ASSIGNMENT - 2

GROUP - 5

Premnath N CS14B022
R Hemanth Reddy CS14B024
M.Uday Theja CS14B044

Perceptron Classifier:

- It is a binary classifier i.e it determines whether the input represented feature vector belongs to a particular class or not .
- The function maps an input **x** with an output value **f(x)** which is a single binary value .

  f(x) = 1 if w.x + b > 0
  = 0 otherwise

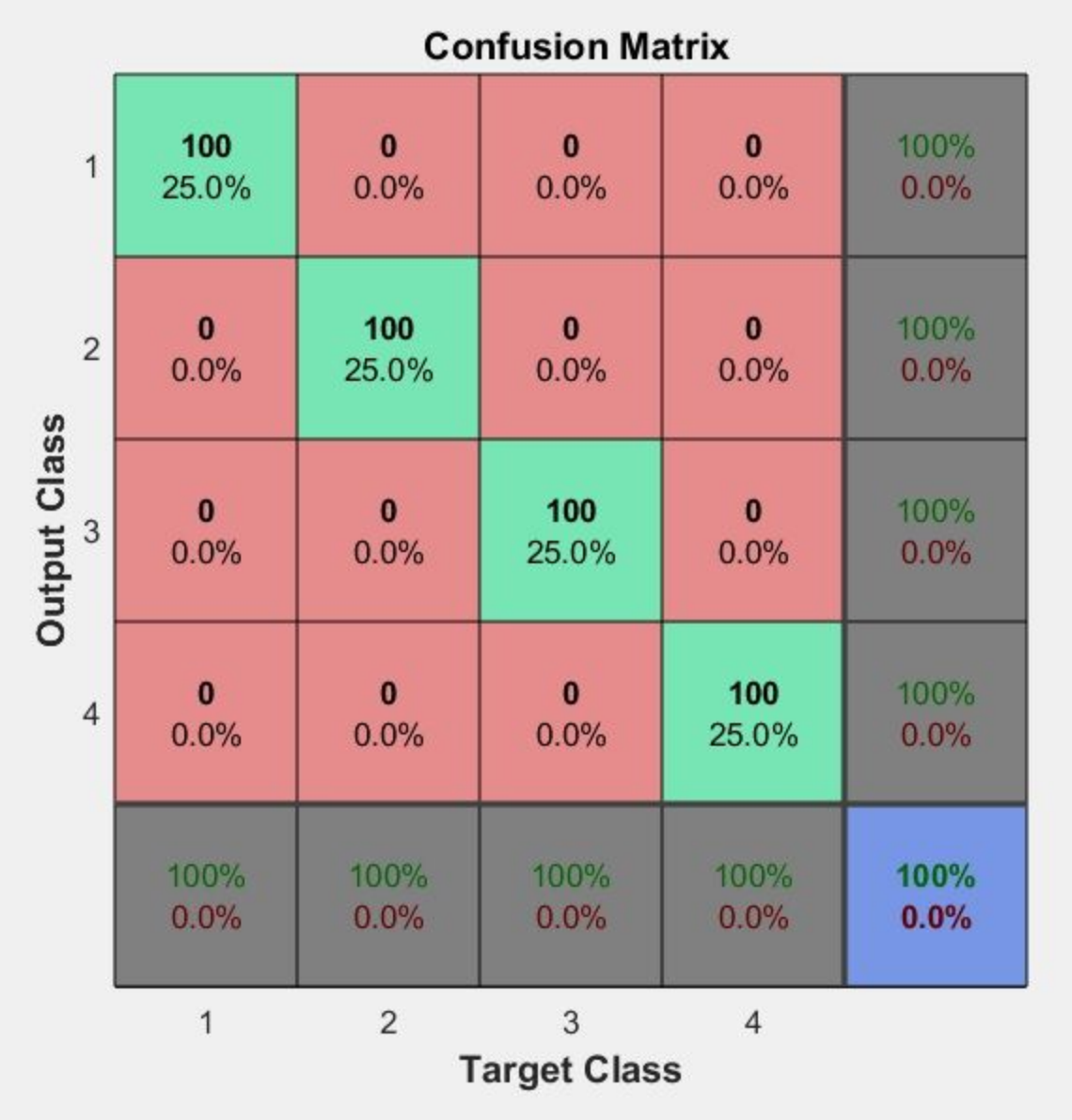
  Here **w** and **x** are vectors and **w.x** denotes the dot product between them and **b** is the bias .
- Weight update equation :

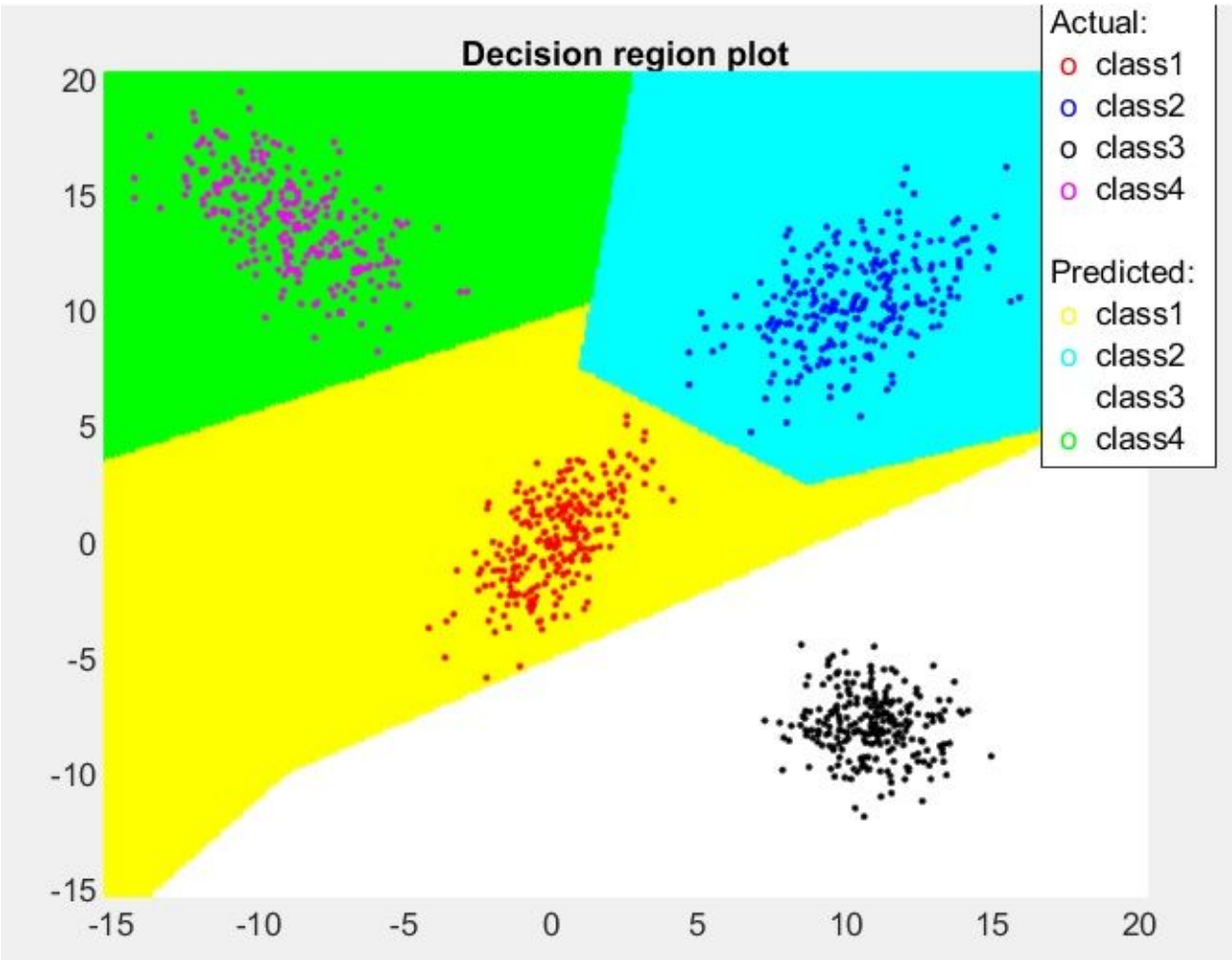
  $w_i(t+1) = w_i(t) + (d_j - y_j(t))*x_{j,i}$ for all features 0 <= i <= d.

  Where, $y_j(t) = f[w(t).x_j]$
  d_j = **desired output**

Perceptron(One vs One) for Linearly Separable Data

- Train ***K* (*K* − 1) / 2** binary classifiers for a *K*-way multiclass problem.
- Each receives the samples of a pair of classes from the original training set, and learns to distinguish these two classes.
- At prediction time, all *K* (*K* − 1) / 2 classifiers are applied to an unseen sample and the class that gets the highest number of "+1" predictions gets predicted by the combined classifier.





Nature of decision surface:

- Decision surfaces are **hyperplanes.**
- The equation of hyperplane is **w.x + b = 0**

Multilayer Feed Forward Neural Networks :

- Artificial neural networks are networks of simple processing elements called neurons that operate on their local data and communicate with other similar elements .
- A typical feed forward neural network consists of an input layer , an output layer and one or more hidden layers .
- The connection between the i^{th} and j^{th} neuron is given by the weight $\mathbf{w_{ij}}$ and the i^{th} neuron by the threshold $\mathbf{\upsilon_i}$. The weight function determines the degree of importance of that articular connection in the neural network .

  The output of i^{th} neuron $\mathbf{x_i}$ is given by

  $$\mathbf{x_i = f(\xi_i)} \quad \text{and}$$

  $$\mathbf{\xi_i = \upsilon_i + \Sigma_{over\ j} w_{ij} x_j}$$

  where ξ_i is the potential of the i^{th} neuron and f is the trAccuracyfer function .

  It holds that

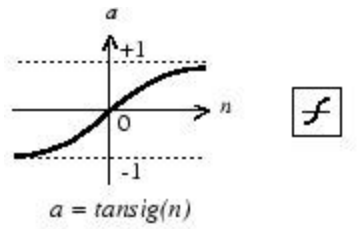
  $$\mathbf{f(\xi) = 1/(1 + \exp(-\xi))}$$

- We vary the thresholds and weights to minimise the difference between the computed and the required output values . Thus , the objective function E is minimised .

  $$\mathbf{E = \Sigma_{over\ outputs}\ 0.5 * (x_o - x^{\Lambda}_o)^2}$$

  where $\mathbf{x_o}$ and $\mathbf{x^{\Lambda}_o}$ are the computed and required values of the output neurons and summation runs over all the output neurons **o .**

Feed Forward Neural Networks :
Transfer Function: Hyperbolic tangent sigmoid transfer function



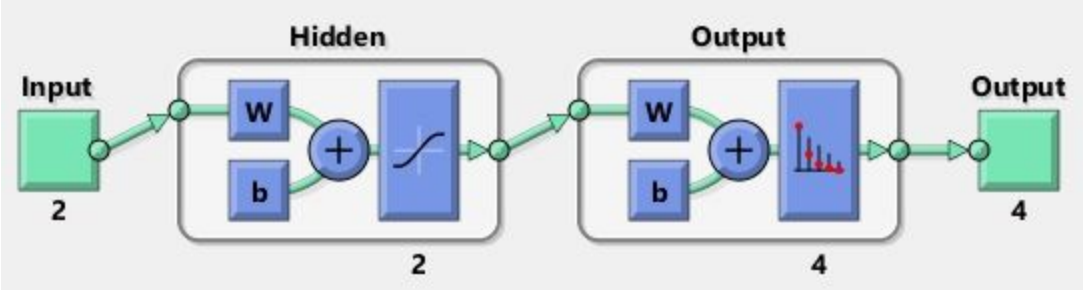
Tan-Sigmoid Transfer Function

DataDivision : Index (divideind)
Training : Scaled Conjugate Gradient (trainscg)
Performance : Cross-Entropy (crossentropy)
Calculations : MEX

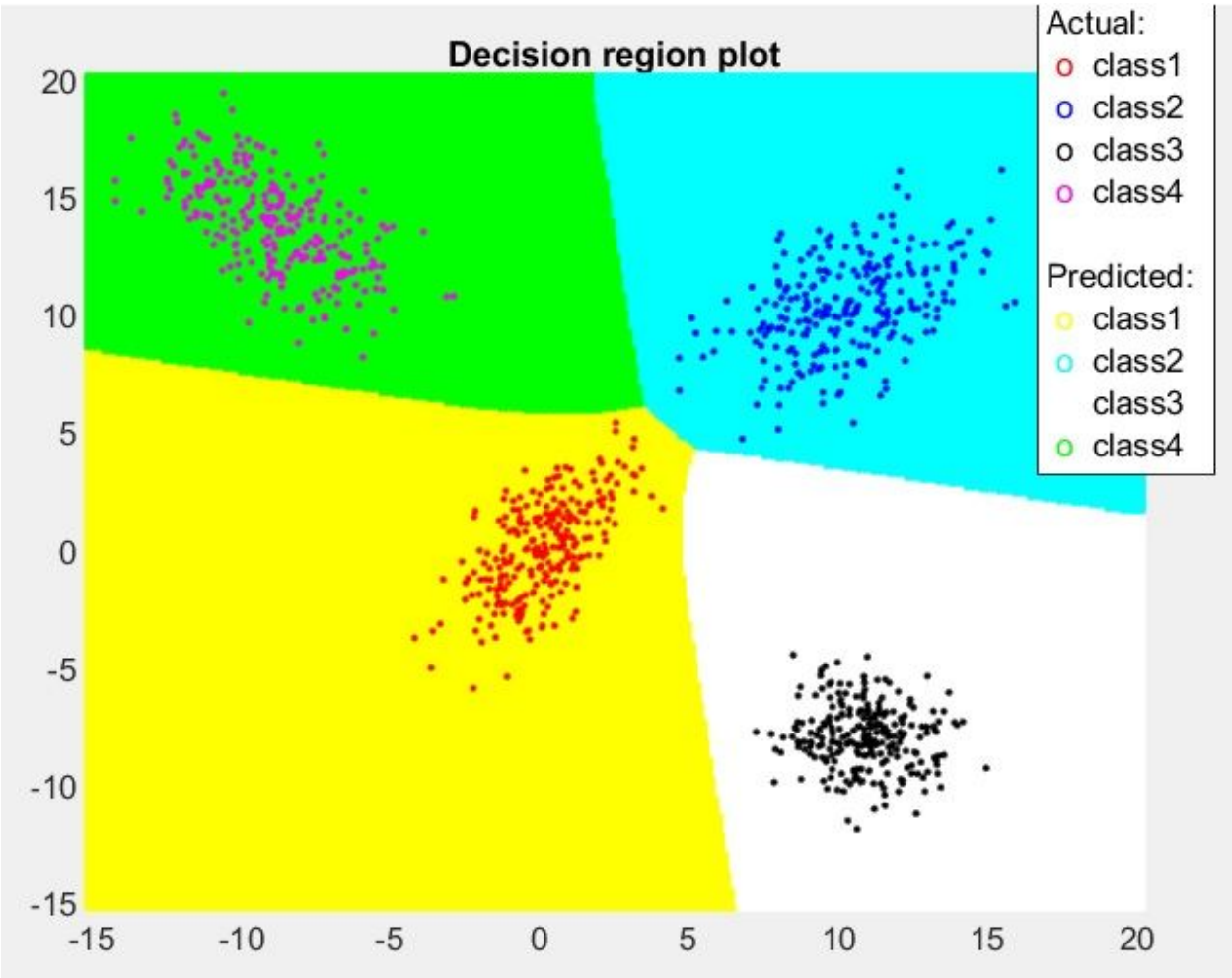
Linearly Separable Data

Epoch = 35 Iterations
Performance = 0.000105
Gradient = 0.000241
Validation Checks = 6

Best Validation Performance is 0.00041847 at epoch 29

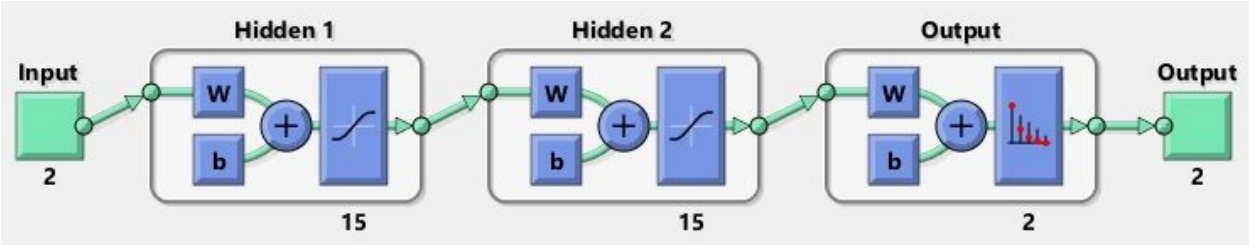

Input
2
Hidden
W
b
2
Output
W
b
4
Output
4


Training Confusion Matrix
Output Class
Target Class
1 250 25.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%
2 0 0.0% 250 25.0% 0 0.0% 0 0.0% 100% 0.0%
3 0 0.0% 0 0.0% 250 25.0% 0 0.0% 100% 0.0%
4 0 0.0% 0 0.0% 0 0.0% 250 25.0% 100% 0.0%
100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0%
1 2 3 4
Validation Confusion Matrix
Output Class
Target Class
1 150 25.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%
2 0 0.0% 150 25.0% 0 0.0% 0 0.0% 100% 0.0%
3 0 0.0% 0 0.0% 150 25.0% 0 0.0% 100% 0.0%
4 0 0.0% 0 0.0% 0 0.0% 150 25.0% 100% 0.0%
100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0%
1 2 3 4
Test Confusion Matrix
Output Class
Target Class
1 100 25.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%
2 0 0.0% 100 25.0% 0 0.0% 0 0.0% 100% 0.0%
3 0 0.0% 0 0.0% 100 25.0% 0 0.0% 100% 0.0%
4 0 0.0% 0 0.0% 0 0.0% 100 25.0% 100% 0.0%
100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0%
1 2 3 4
All Confusion Matrix
Output Class
Target Class
1 500 25.0% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%
2 0 0.0% 500 25.0% 0 0.0% 0 0.0% 100% 0.0%
3 0 0.0% 0 0.0% 500 25.0% 0 0.0% 100% 0.0%
4 0 0.0% 0 0.0% 0 0.0% 500 25.0% 100% 0.0%
100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0% 100% 0.0%
1 2 3 4



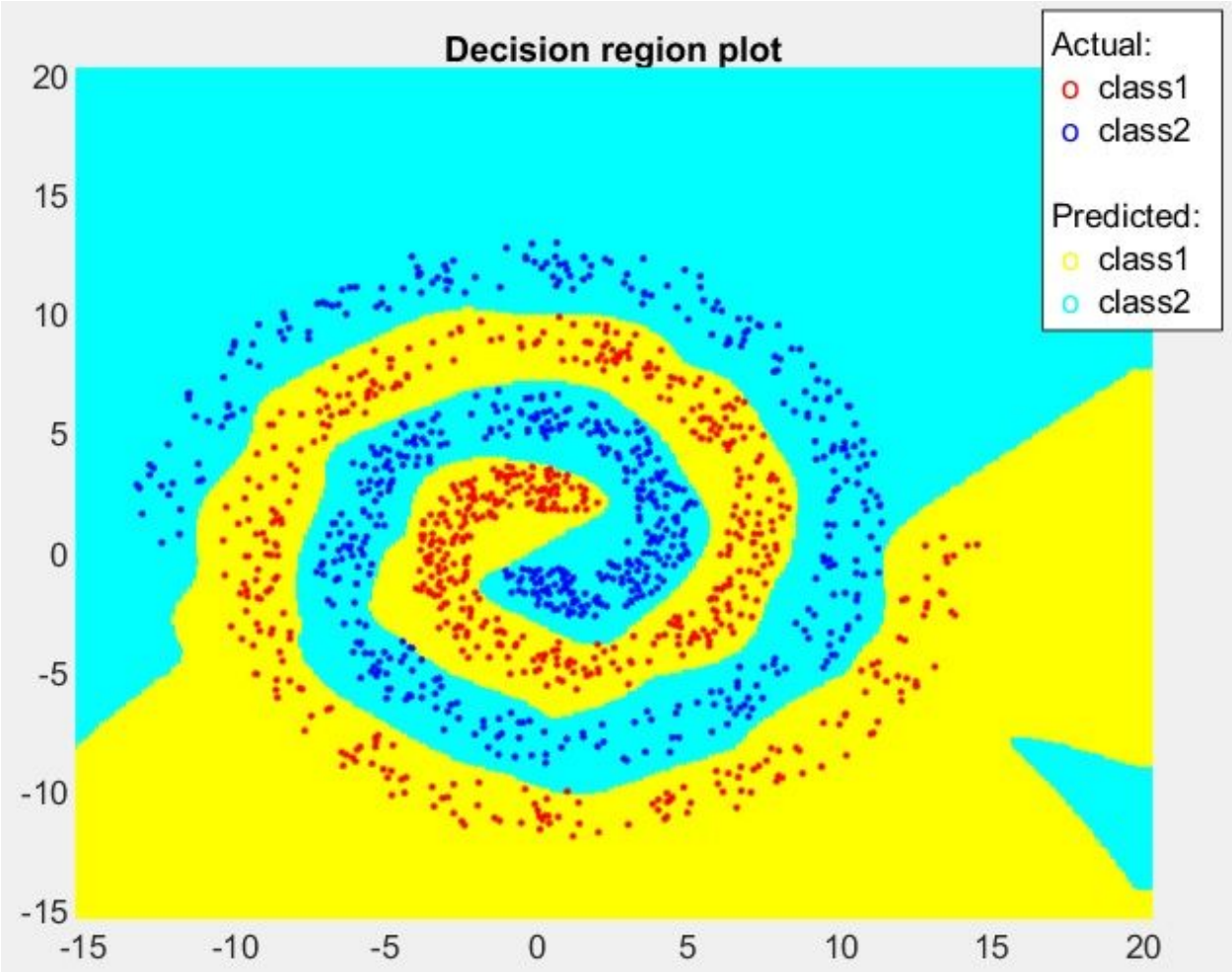
Non Linearly Separable Data

Two Hidden Layers:

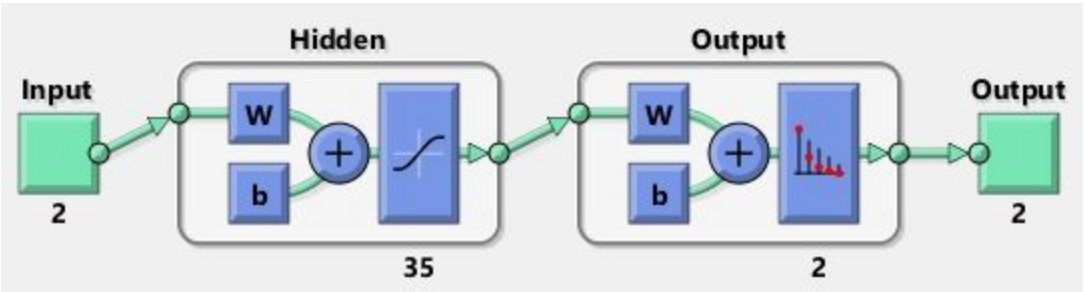


Epoch = 139 Iterations
Performance = 0.000278
Gradient = 0.00138
Validation Checks = 6

