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 .

$$f(x) = 1 \text{ if } w.x + b > 0$$
$$= 0 \text{ otherwise}$$

Here **w** and **x** are vectors and **w.x** denotes the dot product between them and **b** is the bias .

- Weight update equation :

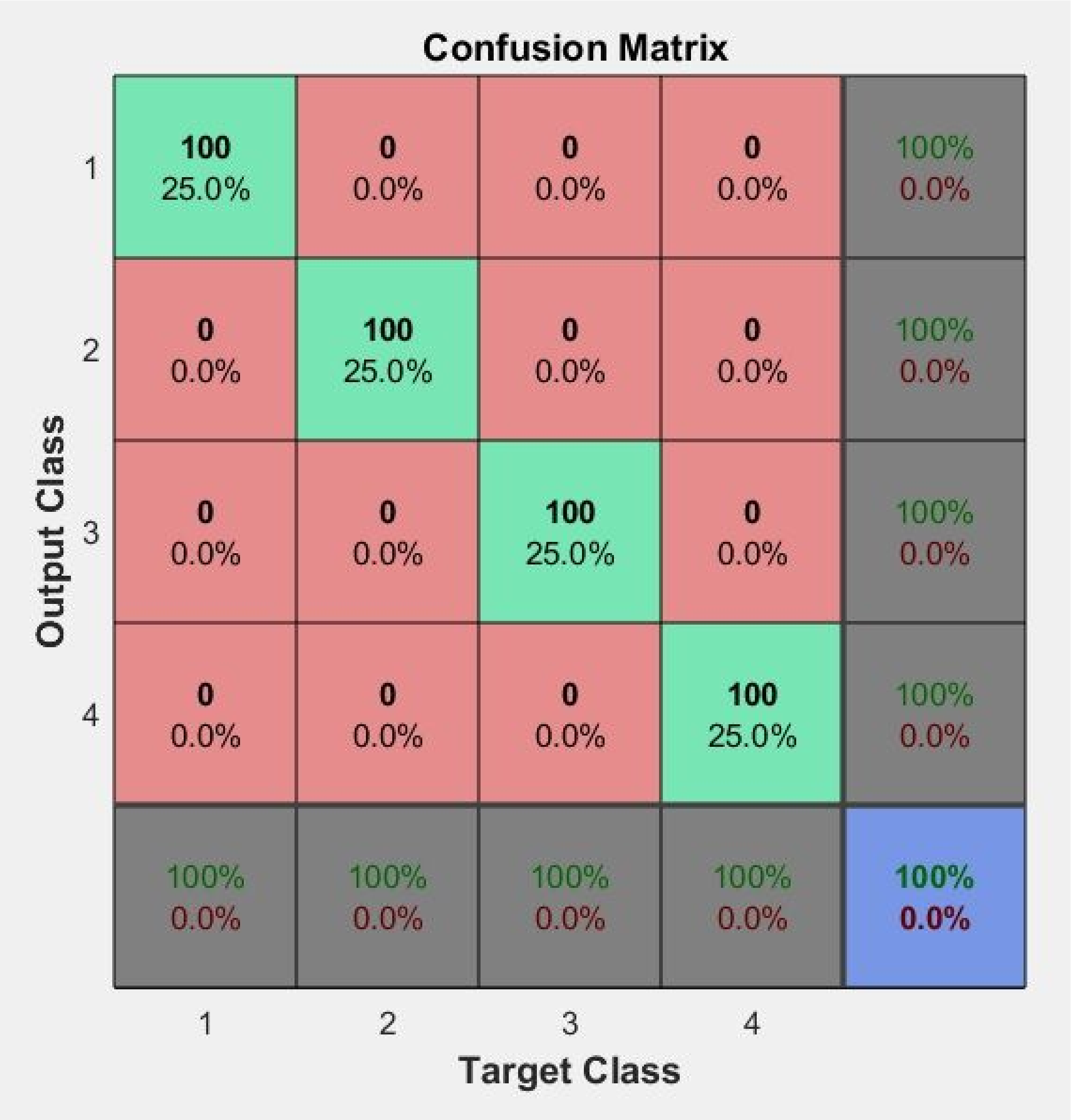
$$w_i(t+1) = w_i(t) + (d_j - y_j(t))*x_{j,i} \qquad \text{for all features } 0 <= i <= d.$$

Where, $y_j(t) = f[w(t).x_j]$

d_j = desired output

Perceptron(One vs One) for Linearly Separable Data

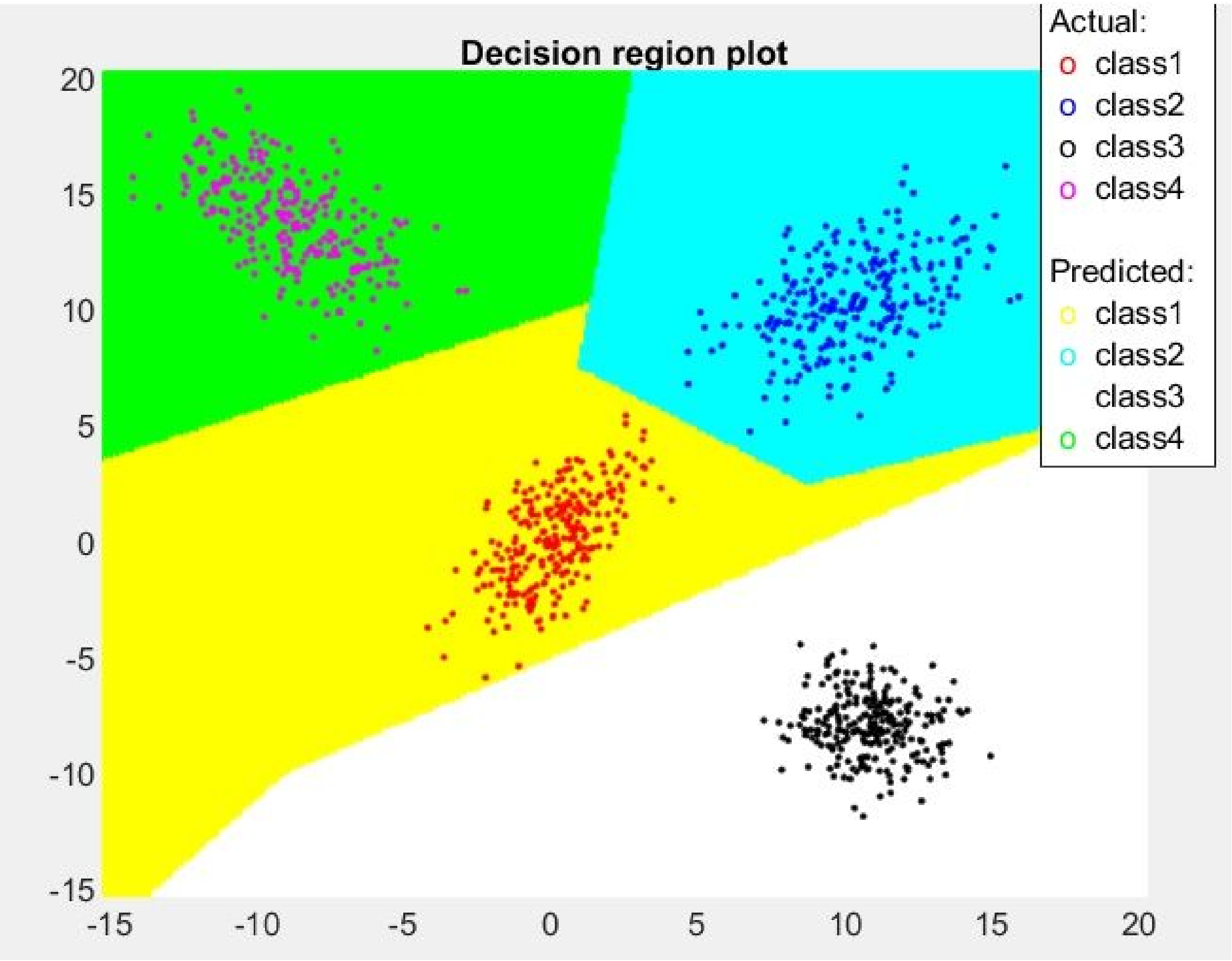
- Train ***K* (*K* − 1) / 2** binary classifiers for a *K*-way multiclass problem.
- Each receives the samples of a pair of classes from the original training set, and learns to distinguish these two classes.
- At prediction time, all *K* (*K* − 1) / 2 classifiers are applied to an unseen sample and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



Confusion Matrix

Output Class	1	2	3	4	
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	

Target Class



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 ith and jth neuron is given by the weight $\mathbf{w_{ij}}$ and the ith neuron by the threshold $\boldsymbol{\upsilon_i}$. The weight function determines the degree of importance of that articular connection in the neural network .

  The output of ith neuron $\mathbf{x_i}$ is given by

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

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

  where ξ_i is the potential of the ith neuron and f is the transfer function .

  It holds that

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

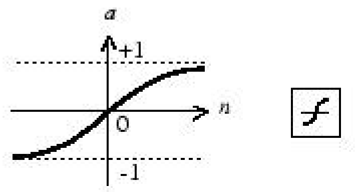
- We vary the thresholds and weights to minimise the difference between the computed and the required output values . Thus , the objective function E is minimised .

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

  where $\mathbf{x_o}$ and $\mathbf{x^{\wedge}_o}$ 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



$a = tansig(n)$

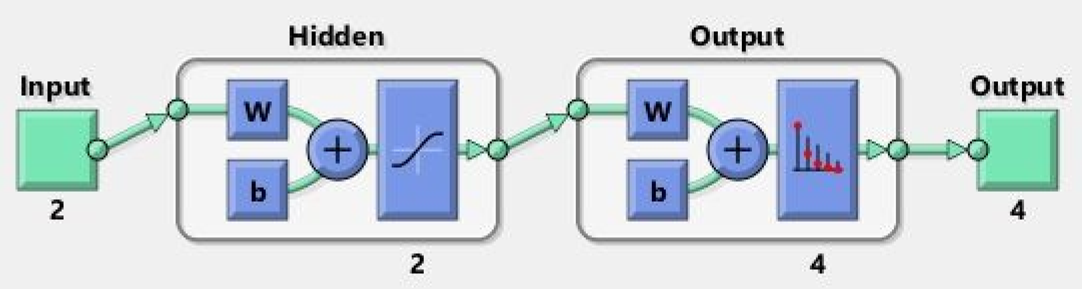
Tan-Sigmoid Transfer Function

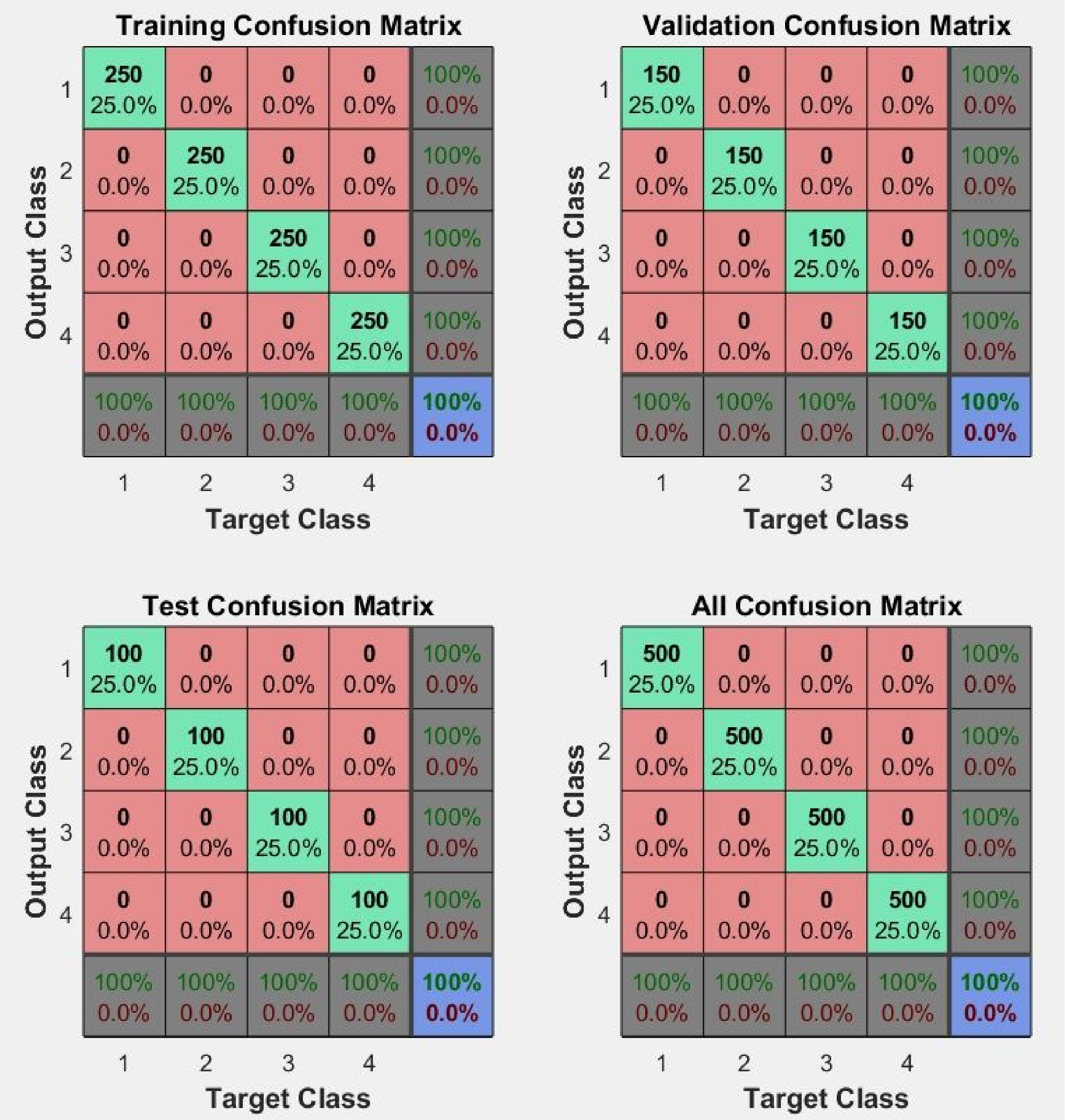
DataDivision : Index (divideind)
Training : Scaled Conjugate Gradient (trainscg)
Performance : Cross-Entropy (crossentropy)
Calculations : MEX

Linearly Separable Data

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

Best Validation Performance is 0.00041847 at epoch 29





Training Confusion Matrix

Output Class \ Target Class	1	2	3	4	
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%

Validation Confusion Matrix

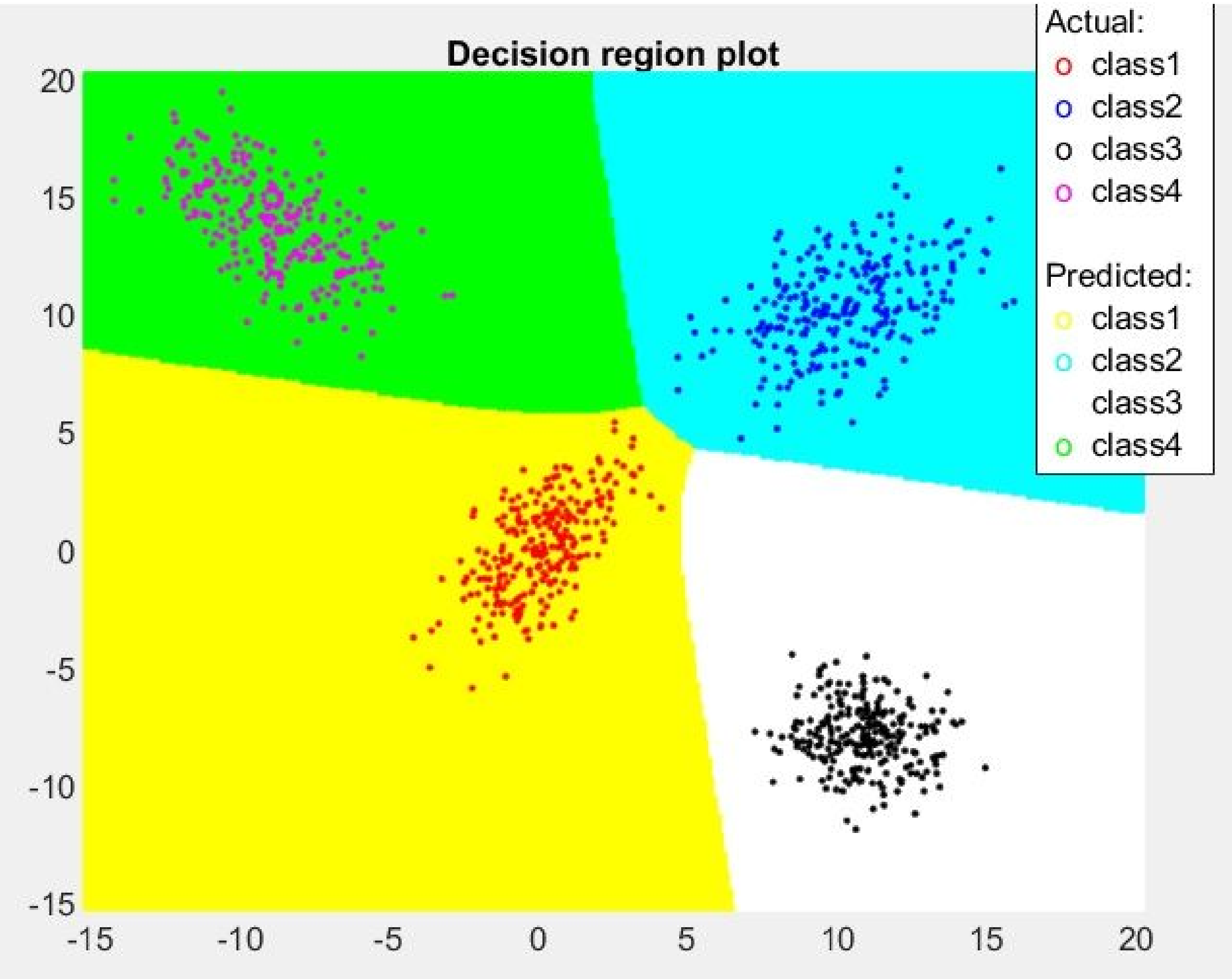
Output Class \ Target Class	1	2	3	4	
1	150 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	150 25.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	150 25.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	150 25.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Test Confusion Matrix

Output Class \ Target Class	1	2	3	4	
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%

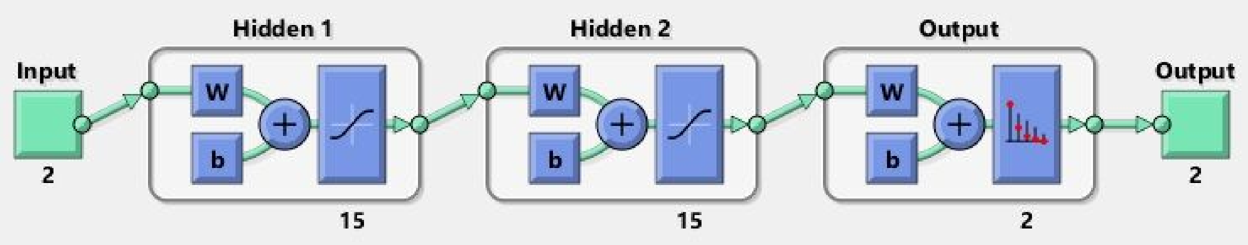
All Confusion Matrix

Output Class \ Target Class	1	2	3	4	
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%



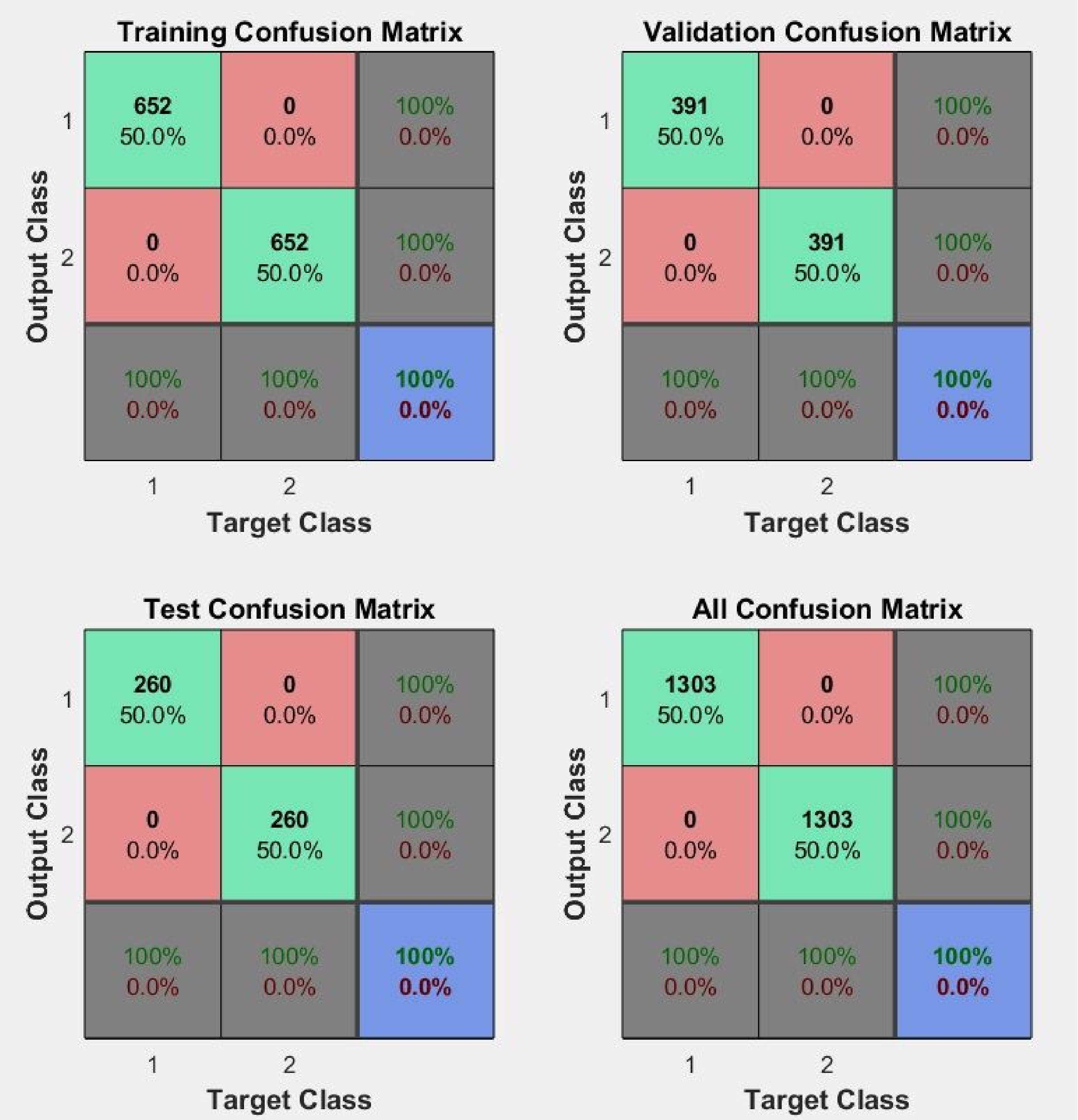
Non Linearly Separable Data

Two Hidden Layers:



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

Best Validation Performance is 0.0020838 at epoch 133



Training Confusion Matrix

Output Class \ Target Class	1	2	
1	652 50.0%	0 0.0%	100% 0.0%
2	0 0.0%	652 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

