Introduction to Intelligent Systems: Project 2

Lee Avital and Mayur Sanghavi

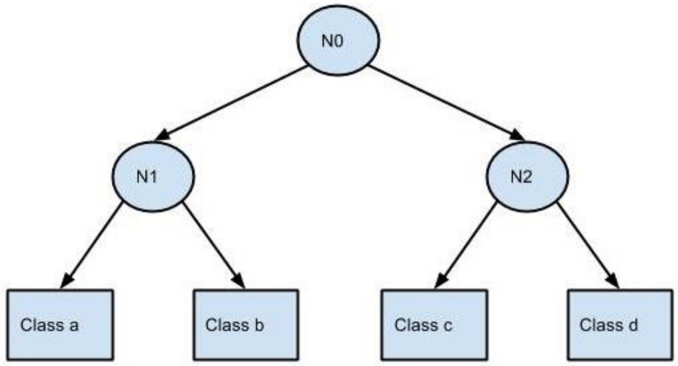
May 2, 2014

1 Classifier Designs

1.1 Random Forest Classifier

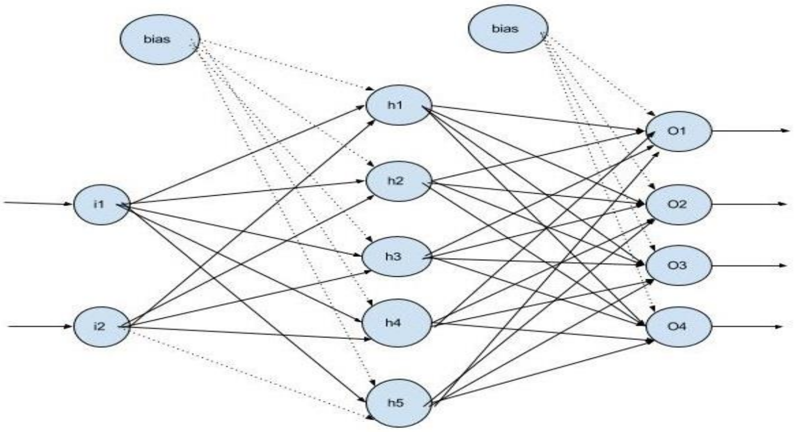
We used a random forest classier that generated a variable number of decision trees that all went down three nodes deep. Since the input variables were continuous, we needed to generate random splits to emulate the decision boundaries necessary for using decision trees.

Following is the structure of a single decision tree:



1.2 Multi-Layer Perceptron Classifier

We also used a 2-5-4 multi-layer perceptron (illustrated in full below) feeding a bias layer into both the hidden and output nodes.