Best Validation Performance is 0.0020838 at epoch 133

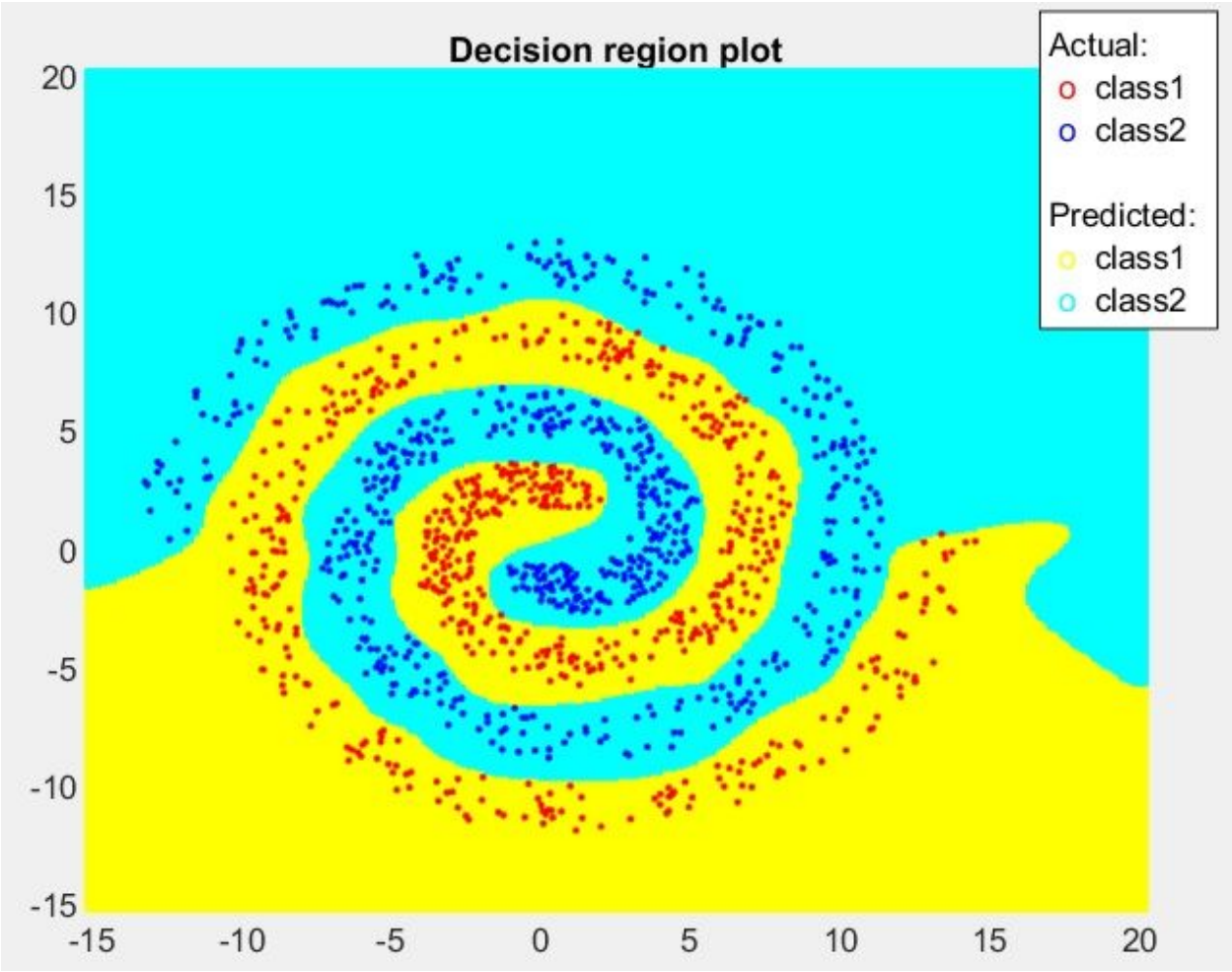


Single Hidden Layer



Epoch = 123 Iterations
Performance = 0.000529
Gradient = 0.000630
Validation Checks = 6

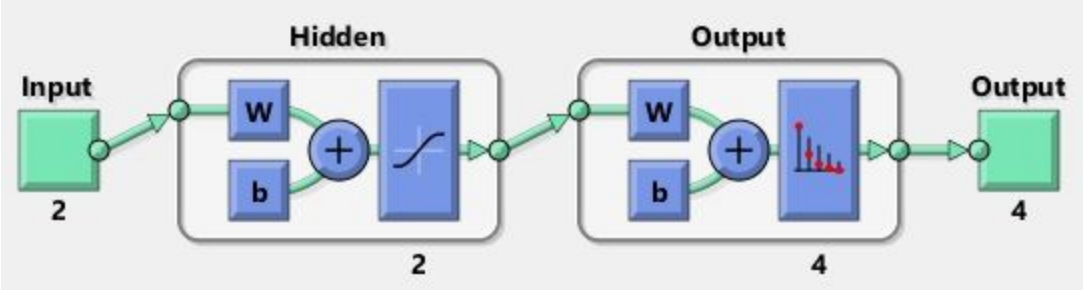
Best Validation Performance is 0.0017486 at epoch 117

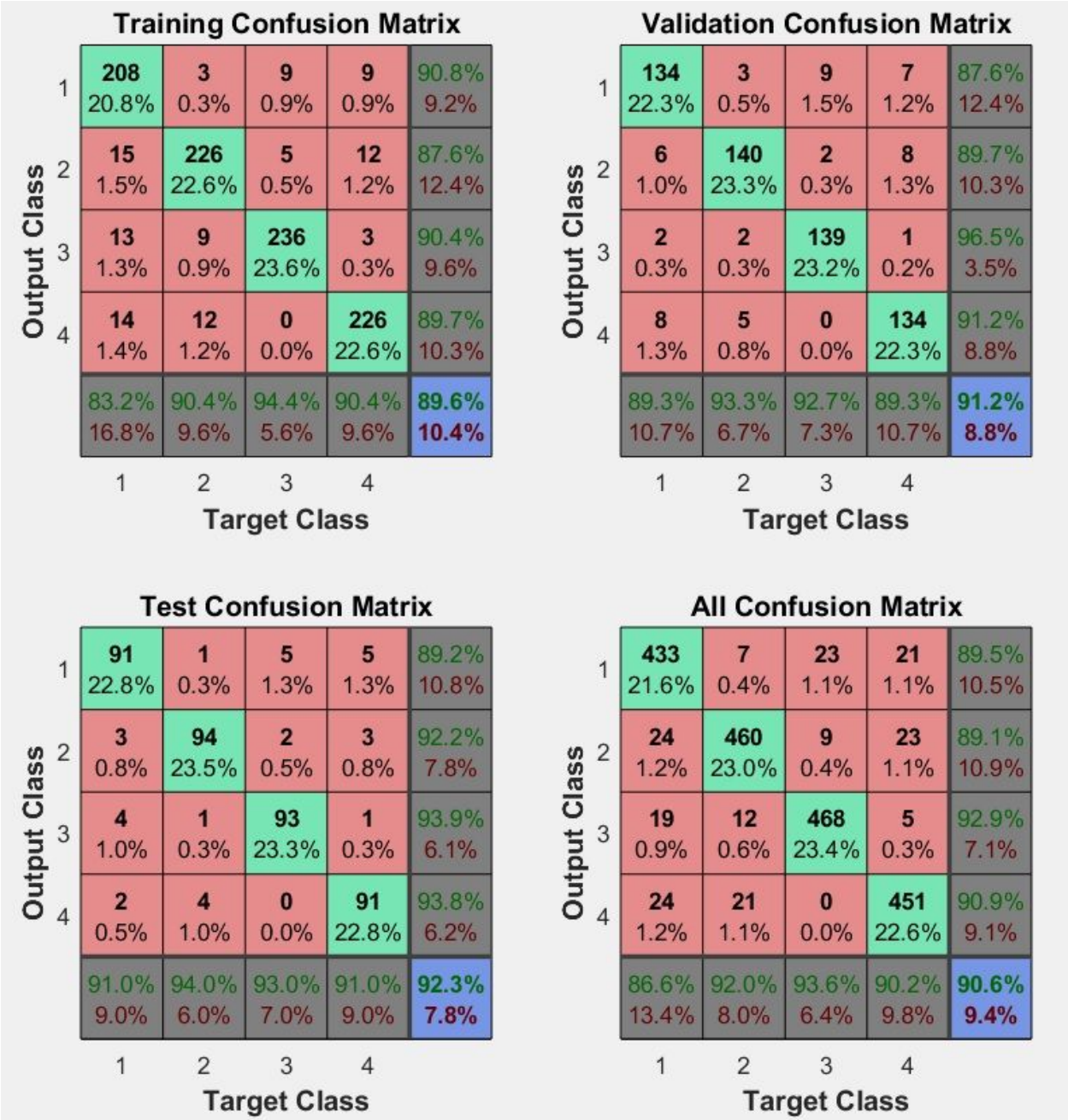


Overlapping Data

Epoch = 35 Iterations
Performance = 0.0705
Gradient = 0.00568
Validation Checks = 6

Best Validation Performance is 0.061296 at epoch 29


Input
2
Hidden
W
b
2
Output
W
b
4
Output
4


Training Confusion Matrix
Output Class
Target Class
208 20.8% | 3 0.3% | 9 0.9% | 9 0.9% | 90.8% 9.2%
15 1.5% | 226 22.6% | 5 0.5% | 12 1.2% | 87.6% 12.4%
13 1.3% | 9 0.9% | 236 23.6% | 3 0.3% | 90.4% 9.6%
14 1.4% | 12 1.2% | 0 0.0% | 226 22.6% | 89.7% 10.3%
83.2% 16.8% | 90.4% 9.6% | 94.4% 5.6% | 90.4% 9.6% | 89.6% 10.4%
Validation Confusion Matrix
Output Class
Target Class
134 22.3% | 3 0.5% | 9 1.5% | 7 1.2% | 87.6% 12.4%
6 1.0% | 140 23.3% | 2 0.3% | 8 1.3% | 89.7% 10.3%
2 0.3% | 2 0.3% | 139 23.2% | 1 0.2% | 96.5% 3.5%
8 1.3% | 5 0.8% | 0 0.0% | 134 22.3% | 91.2% 8.8%
89.3% 10.7% | 93.3% 6.7% | 92.7% 7.3% | 89.3% 10.7% | 91.2% 8.8%
Test Confusion Matrix
Output Class
Target Class
91 22.8% | 1 0.3% | 5 1.3% | 5 1.3% | 89.2% 10.8%
3 0.8% | 94 23.5% | 2 0.5% | 3 0.8% | 92.2% 7.8%
4 1.0% | 1 0.3% | 93 23.3% | 1 0.3% | 93.9% 6.1%
2 0.5% | 4 1.0% | 0 0.0% | 91 22.8% | 93.8% 6.2%
91.0% 9.0% | 94.0% 6.0% | 93.0% 7.0% | 91.0% 9.0% | 92.3% 7.8%
All Confusion Matrix
Output Class
Target Class
433 21.6% | 7 0.4% | 23 1.1% | 21 1.1% | 89.5% 10.5%
24 1.2% | 460 23.0% | 9 0.4% | 23 1.1% | 89.1% 10.9%
19 0.9% | 12 0.6% | 468 23.4% | 5 0.3% | 92.9% 7.1%
24 1.2% | 21 1.1% | 0 0.0% | 451 22.6% | 90.9% 9.1%
86.6% 13.4% | 92.0% 8.0% | 93.6% 6.4% | 90.2% 9.8% | 90.6% 9.4%

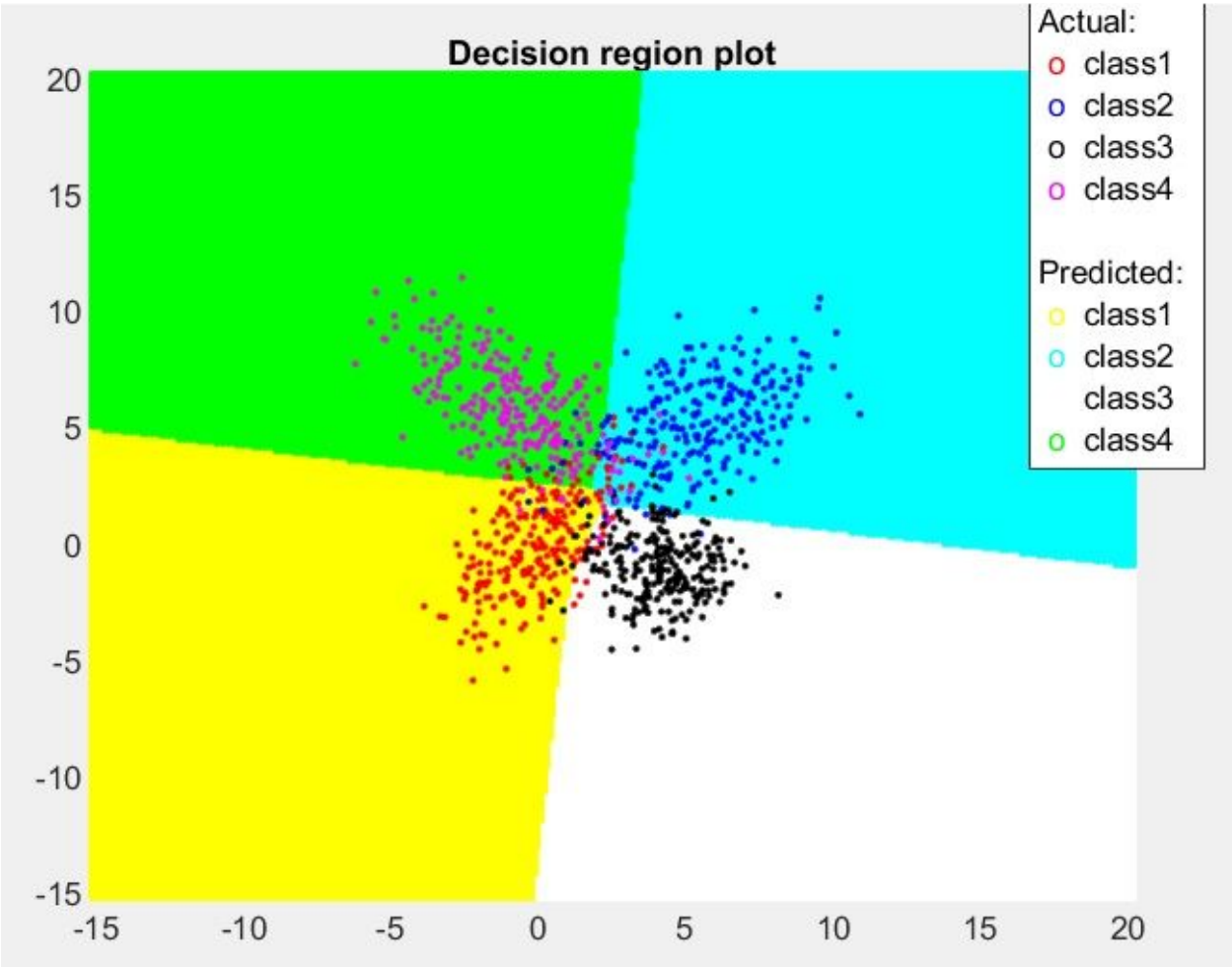
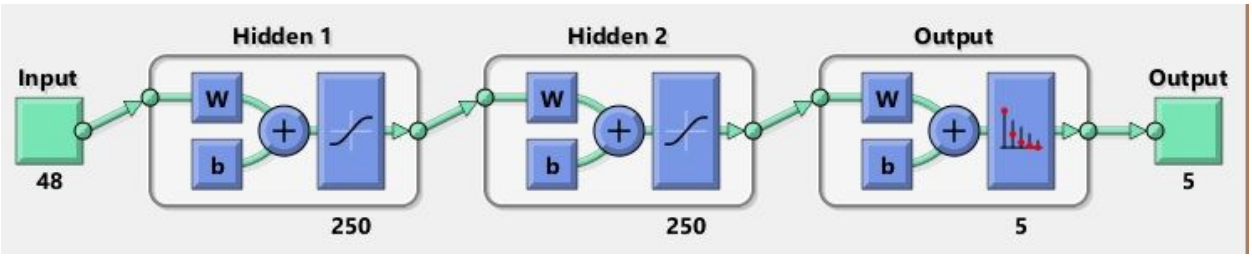


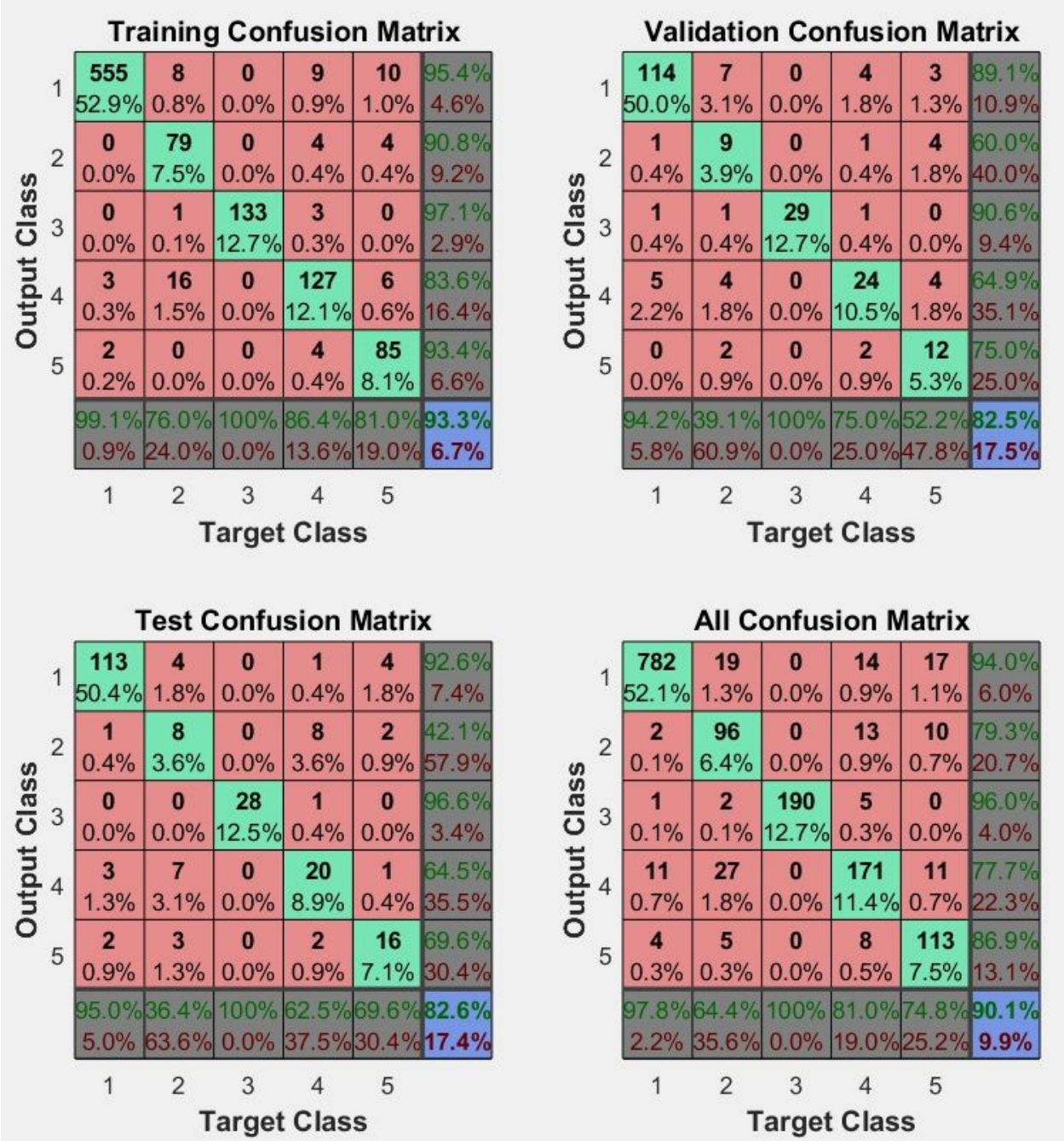
Image Dataset

48 Dimension



Epoch = 3500 Iterations
Performance = 0.0388
Gradient = 0.0718

Best Validation Performance is 0.14379 at epoch 2237



Dimensionality Reduction

Data Preprocessing (Feature Scaling / Mean Normalisation):

- Scale the features to have comparable range of values.
- Standardisation of data:

  If we are given n data points with d features each, then for every feature, we calculate the Mean(μ) and Sigma (sqrt(variance)) and transform each data point using the formula:

  X = X- μ / Sigma .

Principal Component Analysis (PCA) Algorithm

- Performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.
- Compute Covariance matrix (Sigma).
- Compute the **Eigenvectors** of matrix Sigma.

  [U,S,V] = svd(Sigma) (svd: Singular Valued Decomposition)

  The columns of U matrix are the eigenvectors.
- Choose the first k eigenvectors and reconstruct the data.