Validation Confusion Matrix

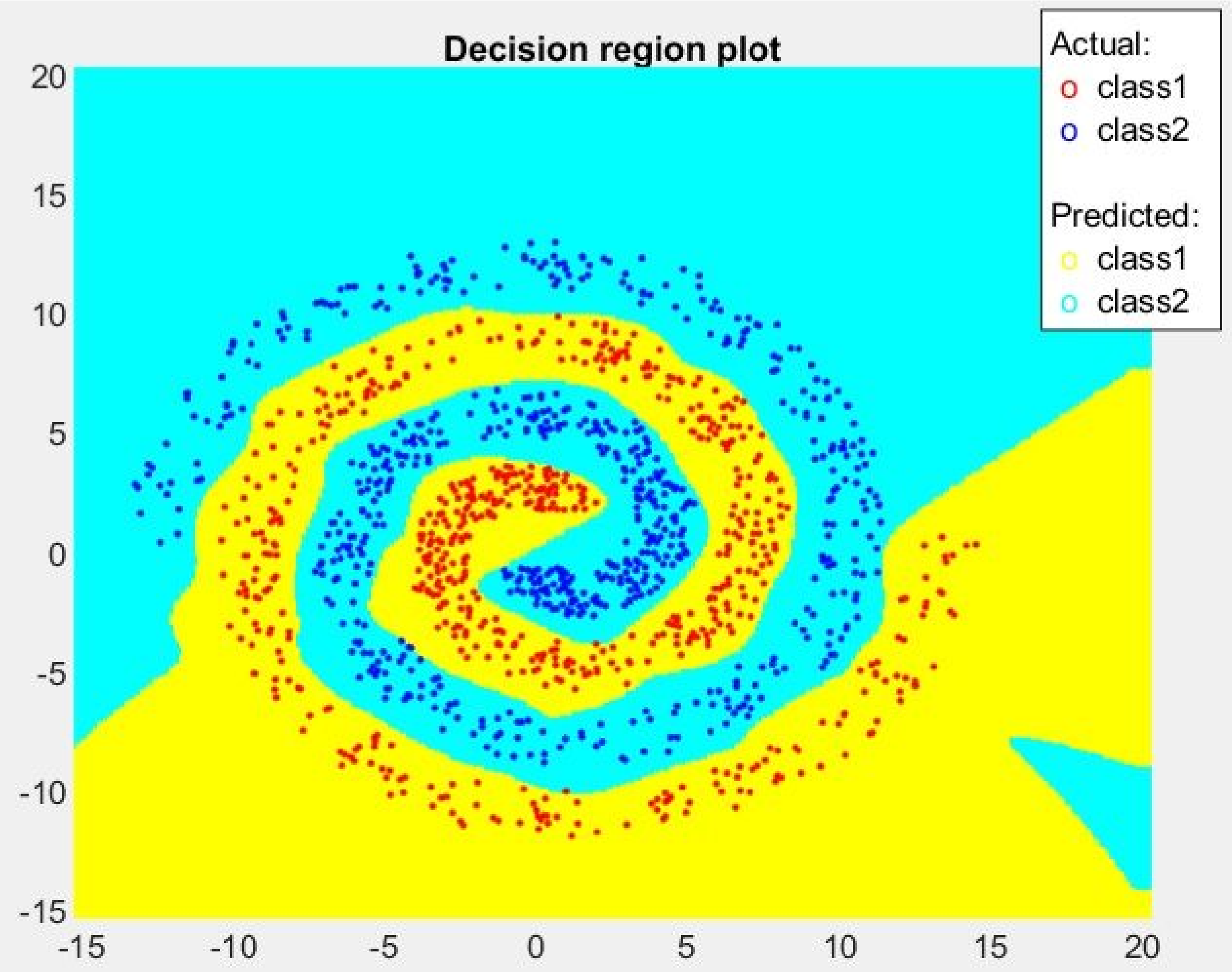
Output Class \ Target Class	1	2	
1	391 50.0%	0 0.0%	100% 0.0%
2	0 0.0%	391 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

Test Confusion Matrix

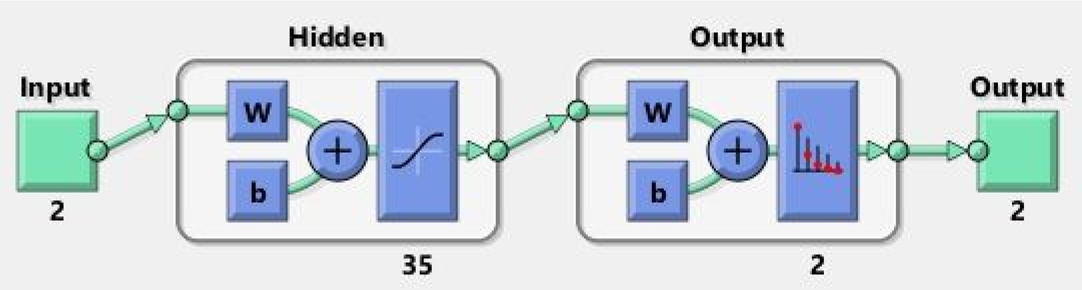
Output Class \ Target Class	1	2	
1	260 50.0%	0 0.0%	100% 0.0%
2	0 0.0%	260 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

All Confusion Matrix

Output Class \ Target Class	1	2	
1	1303 50.0%	0 0.0%	100% 0.0%
2	0 0.0%	1303 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

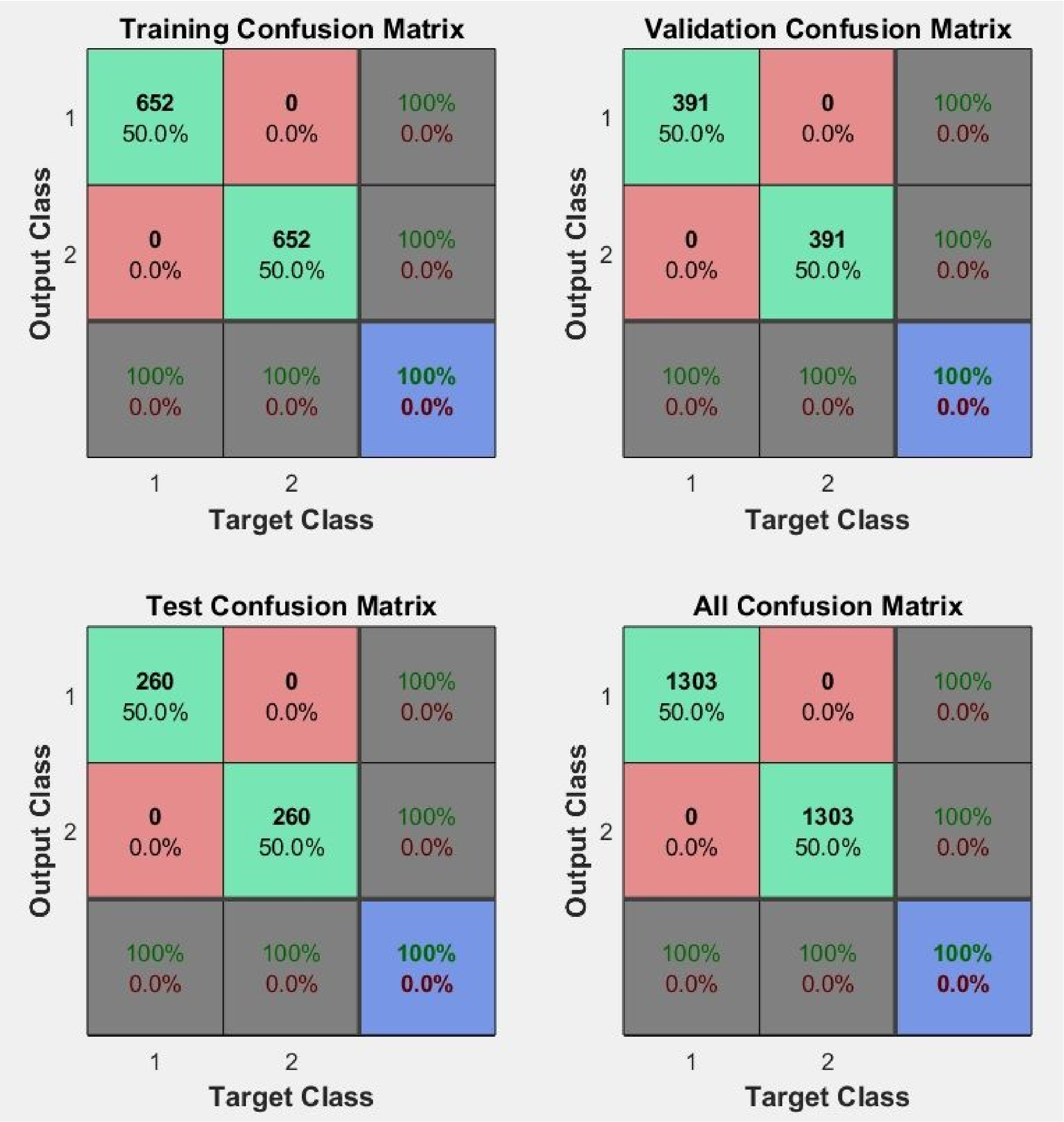


Single Hidden Layer



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

Best Validation Performance is 0.0017486 at epoch 117



Training Confusion Matrix

Output Class	1	2	
1	**652** 50.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**652** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Target Class

Validation Confusion Matrix

Output Class	1	2	
1	**391** 50.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**391** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Target Class

Test Confusion Matrix

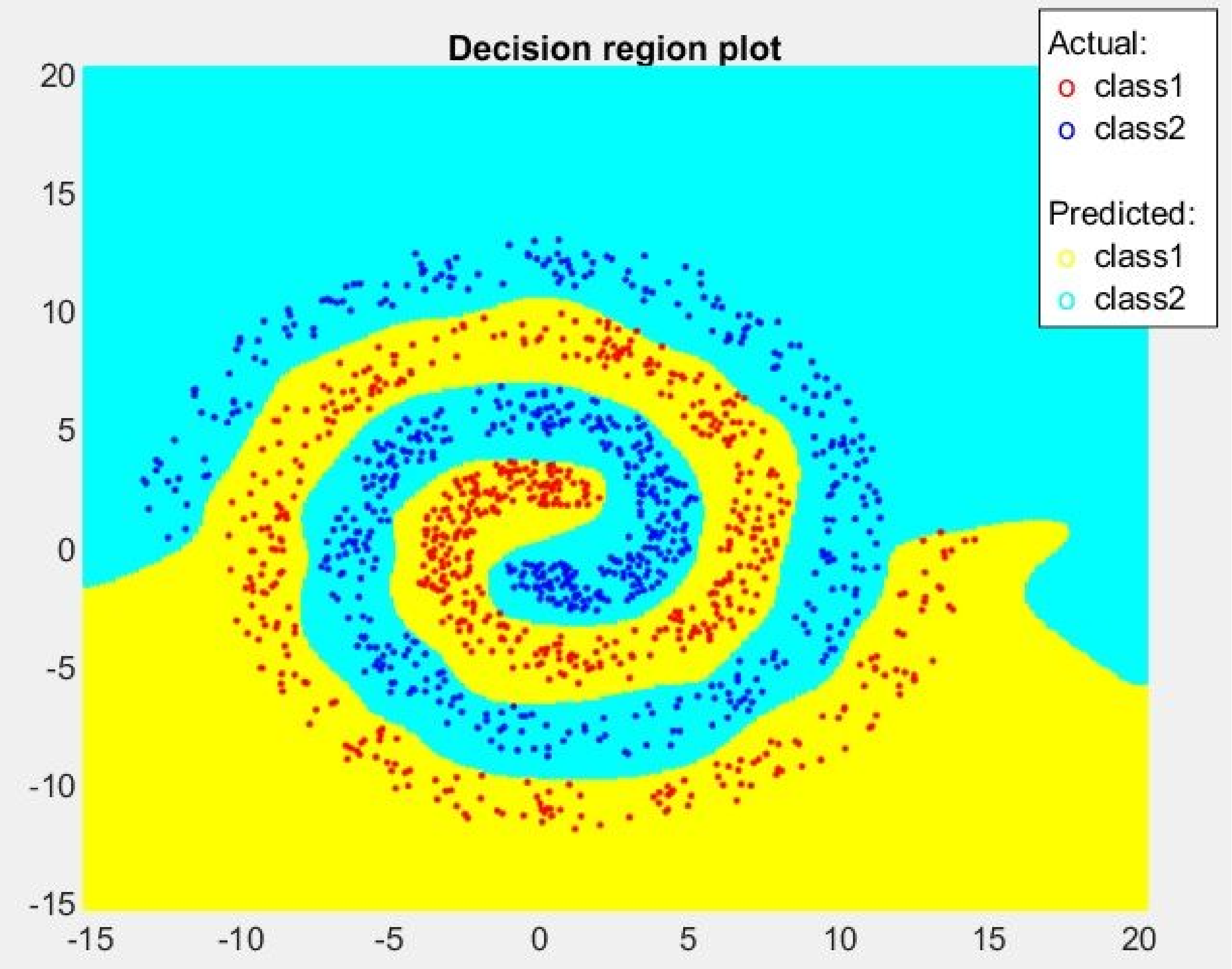
Output Class	1	2	
1	**260** 50.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**260** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Target Class

All Confusion Matrix

Output Class	1	2	
1	**1303** 50.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**1303** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

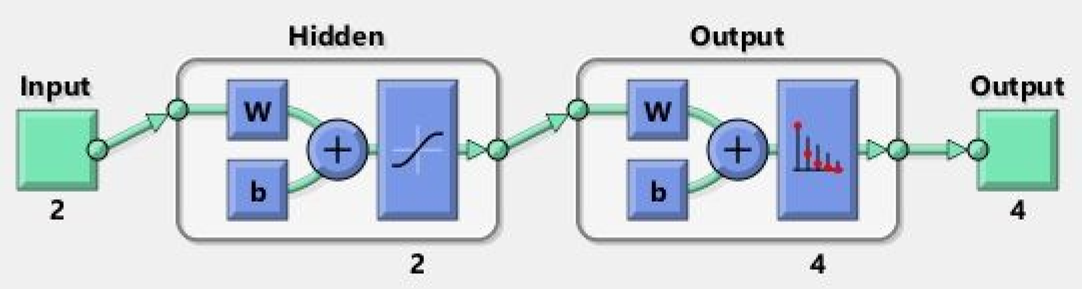
Target Class

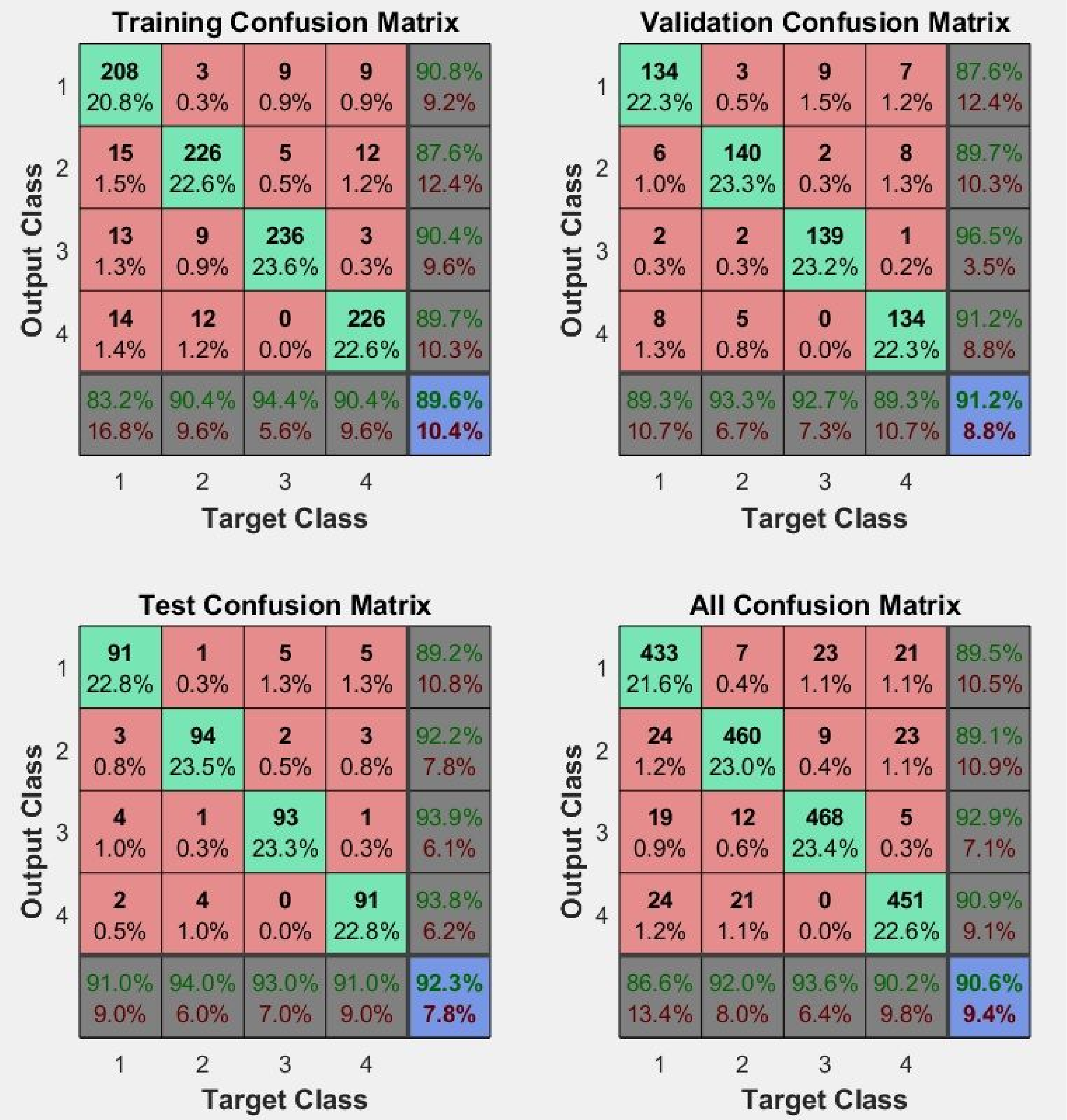


Overlapping Data

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

Best Validation Performance is 0.061296 at epoch 29





Training Confusion Matrix

Output Class \ Target Class	1	2	3	4	
1	208 20.8%	3 0.3%	9 0.9%	9 0.9%	90.8% 9.2%
2	15 1.5%	226 22.6%	5 0.5%	12 1.2%	87.6% 12.4%
3	13 1.3%	9 0.9%	236 23.6%	3 0.3%	90.4% 9.6%
4	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	1	2	3	4	
1	134 22.3%	3 0.5%	9 1.5%	7 1.2%	87.6% 12.4%
2	6 1.0%	140 23.3%	2 0.3%	8 1.3%	89.7% 10.3%
3	2 0.3%	2 0.3%	139 23.2%	1 0.2%	96.5% 3.5%
4	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	1	2	3	4	
1	91 22.8%	1 0.3%	5 1.3%	5 1.3%	89.2% 10.8%
2	3 0.8%	94 23.5%	2 0.5%	3 0.8%	92.2% 7.8%
3	4 1.0%	1 0.3%	93 23.3%	1 0.3%	93.9% 6.1%
4	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	1	2	3	4	
1	433 21.6%	7 0.4%	23 1.1%	21 1.1%	89.5% 10.5%
2	24 1.2%	460 23.0%	9 0.4%	23 1.1%	89.1% 10.9%
3	19 0.9%	12 0.6%	468 23.4%	5 0.3%	92.9% 7.1%
4	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%

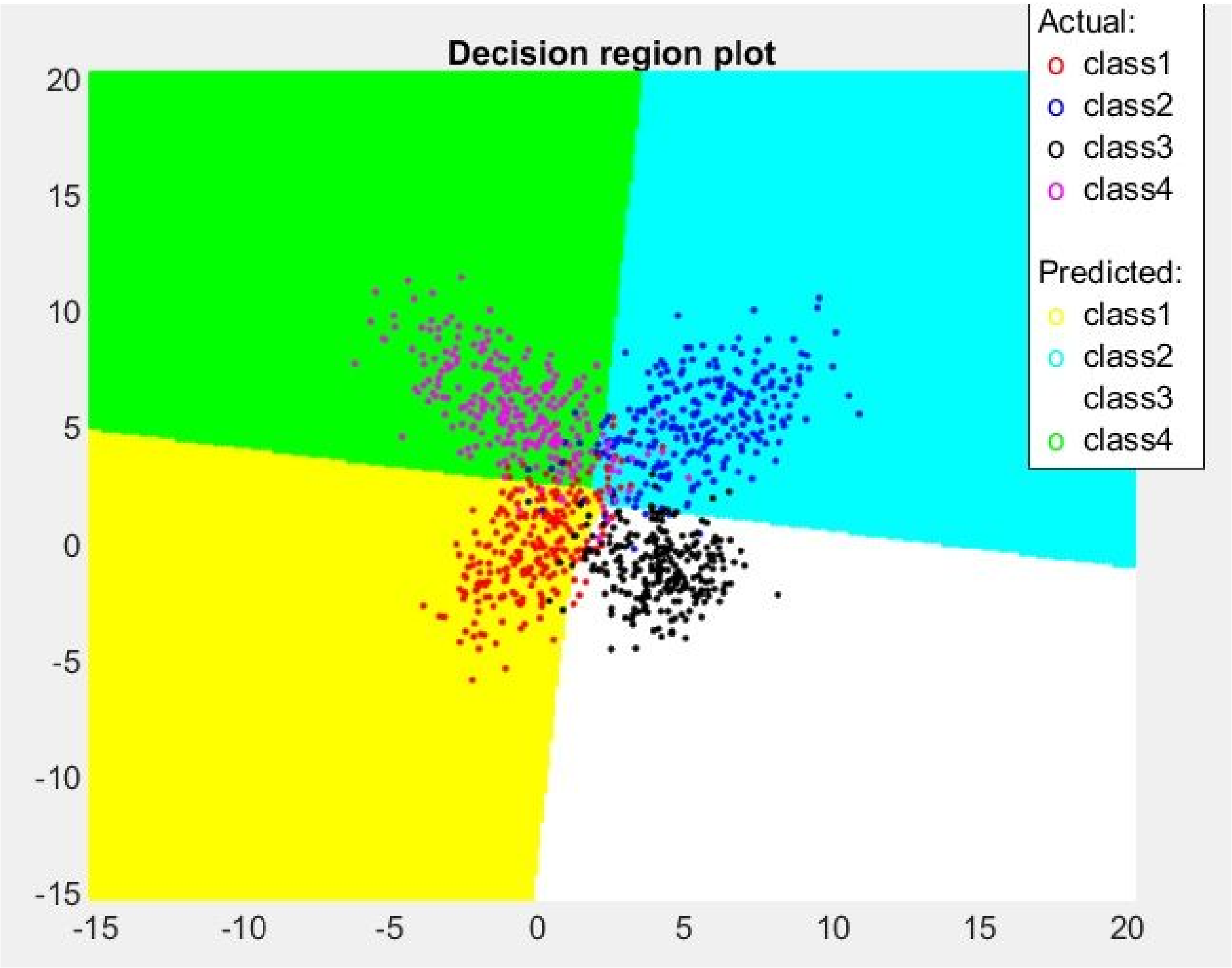
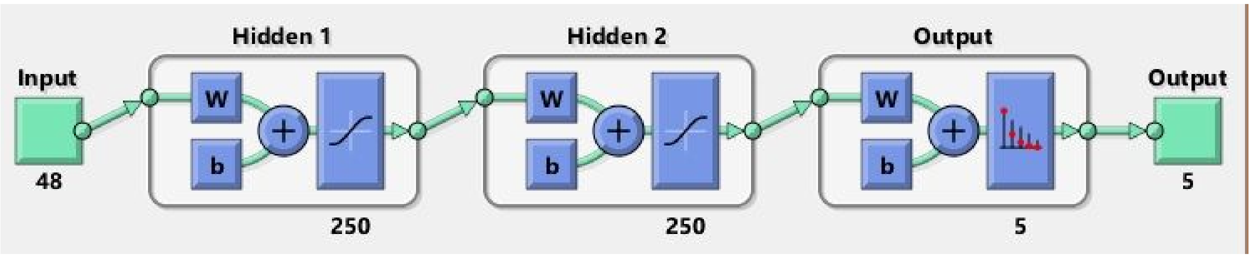