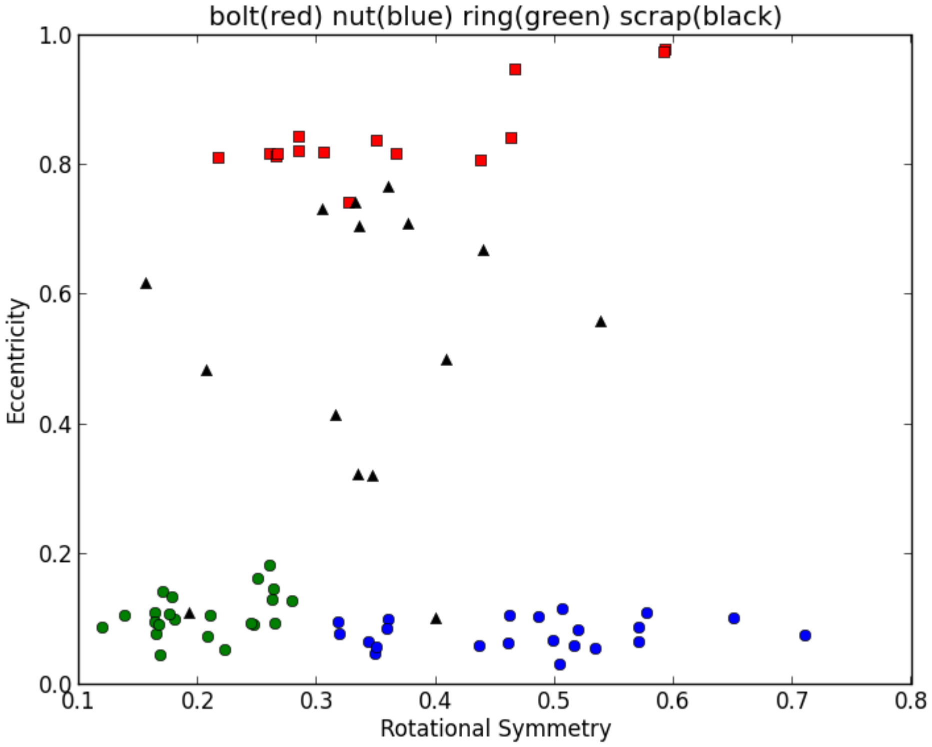
1.3 Hypothesis

The decision tree classifies the given input by running it through number of trees and then considering the class with maximum average probability. Whereas the neural network calculates the output of final output layer neurons using the learned weights and the class having maximum output is assigned.

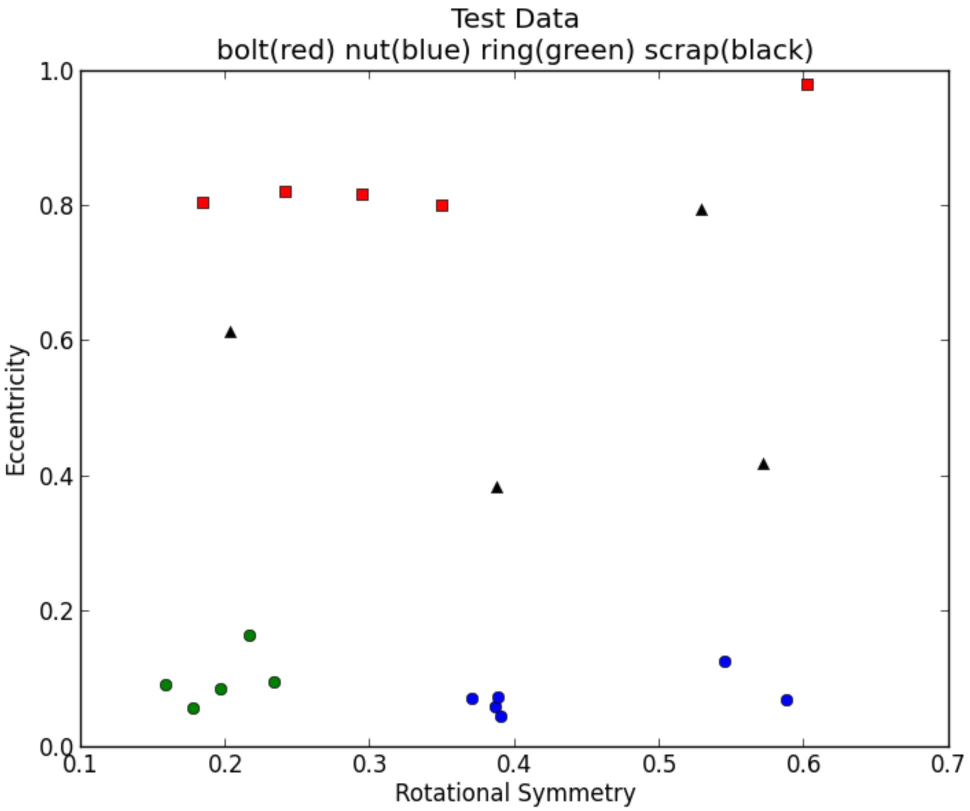
The decision tree and neural network both were able to classify the data in a fairly accurate manner after training for sufficient epochs/trees. When running both classifiers for a short time (i.e. 100 trees for the random forest and 100 epochs for the multi-layer perceptron) we expected the multi-layer perceptron to outperform the random forest. This was because the multi-layer perceptron, by its nature, worked on continuous inputs while the random forest needed to extrapolate discrete decision classes.

We expected the multi-layer perceptron to outperform the random forest after running a long time (i.e. 200 trees or 10,000 epochs.) This was because the random forest was likely to overfit when the number of trees increased and if most of them had similar split points. On the other hand the Neural Network trains on the error after each iteration and also converges for non-linear inputs.