Choosing number of Principal Components *k*

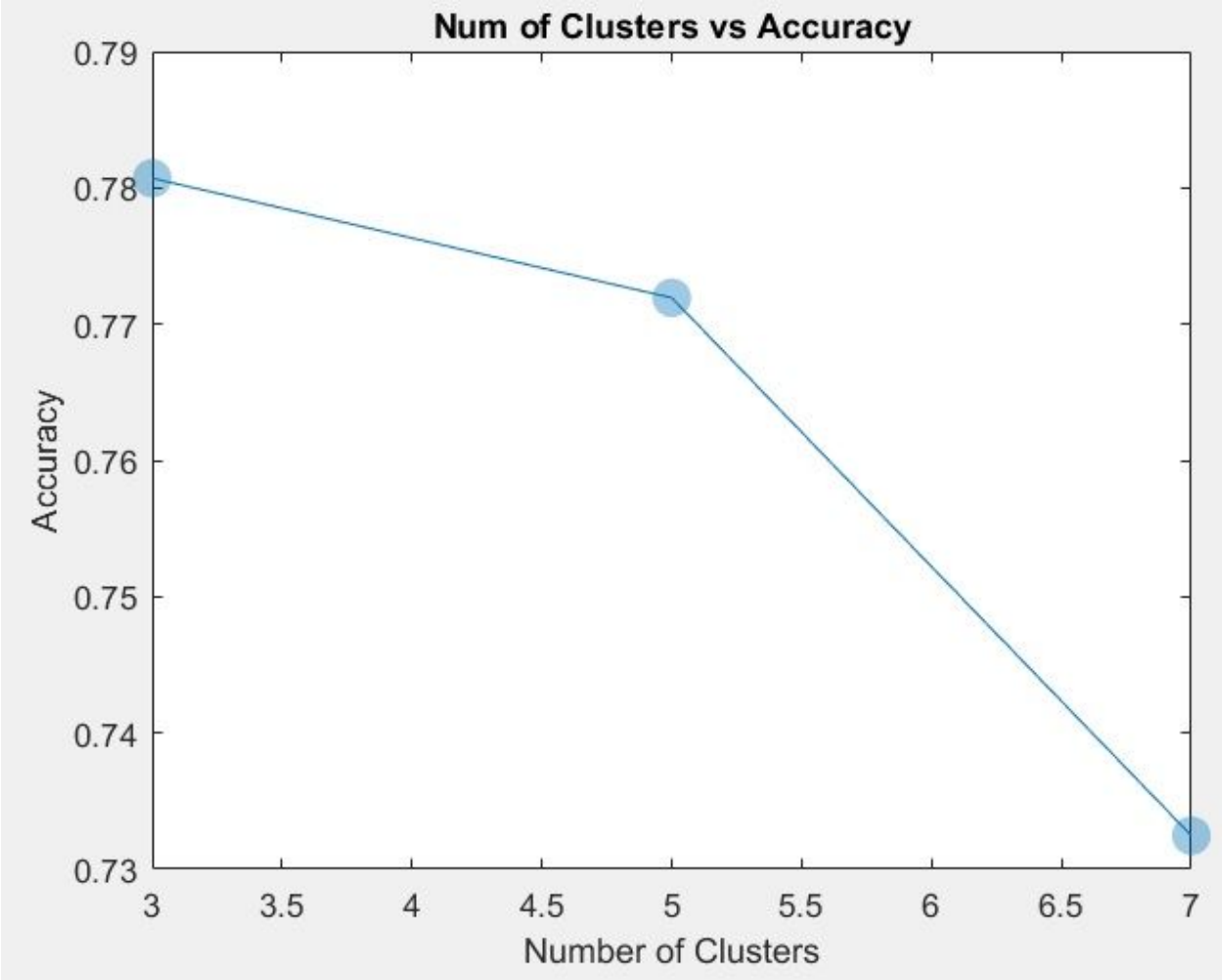
- Average Squared projection error = $1/N \Sigma || x^{(i)} - x^{(i)}_{approx} ||^2$
- Total variation in data = $1/N \Sigma || x^{(i)} ||^2$
- Choose k to be smallest such that:

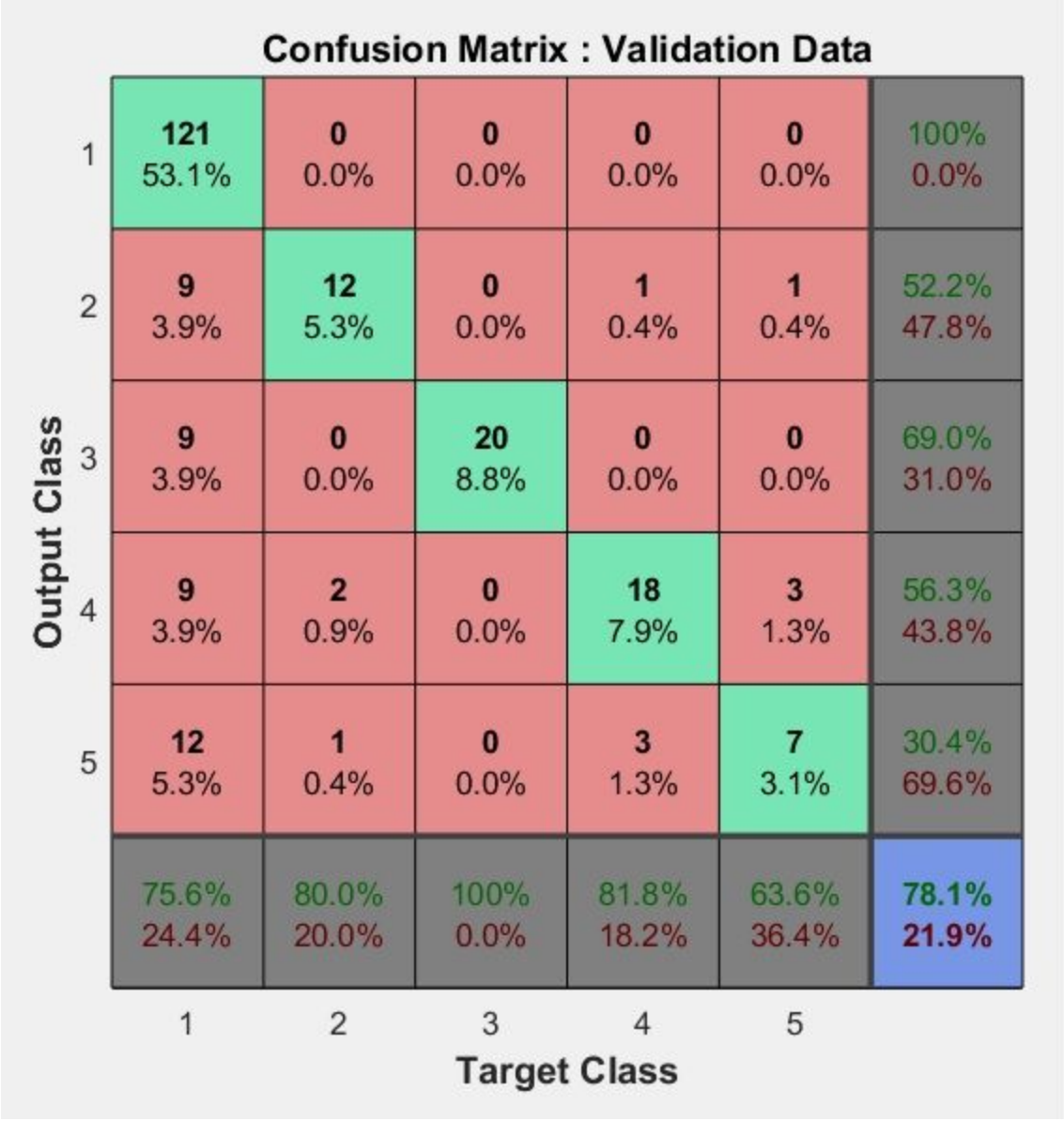
  Average Squared projection error / Total variation in data <= 0.01

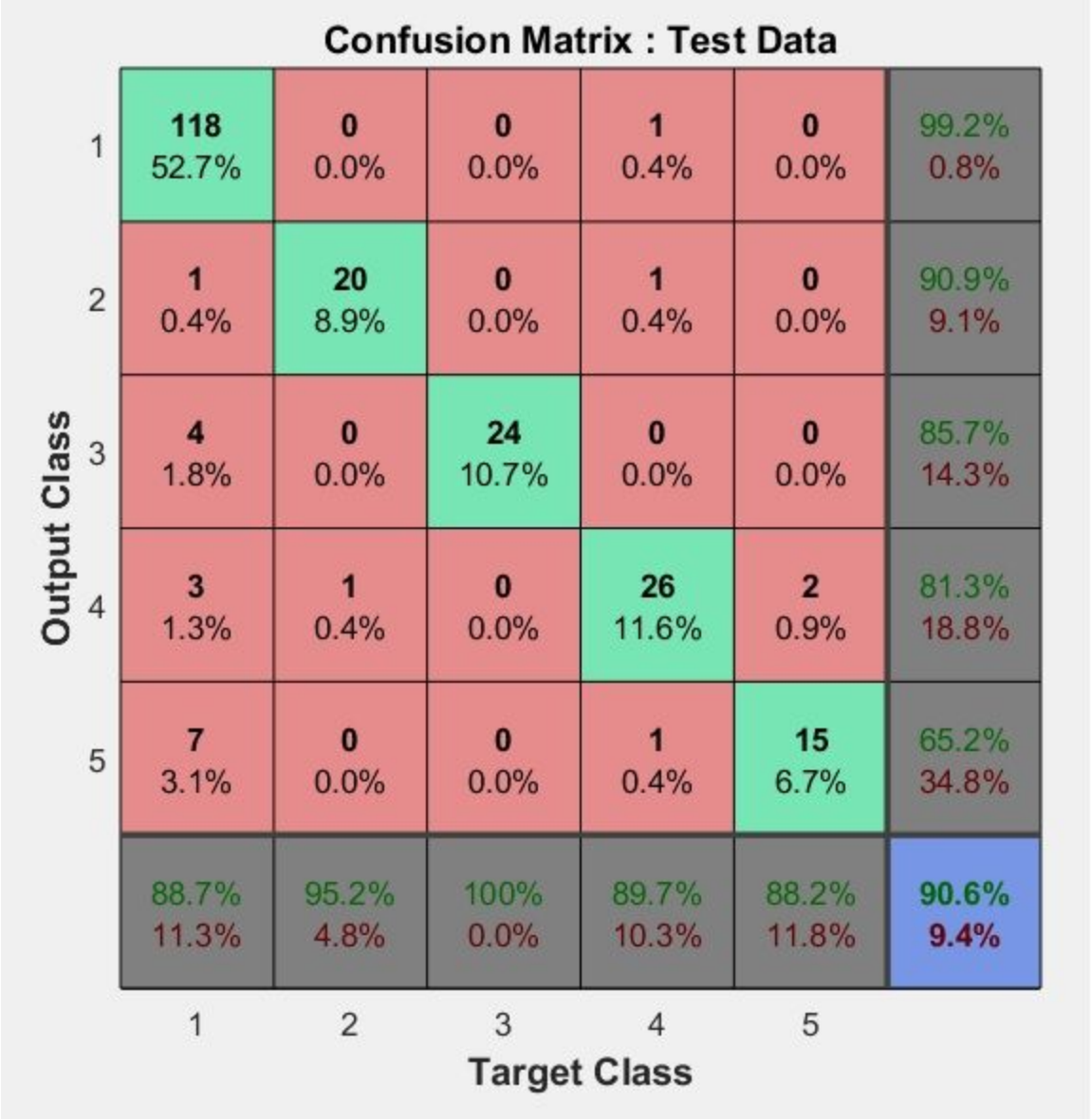
  99% variance is retained.

Image Dataset
26 Dimensions
90% Variance Retained

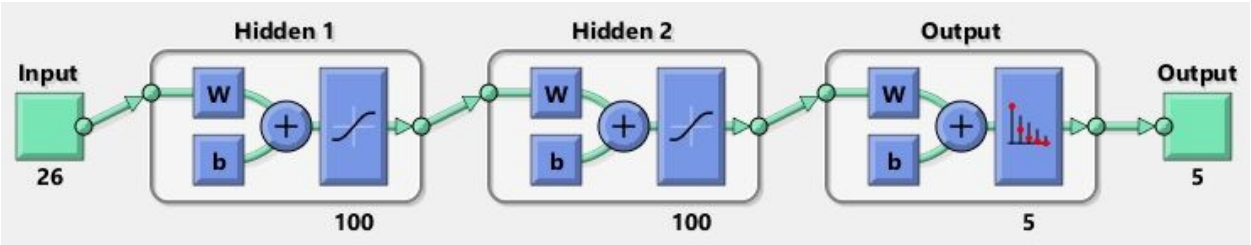
GMM




Confusion Matrix : Validation Data
Output Class
Target Class
1
2
3
4
5
121
53.1%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
9
3.9%
12
5.3%
0
0.0%
1
0.4%
1
0.4%
52.2%
47.8%
9
3.9%
0
0.0%
20
8.8%
0
0.0%
0
0.0%
69.0%
31.0%
9
3.9%
2
0.9%
0
0.0%
18
7.9%
3
1.3%
56.3%
43.8%
12
5.3%
1
0.4%
0
0.0%
3
1.3%
7
3.1%
30.4%
69.6%
75.6%
24.4%
80.0%
20.0%
100%
0.0%
81.8%
18.2%
63.6%
36.4%
78.1%
21.9%


Confusion Matrix : Test Data
Output Class
Target Class
1
2
3
4
5
118
52.7%
0
0.0%
0
0.0%
1
0.4%
0
0.0%
99.2%
0.8%
1
0.4%
20
8.9%
0
0.0%
1
0.4%
0
0.0%
90.9%
9.1%
4
1.8%
0
0.0%
24
10.7%
0
0.0%
0
0.0%
85.7%
14.3%
3
1.3%
1
0.4%
0
0.0%
26
11.6%
2
0.9%
81.3%
18.8%
7
3.1%
0
0.0%
0
0.0%
1
0.4%
15
6.7%
65.2%
34.8%
88.7%
11.3%
95.2%
4.8%
100%
0.0%
89.7%
10.3%
88.2%
11.8%
90.6%
9.4%

MLFFNN


Input
26
Hidden 1
W
b
100
Hidden 2
W
b
100
Output
W
b
5
Output
5

Epoch = 487 Iterations
Performance = 0.0613
Gradient = 9.04e-08

Best Validation Performance is 0.1418 at epoch 255

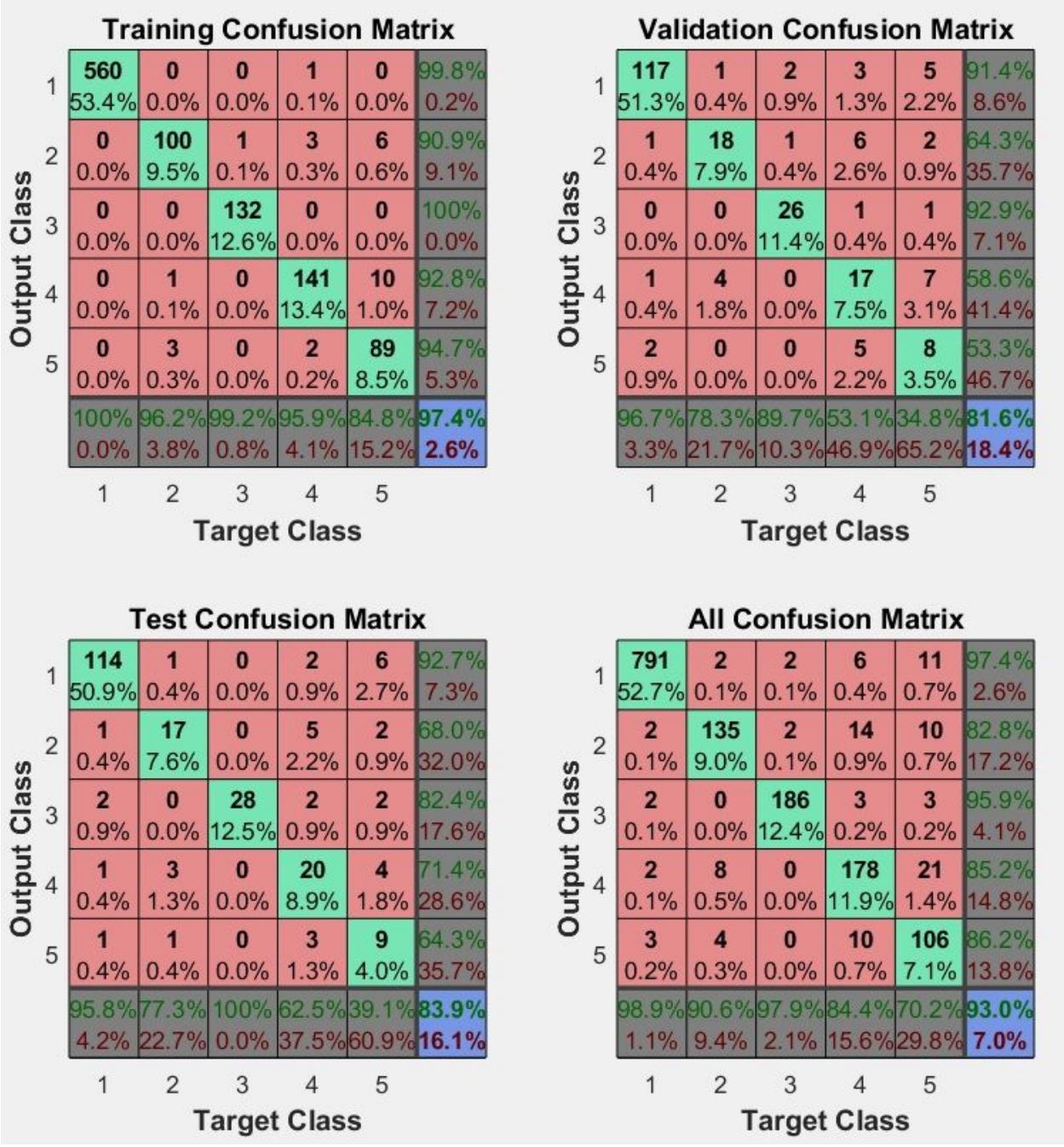
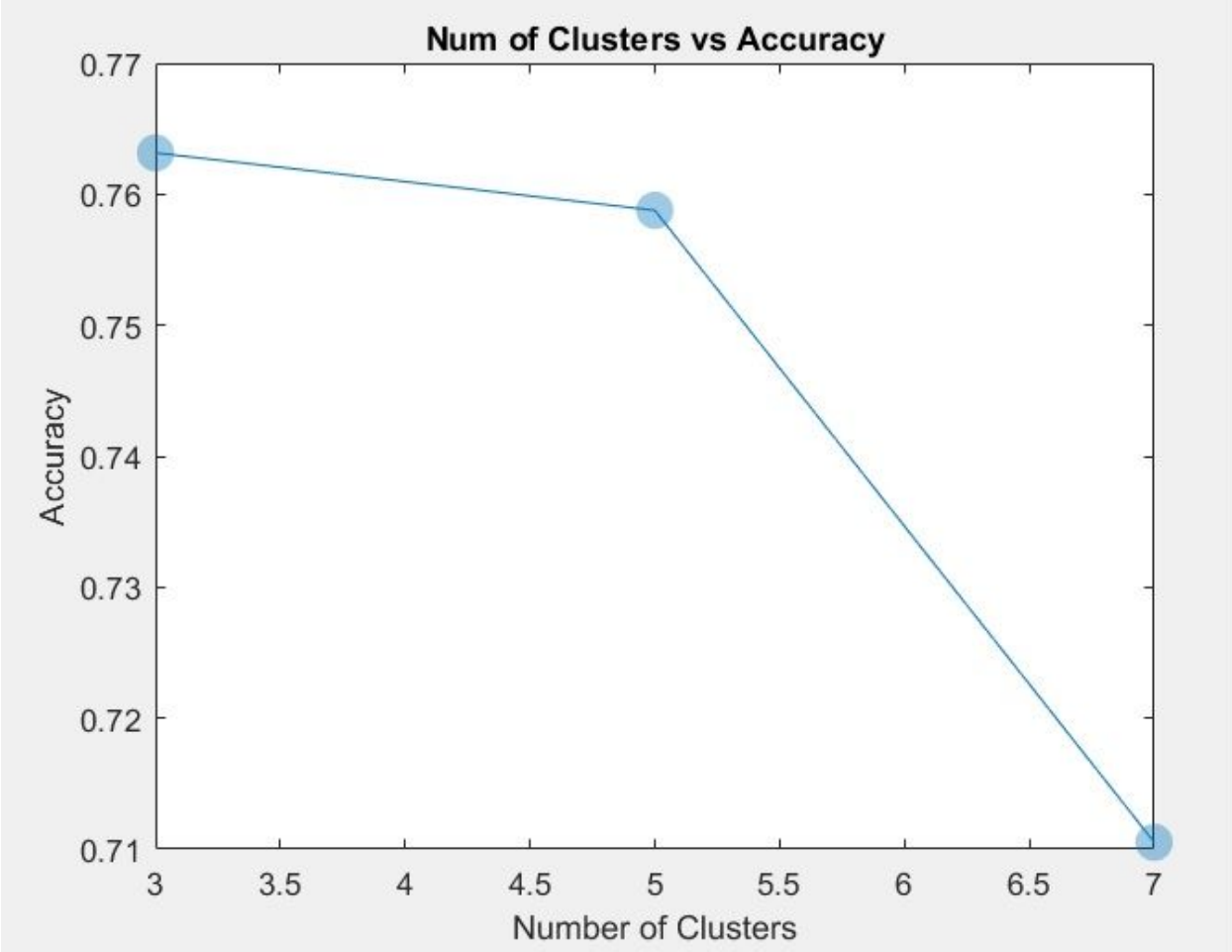
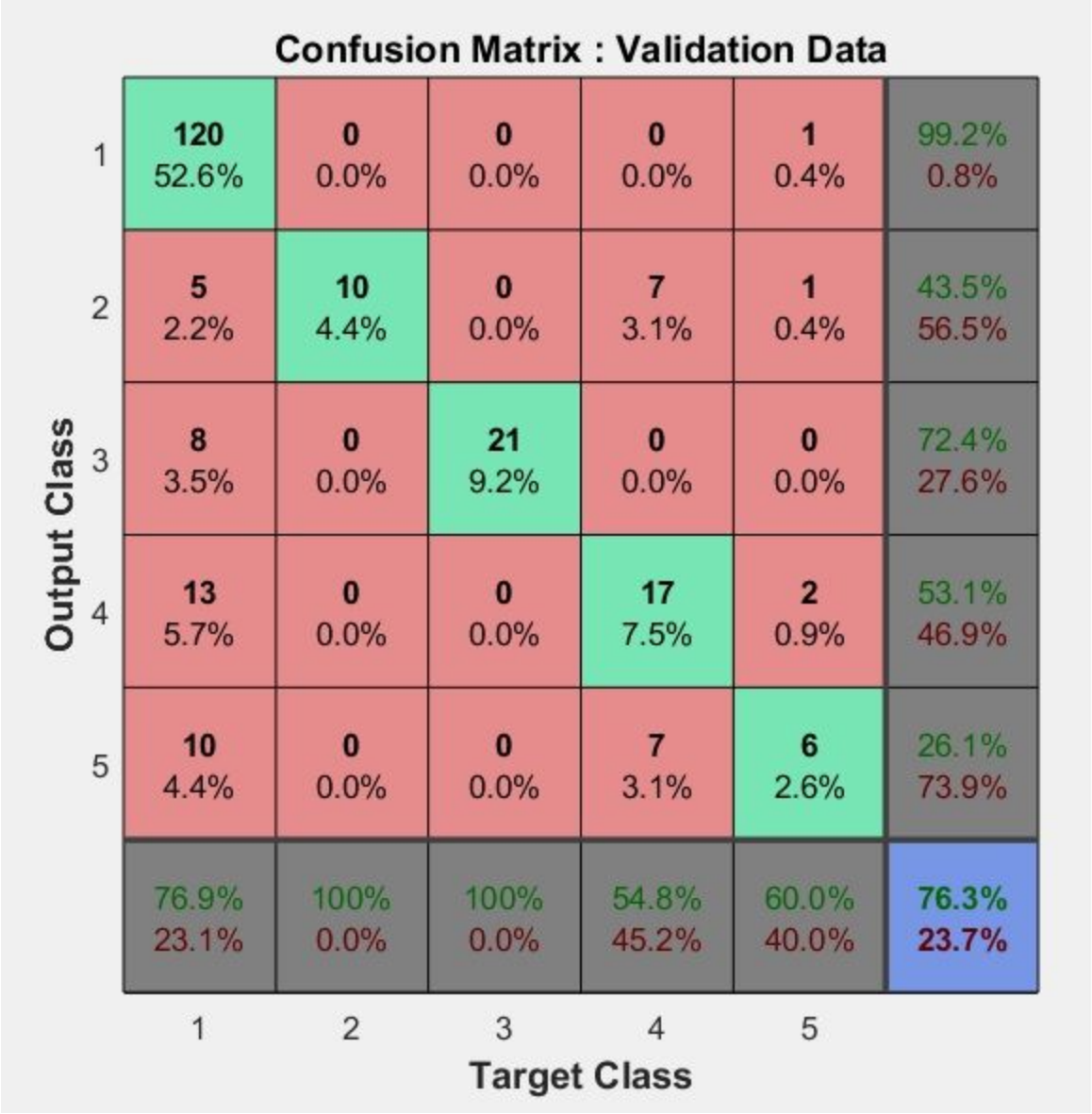
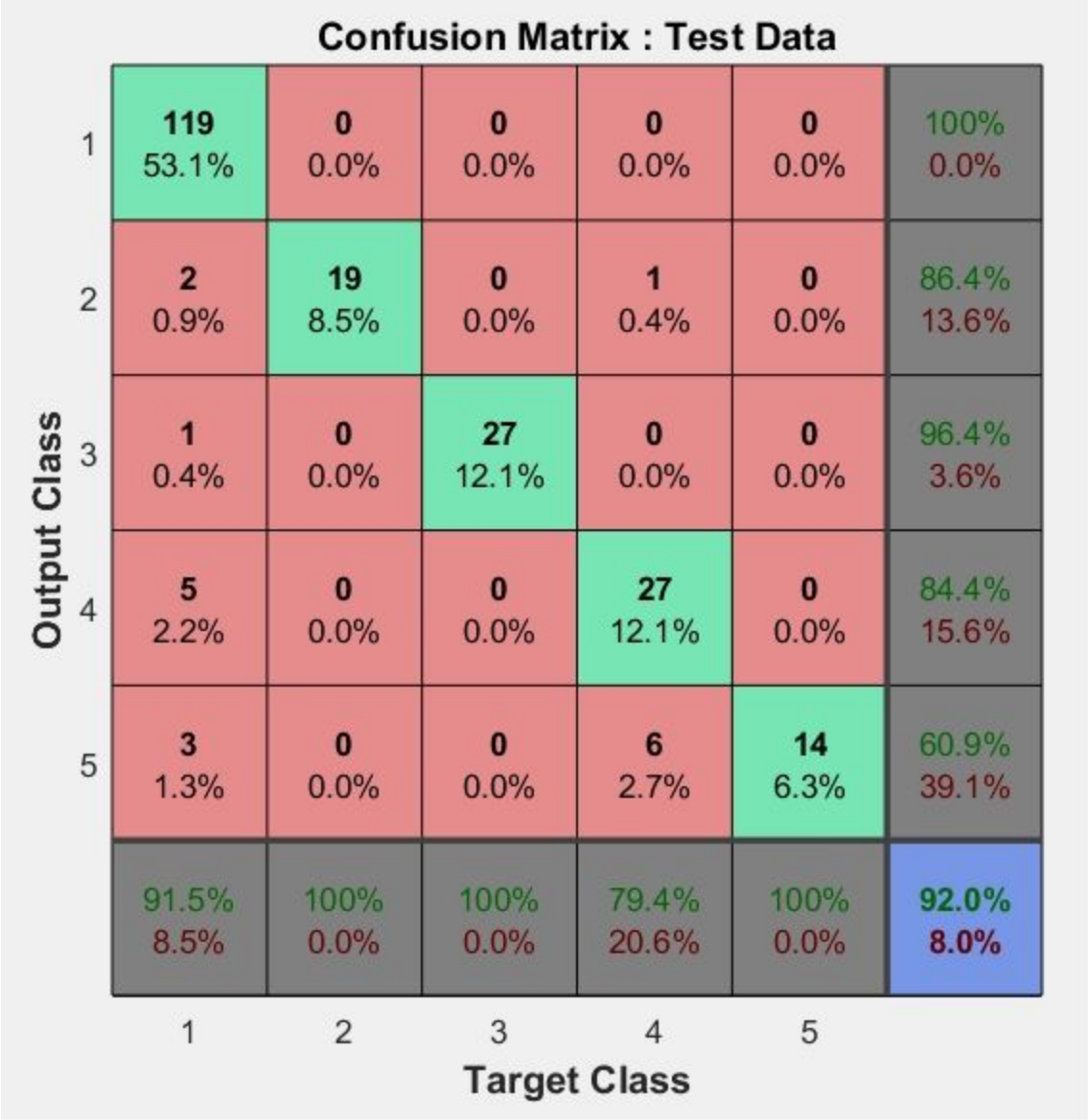