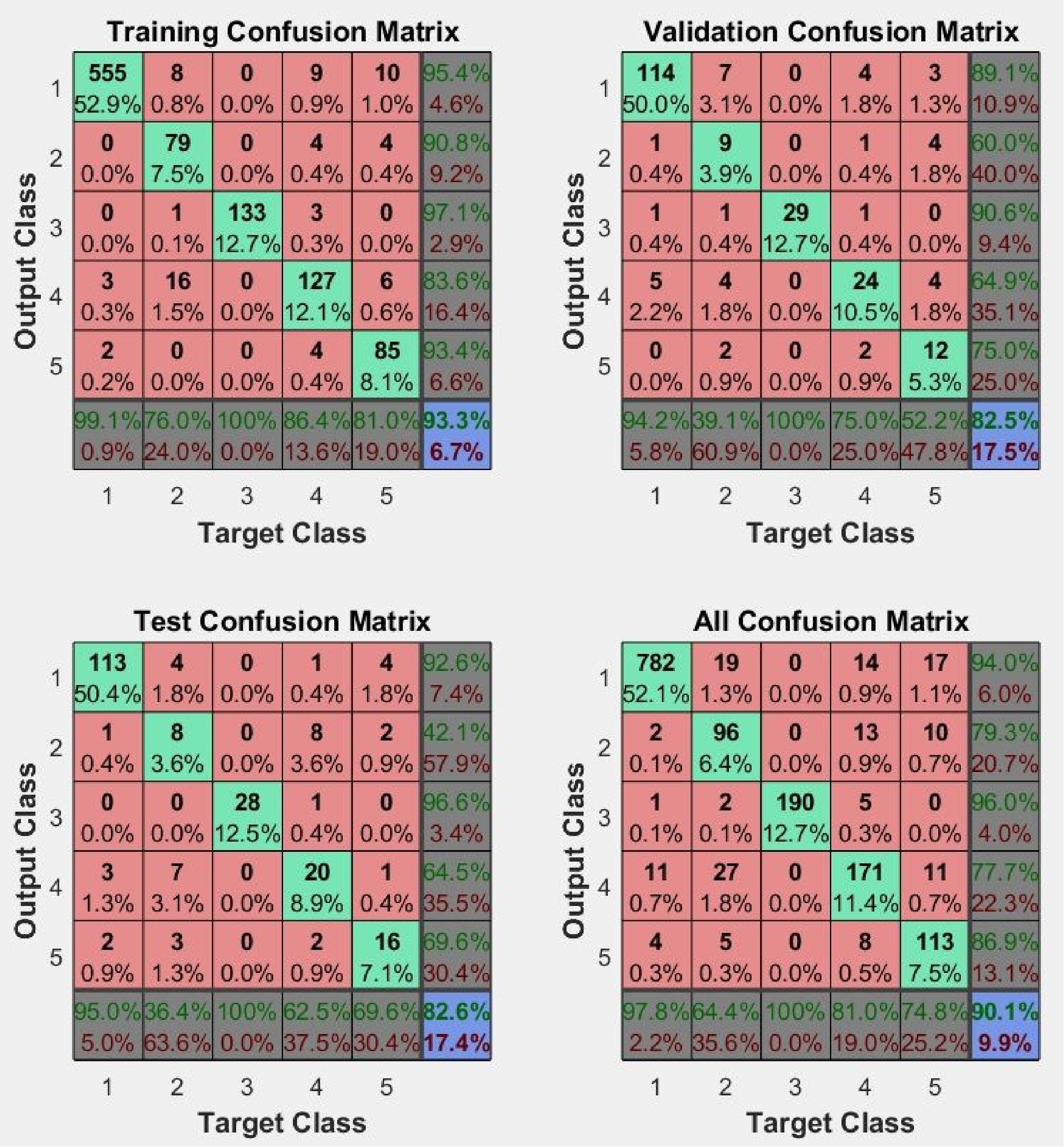
Image Dataset

48 Dimension



Epoch = 3500 Iterations
Performance = 0.0388
Gradient = 0.0718

Best Validation Performance is 0.14379 at epoch 2237



Training Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	555 52.9%	8 0.8%	0 0.0%	9 0.9%	10 1.0%	95.4% 4.6%
2	0 0.0%	79 7.5%	0 0.0%	4 0.4%	4 0.4%	90.8% 9.2%
3	0 0.0%	1 0.1%	133 12.7%	3 0.3%	0 0.0%	97.1% 2.9%
4	3 0.3%	16 1.5%	0 0.0%	127 12.1%	6 0.6%	83.6% 16.4%
5	2 0.2%	0 0.0%	0 0.0%	4 0.4%	85 8.1%	93.4% 6.6%
	99.1% 0.9%	76.0% 24.0%	100% 0.0%	86.4% 13.6%	81.0% 19.0%	93.3% 6.7%

Validation Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	114 50.0%	7 3.1%	0 0.0%	4 1.8%	3 1.3%	89.1% 10.9%
2	1 0.4%	9 3.9%	0 0.0%	1 0.4%	4 1.8%	60.0% 40.0%
3	1 0.4%	1 0.4%	29 12.7%	1 0.4%	0 0.0%	90.6% 9.4%
4	5 2.2%	4 1.8%	0 0.0%	24 10.5%	4 1.8%	64.9% 35.1%
5	0 0.0%	2 0.9%	0 0.0%	2 0.9%	12 5.3%	75.0% 25.0%
	94.2% 5.8%	39.1% 60.9%	100% 0.0%	75.0% 25.0%	52.2% 47.8%	82.5% 17.5%

Test Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	113 50.4%	4 1.8%	0 0.0%	1 0.4%	4 1.8%	92.6% 7.4%
2	1 0.4%	8 3.6%	0 0.0%	8 3.6%	2 0.9%	42.1% 57.9%
3	0 0.0%	0 0.0%	28 12.5%	1 0.4%	0 0.0%	96.6% 3.4%
4	3 1.3%	7 3.1%	0 0.0%	20 8.9%	1 0.4%	64.5% 35.5%
5	2 0.9%	3 1.3%	0 0.0%	2 0.9%	16 7.1%	69.6% 30.4%
	95.0% 5.0%	36.4% 63.6%	100% 0.0%	62.5% 37.5%	69.6% 30.4%	82.6% 17.4%

All Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	782 52.1%	19 1.3%	0 0.0%	14 0.9%	17 1.1%	94.0% 6.0%
2	2 0.1%	96 6.4%	0 0.0%	13 0.9%	10 0.7%	79.3% 20.7%
3	1 0.1%	2 0.1%	190 12.7%	5 0.3%	0 0.0%	96.0% 4.0%
4	11 0.7%	27 1.8%	0 0.0%	171 11.4%	11 0.7%	77.7% 22.3%
5	4 0.3%	5 0.3%	0 0.0%	8 0.5%	113 7.5%	86.9% 13.1%
	97.8% 2.2%	64.4% 35.6%	100% 0.0%	81.0% 19.0%	74.8% 25.2%	90.1% 9.9%

Dimensionality Reduction

Data Preprocessing (Feature Scaling / Mean Normalisation):

- Mean Normalisation of data:

  $E[x] = \mu$

  $y = x - \mu$

  $E[y] = 0$

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:

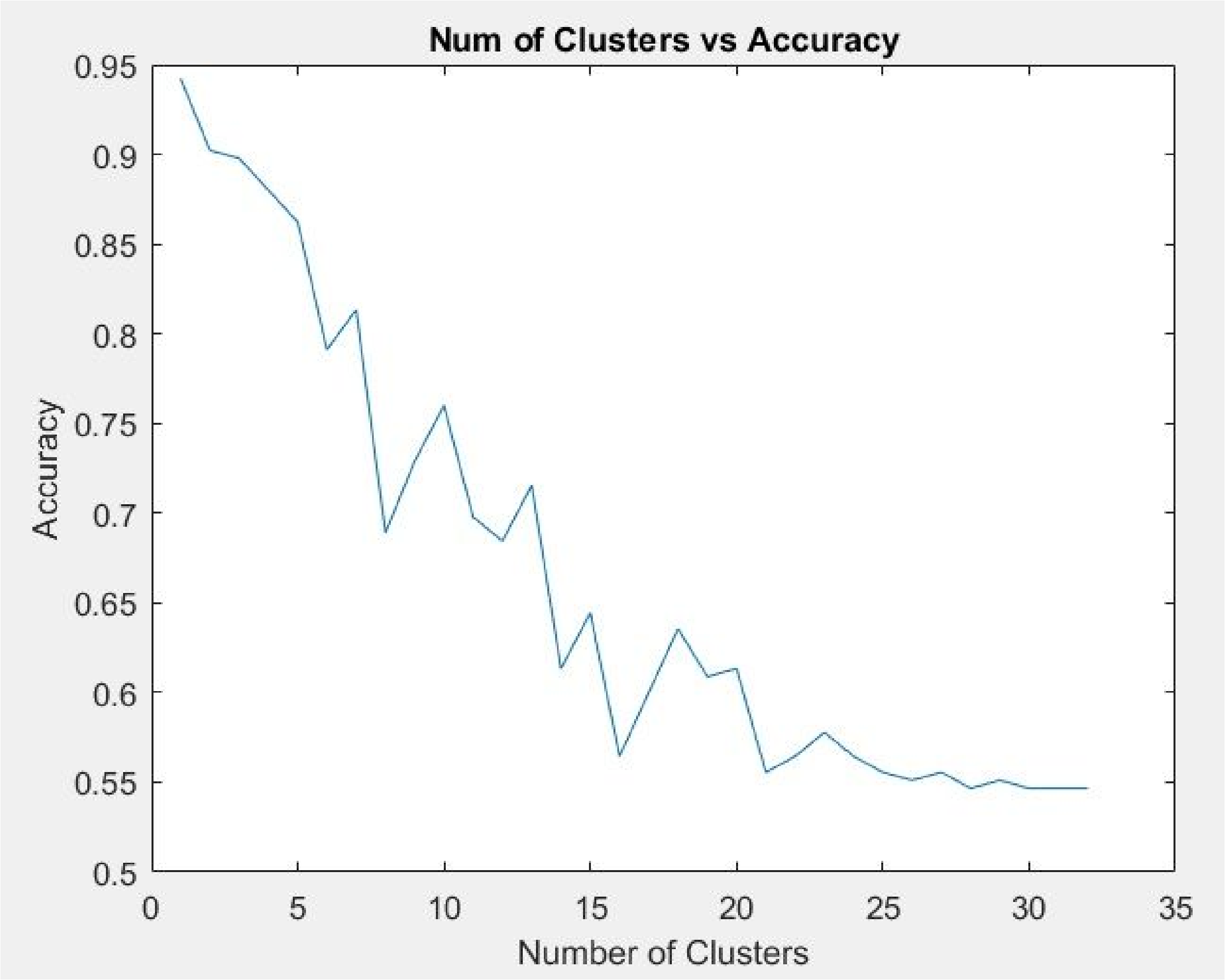
  Average Squared projection error / Total variation in data <= 0.01

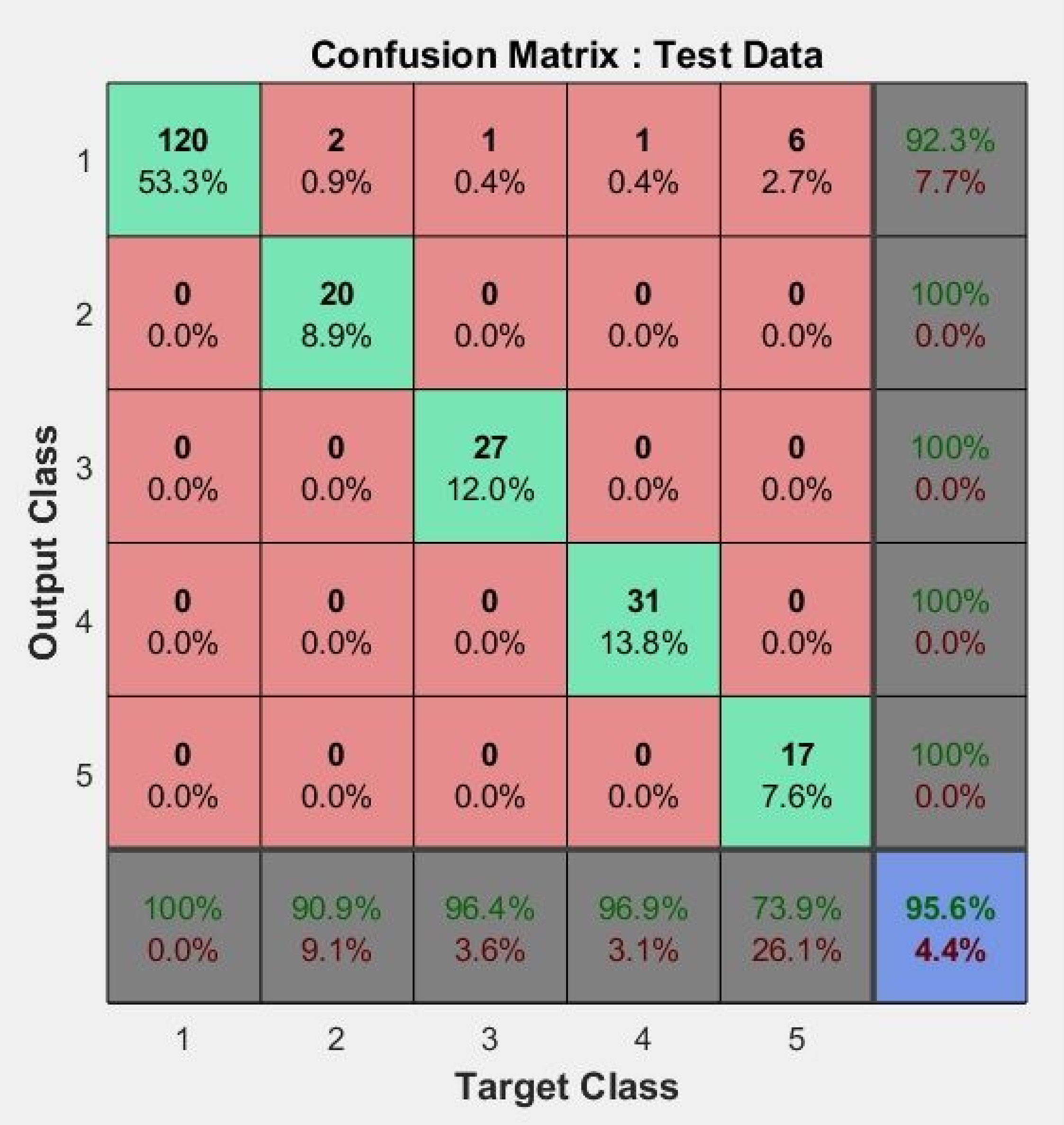
  99% variance is retained.

Image Dataset

48 Dimensions

100% Variance Retained





Confusion Matrix : Test Data

Output Class	1	2	3	4	5	
1	**120** 53.3%	**2** 0.9%	**1** 0.4%	**1** 0.4%	**6** 2.7%	92.3% 7.7%
2	**0** 0.0%	**20** 8.9%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
3	**0** 0.0%	**0** 0.0%	**27** 12.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**31** 13.8%	**0** 0.0%	100% 0.0%
5	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**17** 7.6%	100% 0.0%
	100% 0.0%	90.9% 9.1%	96.4% 3.6%	96.9% 3.1%	73.9% 26.1%	**95.6%** **4.4%**