2 Data Sets:



Training Data



Test Data

As can be seen from the above plots, training data seems to be fairly separable, thus resulting in good trained models of MLP and decision tree. Though few plots of scrap objects are scattered over other regions but as the number of such points are few, in all it's a good training data. The test data can be said to be linearly separable if you exclude the single scrap plot in the region of Eccentricity = 0.8. Due to which Decision Trees get 95% accuracy i.e. they wrongly classify that point, but that is not a problem for neural network.

Even by just looking at plots one can determine that if Eccentricity of an object is below 0.3 it can be only either a ring or a nut, and if its Rotational Symmetry is less than 0.3 it most probably will be classified as a ring.

3 Results:

a) MLP:

Epochs	Recognition Rate(%)	Profit ($)
0	30.0	-0.08
10	25.0	-0.8
100	55.0	0.77
1000	100.0	2.03
10000	100.0	2.03

Test Output for MLP

Confusion Matrices for MLP:

	Bolt	Nut	Ring	Scrap
Bolt	0	0	0	0
Nut	5	6	5	4
Ring	0	0	0	0
Scrap	0	0	0	0

0th Epoch

	Bolt	Nut	Ring	Scrap
Bolt	0	0	0	0
Nut	0	0	0	0
Ring	5	6	5	4
Scrap	0	0	0	0

10th Epoch

	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	2
Nut	0	1	0	0
Ring	0	5	5	2
Scrap	0	0	0	0

100th Epoch

	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	0
Nut	0	6	0	0
Ring	0	0	5	0
Scrap	0	0	0	4

1000th Epoch

	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	0
Nut	0	6	0	0
Ring	0	0	5	0
Scrap	0	0	0	4

10000th Epoch

b) Decision Trees:

No. of Trees	Recognition Rate(%)	Profit ($)
1	60.0	0.8
10	90.0	1.95
100	95.0	1.99
175	95.0	1.99

Test Output for Decision Trees

Confusion Matrices for Decision Trees:

	Bolt	Nut	Ring	Scrap
Bolt	4	0	0	3
Nut	0	2	0	0
Ring	0	4	5	0
Scrap	1	0	0	1

1 Tree

	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	2
Nut	0	6	0	0
Ring	0	0	5	0
Scrap	0	0	0	2