Image Dataset
33 Dimensions
95% Variance Retained

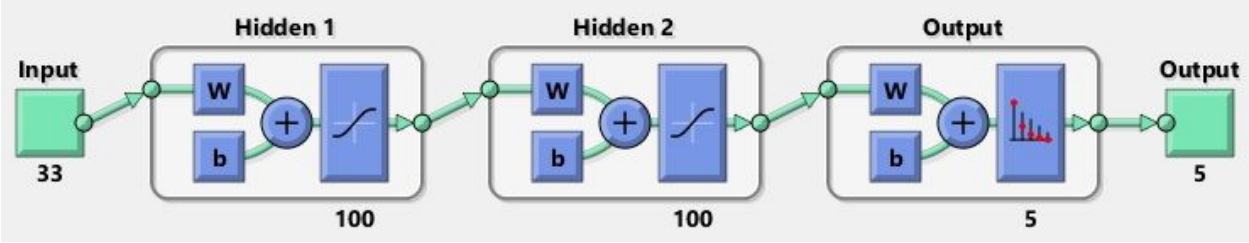
GMM




Confusion Matrix : Validation Data
Output Class
Target Class
1
2
3
4
5
120 52.6%
0 0.0%
0 0.0%
0 0.0%
1 0.4%
99.2% 0.8%
5 2.2%
10 4.4%
0 0.0%
7 3.1%
1 0.4%
43.5% 56.5%
8 3.5%
0 0.0%
21 9.2%
0 0.0%
0 0.0%
72.4% 27.6%
13 5.7%
0 0.0%
0 0.0%
17 7.5%
2 0.9%
53.1% 46.9%
10 4.4%
0 0.0%
0 0.0%
7 3.1%
6 2.6%
26.1% 73.9%
76.9% 23.1%
100% 0.0%
100% 0.0%
54.8% 45.2%
60.0% 40.0%
76.3% 23.7%



MLFFNN



Epoch = 452 Iterations

Performance = 0.0630
Gradient = 9.45e-07

Best Validation Performance is 0.155 at epoch 245

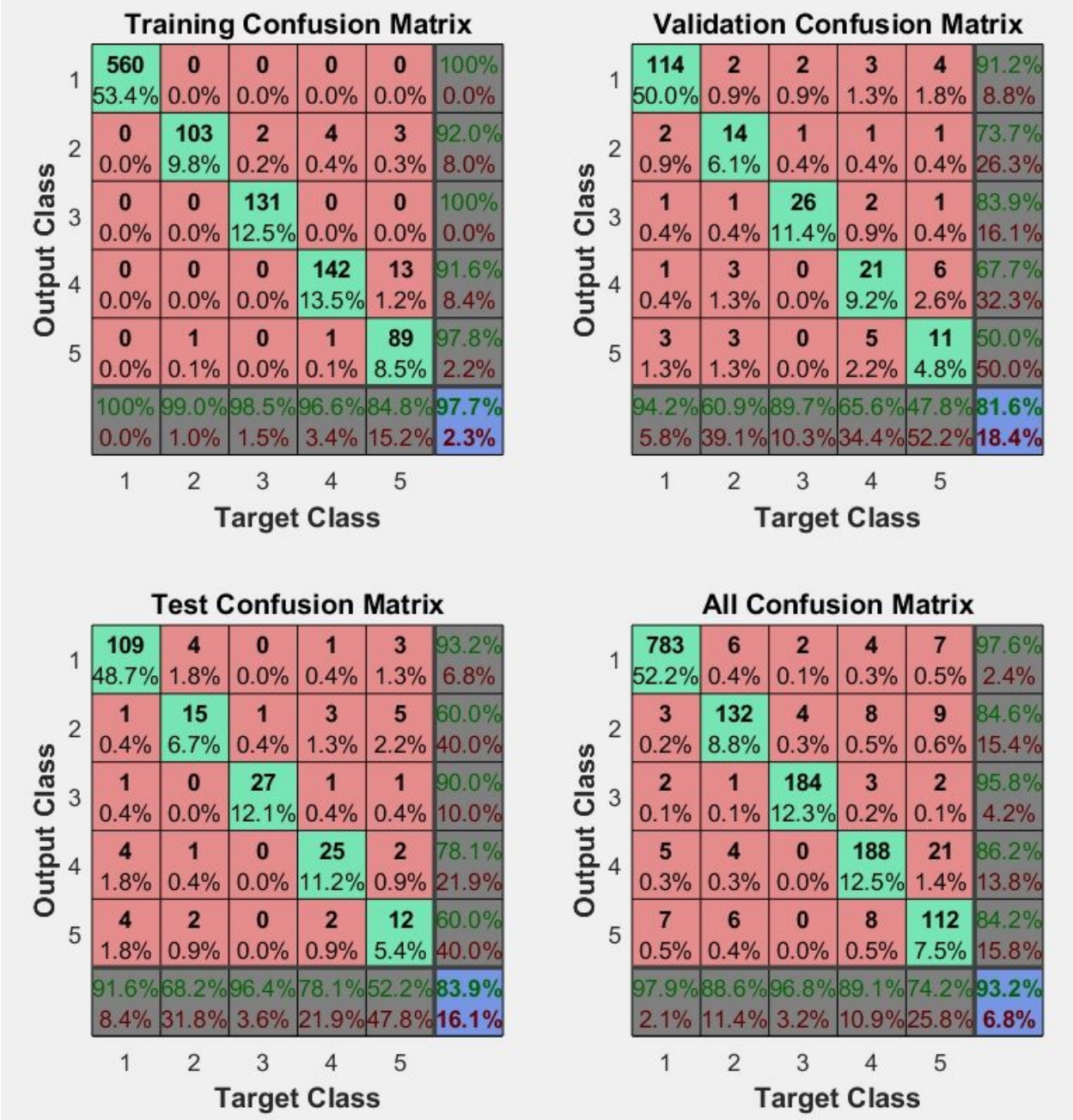
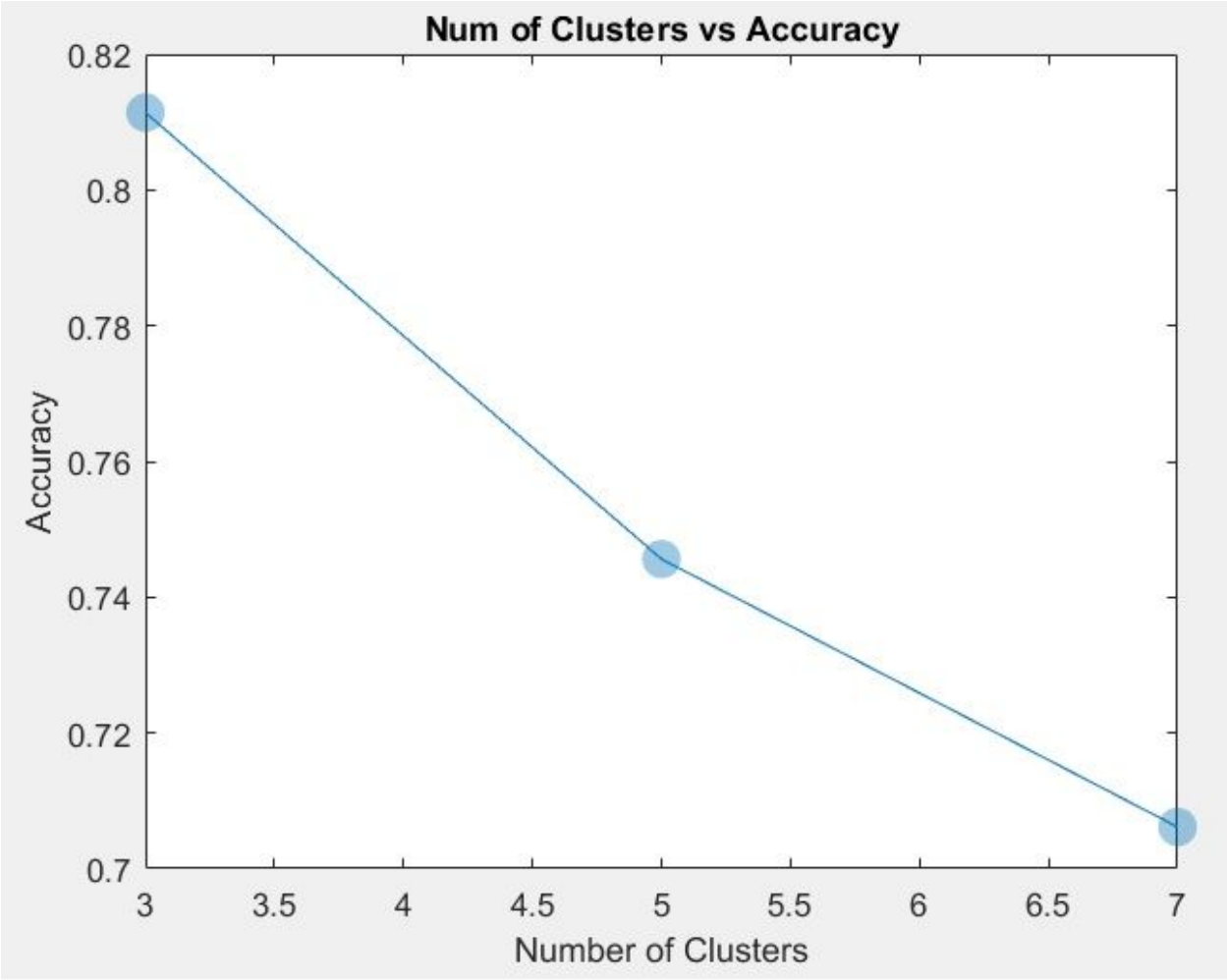
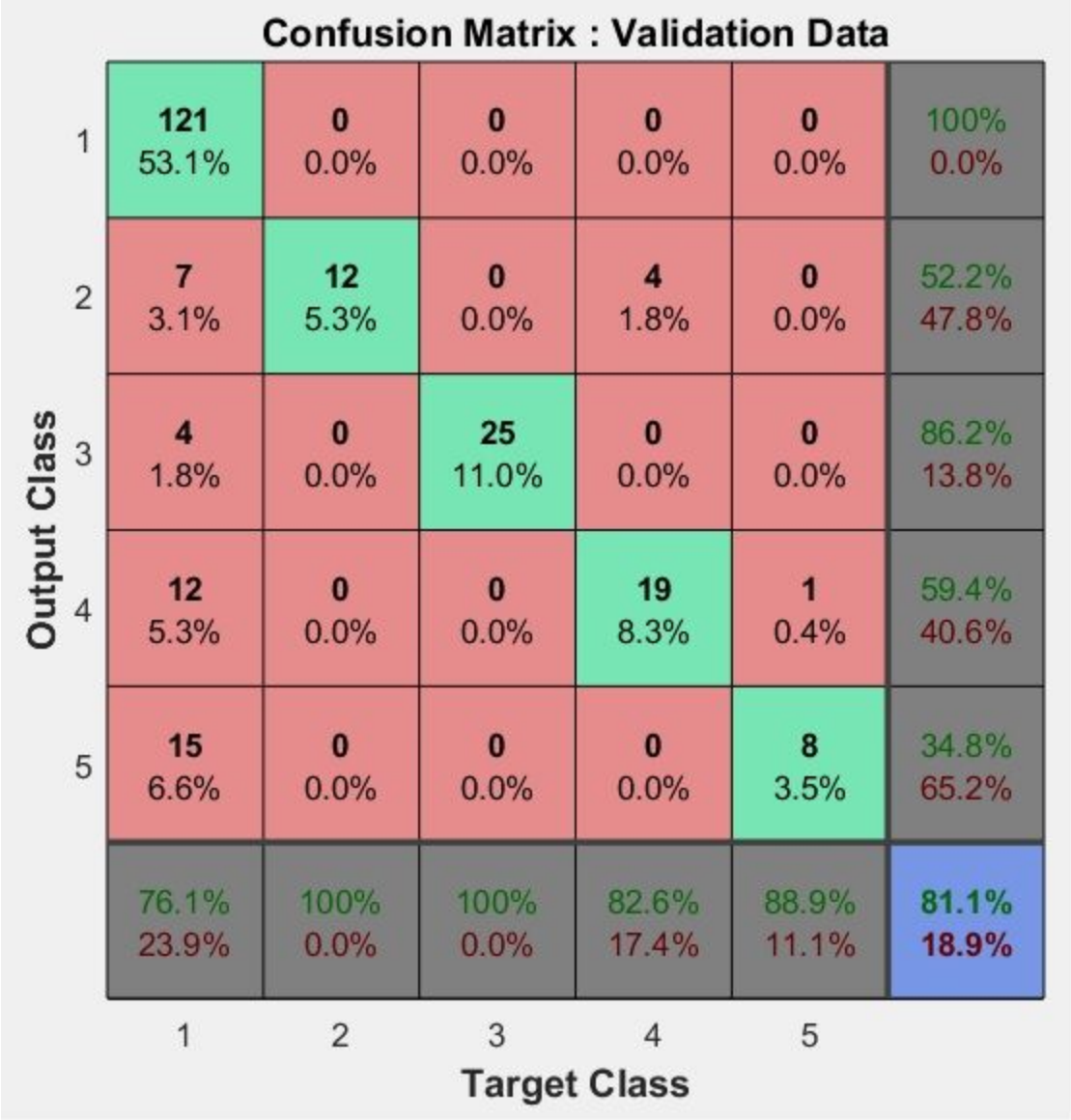
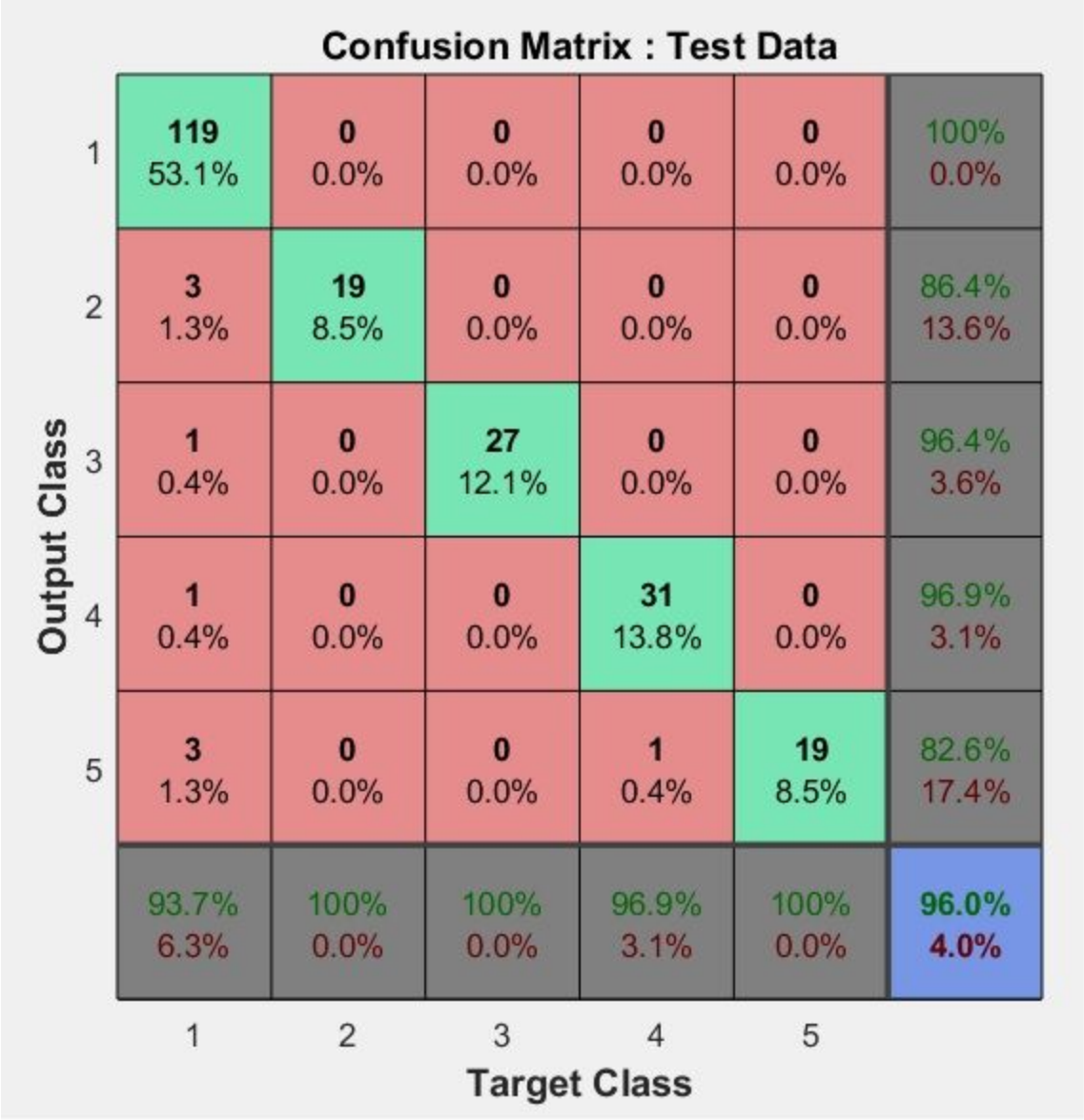


Image Dataset
43 Dimensions
99% Variance Retained

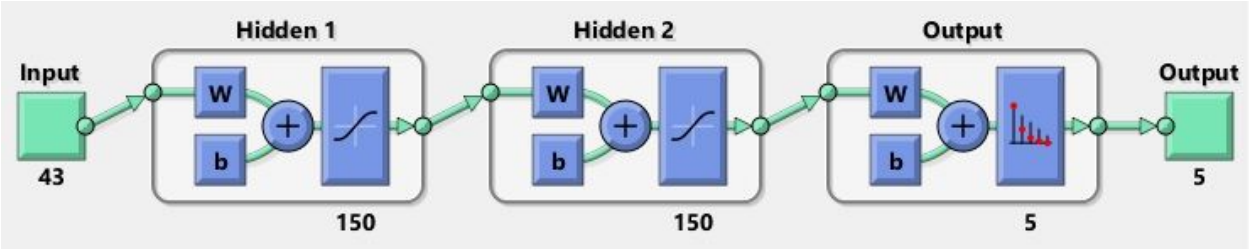
GMM




Confusion Matrix : Validation Data
Output Class
Target Class
1
2
3
4
5
121
53.1%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
7
3.1%
12
5.3%
0
0.0%
4
1.8%
0
0.0%
52.2%
47.8%
4
1.8%
0
0.0%
25
11.0%
0
0.0%
0
0.0%
86.2%
13.8%
12
5.3%
0
0.0%
0
0.0%
19
8.3%
1
0.4%
59.4%
40.6%
15
6.6%
0
0.0%
0
0.0%
0
0.0%
8
3.5%
34.8%
65.2%
76.1%
23.9%
100%
0.0%
100%
0.0%
82.6%
17.4%
88.9%
11.1%
81.1%
18.9%