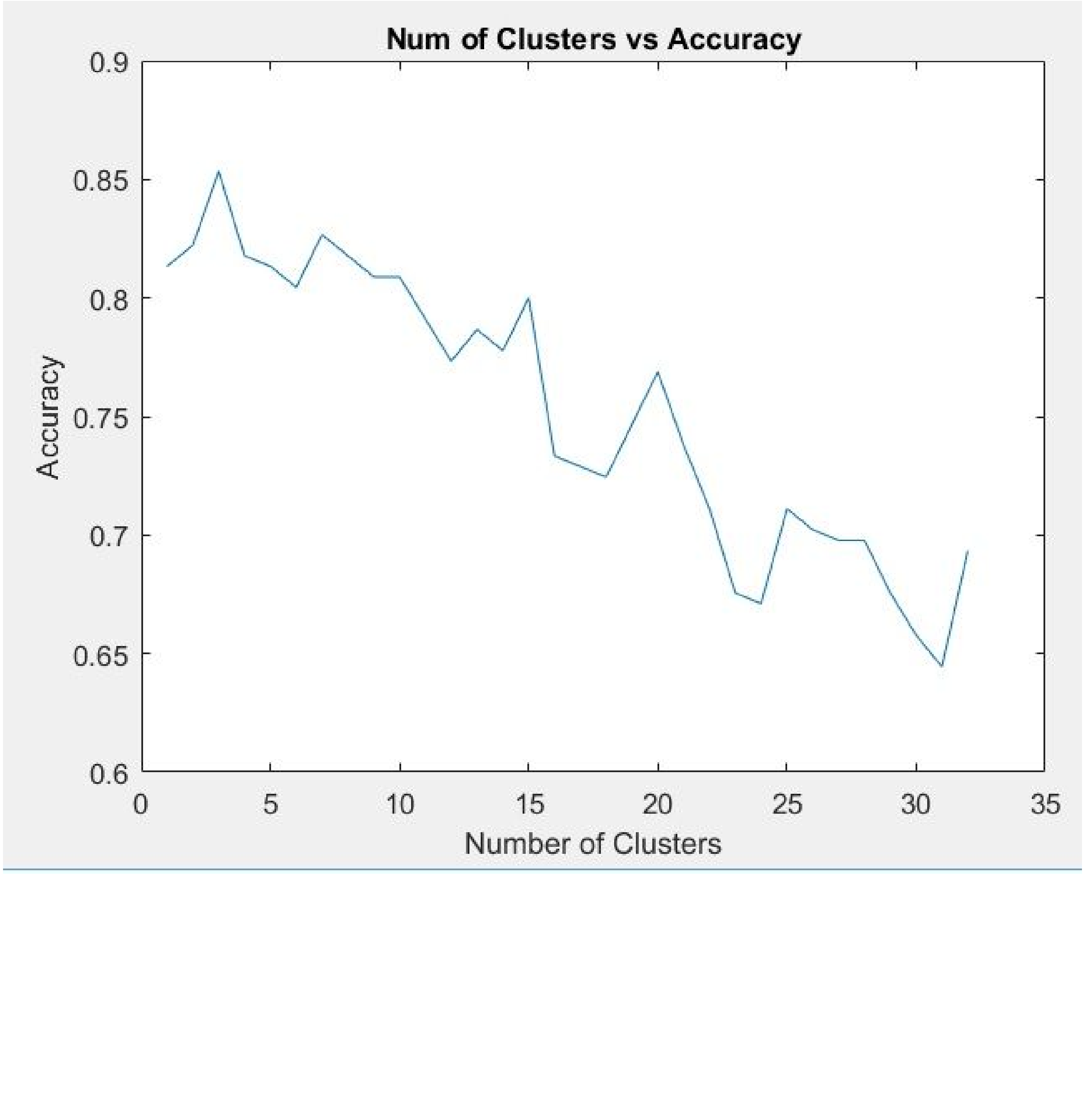
Target Class

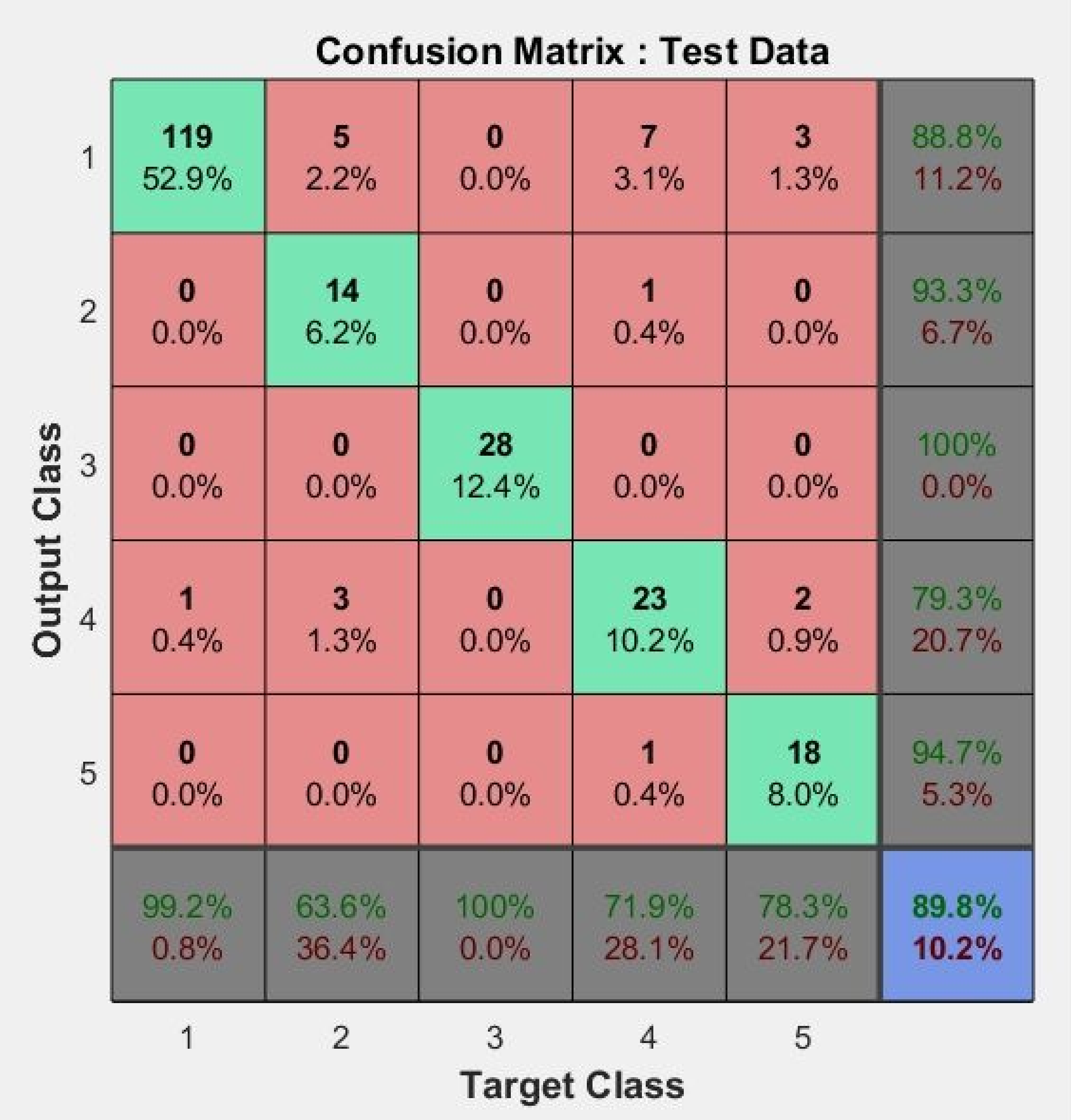
Image Dataset

27 Dimensions

99.0661% Variance Retained

GMM

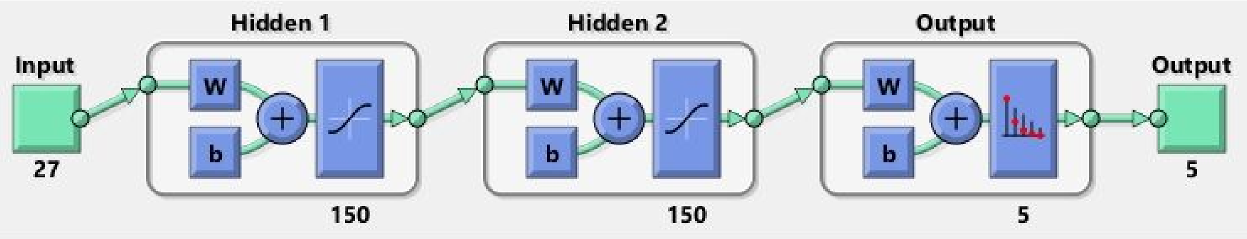




Confusion Matrix : Test Data

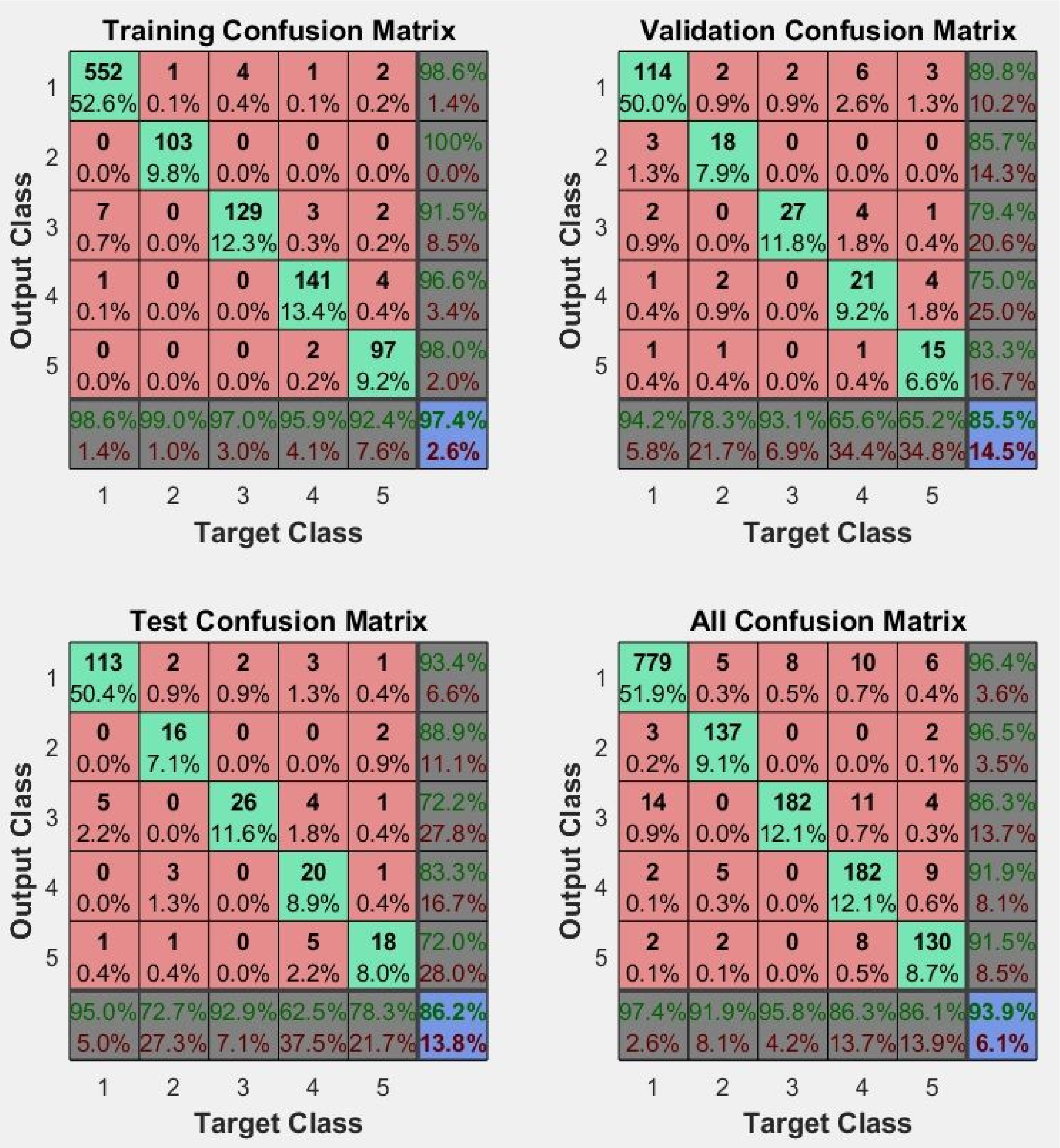
Output Class \ Target Class	1	2	3	4	5	
1	119 52.9%	5 2.2%	0 0.0%	7 3.1%	3 1.3%	88.8% 11.2%
2	0 0.0%	14 6.2%	0 0.0%	1 0.4%	0 0.0%	93.3% 6.7%
3	0 0.0%	0 0.0%	28 12.4%	0 0.0%	0 0.0%	100% 0.0%
4	1 0.4%	3 1.3%	0 0.0%	23 10.2%	2 0.9%	79.3% 20.7%
5	0 0.0%	0 0.0%	0 0.0%	1 0.4%	18 8.0%	94.7% 5.3%
	99.2% 0.8%	63.6% 36.4%	100% 0.0%	71.9% 28.1%	78.3% 21.7%	**89.8%** **10.2%**

MLFFNN



Epoch = 821 Iterations
Performance = 0.0556
Gradient = 8.16e-07

Best Validation Performance is 0.1028 at epoch 393



Training Confusion Matrix

Output Class	1	2	3	4	5	
1	552 52.6%	1 0.1%	4 0.4%	1 0.1%	2 0.2%	98.6% 1.4%
2	0 0.0%	103 9.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	7 0.7%	0 0.0%	129 12.3%	3 0.3%	2 0.2%	91.5% 8.5%
4	1 0.1%	0 0.0%	0 0.0%	141 13.4%	4 0.4%	96.6% 3.4%
5	0 0.0%	0 0.0%	0 0.0%	2 0.2%	97 9.2%	98.0% 2.0%
	98.6% 1.4%	99.0% 1.0%	97.0% 3.0%	95.9% 4.1%	92.4% 7.6%	97.4% 2.6%

Target Class

Validation Confusion Matrix

Output Class	1	2	3	4	5	
1	114 50.0%	2 0.9%	2 0.9%	6 2.6%	3 1.3%	89.8% 10.2%
2	3 1.3%	18 7.9%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
3	2 0.9%	0 0.0%	27 11.8%	4 1.8%	1 0.4%	79.4% 20.6%
4	1 0.4%	2 0.9%	0 0.0%	21 9.2%	4 1.8%	75.0% 25.0%
5	1 0.4%	1 0.4%	0 0.0%	1 0.4%	15 6.6%	83.3% 16.7%
	94.2% 5.8%	78.3% 21.7%	93.1% 6.9%	65.6% 34.4%	65.2% 34.8%	85.5% 14.5%

Target Class

Test Confusion Matrix

Output Class	1	2	3	4	5	
1	113 50.4%	2 0.9%	2 0.9%	3 1.3%	1 0.4%	93.4% 6.6%
2	0 0.0%	16 7.1%	0 0.0%	0 0.0%	2 0.9%	88.9% 11.1%
3	5 2.2%	0 0.0%	26 11.6%	4 1.8%	1 0.4%	72.2% 27.8%
4	0 0.0%	3 1.3%	0 0.0%	20 8.9%	1 0.4%	83.3% 16.7%
5	1 0.4%	1 0.4%	0 0.0%	5 2.2%	18 8.0%	72.0% 28.0%
	95.0% 5.0%	72.7% 27.3%	92.9% 7.1%	62.5% 37.5%	78.3% 21.7%	86.2% 13.8%

Target Class

All Confusion Matrix

Output Class	1	2	3	4	5	
1	779 51.9%	5 0.3%	8 0.5%	10 0.7%	6 0.4%	96.4% 3.6%
2	3 0.2%	137 9.1%	0 0.0%	0 0.0%	2 0.1%	96.5% 3.5%
3	14 0.9%	0 0.0%	182 12.1%	11 0.7%	4 0.3%	86.3% 13.7%
4	2 0.1%	5 0.3%	0 0.0%	182 12.1%	9 0.6%	91.9% 8.1%
5	2 0.1%	2 0.1%	0 0.0%	8 0.5%	130 8.7%	91.5% 8.5%
	97.4% 2.6%	91.9% 8.1%	95.8% 4.2%	86.3% 13.7%	86.1% 13.9%	93.9% 6.1%

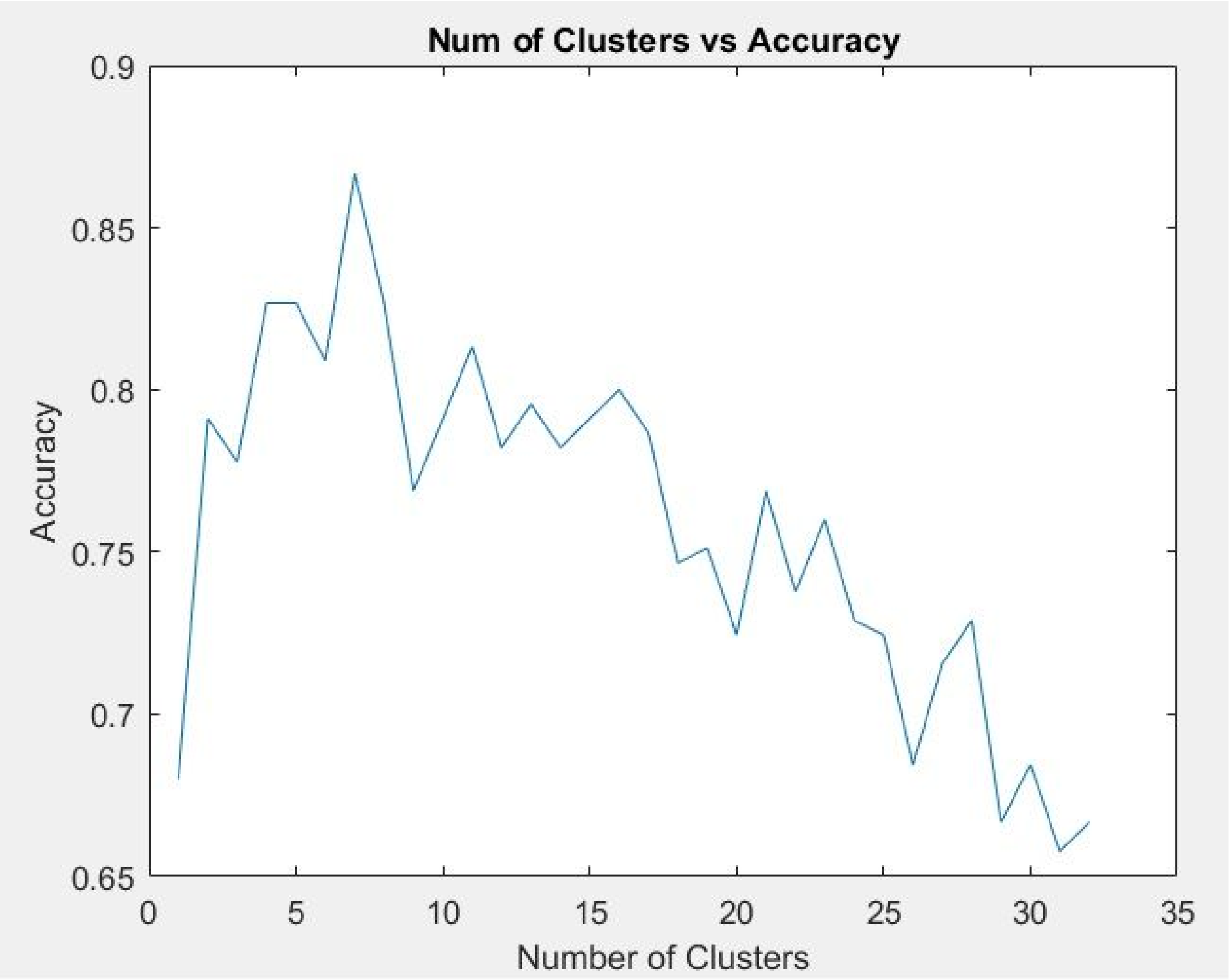
Target Class

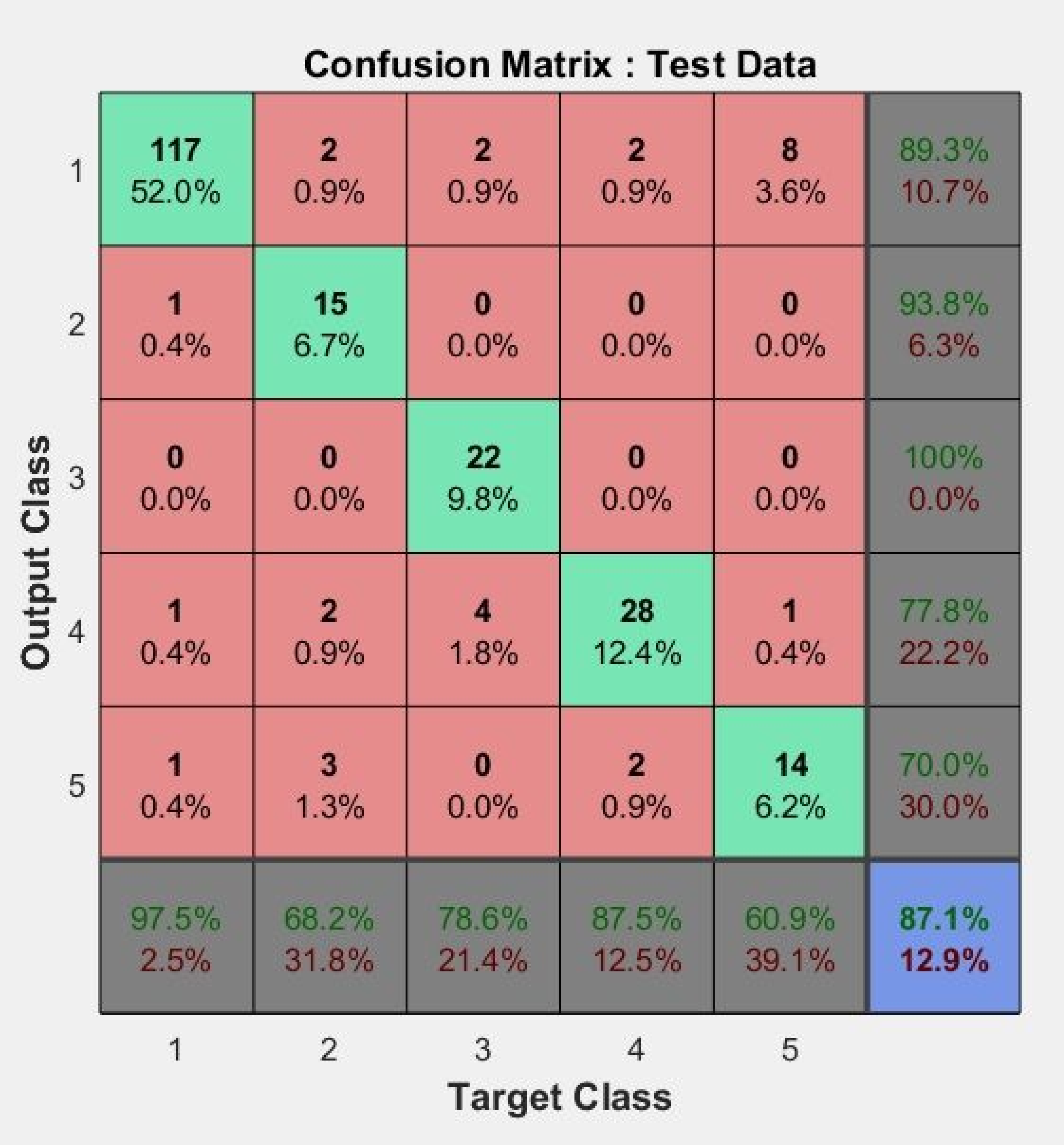
Image Dataset

11 Dimensions

95.4776% Variance Retained

GMM

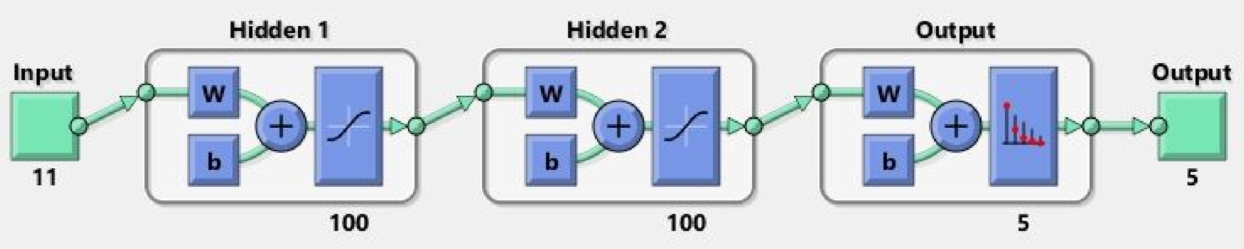




Confusion Matrix : Test Data

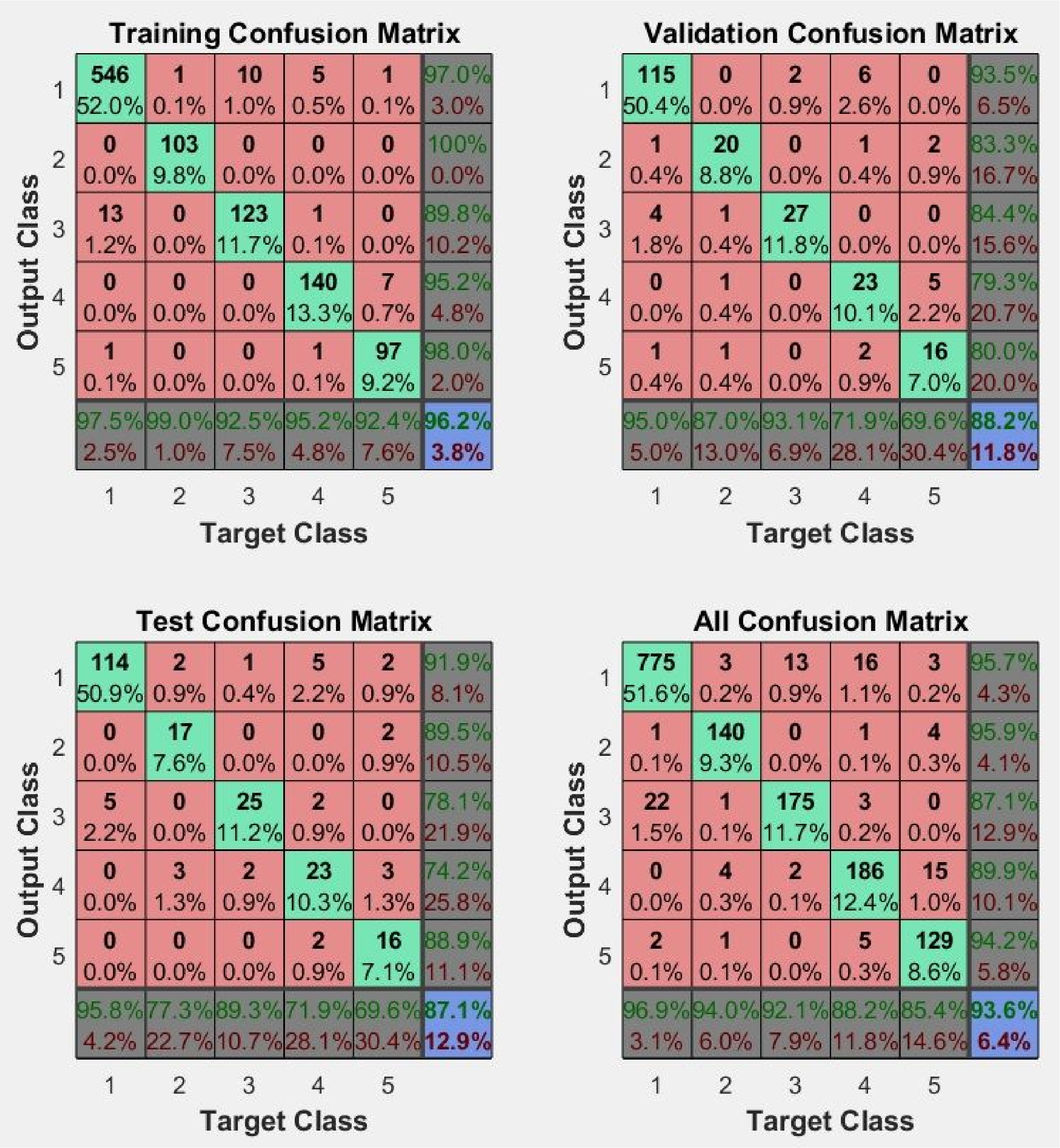
Output Class \ Target Class	1	2	3	4	5	
1	117 52.0%	2 0.9%	2 0.9%	2 0.9%	8 3.6%	89.3% 10.7%
2	1 0.4%	15 6.7%	0 0.0%	0 0.0%	0 0.0%	93.8% 6.3%
3	0 0.0%	0 0.0%	22 9.8%	0 0.0%	0 0.0%	100% 0.0%
4	1 0.4%	2 0.9%	4 1.8%	28 12.4%	1 0.4%	77.8% 22.2%
5	1 0.4%	3 1.3%	0 0.0%	2 0.9%	14 6.2%	70.0% 30.0%
	97.5% 2.5%	68.2% 31.8%	78.6% 21.4%	87.5% 12.5%	60.9% 39.1%	87.1% 12.9%

MLFFNN



Epoch = 1000 Iterations
Performance = 0.0528
Gradient = 0.00531

Best Validation Performance is 0.12491 at epoch 515



Training Confusion Matrix

Output Class

1	546 52.0%	1 0.1%	10 1.0%	5 0.5%	1 0.1%	97.0% 3.0%
2	0 0.0%	103 9.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	13 1.2%	0 0.0%	123 11.7%	1 0.1%	0 0.0%	89.8% 10.2%
4	0 0.0%	0 0.0%	0 0.0%	140 13.3%	7 0.7%	95.2% 4.8%
5	1 0.1%	0 0.0%	0 0.0%	1 0.1%	97 9.2%	98.0% 2.0%
	97.5% 2.5%	99.0% 1.0%	92.5% 7.5%	95.2% 4.8%	92.4% 7.6%	**96.2%** **3.8%**
	1	2	3	4	5	

Target Class

Validation Confusion Matrix

Output Class

1	115 50.4%	0 0.0%	2 0.9%	6 2.6%	0 0.0%	93.5% 6.5%
2	1 0.4%	20 8.8%	0 0.0%	1 0.4%	2 0.9%	83.3% 16.7%
3	4 1.8%	1 0.4%	27 11.8%	0 0.0%	0 0.0%	84.4% 15.6%
4	0 0.0%	1 0.4%	0 0.0%	23 10.1%	5 2.2%	79.3% 20.7%
5	1 0.4%	1 0.4%	0 0.0%	2 0.9%	16 7.0%	80.0% 20.0%
	95.0% 5.0%	87.0% 13.0%	93.1% 6.9%	71.9% 28.1%	69.6% 30.4%	**88.2%** **11.8%**
	1	2	3	4	5	

