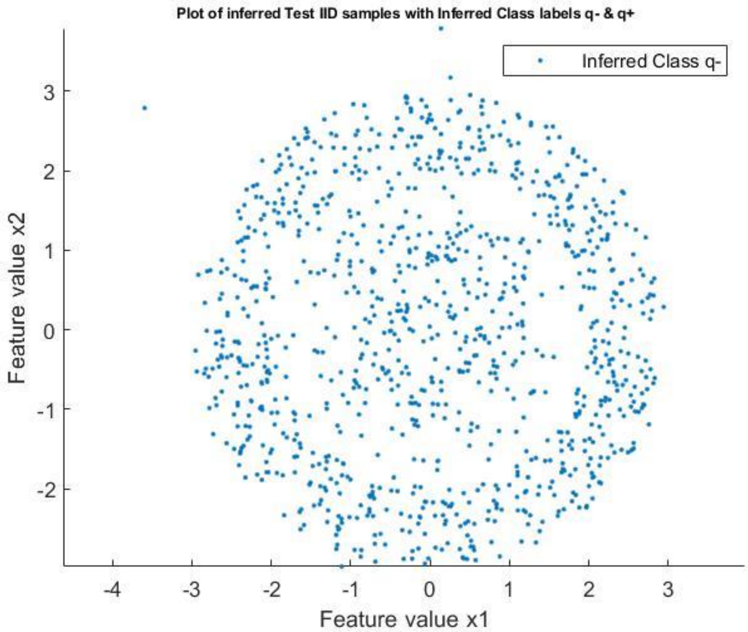
The number of Misclassification errors in training set: 368

Probability of error in training set:
0.3680

Number of samples inferred as Class1:
0

Number of samples inferred as Class:2
1000




Number of Test samples generated for Class 1
355

Number of Test samples generated for Class 2
645

The number of Misclassification errors in test set:
355

Probability of error in training set:
0.3550

Number of Test samples inferred as Class1:
0

Number of Test samples inferred as Class:2
1000