Confusion Matrix : Test Data
Output Class
Target Class
1
2
3
4
5
119
53.1%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
3
1.3%
19
8.5%
0
0.0%
0
0.0%
0
0.0%
86.4%
13.6%
1
0.4%
0
0.0%
27
12.1%
0
0.0%
0
0.0%
96.4%
3.6%
1
0.4%
0
0.0%
0
0.0%
31
13.8%
0
0.0%
96.9%
3.1%
3
1.3%
0
0.0%
0
0.0%
1
0.4%
19
8.5%
82.6%
17.4%
93.7%
6.3%
100%
0.0%
100%
0.0%
96.9%
3.1%
100%
0.0%
96.0%
4.0%

MLFFNN


Input
43
Hidden 1
W
b
150
Hidden 2
W
b
150
Output
W
b
5
Output
5

Epoch = 400 Iterations
Performance = 0.0596
Gradient = 8.92e-07

Best Validation Performance is 0.15478 at epoch 231

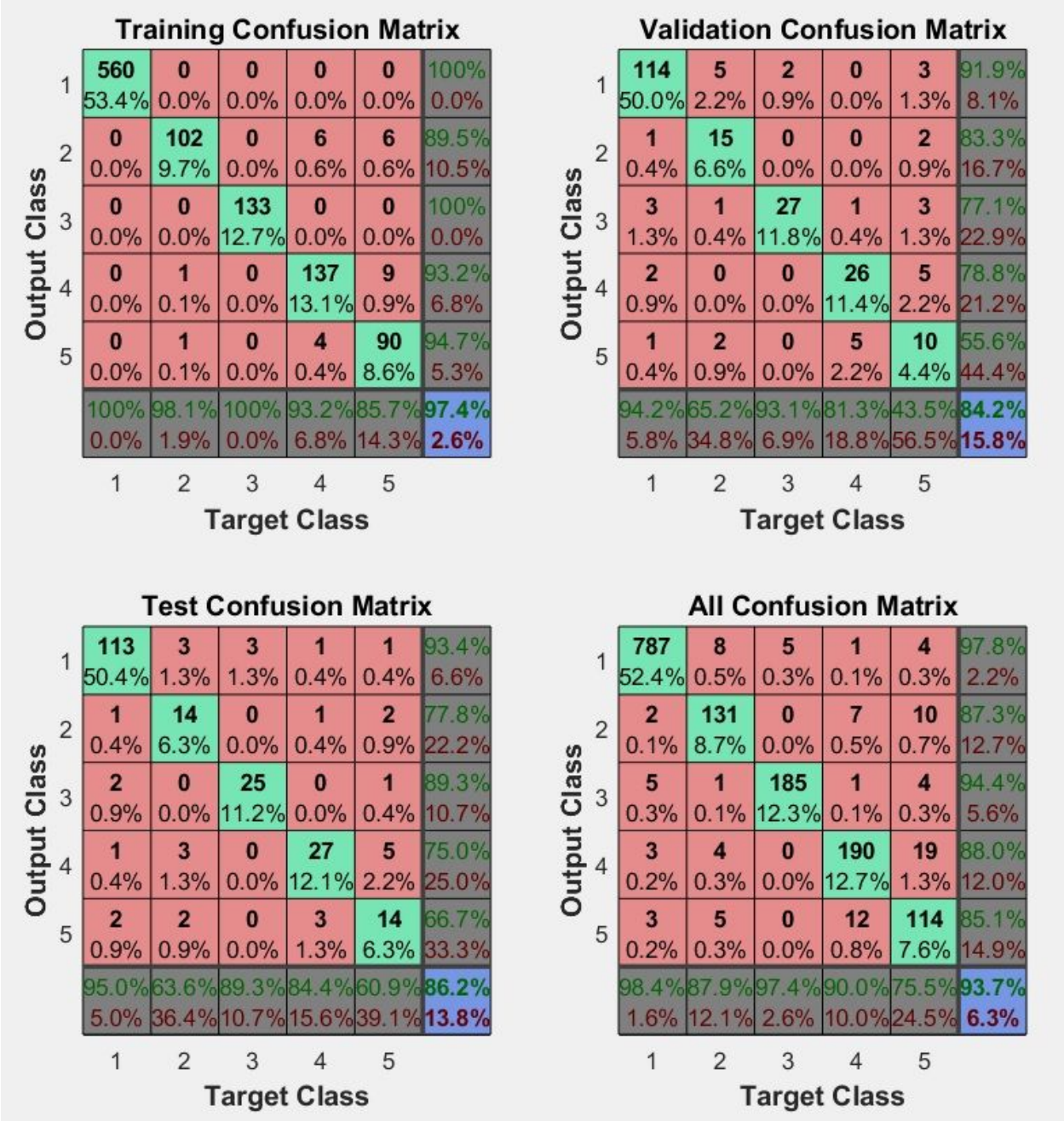
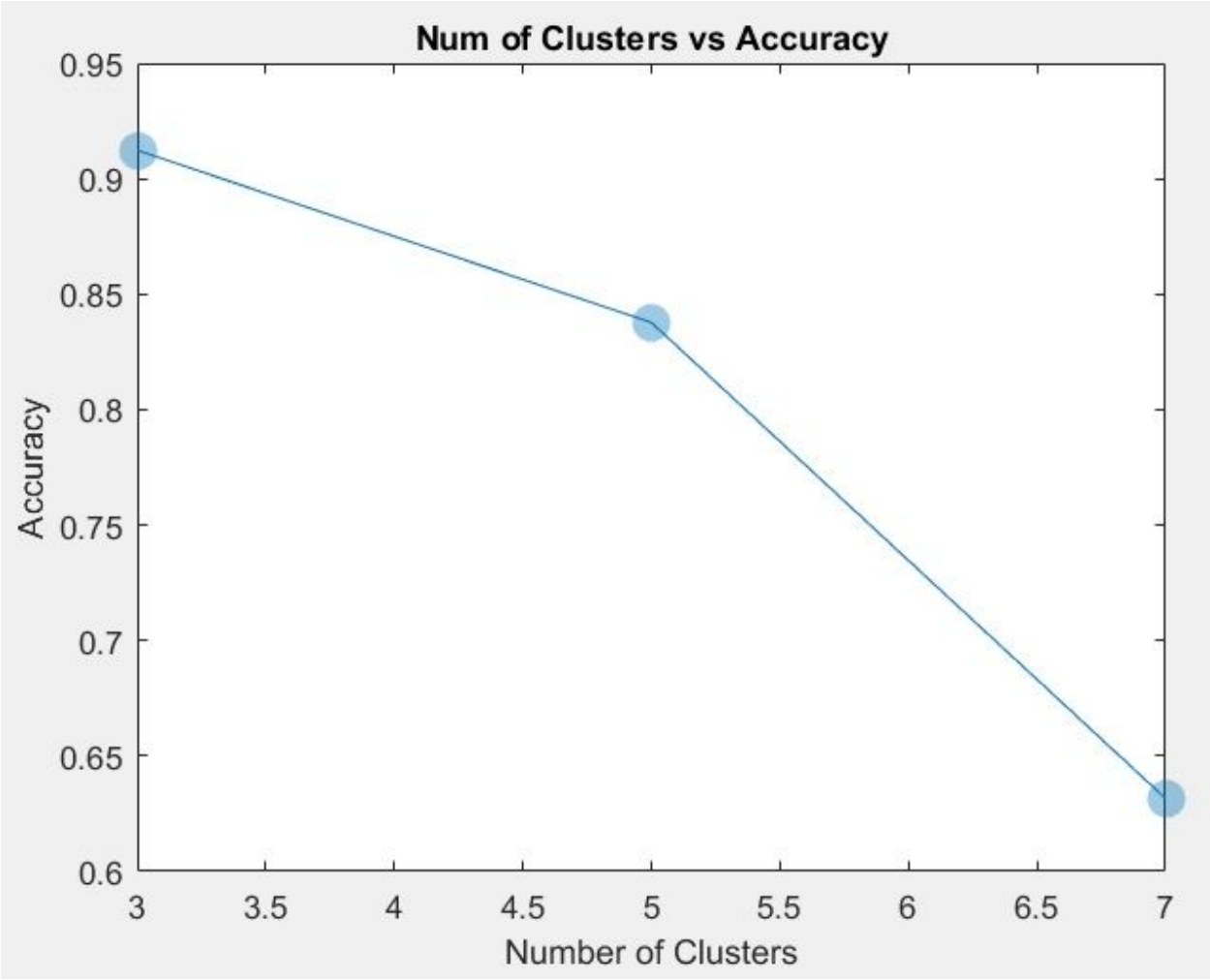
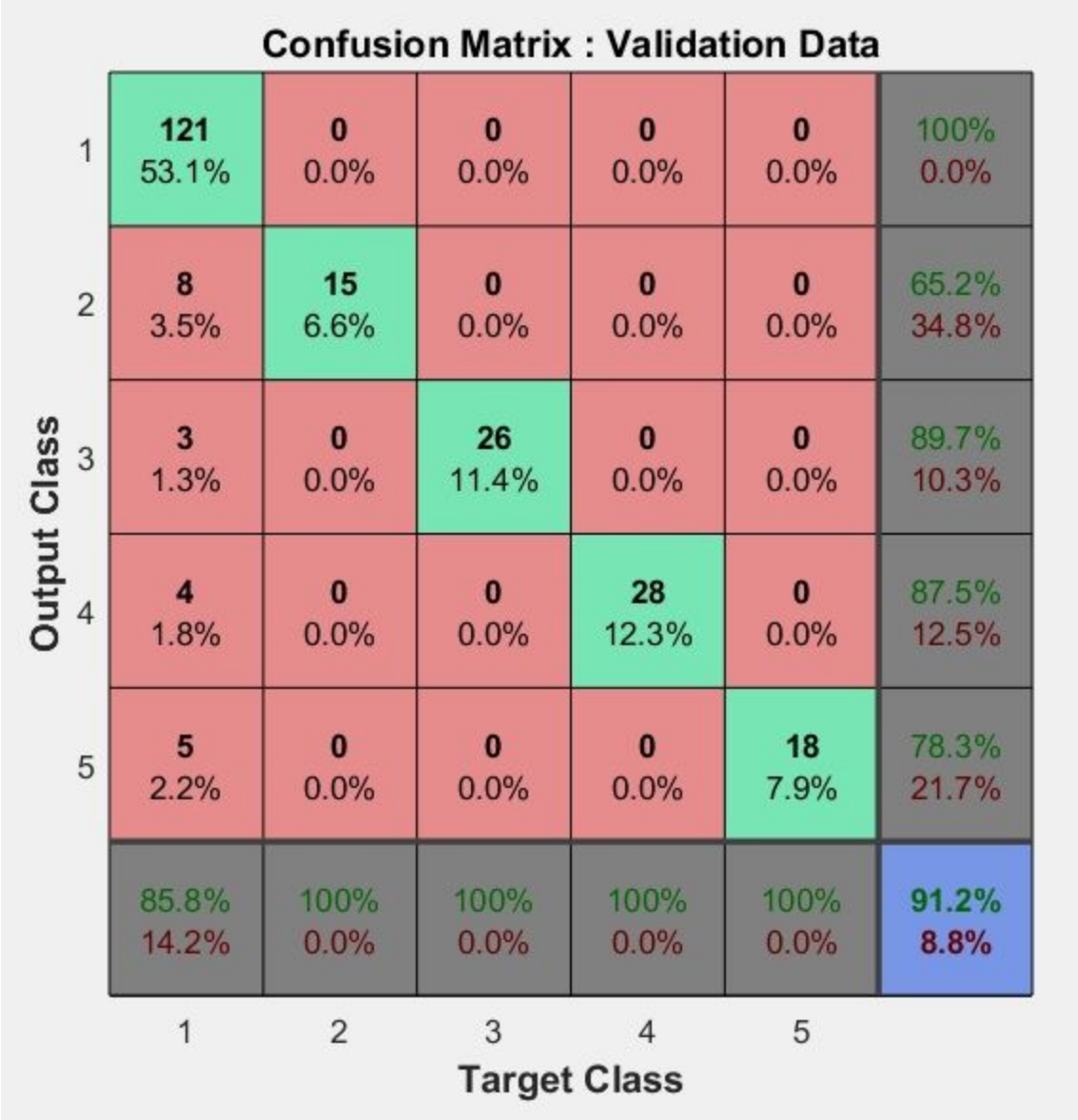
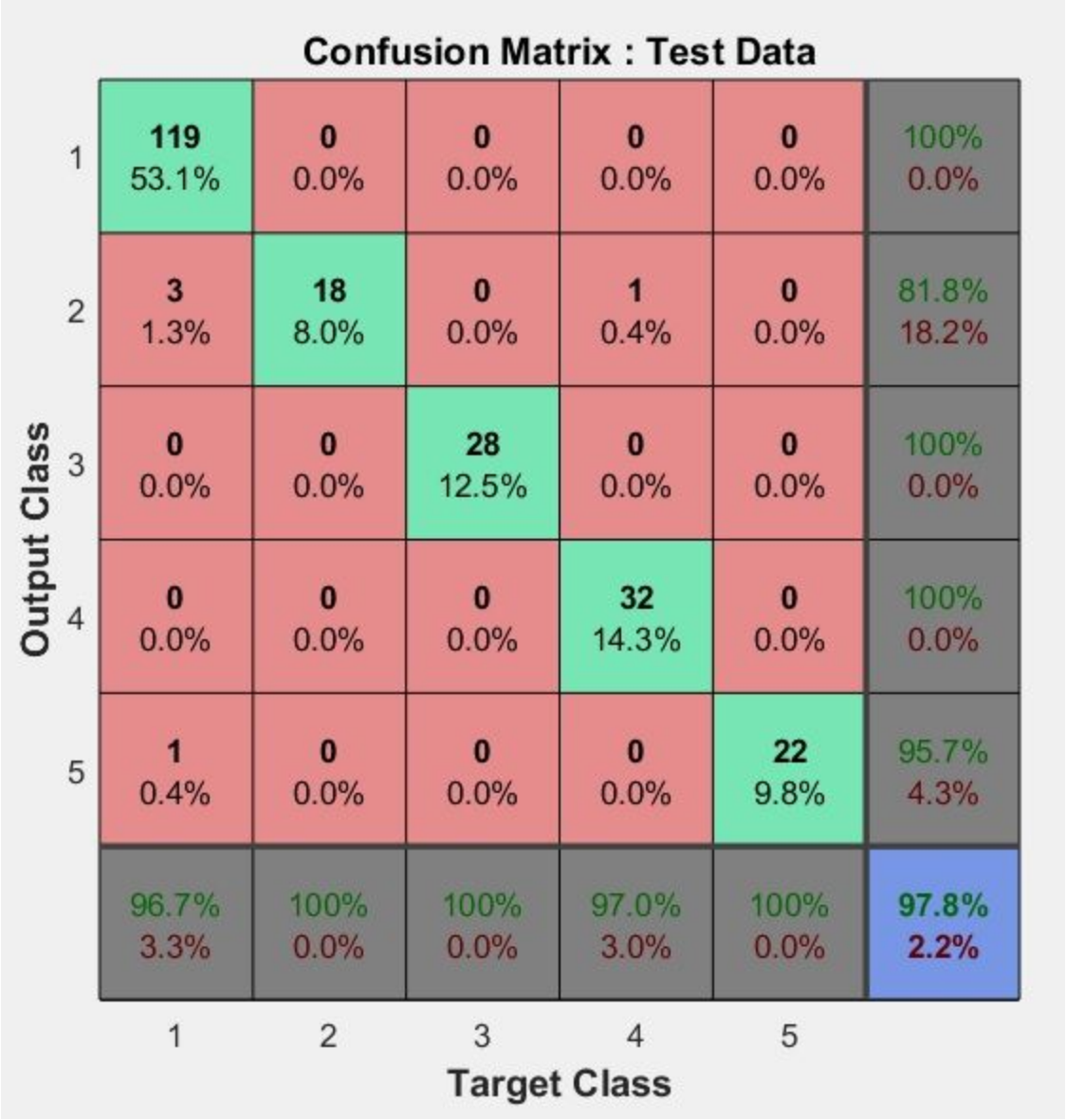


Image Dataset
48 Dimensions
Original Dataset

GMM



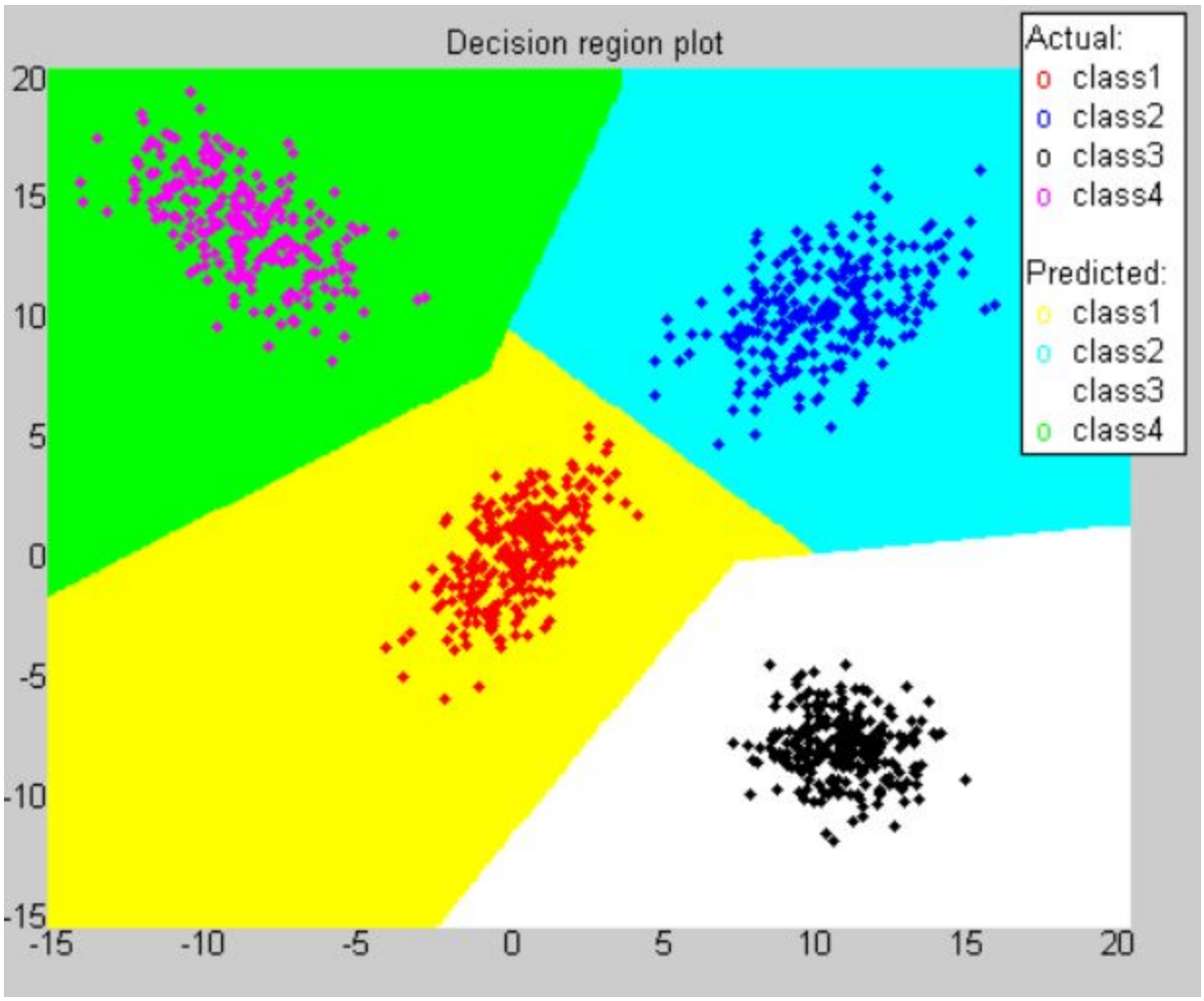

Confusion Matrix : Validation Data
Output Class
Target Class
1
2
3
4
5
121
53.1%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
8
3.5%
15
6.6%
0
0.0%
0
0.0%
0
0.0%
65.2%
34.8%
3
1.3%
0
0.0%
26
11.4%
0
0.0%
0
0.0%
89.7%
10.3%
4
1.8%
0
0.0%
0
0.0%
28
12.3%
0
0.0%
87.5%
12.5%
5
2.2%
0
0.0%
0
0.0%
0
0.0%
18
7.9%
78.3%
21.7%
85.8%
14.2%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
91.2%
8.8%

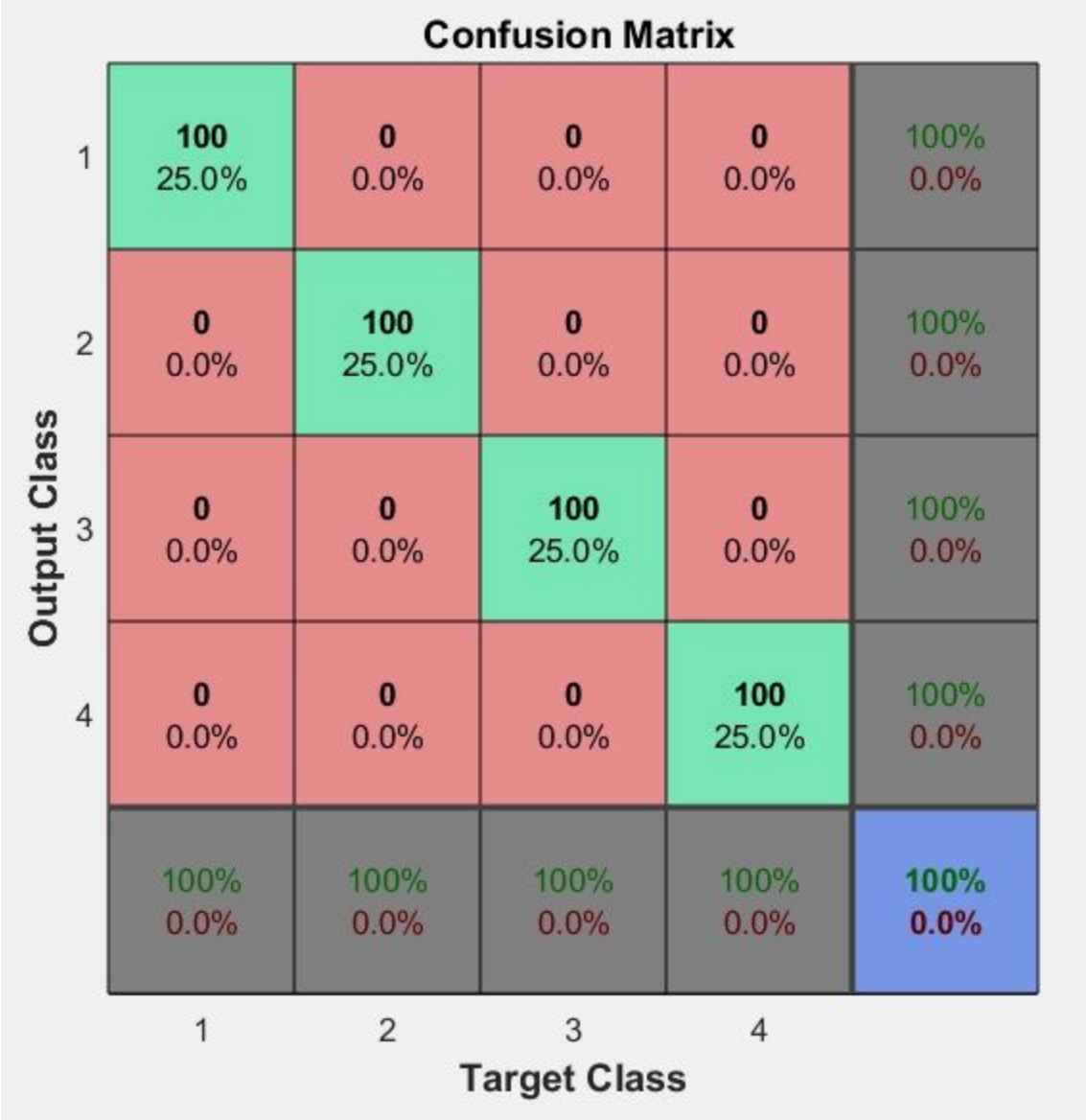

Confusion Matrix : Test Data
Output Class
Target Class
1
2
3
4
5
119
53.1%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
3
1.3%
18
8.0%
0
0.0%
1
0.4%
0
0.0%
81.8%
18.2%
0
0.0%
0
0.0%
28
12.5%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
32
14.3%
0
0.0%
100%
0.0%
1
0.4%
0
0.0%
0
0.0%
0
0.0%
22
9.8%
95.7%
4.3%
96.7%
3.3%
100%
0.0%
100%
0.0%
97.0%
3.0%
100%
0.0%
97.8%
2.2%