Target Class

Test Confusion Matrix

Output Class

1	114 50.9%	2 0.9%	1 0.4%	5 2.2%	2 0.9%	91.9% 8.1%
2	0 0.0%	17 7.6%	0 0.0%	0 0.0%	2 0.9%	89.5% 10.5%
3	5 2.2%	0 0.0%	25 11.2%	2 0.9%	0 0.0%	78.1% 21.9%
4	0 0.0%	3 1.3%	2 0.9%	23 10.3%	3 1.3%	74.2% 25.8%
5	0 0.0%	0 0.0%	0 0.0%	2 0.9%	16 7.1%	88.9% 11.1%
	95.8% 4.2%	77.3% 22.7%	89.3% 10.7%	71.9% 28.1%	69.6% 30.4%	**87.1%** **12.9%**
	1	2	3	4	5	

Target Class

All Confusion Matrix

Output Class

1	775 51.6%	3 0.2%	13 0.9%	16 1.1%	3 0.2%	95.7% 4.3%
2	1 0.1%	140 9.3%	0 0.0%	1 0.1%	4 0.3%	95.9% 4.1%
3	22 1.5%	1 0.1%	175 11.7%	3 0.2%	0 0.0%	87.1% 12.9%
4	0 0.0%	4 0.3%	2 0.1%	186 12.4%	15 1.0%	89.9% 10.1%
5	2 0.1%	1 0.1%	0 0.0%	5 0.3%	129 8.6%	94.2% 5.8%
	96.9% 3.1%	94.0% 6.0%	92.1% 7.9%	88.2% 11.8%	85.4% 14.6%	**93.6%** **6.4%**
	1	2	3	4	5	

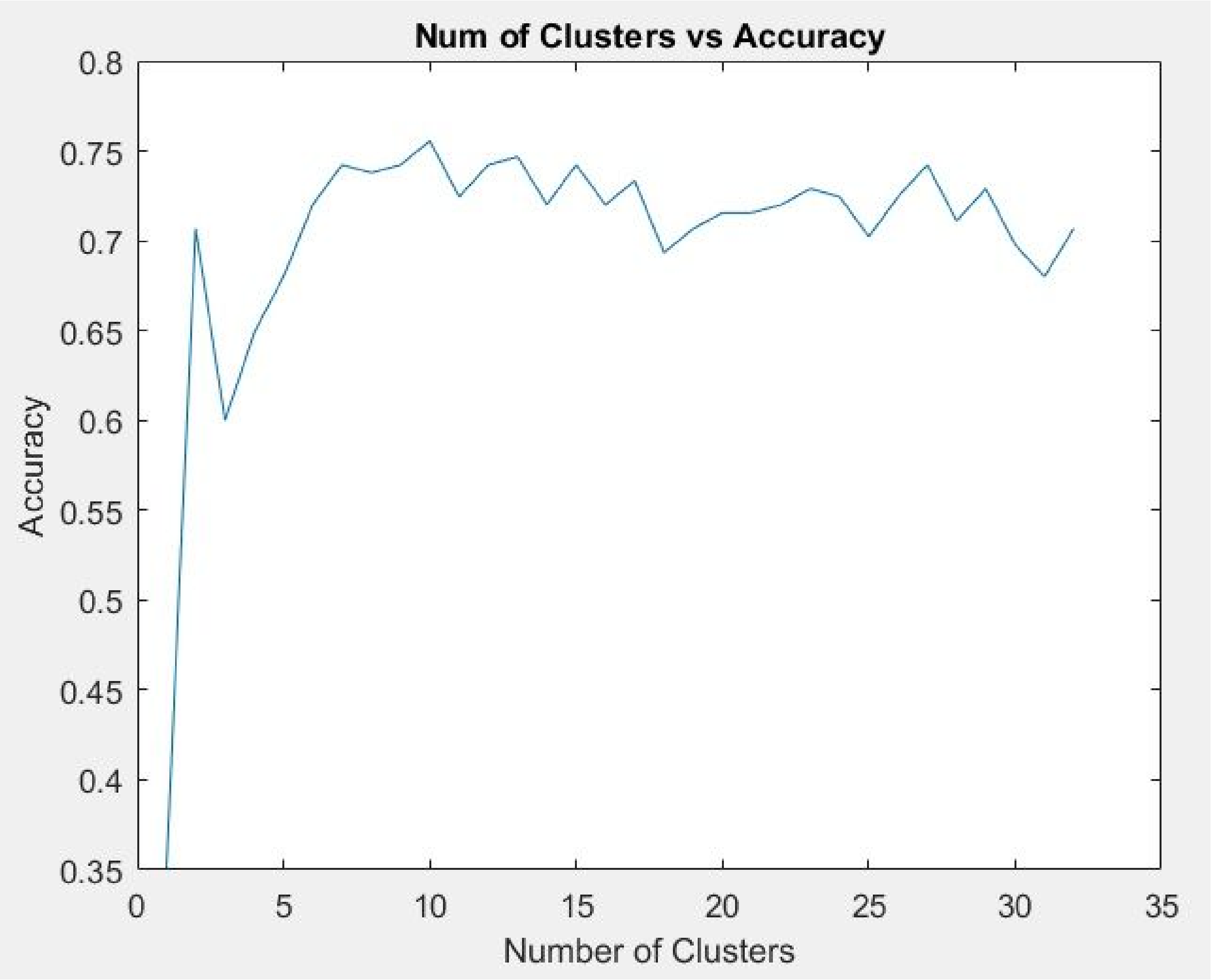
Target Class

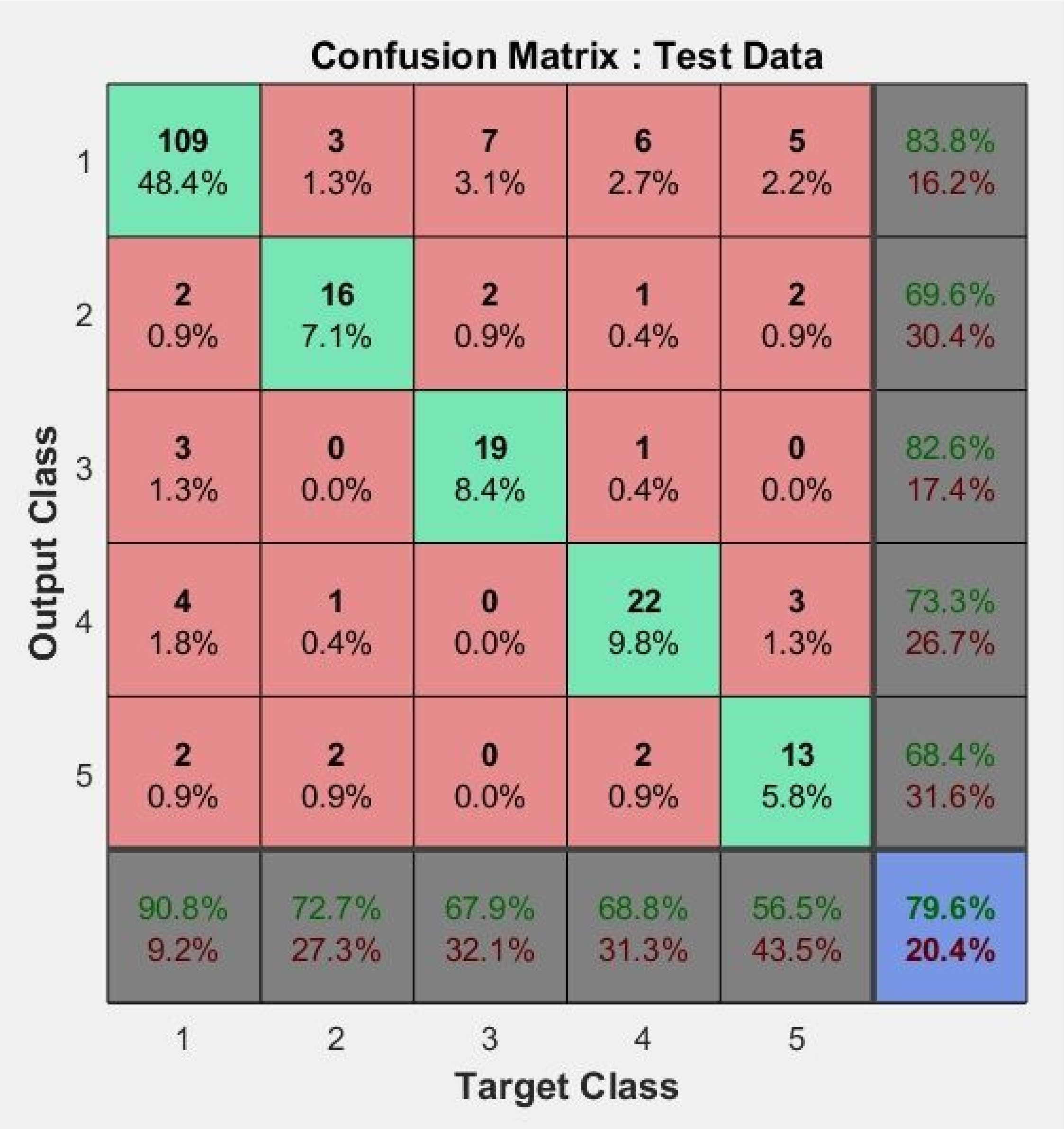
Image Dataset

5 Dimensions

90.0455% Variance Retained

GMM





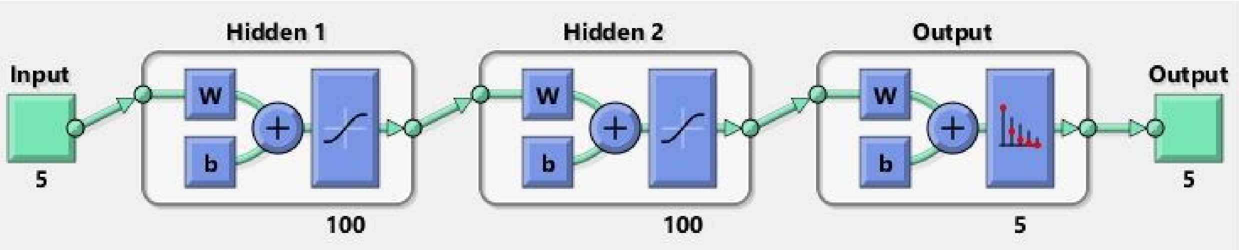
Confusion Matrix : Test Data

Output Class

Target Class

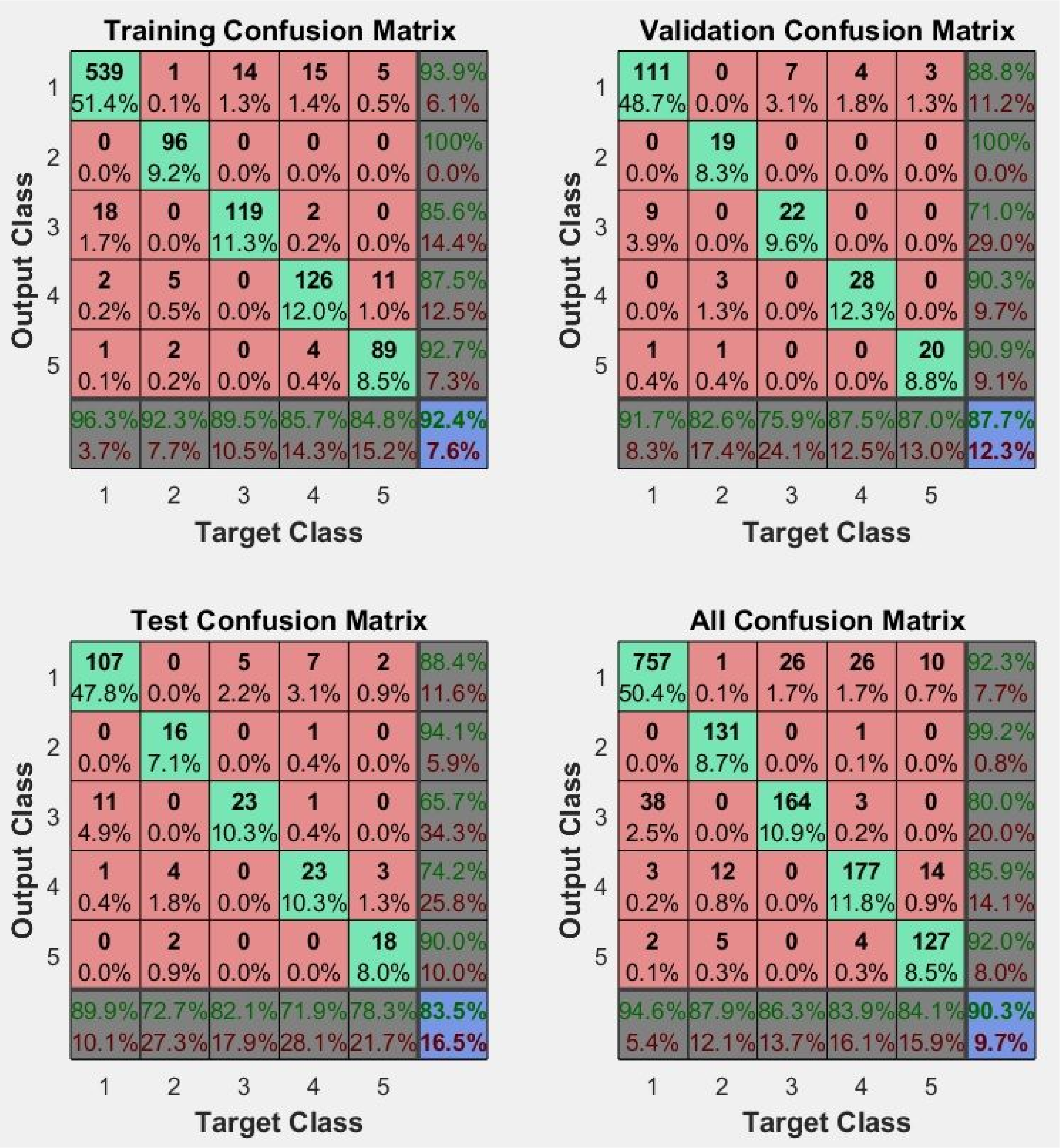
	1	2	3	4	5	
1	**109** 48.4%	**3** 1.3%	**7** 3.1%	**6** 2.7%	**5** 2.2%	83.8% 16.2%
2	**2** 0.9%	**16** 7.1%	**2** 0.9%	**1** 0.4%	**2** 0.9%	69.6% 30.4%
3	**3** 1.3%	**0** 0.0%	**19** 8.4%	**1** 0.4%	**0** 0.0%	82.6% 17.4%
4	**4** 1.8%	**1** 0.4%	**0** 0.0%	**22** 9.8%	**3** 1.3%	73.3% 26.7%
5	**2** 0.9%	**2** 0.9%	**0** 0.0%	**2** 0.9%	**13** 5.8%	68.4% 31.6%
	90.8% 9.2%	72.7% 27.3%	67.9% 32.1%	68.8% 31.3%	56.5% 43.5%	**79.6%** **20.4%**

MLFFNN



Epoch = 1000 Iterations
Performance = 0.0706
Gradient = 0.0275

Best Validation Performance is 0.08308 at epoch 987


Output Class
Target Class

Training Confusion Matrix

	1	2	3	4	5	
1	539 51.4%	1 0.1%	14 1.3%	15 1.4%	5 0.5%	93.9% 6.1%
2	0 0.0%	96 9.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	18 1.7%	0 0.0%	119 11.3%	2 0.2%	0 0.0%	85.6% 14.4%
4	2 0.2%	5 0.5%	0 0.0%	126 12.0%	11 1.0%	87.5% 12.5%
5	1 0.1%	2 0.2%	0 0.0%	4 0.4%	89 8.5%	92.7% 7.3%
	96.3% 3.7%	92.3% 7.7%	89.5% 10.5%	85.7% 14.3%	84.8% 15.2%	92.4% 7.6%

Validation Confusion Matrix

	1	2	3	4	5	
1	111 48.7%	0 0.0%	7 3.1%	4 1.8%	3 1.3%	88.8% 11.2%
2	0 0.0%	19 8.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	9 3.9%	0 0.0%	22 9.6%	0 0.0%	0 0.0%	71.0% 29.0%
4	0 0.0%	3 1.3%	0 0.0%	28 12.3%	0 0.0%	90.3% 9.7%
5	1 0.4%	1 0.4%	0 0.0%	0 0.0%	20 8.8%	90.9% 9.1%
	91.7% 8.3%	82.6% 17.4%	75.9% 24.1%	87.5% 12.5%	87.0% 13.0%	87.7% 12.3%

Test Confusion Matrix

	1	2	3	4	5	
1	107 47.8%	0 0.0%	5 2.2%	7 3.1%	2 0.9%	88.4% 11.6%
2	0 0.0%	16 7.1%	0 0.0%	1 0.4%	0 0.0%	94.1% 5.9%
3	11 4.9%	0 0.0%	23 10.3%	1 0.4%	0 0.0%	65.7% 34.3%
4	1 0.4%	4 1.8%	0 0.0%	23 10.3%	3 1.3%	74.2% 25.8%
5	0 0.0%	2 0.9%	0 0.0%	0 0.0%	18 8.0%	90.0% 10.0%
	89.9% 10.1%	72.7% 27.3%	82.1% 17.9%	71.9% 28.1%	78.3% 21.7%	83.5% 16.5%

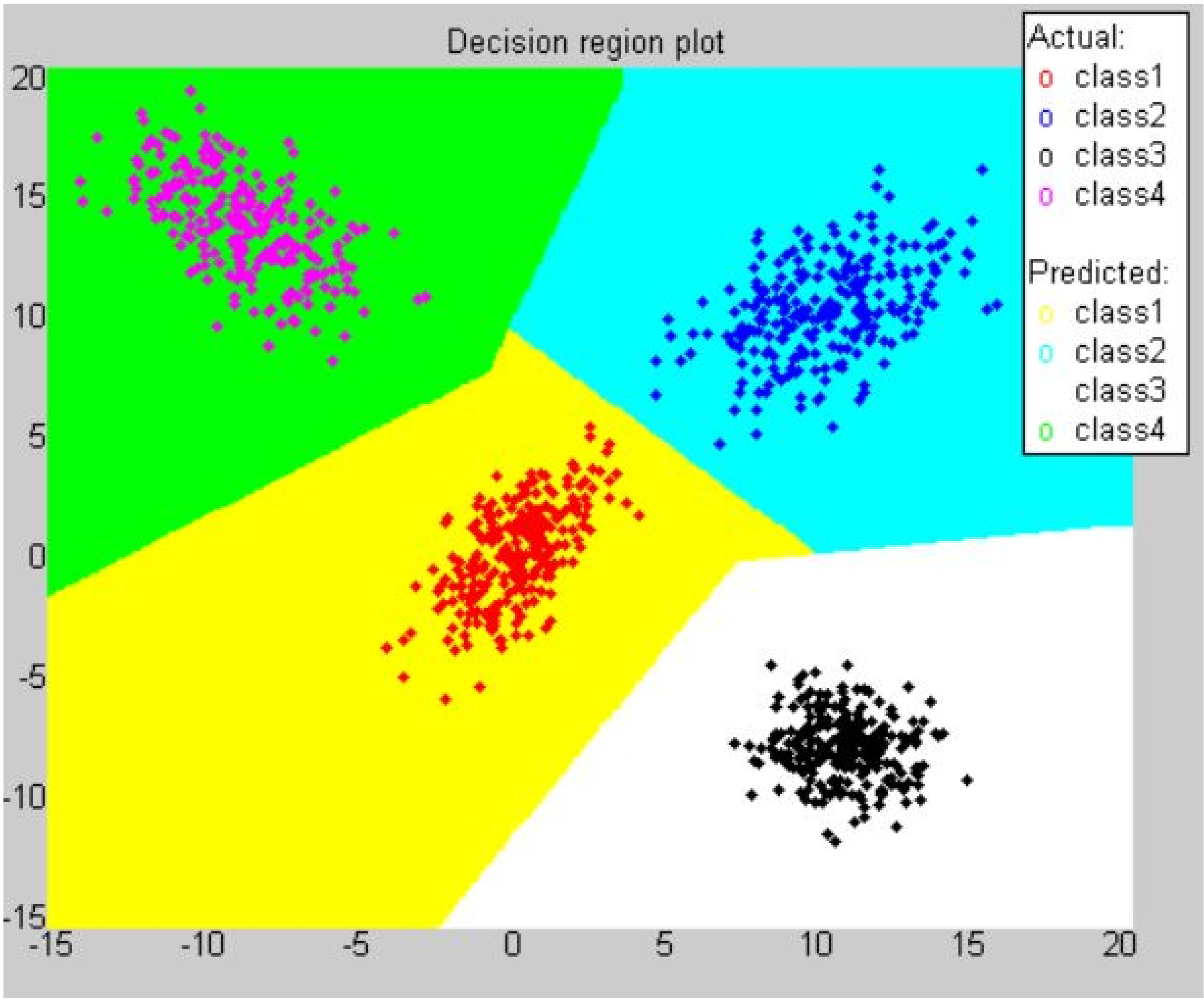
All Confusion Matrix

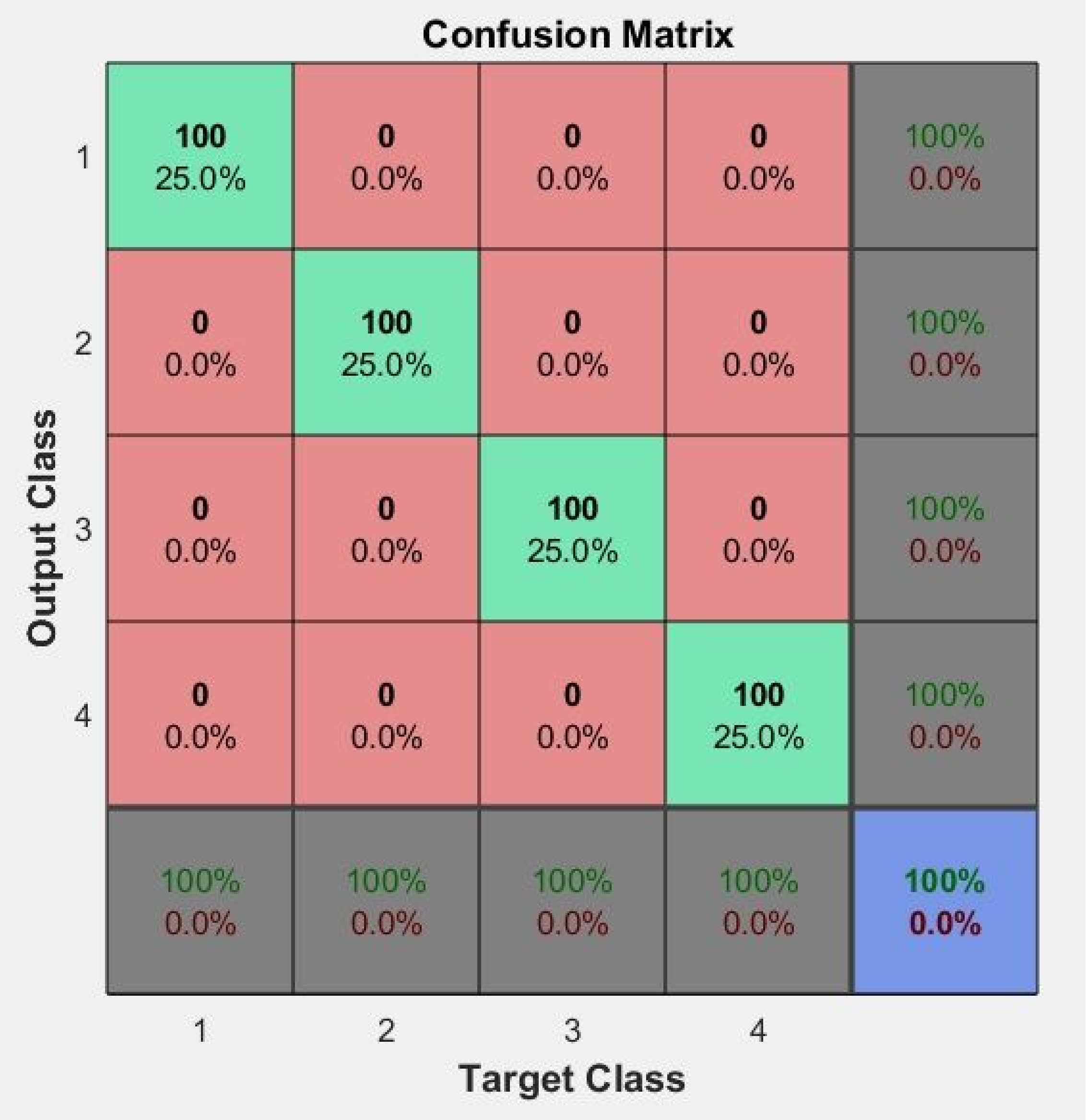
	1	2	3	4	5	
1	757 50.4%	1 0.1%	26 1.7%	26 1.7%	10 0.7%	92.3% 7.7%
2	0 0.0%	131 8.7%	0 0.0%	1 0.1%	0 0.0%	99.2% 0.8%
3	38 2.5%	0 0.0%	164 10.9%	3 0.2%	0 0.0%	80.0% 20.0%
4	3 0.2%	12 0.8%	0 0.0%	177 11.8%	14 0.9%	85.9% 14.1%
5	2 0.1%	5 0.3%	0 0.0%	4 0.3%	127 8.5%	92.0% 8.0%
	94.6% 5.4%	87.9% 12.1%	86.3% 13.7%	83.9% 16.1%	84.1% 15.9%	90.3% 9.7%