10 Trees

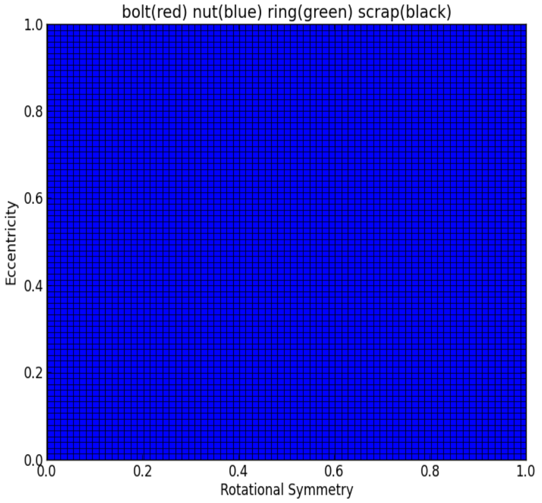
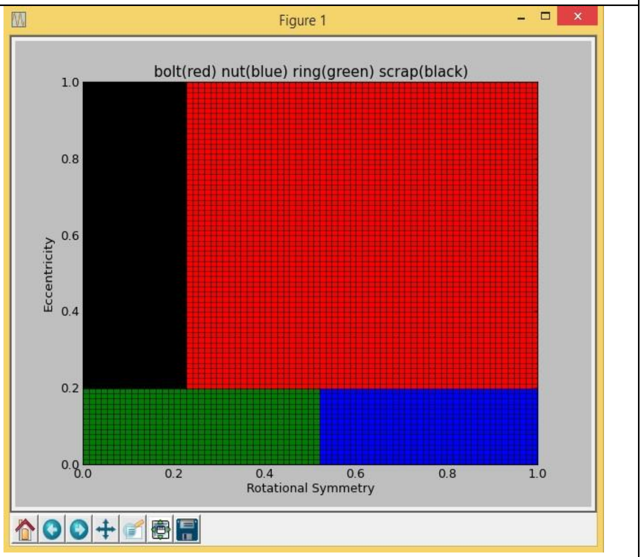
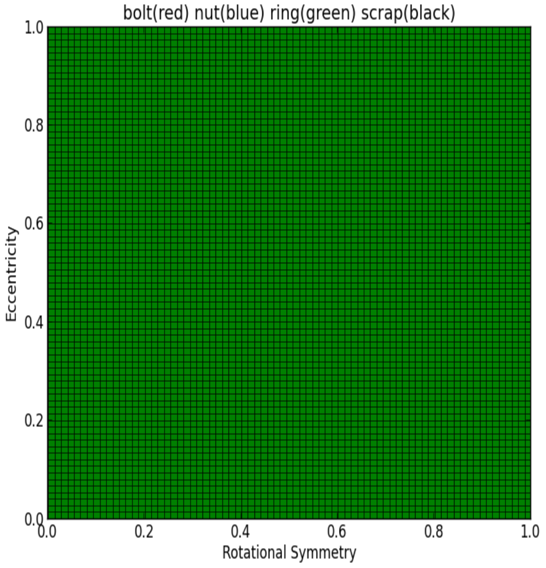
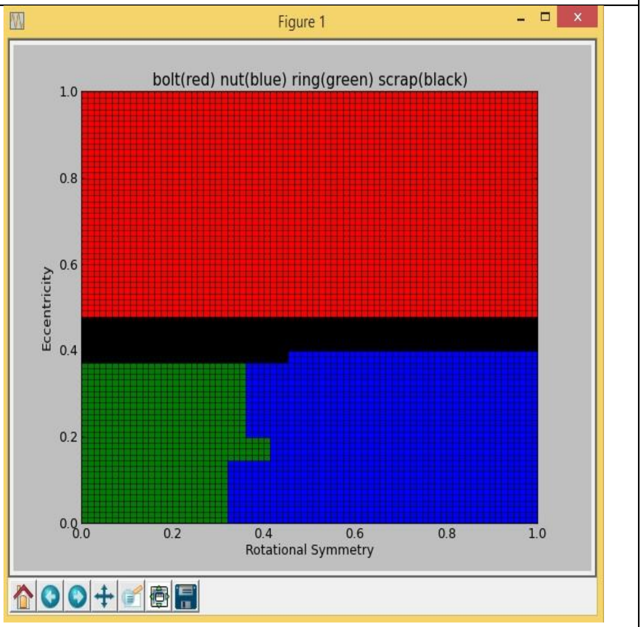
	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	1
Nut	0	6	0	0
Ring	0	0	5	0
Scrap	0	0	0	3

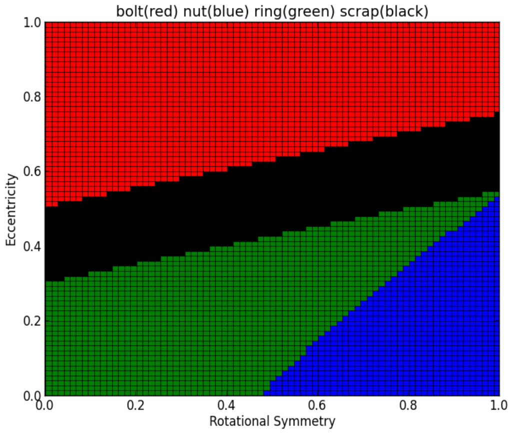
100 Trees

	Bolt	Nut	Ring	Scrap
Bolt	5	0	0	1
Nut	0	6	0	0
Ring	0	0	5	0
Scrap	0	0	0	3

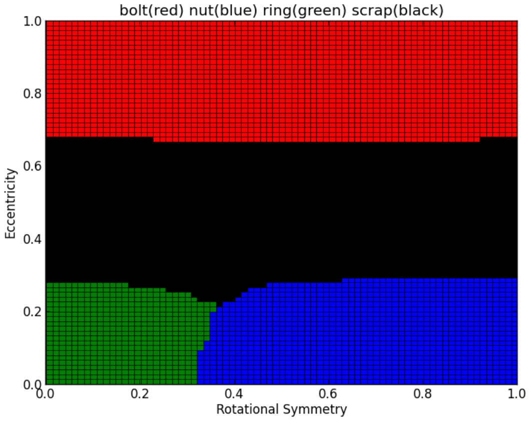
175 Trees

c) Classification Regions:

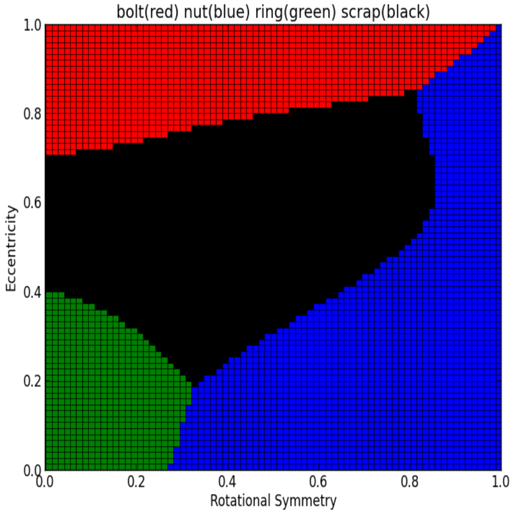
MLP	Decision Trees
 0 epoch	 Figure 1 bolt(red) nut(blue) ring(green) scrap(black) Eccentricity 0.0 0.2 0.4 0.6 0.8 1.0 Rotational Symmetry 1 Tree
 10 epochs	 Figure 1 bolt(red) nut(blue) ring(green) scrap(black) Eccentricity 0.0 0.2 0.4 0.6 0.8 1.0 Rotational Symmetry 10 Trees