Data-set 1

Linear C-SVM on Linearly Separable Data

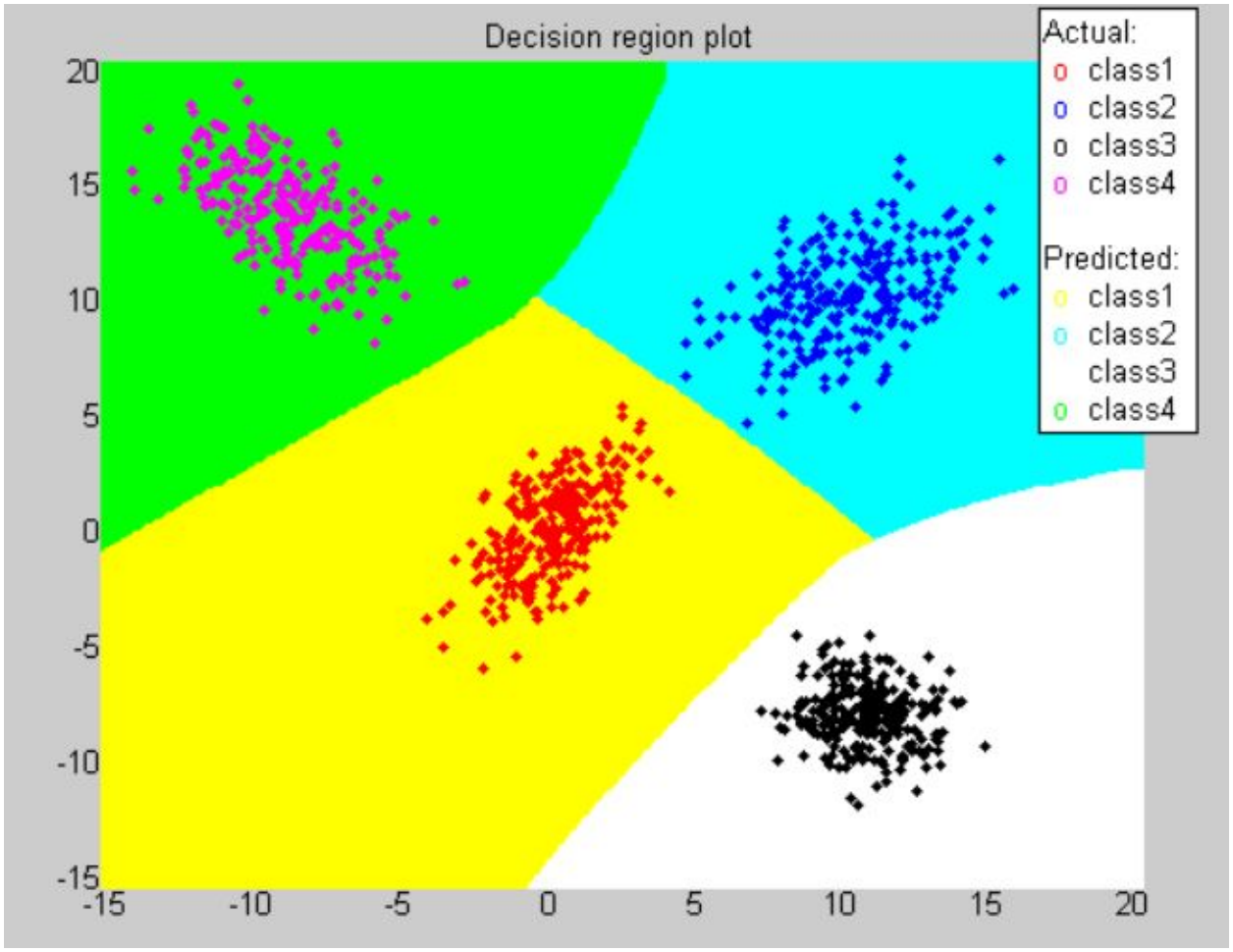
Decision Surface takes form of Hyper-planes

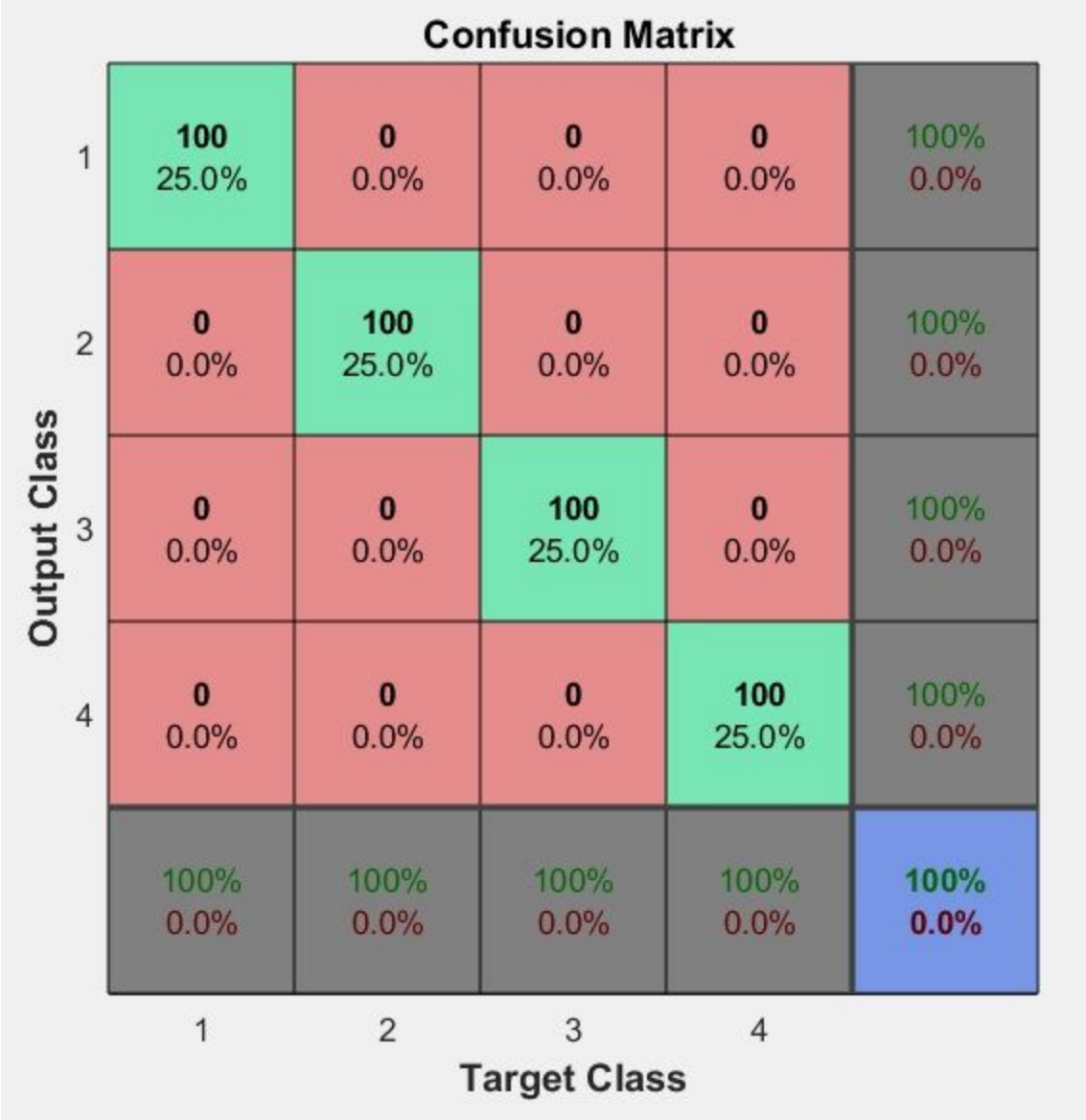



Confusion Matrix
Output Class
Target Class
1
2
3
4
100
25.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
100
25.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
100
25.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
100
25.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%

Polynomial Kernel

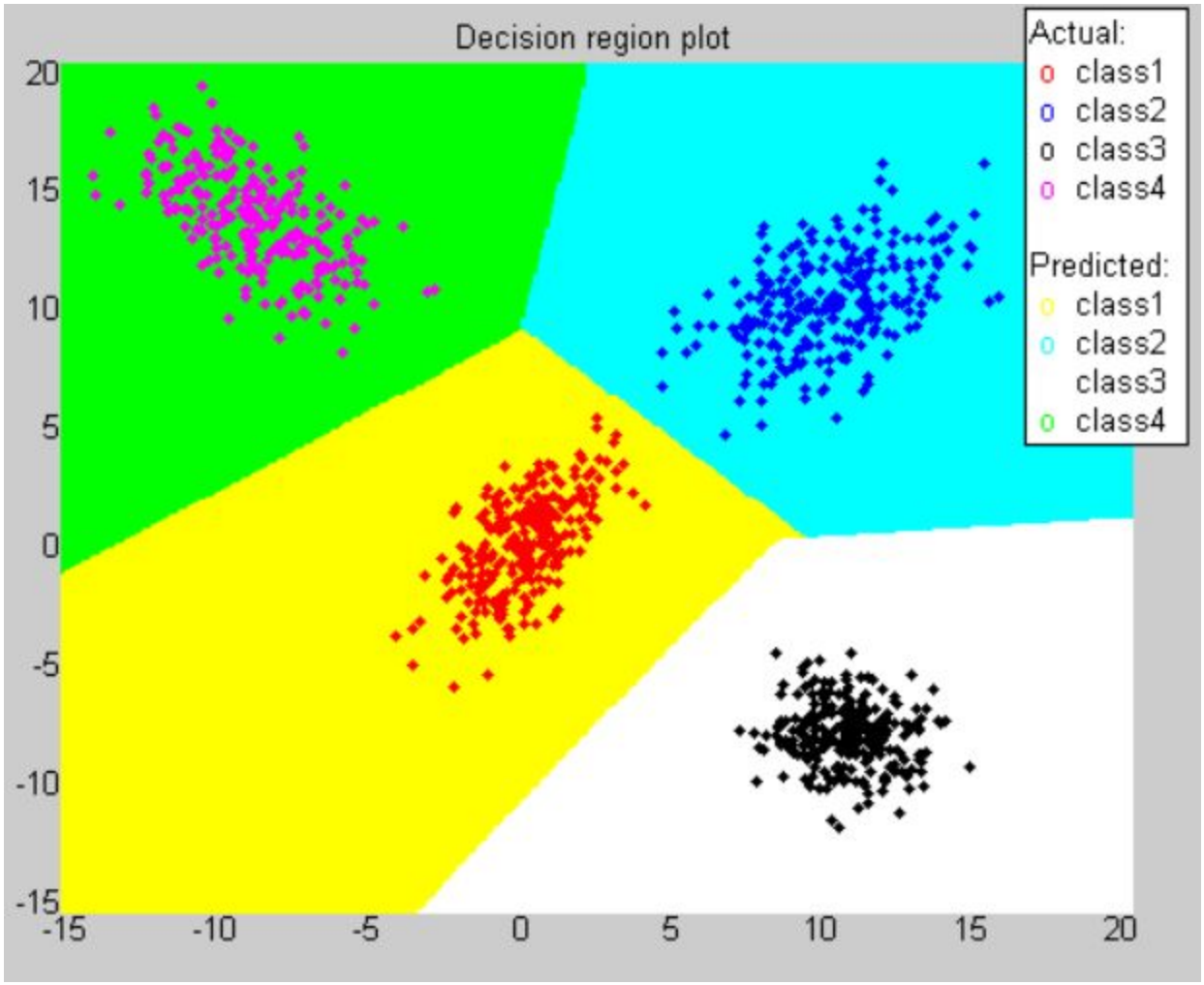
Degree = 3 Cost = 64

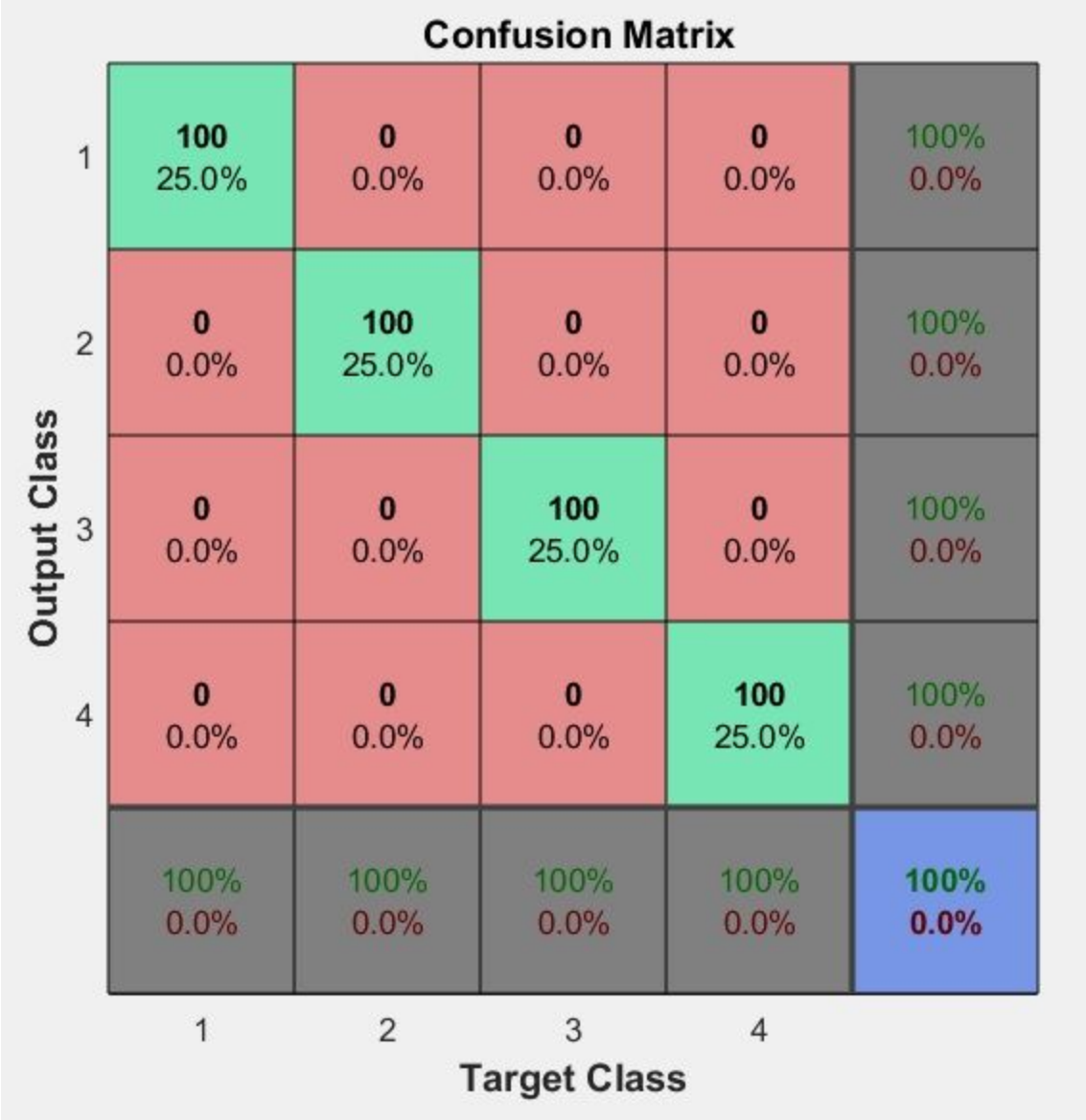



Confusion Matrix
Output Class
Target Class
1
2
3
4
100
25.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
100
25.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
100
25.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
100
25.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%

RBF

Gamma = 0.25 Cost =0.125



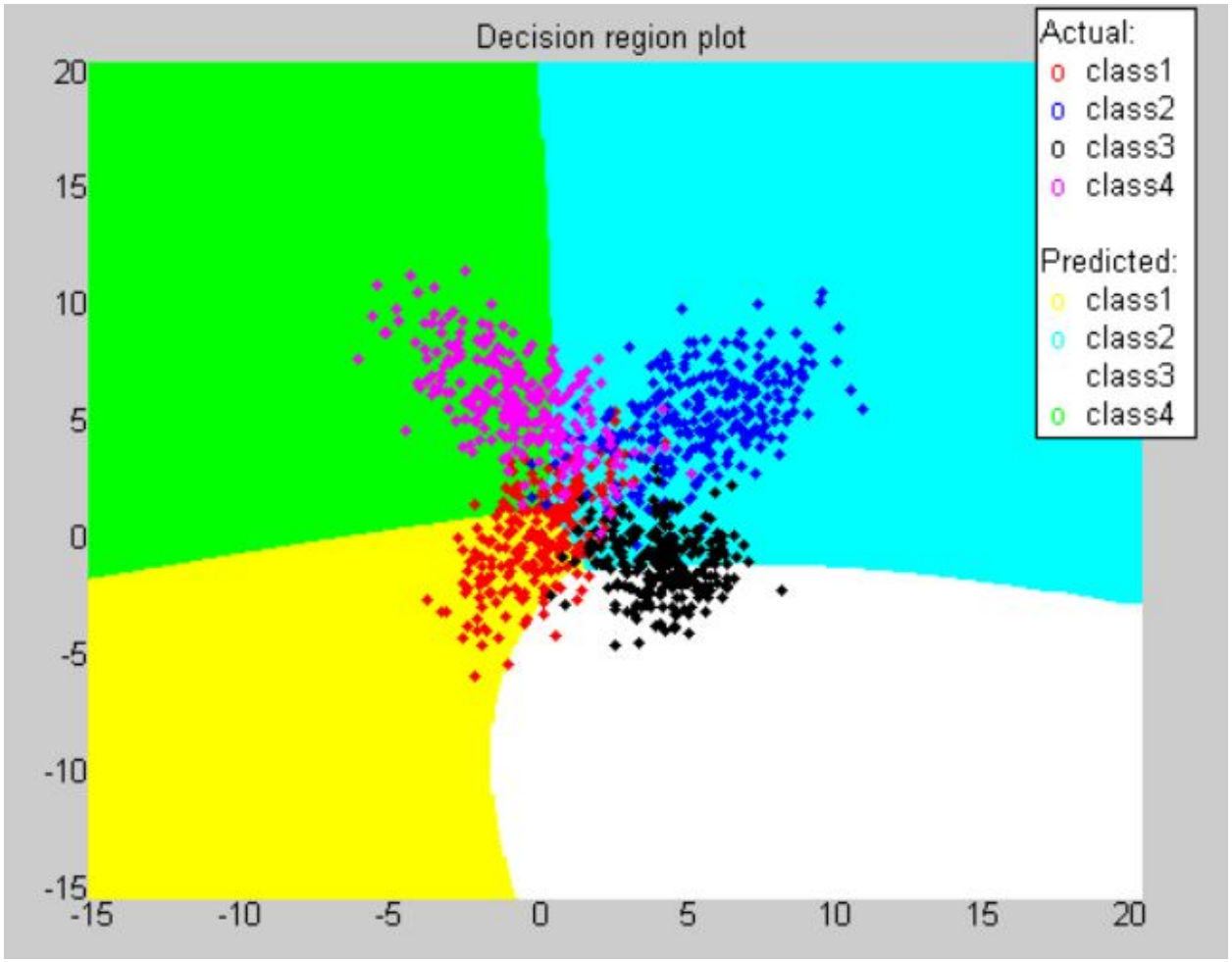

Confusion Matrix
Output Class
Target Class
1
2
3
4
100
25.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
100
25.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
100
25.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
100
25.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%