Data-set 1

Linear C-SVM on Linearly Separable Data

Decision Surface takes form of Hyper-planes

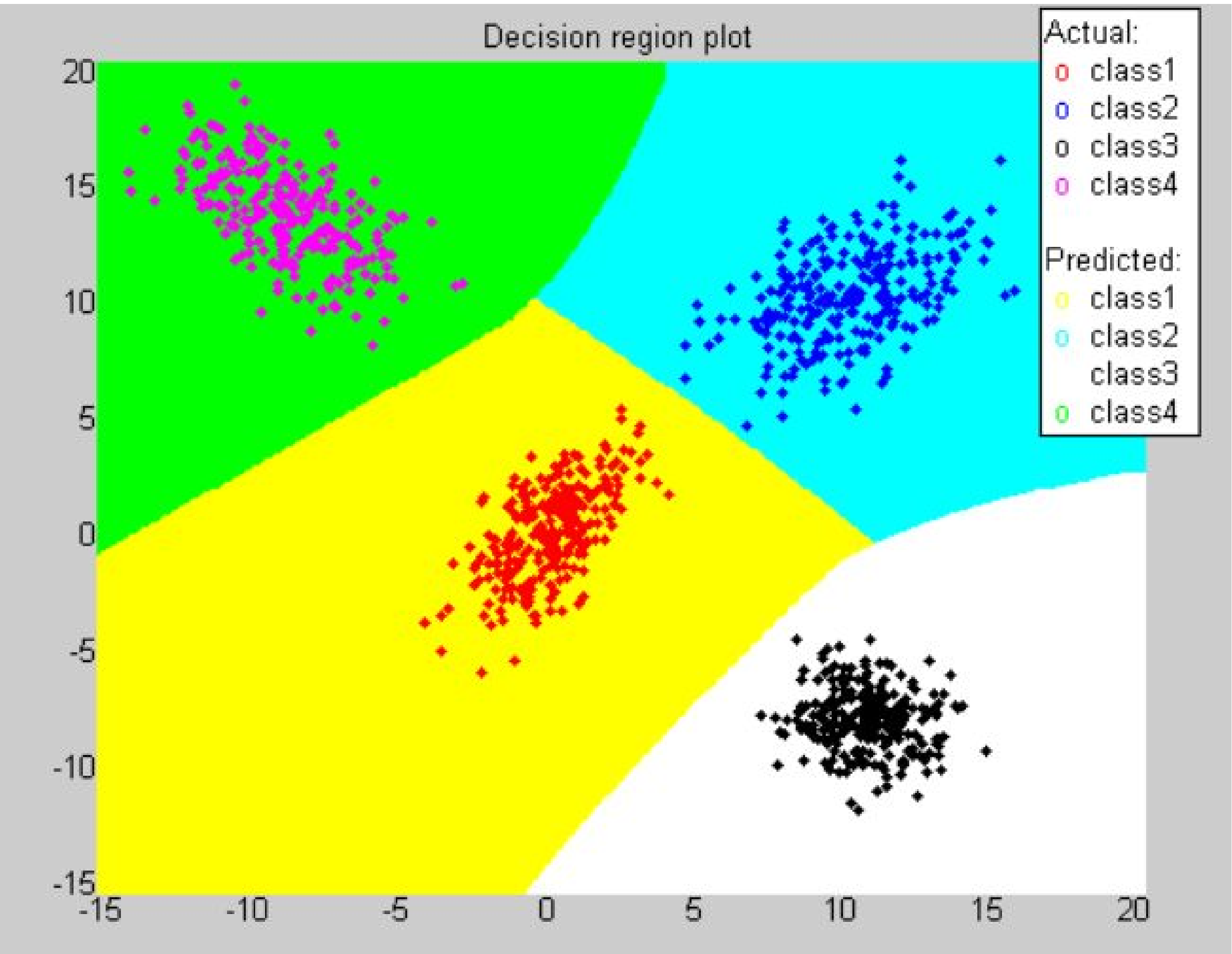


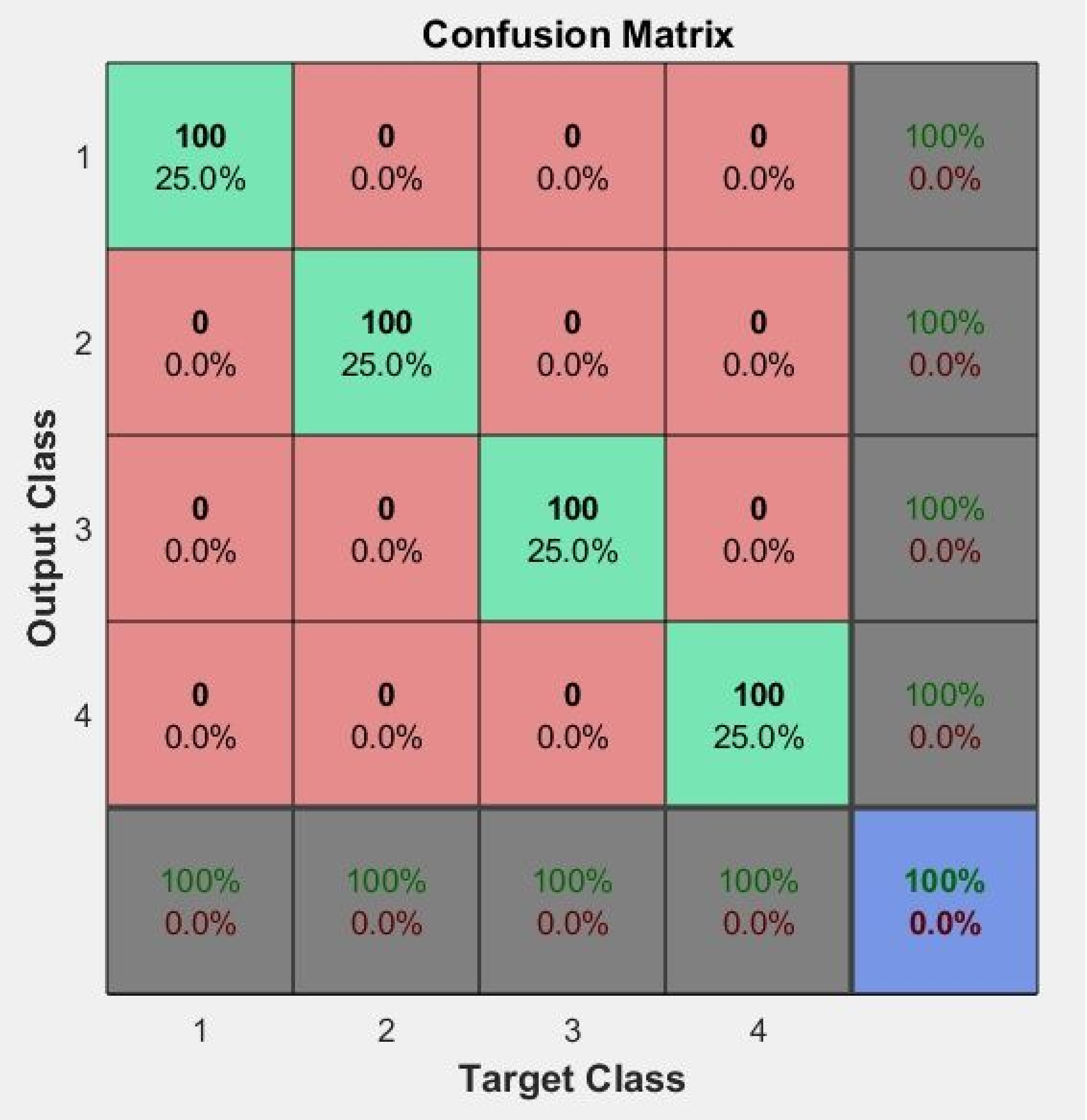


	1	2	3	4	
1	**100** 25.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**100** 25.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
3	**0** 0.0%	**0** 0.0%	**100** 25.0%	**0** 0.0%	100% 0.0%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**100** 25.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	**100%** **0.0%**

Polynomial Kernel

Degree = 3 Cost = 64



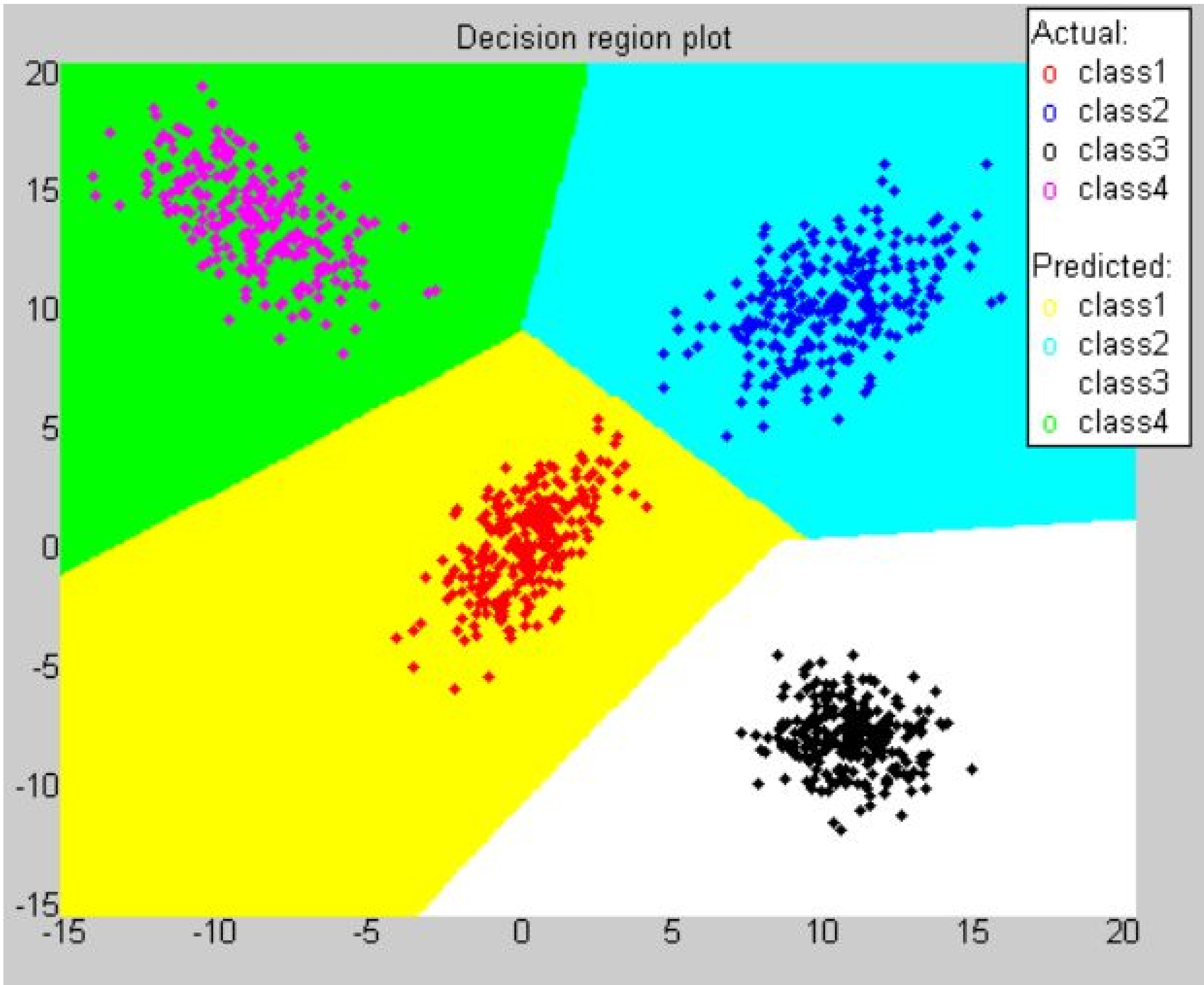


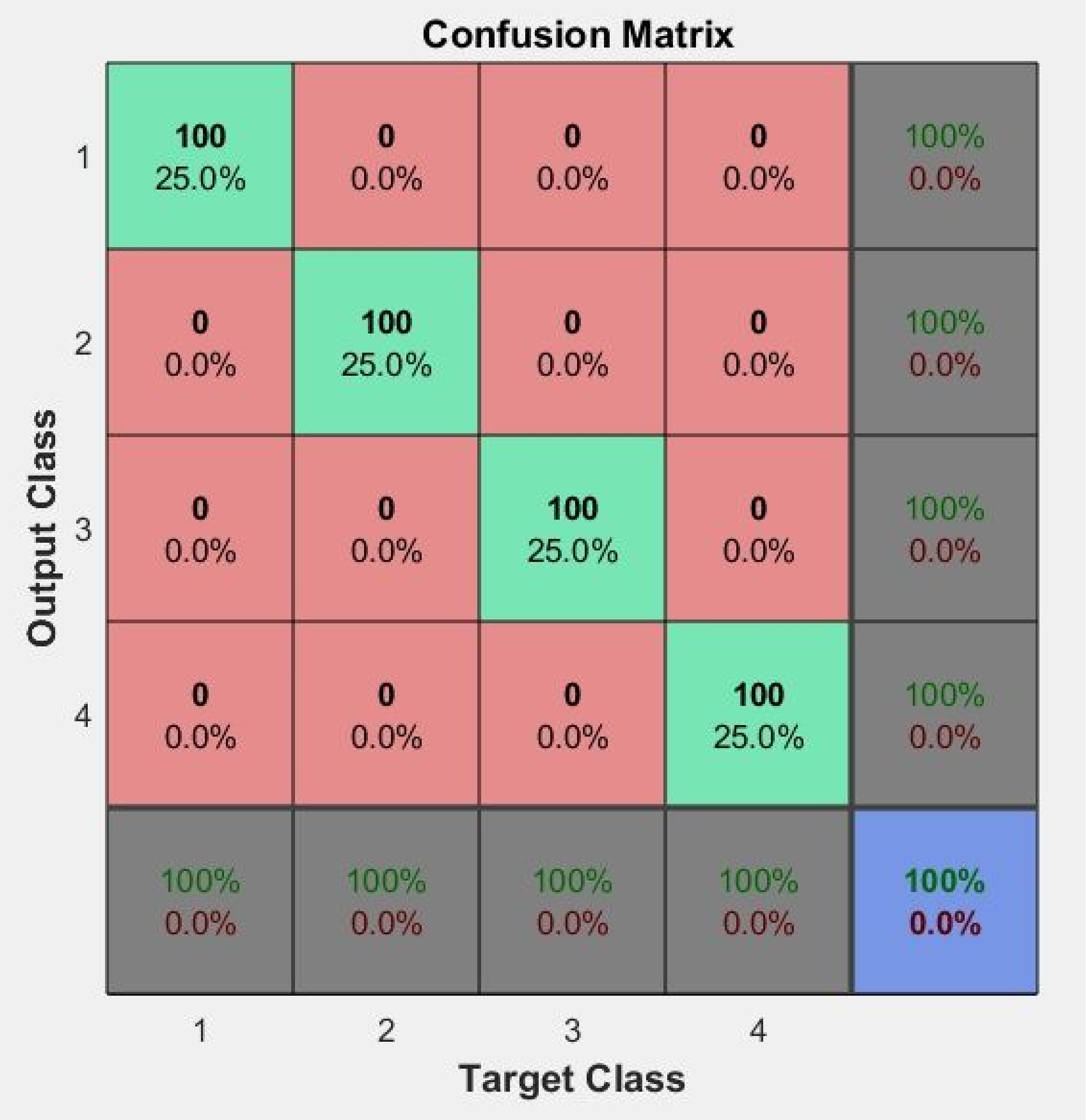
Confusion Matrix

	1	2	3	4	
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%**

RBF

Gamma = 0.25 Cost =0.125





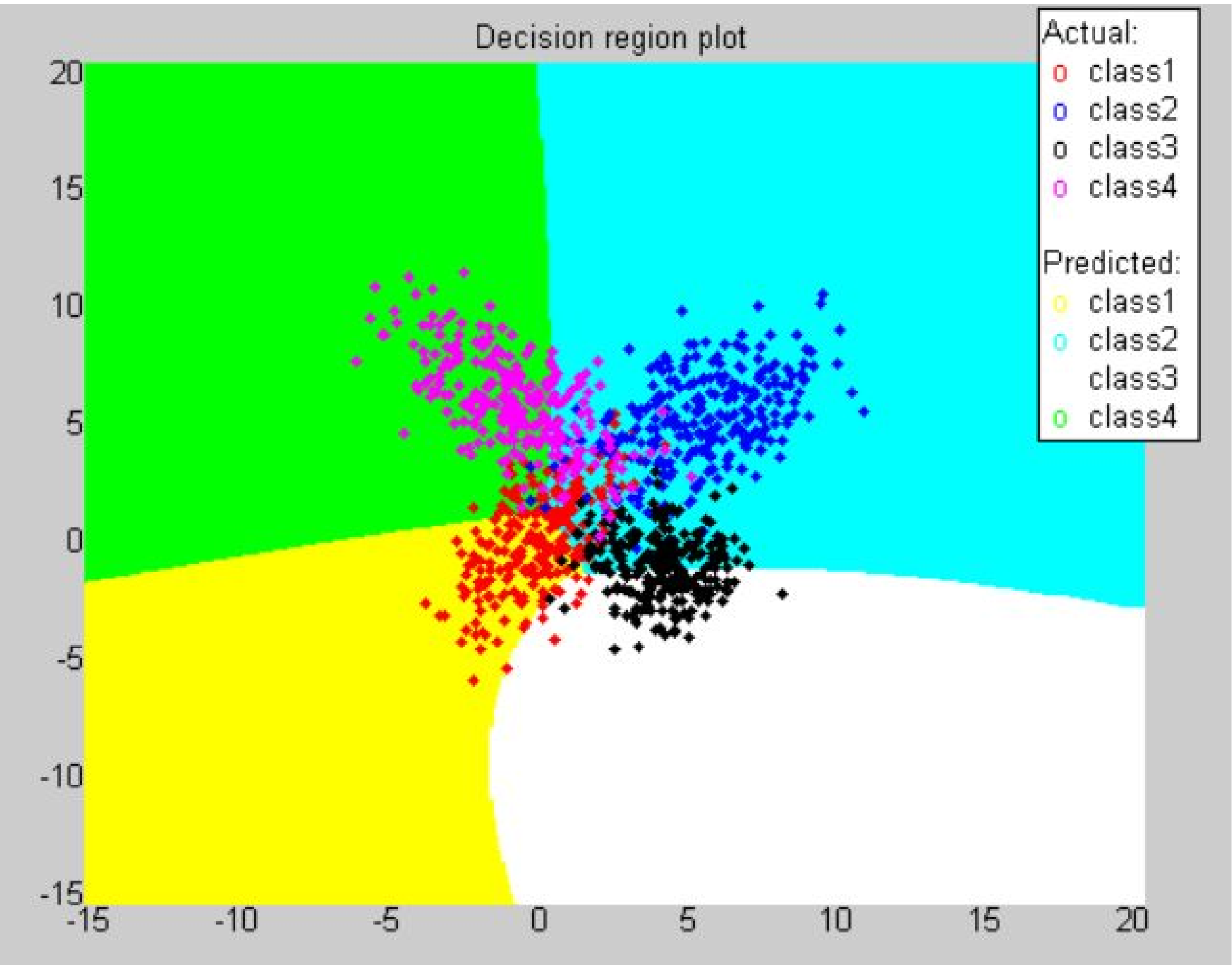
	1	2	3	4	
1	100 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	100 25.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	100 25.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	100 25.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