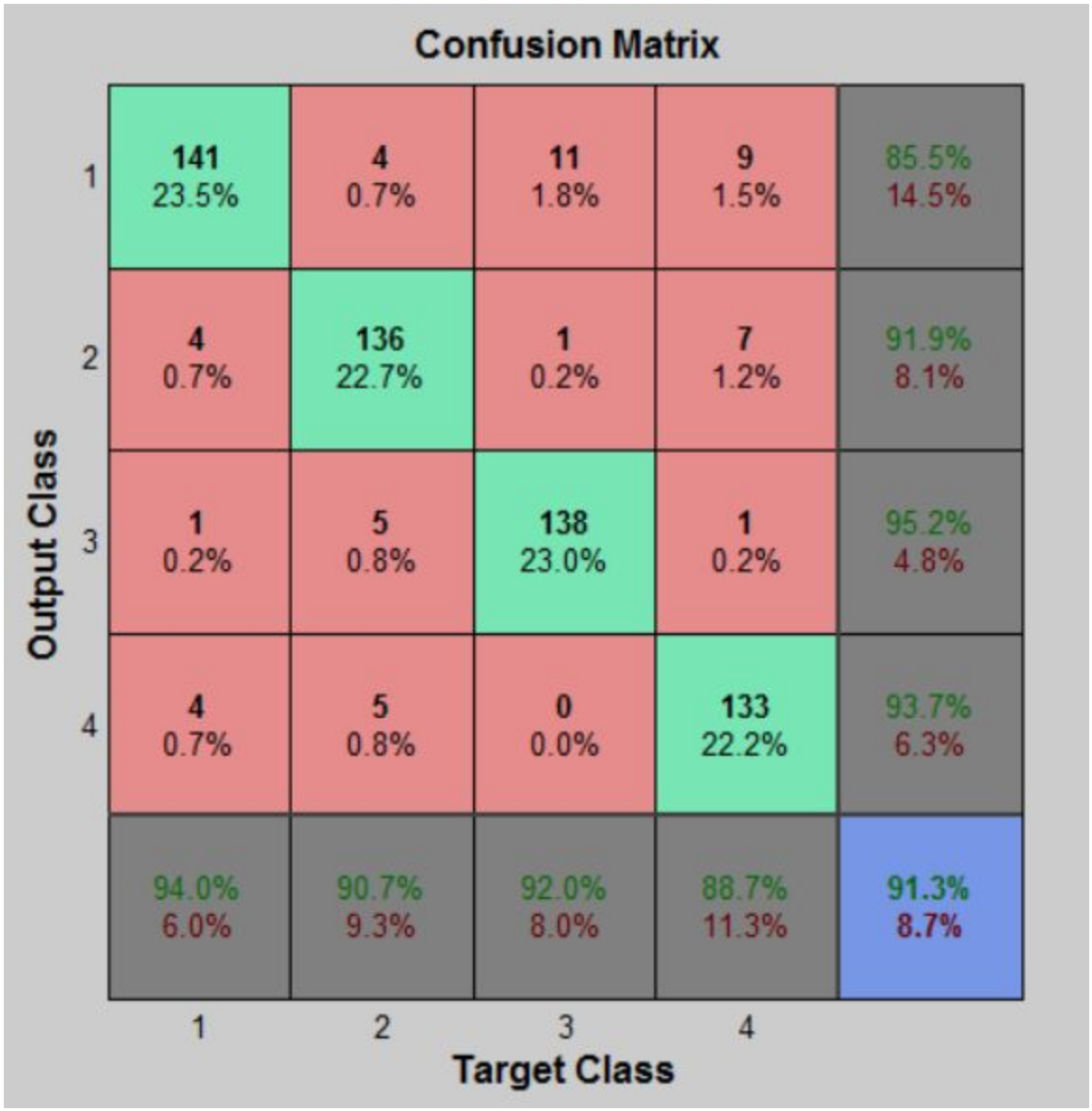
Overlapping Data

Polynomial Function

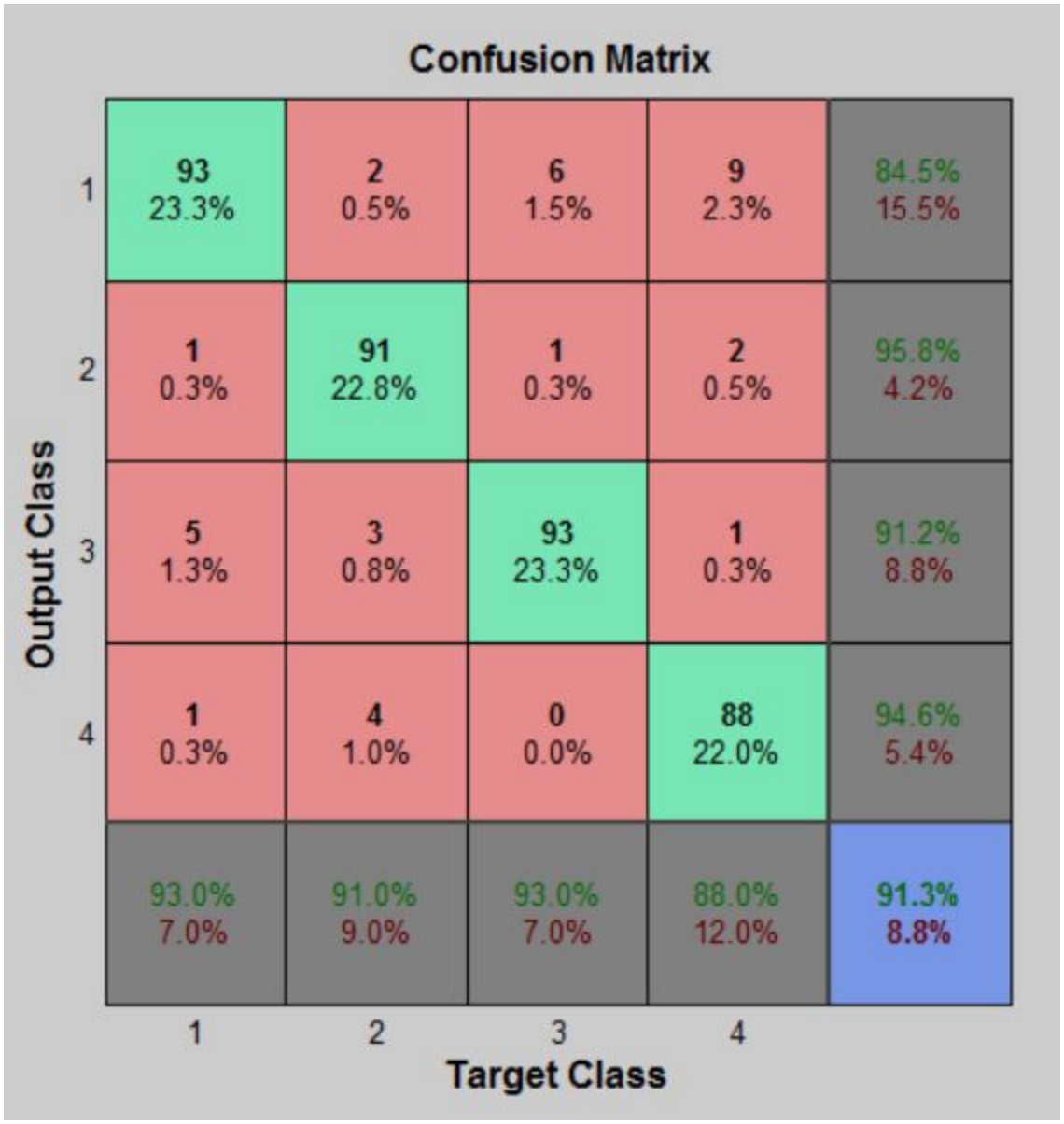
Degree=2 C=64 Coeff=1

Training Data : Accuracy = 89.6% (896/1000) (classification)
Validation Data : Accuracy = 91.3333% (548/600) (classification)
Test Data Accuracy = 91.25% (365/400) (classification)



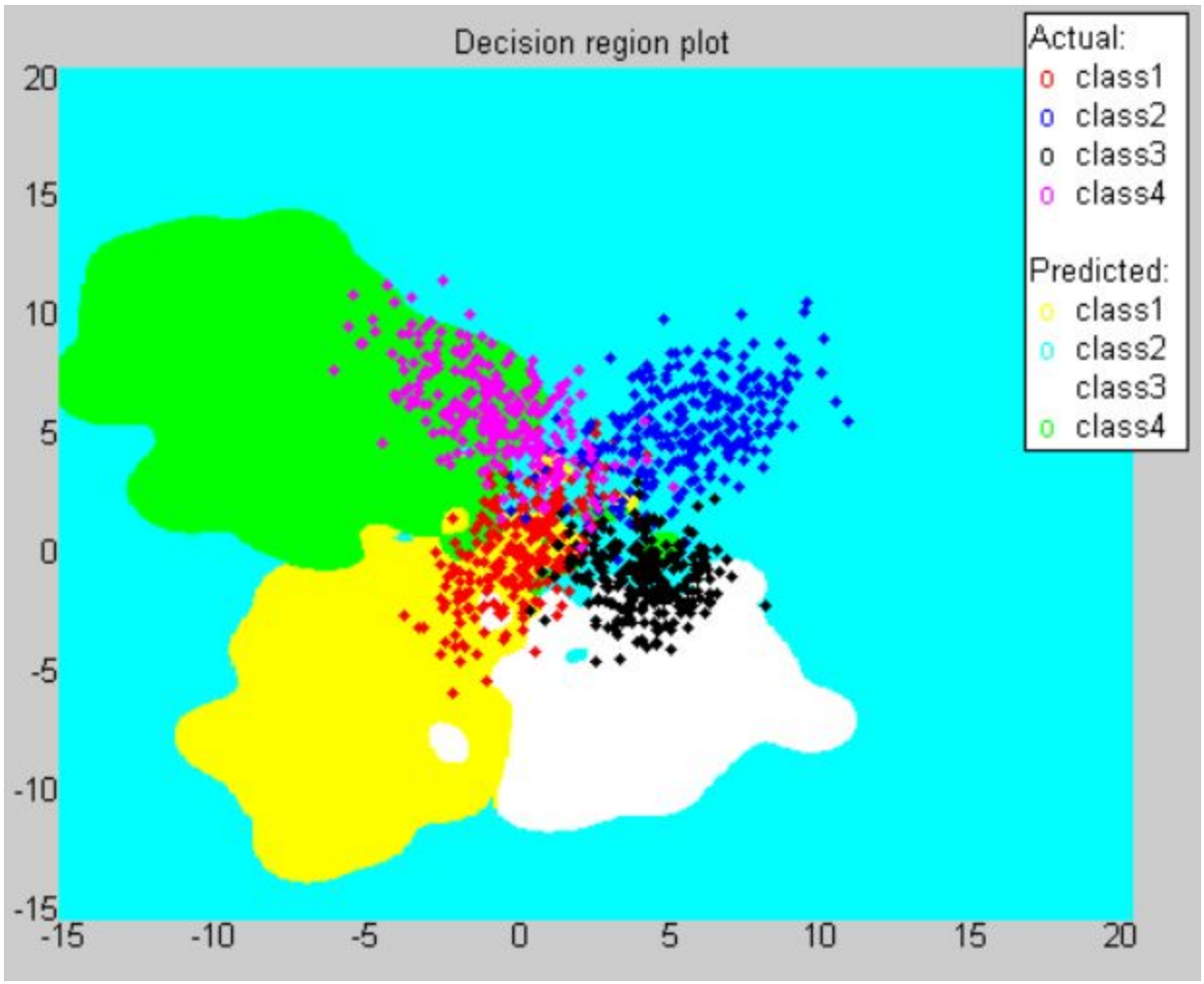


Confusion Matrix For Validation Data

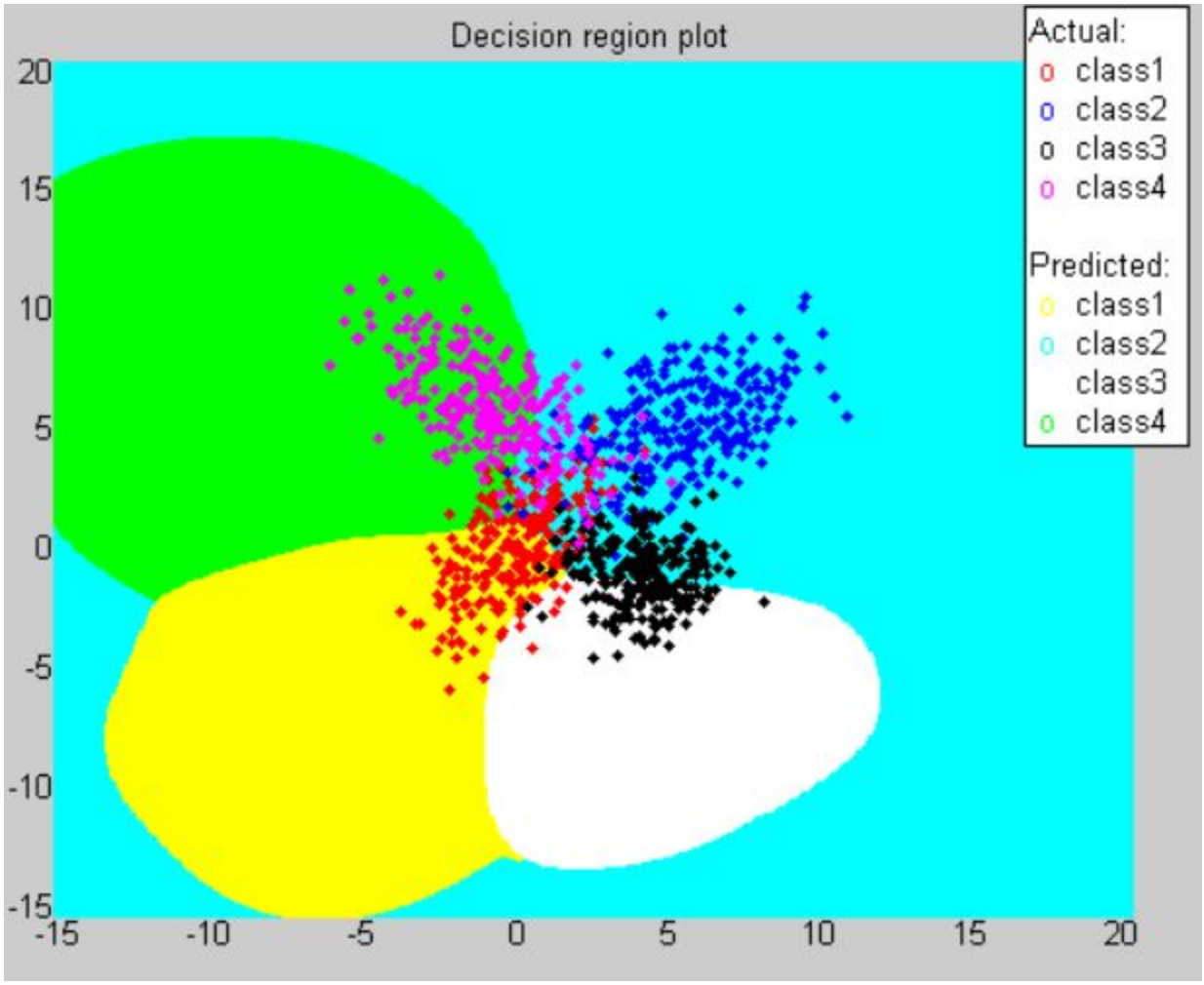


Confusion Matrix for Testing Data

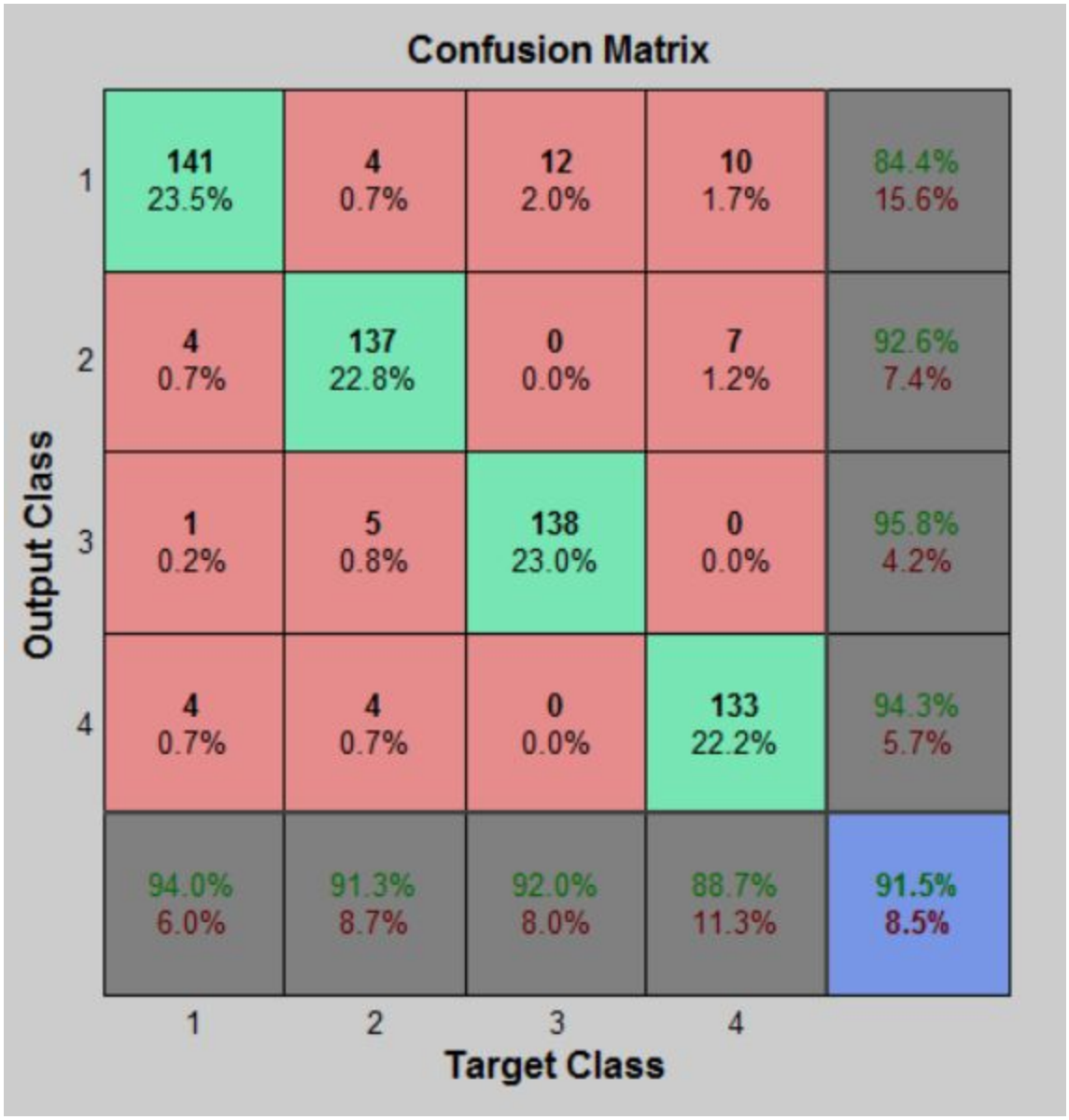
RBF



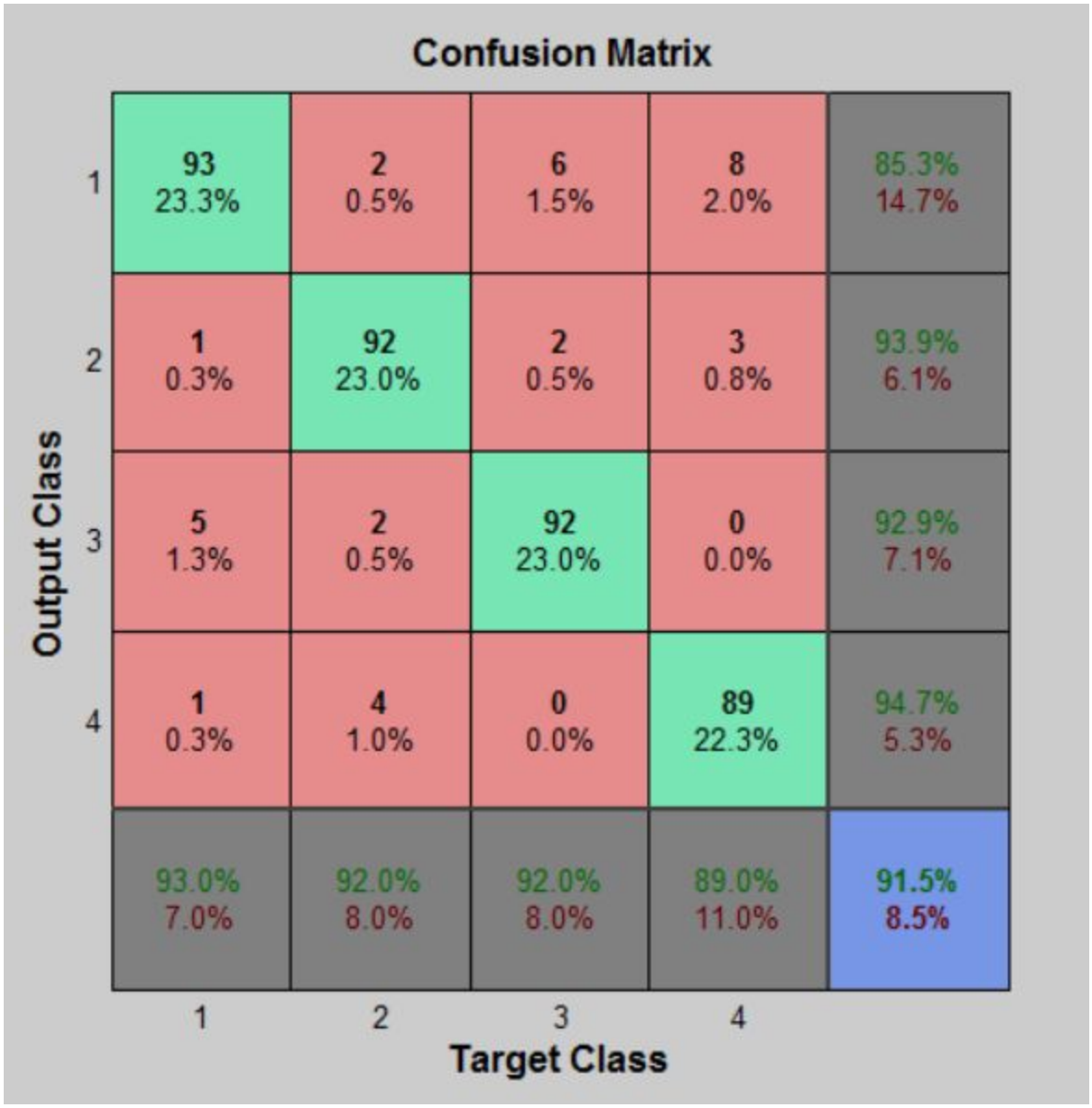
Gamma = 128 Cost =16
Training Data Accuracy = 93.6% (936/1000) (classification)(To check for Over-fitting)
Validation Data Accuracy = 88.1667% (529/600) (classification)
Testing Data Accuracy = 87.25% (349/400) (classification)



Gamma = 16 Cost = 0.25
Training Data Accuracy = 89.7% (897/1000) (classification)(To check for Over-fiting)
Validation Data Accuracy = 91.5% (549/600) (classification)
Testing Data Accuracy = 91.5% (366/400) (classification)



Confusion Matrix for Validation Data

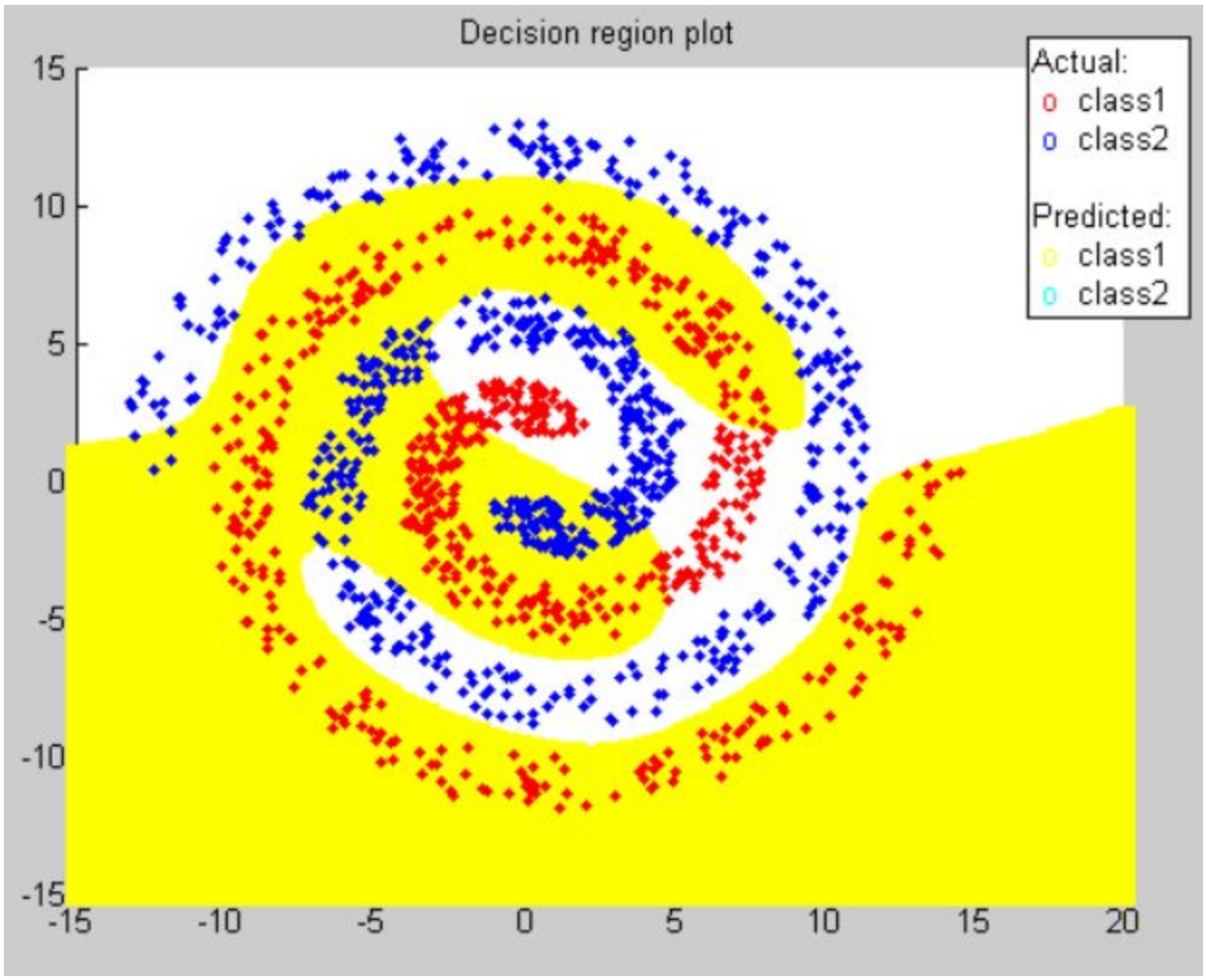


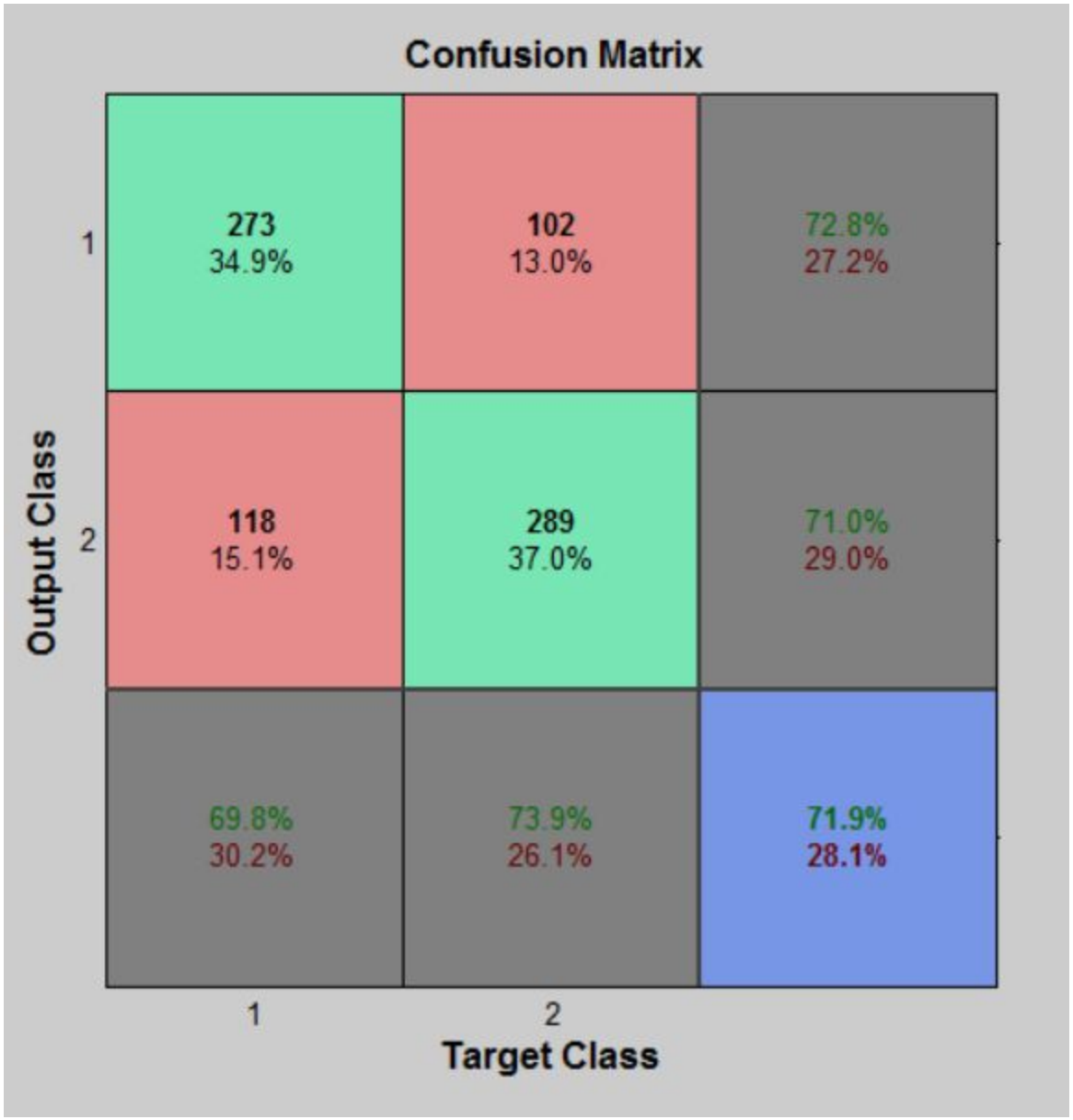
Confusion Matrix for Testing Data

Non-Linearly Separable Data

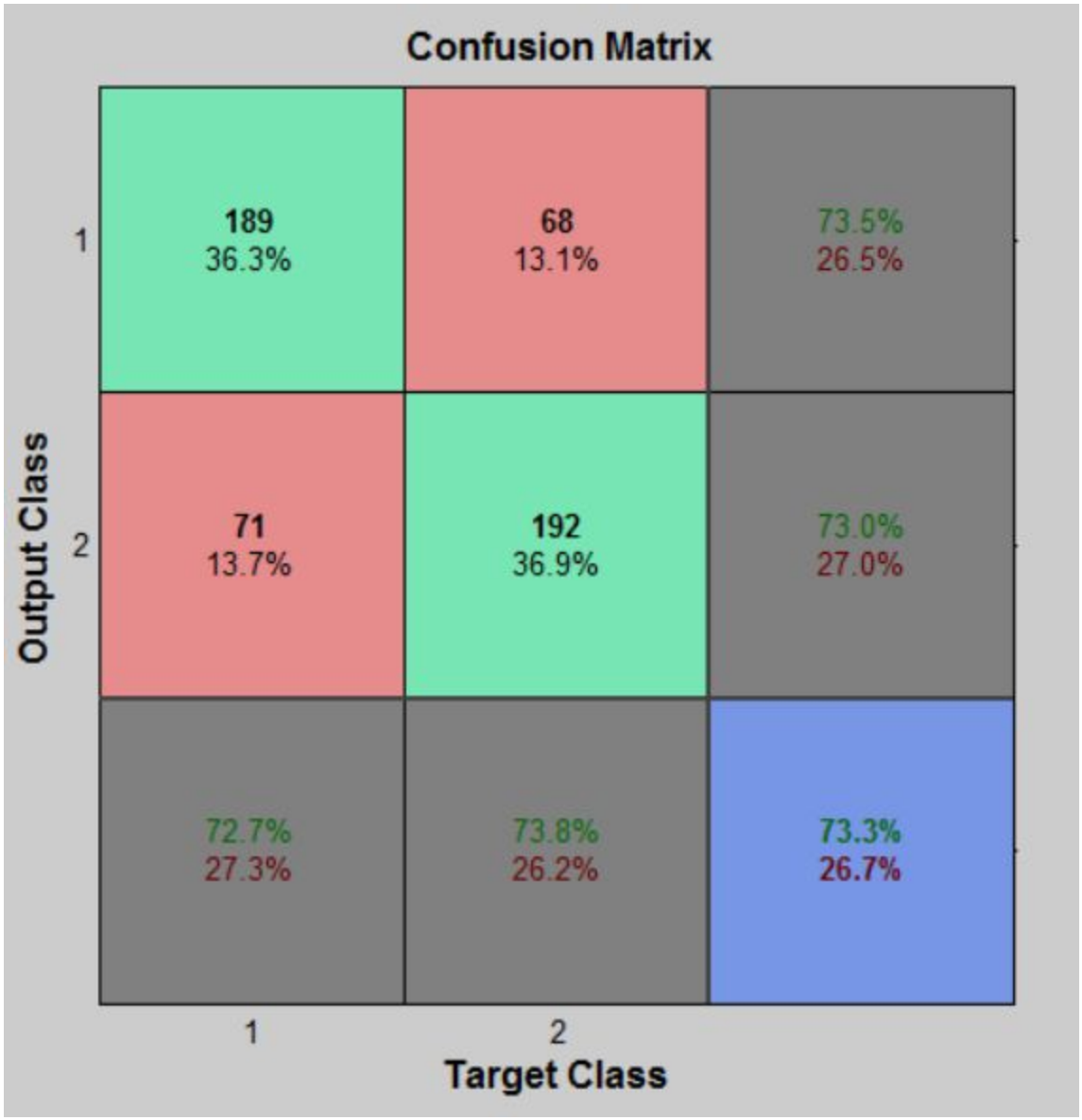
Polynomial Kernel Function

Degree=7 Cost =1024
Training Data : Accuracy = 74.3865% (970/1304) (classification)(Given to check for over-fitting)
Validation Data : Accuracy = 71.867% (562/782) (classification)
Testing Data : Accuracy = 73.2692% (381/520) (classification)



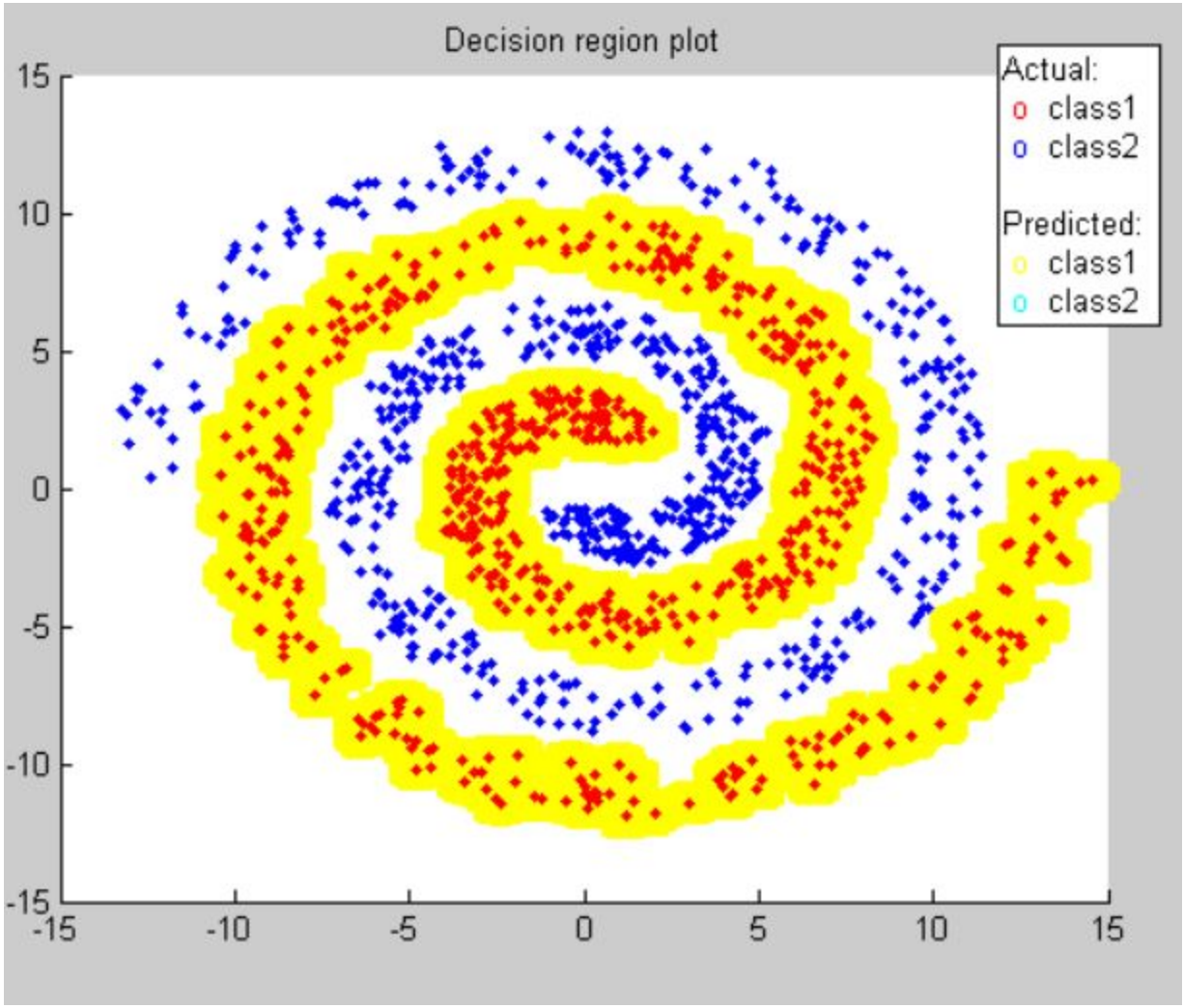


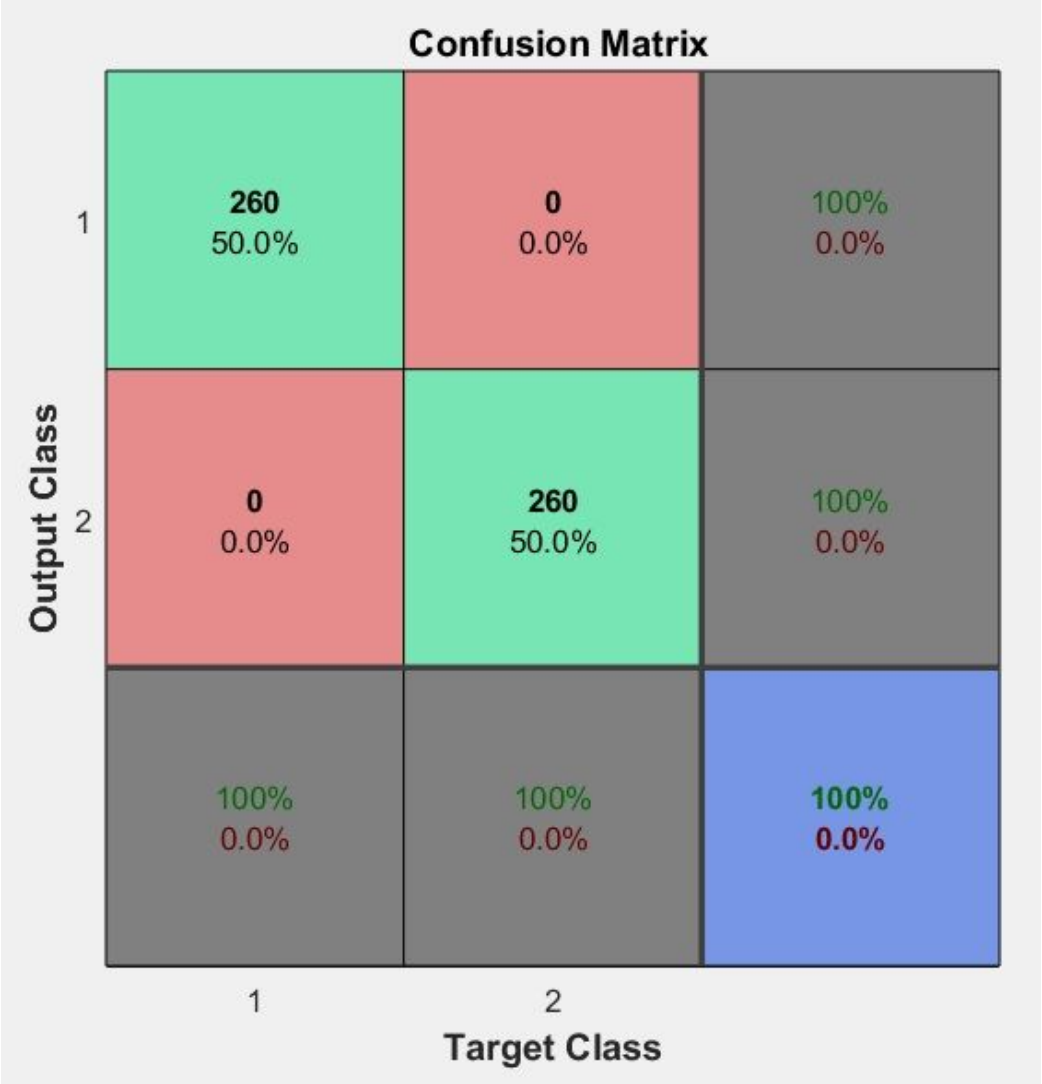
Confusion Matrix for Validation Data



Confusion Matrix For Testing Data

RBF


Decision region plot
Actual:
class1
class2
Predicted:
class1
class2
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15


Confusion Matrix
260
50.0%
0
0.0%
100%
0.0%
0
0.0%
260
50.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
1
2
Output Class
1
2
Target Class

Real World Datasets

Image Classification

5-fold Cross Validation employed after Data Distribution to avoid Over-fitting

C-SVM using RBF Kernel

Original

48 Dimensional Feature Vector
Gamma = 4 c = 64
Accuracy = 100%

Reduced:

43 Dimensional Feature Vector
Gamma = 2 Cost = 128
Accuracy = 100 %

33 Dimensional Feature Vector
Gamma = 1 Cost =256
Accuracy = 100 %

33 Dimensional Feature Vector
Gamma = 1 Cost = 512
Accuracy = 100 %

C-SVM using polynomial Kernel

Original

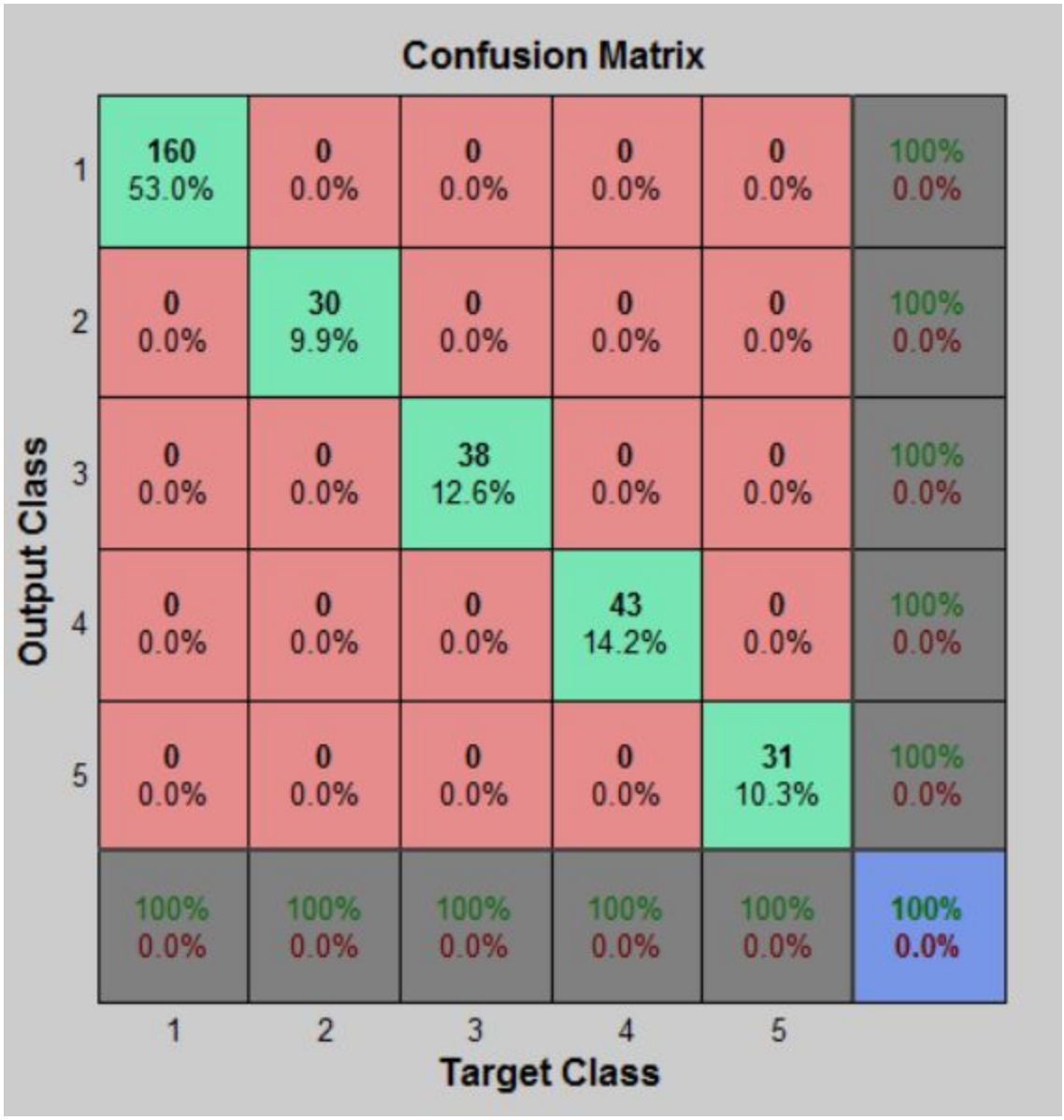
48 Dimensional Feature Vector
Degree =2 ,Cost = 180000
Accuracy= 100 %

Reduced

43 Dimensional Feature Vector
Degree = 2 , Cost = 280000
Accuracy= 100 %

33 Dimensional Feature Vector
Degree =2 Cost= 430000
Accuracy= 100 %

26 Dimensional Feature Vector
Degree=2 Cost= 970000
Accuracy= 100 %


Confusion Matrix
Output Class
Target Class
1
2
3
4
5
160
53.0%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
30
9.9%
0
0.0%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
38
12.6%
0
0.0%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
43
14.2%
0
0.0%
100%
0.0%
0
0.0%
0
0.0%
0
0.0%
0
0.0%
31
10.3%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%
100%
0.0%