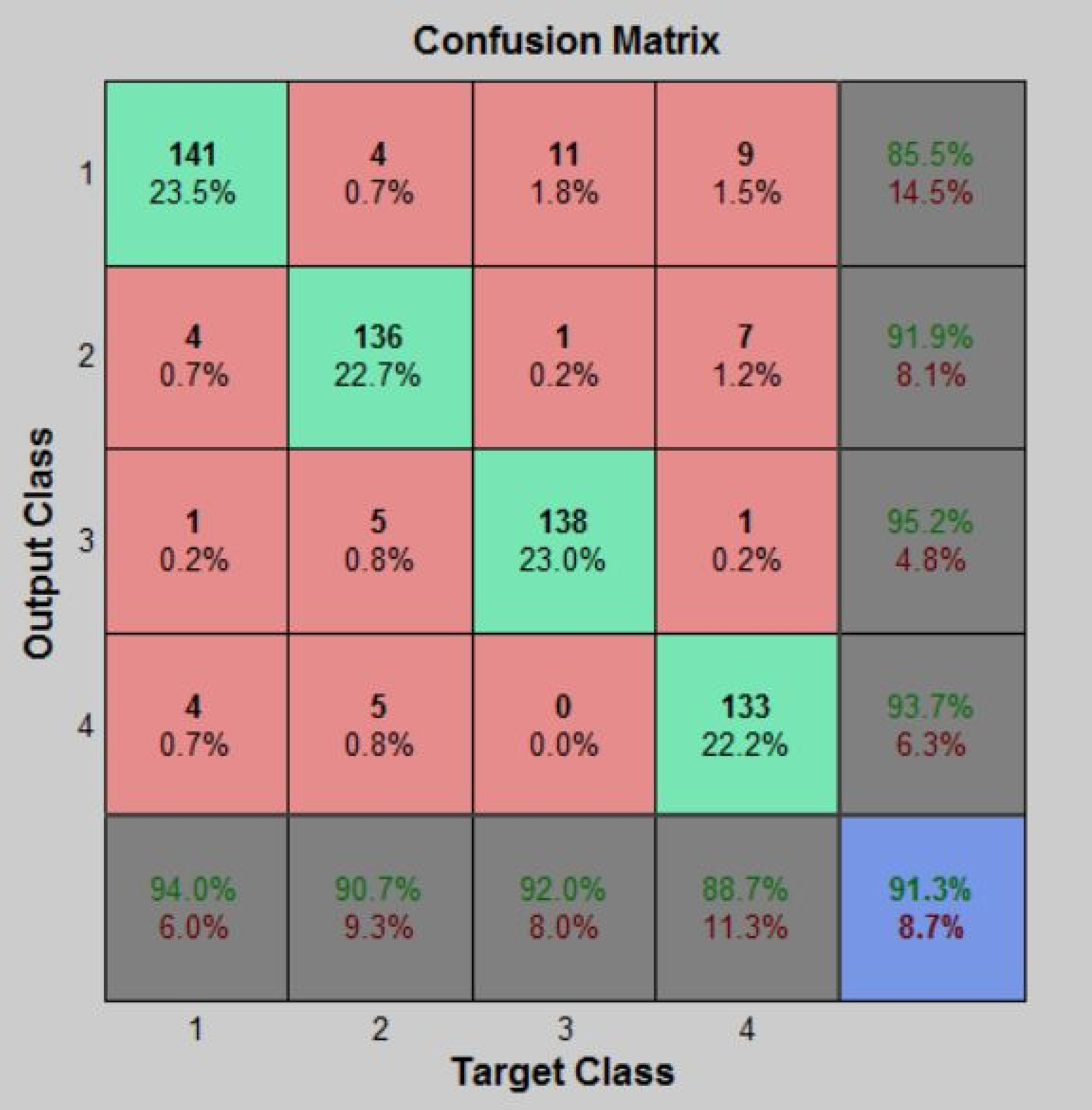
Overlapping Data

Polynomial Function

Degree=2 C=64 Coeff=1

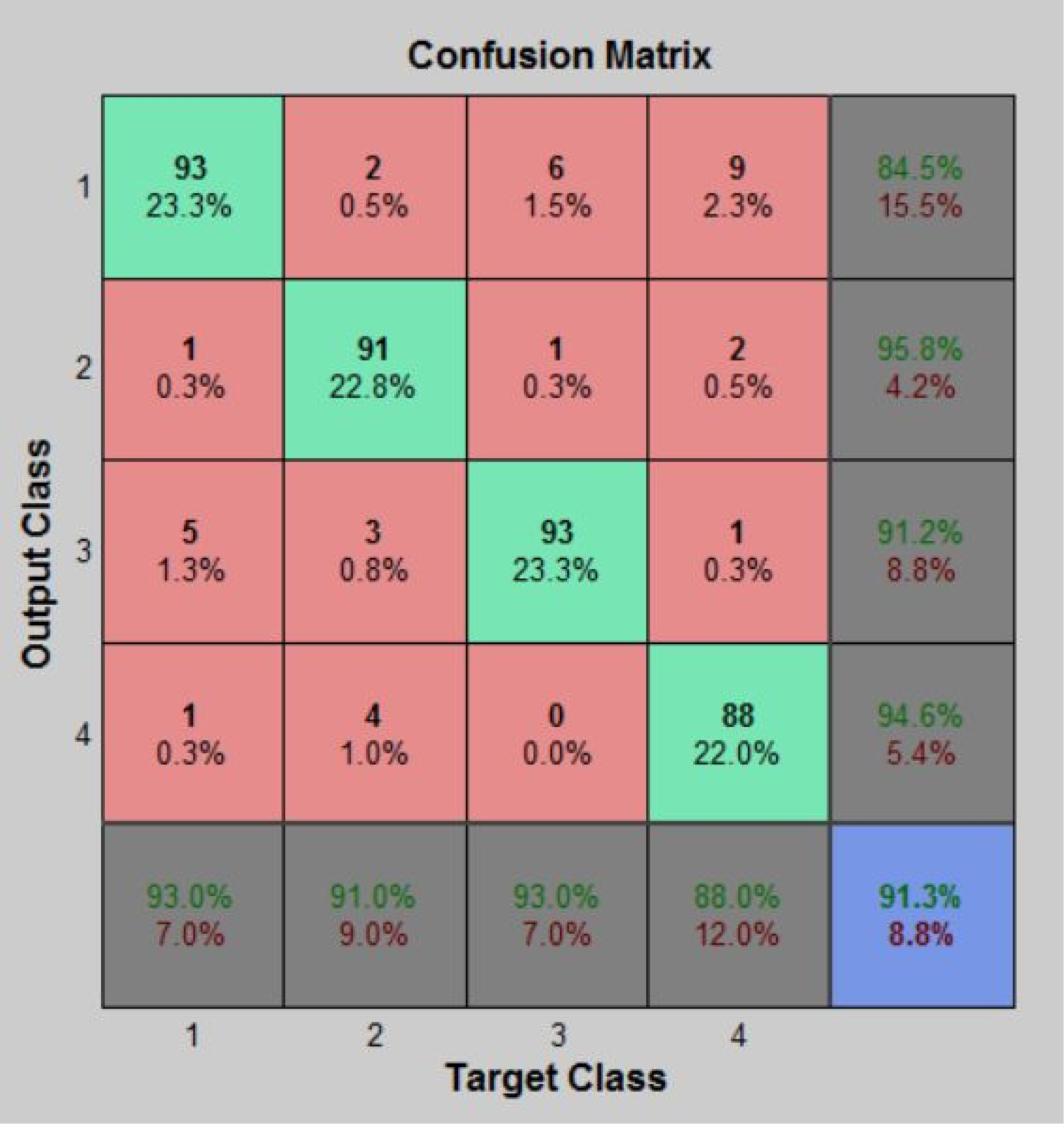
Training Data : Accuracy = 89.6% (896/1000) (classification)
Validation Data : Accuracy = 91.3333% (548/600) (classification)
Test Data Accuracy = 91.25% (365/400) (classification)





	1	2	3	4	
1	141 23.5%	4 0.7%	11 1.8%	9 1.5%	85.5% 14.5%
2	4 0.7%	136 22.7%	1 0.2%	7 1.2%	91.9% 8.1%
3	1 0.2%	5 0.8%	138 23.0%	1 0.2%	95.2% 4.8%
4	4 0.7%	5 0.8%	0 0.0%	133 22.2%	93.7% 6.3%
	94.0% 6.0%	90.7% 9.3%	92.0% 8.0%	88.7% 11.3%	91.3% 8.7%

Confusion Matrix For Validation Data



Confusion Matrix

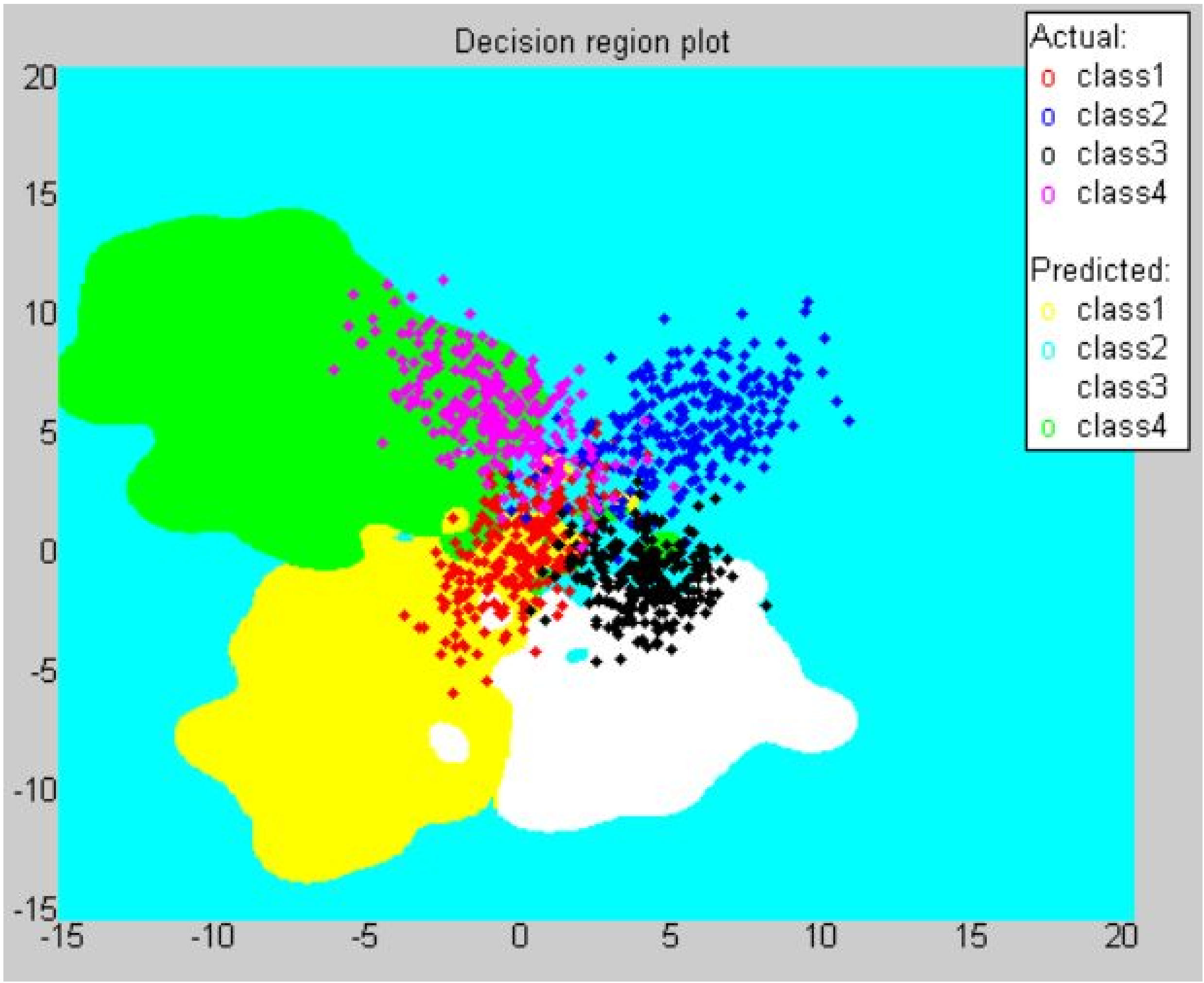
Output Class

	1	2	3	4	
1	93 23.3%	2 0.5%	6 1.5%	9 2.3%	84.5% 15.5%
2	1 0.3%	91 22.8%	1 0.3%	2 0.5%	95.8% 4.2%
3	5 1.3%	3 0.8%	93 23.3%	1 0.3%	91.2% 8.8%
4	1 0.3%	4 1.0%	0 0.0%	88 22.0%	94.6% 5.4%
	93.0% 7.0%	91.0% 9.0%	93.0% 7.0%	88.0% 12.0%	91.3% 8.8%

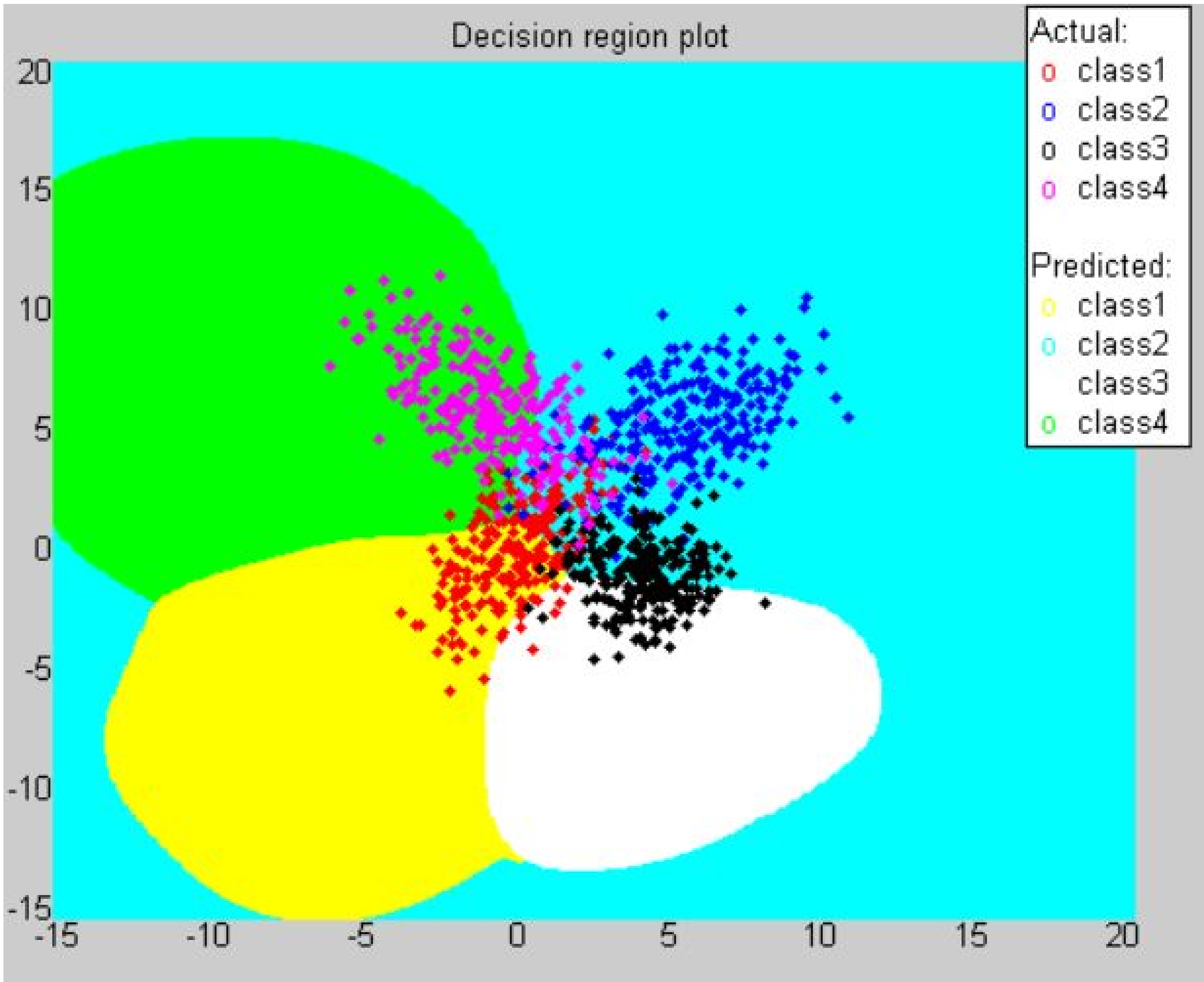
Target Class

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)

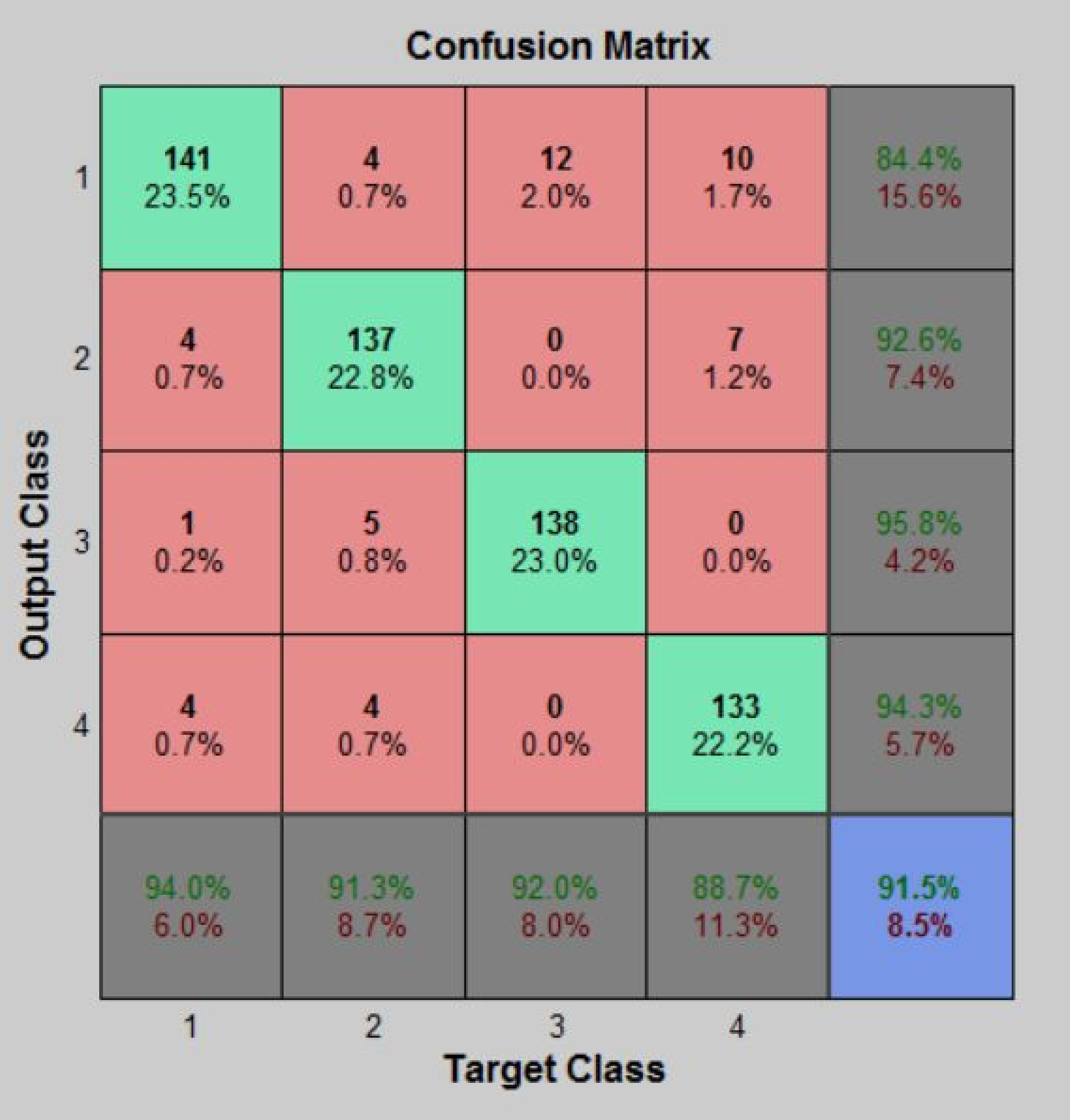


Gamma = 16 Cost = 0.25

Training Data Accuracy = 89.7% (897/1000) (classification)(To check for Over-fiting)

Validation Data Accuracy = 91.5% (549/600) (classification)

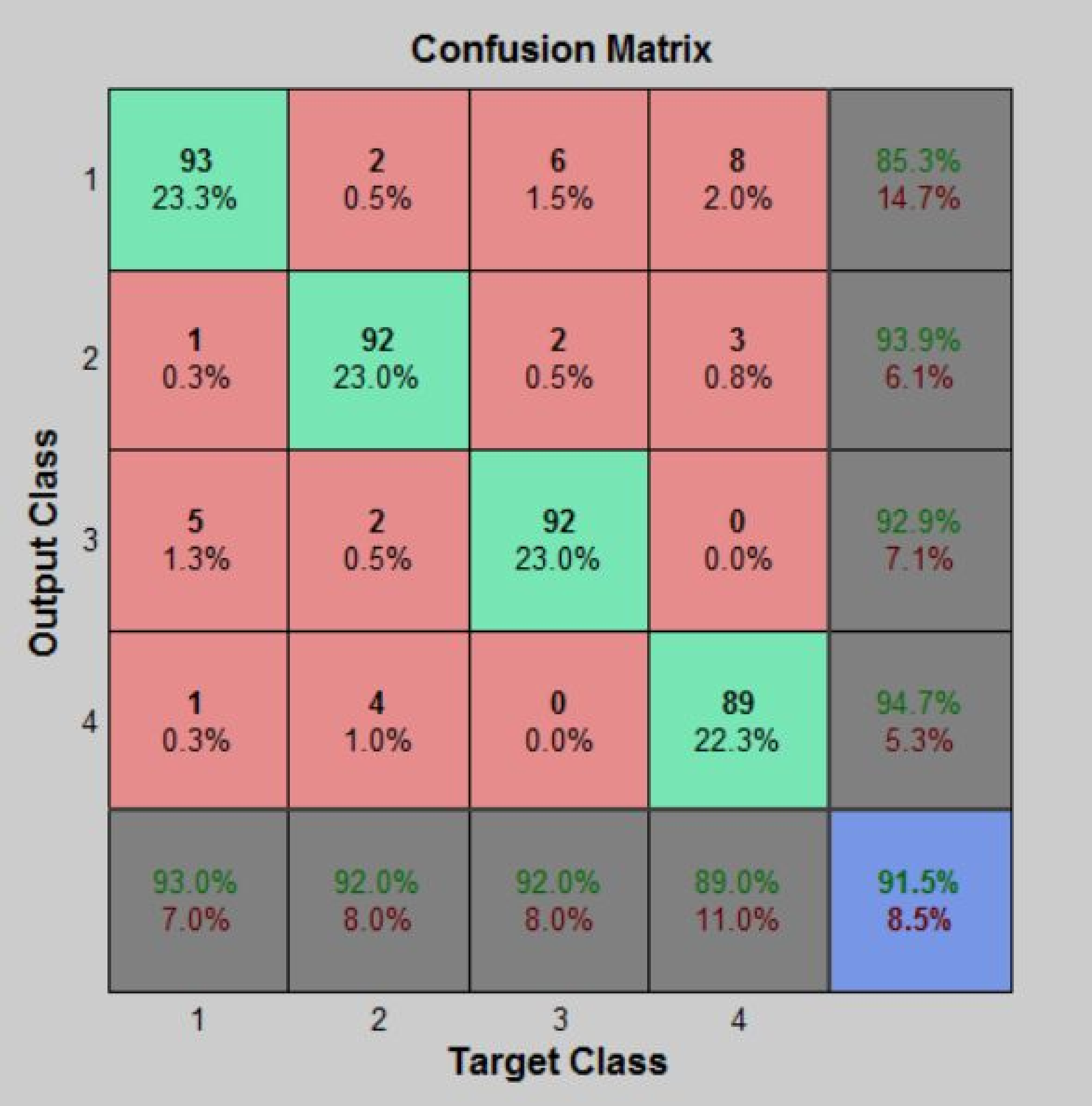
Testing Data Accuracy = 91.5% (366/400) (classification)



Confusion Matrix

Output Class \ Target Class	1	2	3	4	
1	**141** 23.5%	**4** 0.7%	**12** 2.0%	**10** 1.7%	84.4% 15.6%
2	**4** 0.7%	**137** 22.8%	**0** 0.0%	**7** 1.2%	92.6% 7.4%
3	**1** 0.2%	**5** 0.8%	**138** 23.0%	**0** 0.0%	95.8% 4.2%
4	**4** 0.7%	**4** 0.7%	**0** 0.0%	**133** 22.2%	94.3% 5.7%
	94.0% 6.0%	91.3% 8.7%	92.0% 8.0%	88.7% 11.3%	**91.5%** **8.5%**

Confusion Matrix for Validation Data



Confusion Matrix

Output Class	1	2	3	4	
1	93 23.3%	2 0.5%	6 1.5%	8 2.0%	85.3% 14.7%
2	1 0.3%	92 23.0%	2 0.5%	3 0.8%	93.9% 6.1%
3	5 1.3%	2 0.5%	92 23.0%	0 0.0%	92.9% 7.1%
4	1 0.3%	4 1.0%	0 0.0%	89 22.3%	94.7% 5.3%
	93.0% 7.0%	92.0% 8.0%	92.0% 8.0%	89.0% 11.0%	91.5% 8.5%

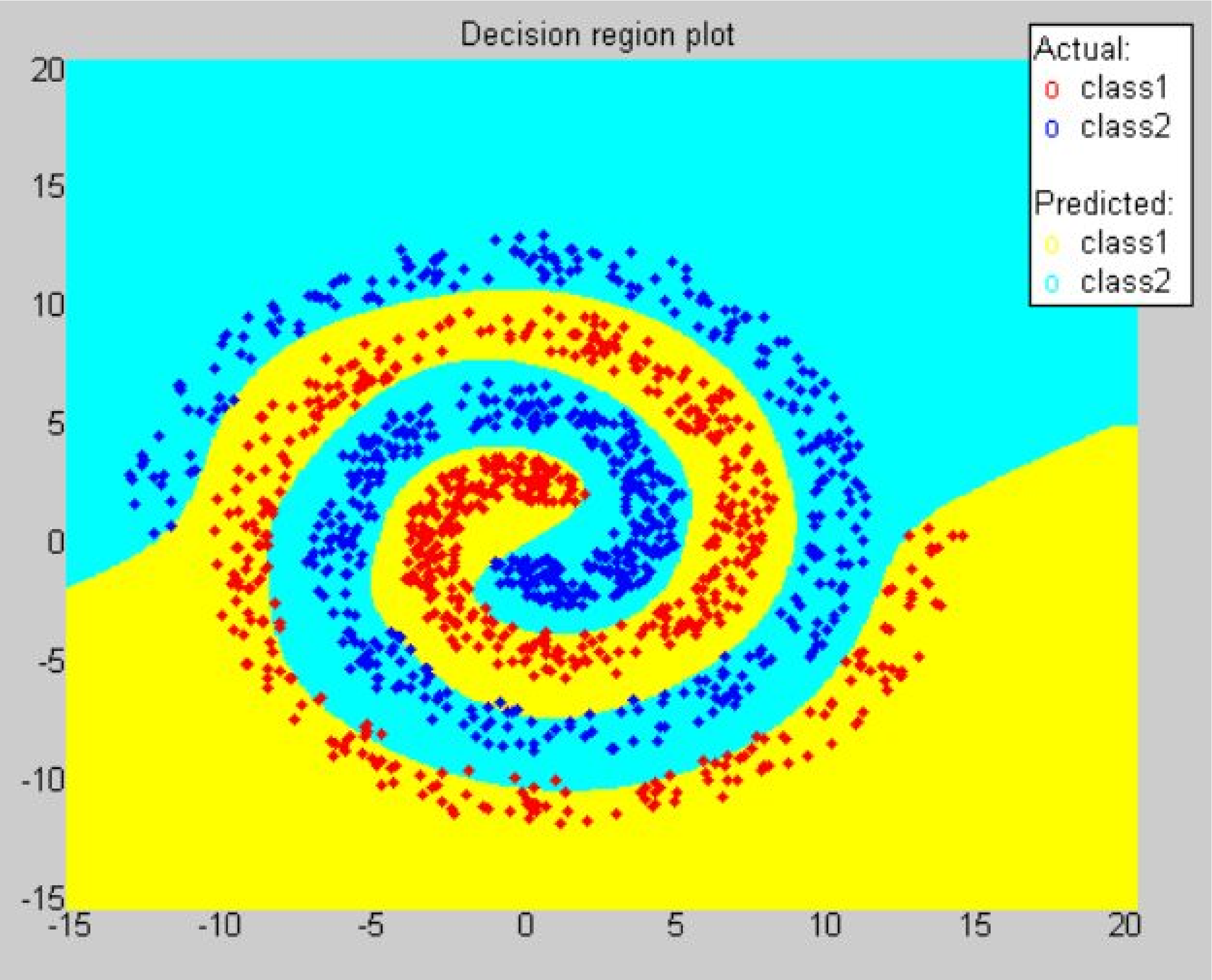
Target Class

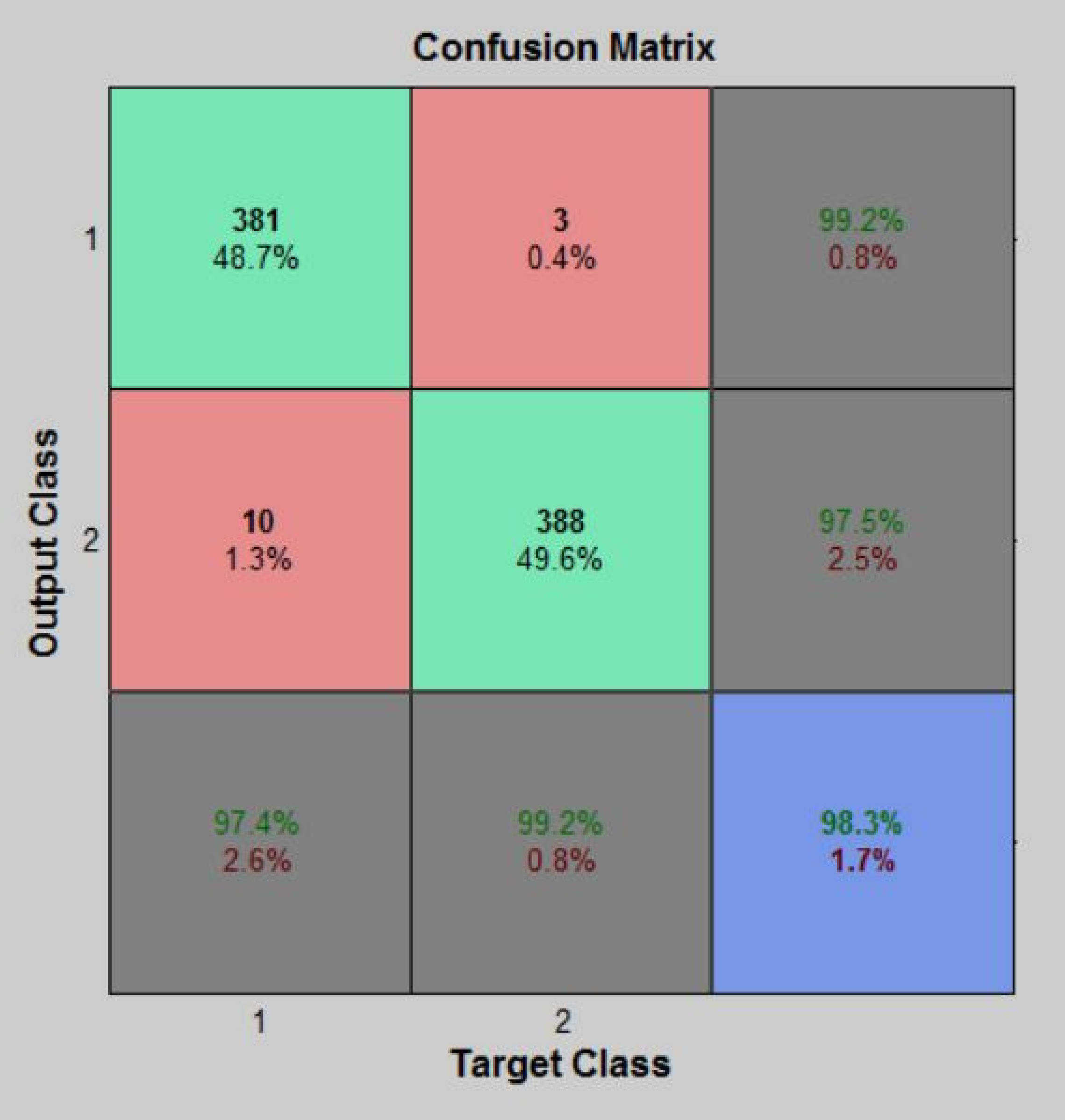
Confusion Matrix for Testing Data

Non-Linearly Separable Data

Polynomial Kernel Function

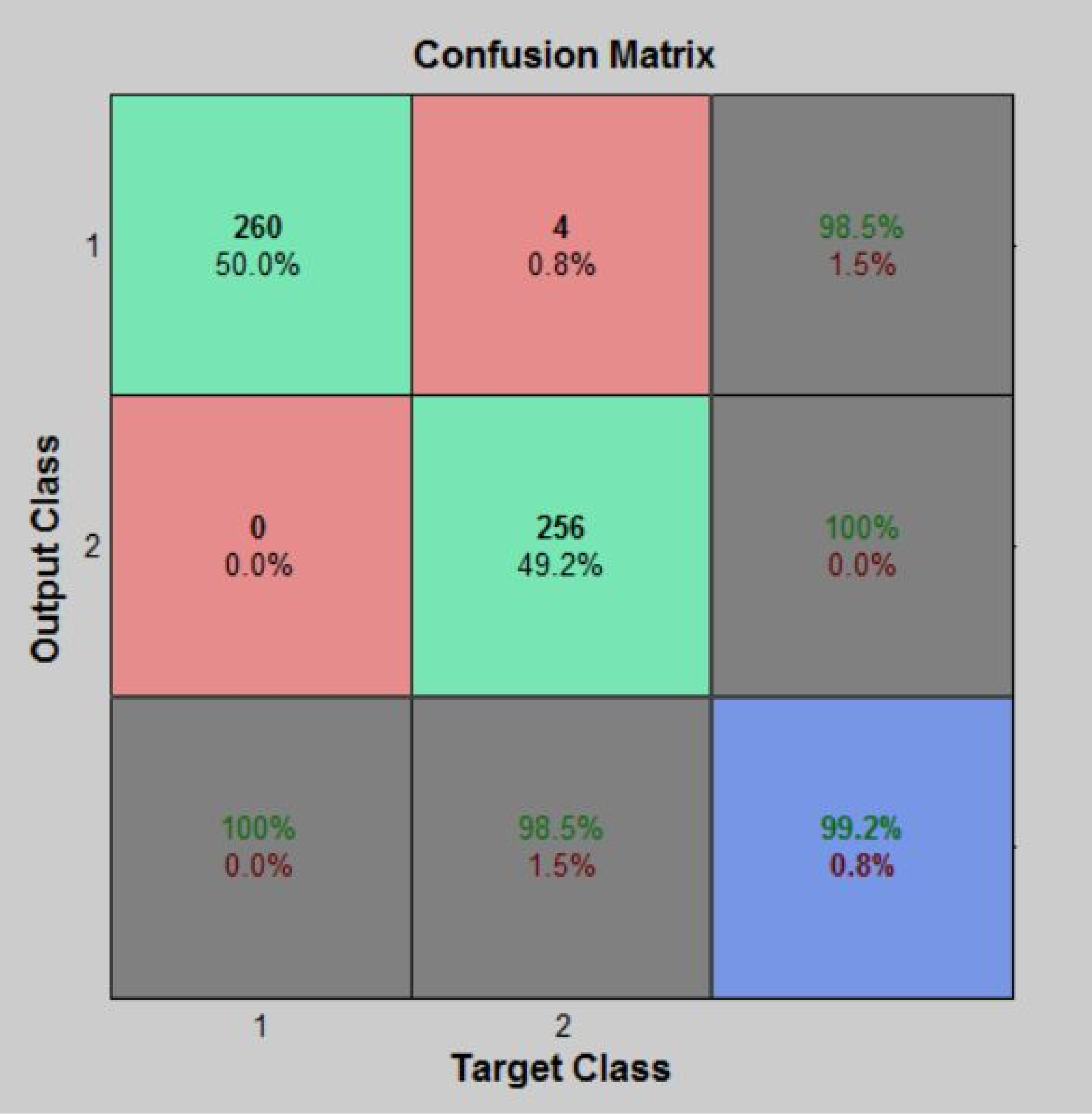
Degree=7 Cost =1024
Training Data : Accuracy = 74.3865% (970/1304) (classification)(Given to check for over-fitting)
Validation Data : Accuracy = 71.867% (562/782) (classification)
Testing Data : Accuracy = 73.2692% (381/520) (classification)





	1	2	
1	381 48.7%	3 0.4%	99.2% 0.8%
2	10 1.3%	388 49.6%	97.5% 2.5%
	97.4% 2.6%	99.2% 0.8%	98.3% 1.7%

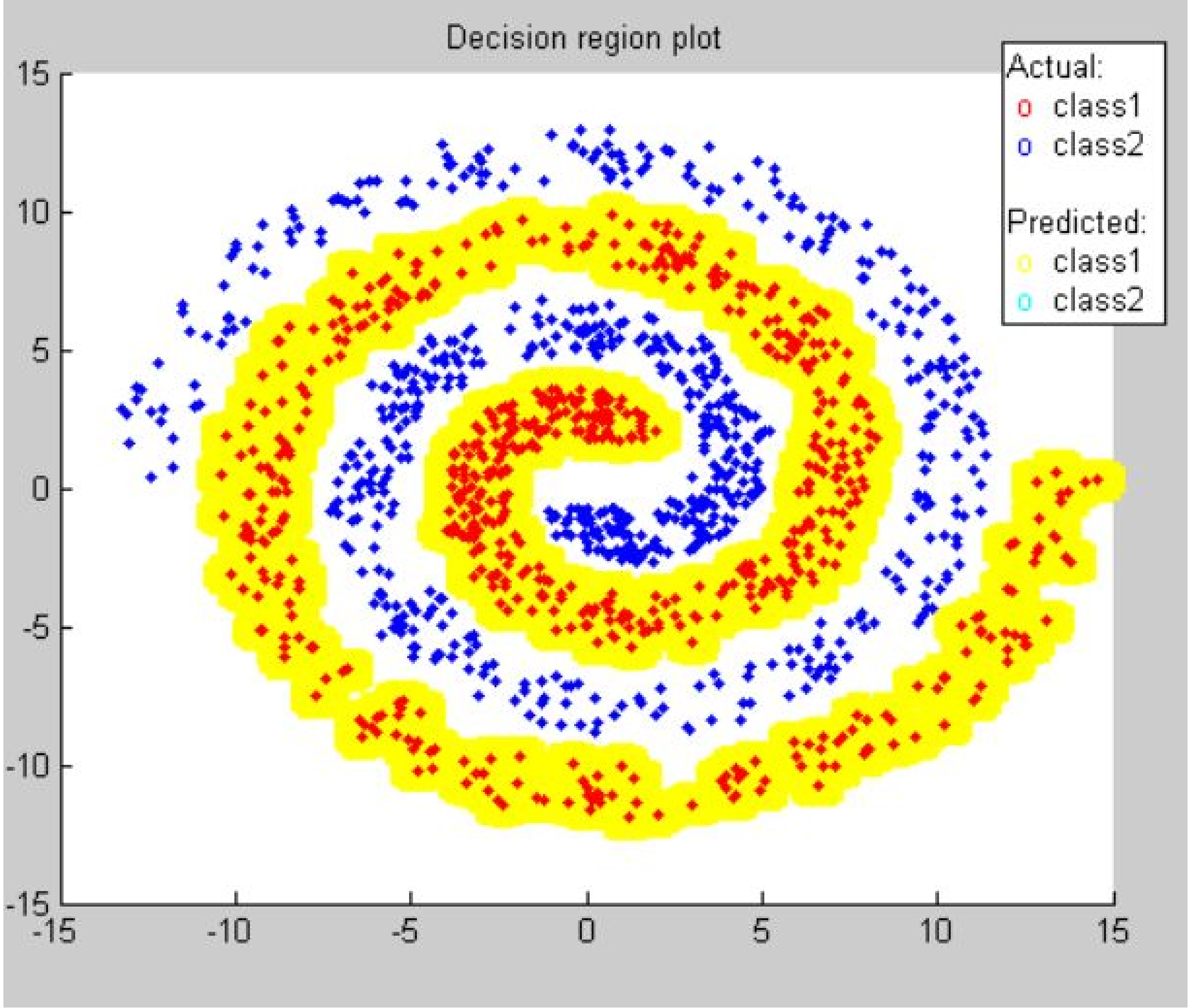
Confusion Matrix for Validation Data

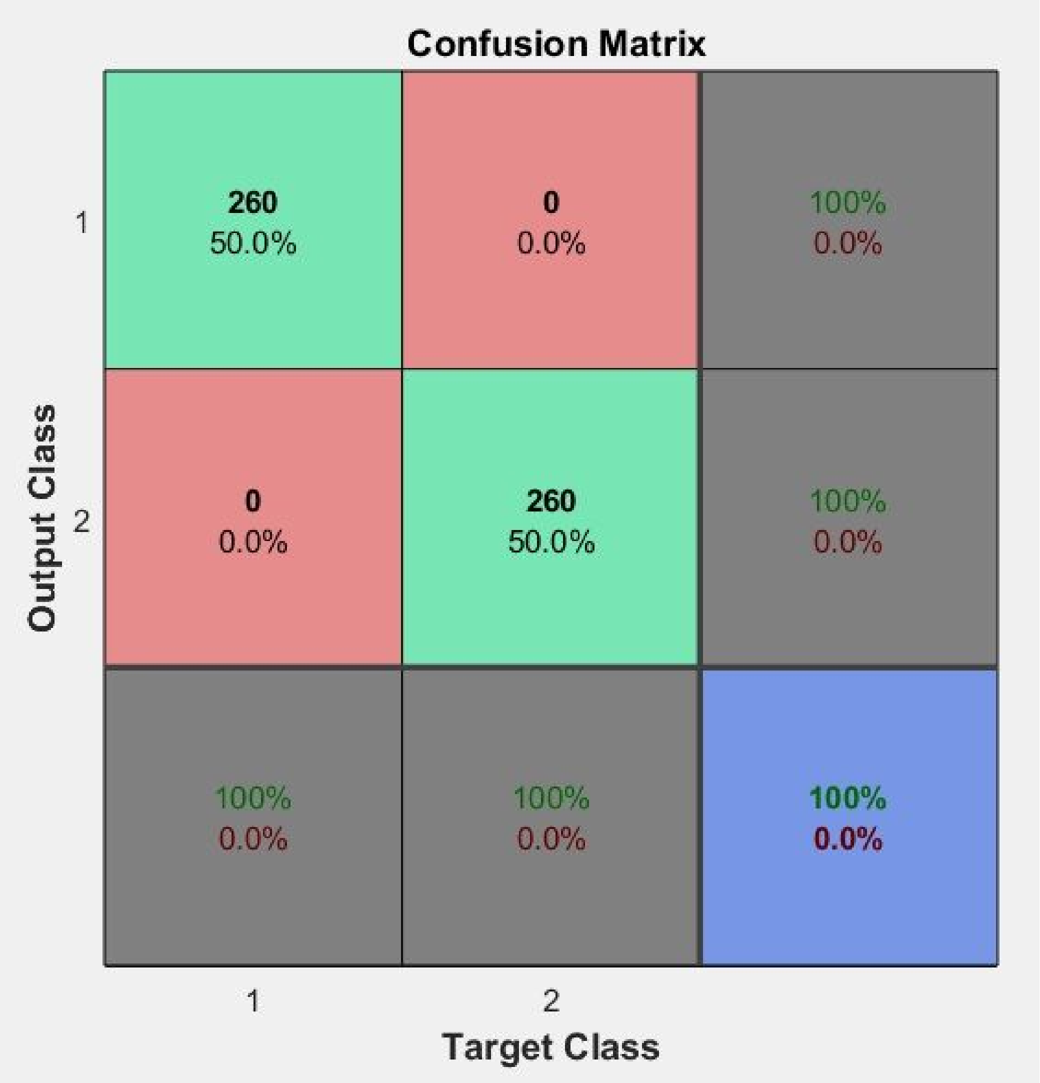


	1	2	
1	260 50.0%	4 0.8%	98.5% 1.5%
2	0 0.0%	256 49.2%	100% 0.0%
	100% 0.0%	98.5% 1.5%	99.2% 0.8%

Confusion Matrix For Testing Data

RBF


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



	1	2	
1	**260** 50.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**260** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

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

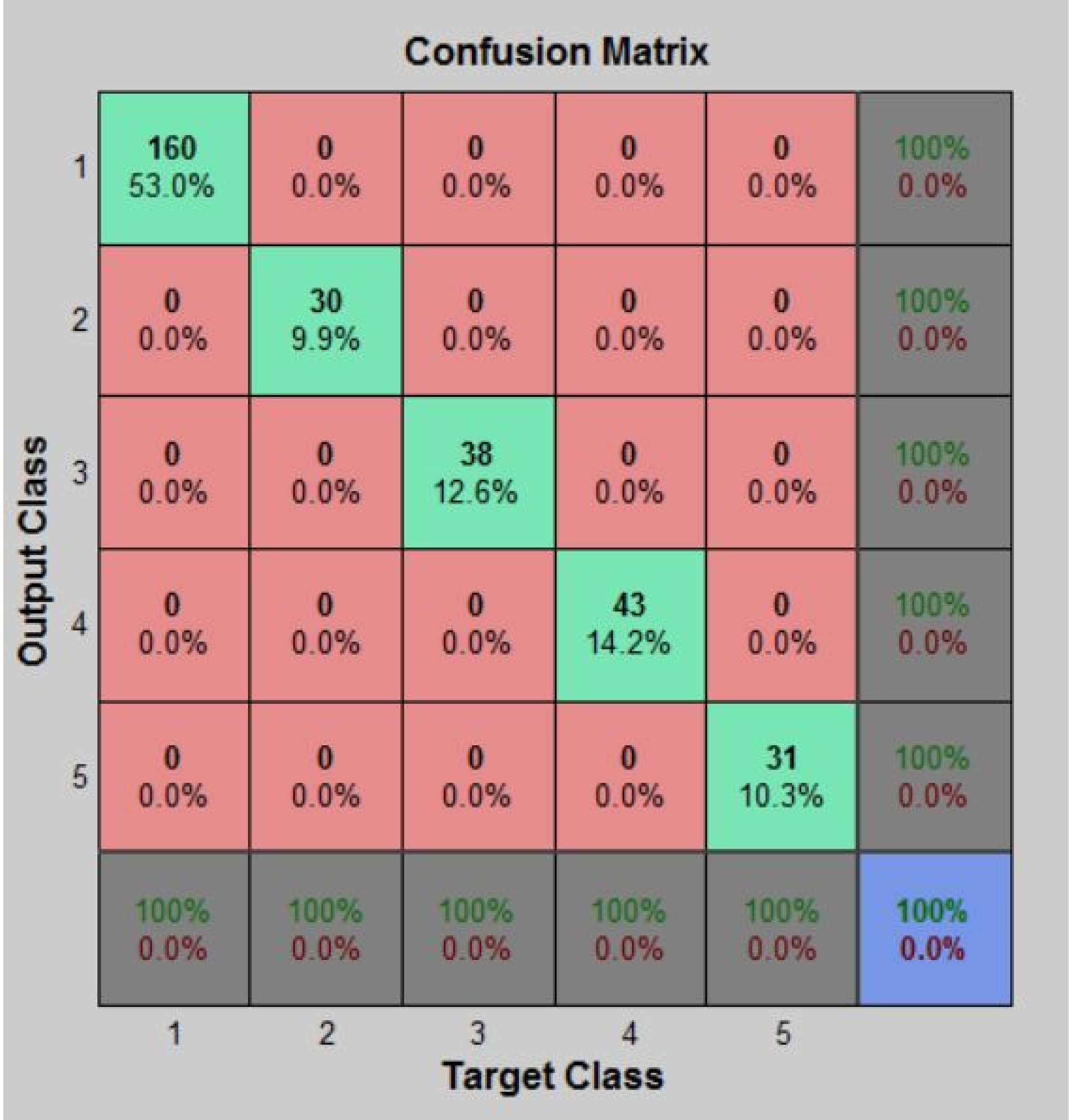
48 Dimensional Feature Vector
Degree =2 ,Cost = 180000
Accuracy= 100 %

Reduced

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

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

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



	1	2	3	4	5	
1	160 53.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	30 9.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	38 12.6%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	43 14.2%	0 0.0%	100% 0.0%
5	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%	100% 0.0%