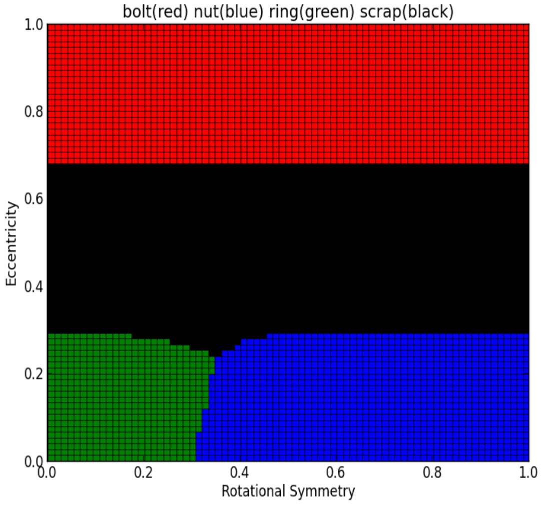
100 Epochs



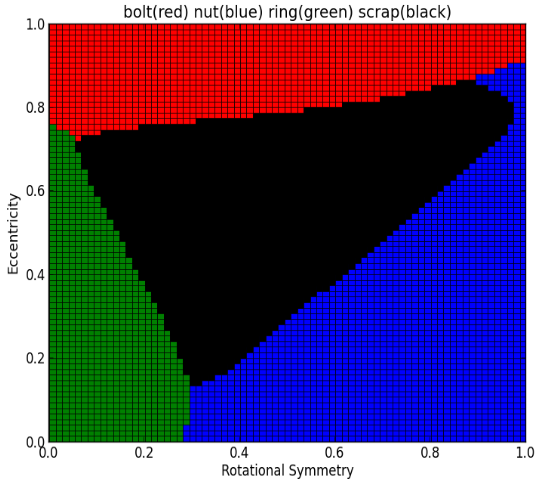
100 Trees



1000 Epochs



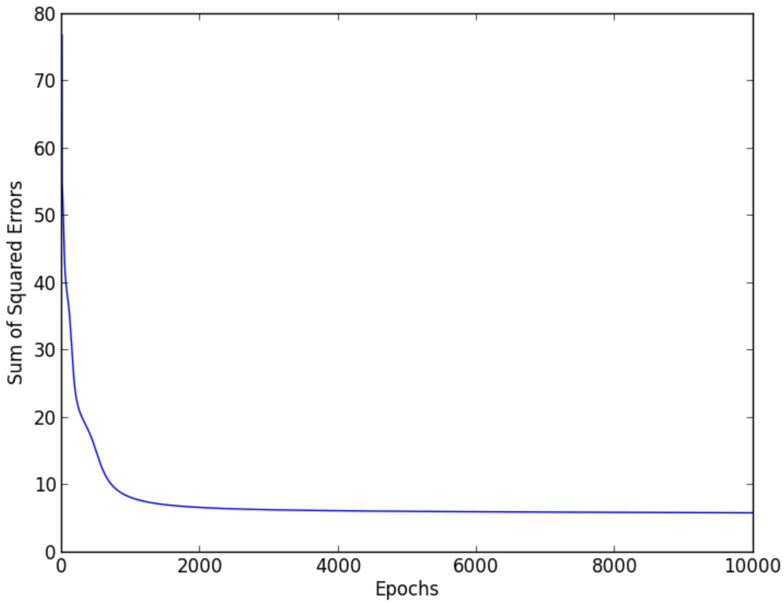
175 Trees



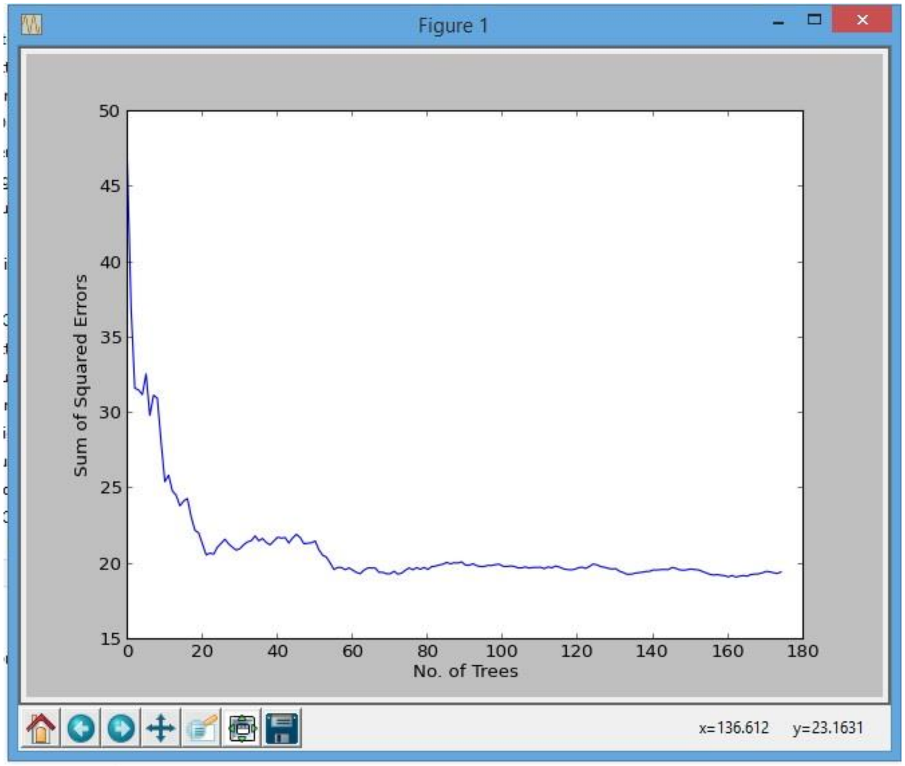
10000 Epochs

d) Learning Curve:

1) MLP:



2) Decision Tree:



4 Discussion

Our predictions ended up being correct. The decision tree based random forest ended up topping out at 95% accuracy while the multi-layer perceptron had 100% accuracy.

For the neural networks we assigned the initial weights between (-1, 1) and the architecture of network being 2 – 5 – 4. The network seems constantly reduce the error as the number of epochs increases and classifies with 100% accuracy even after 1000 epochs. Initially the class region shows only one class for entire region, but the region gradually divides into more accurate classes as the number of epochs increases.

We chose to go with decision trees having depth 2, because it classified equally accurately for this depth as it did for larger depths. Also, for every node we considered 3 random splits of each attribute and chose the best split (k = 3). Interestingly enough, the random forest with 10 sub-trees performed just as well as the one with 100 sub-trees. This suggests that the 100-tree random forest might be overfit, and the 10-tree is good enough. This seems even more true when you look at the error for the test data and notice how the error flattens well before 100 trees. Contrary to MLP, here we get four regions right from initial stages of learning. But the accuracy of regions is very low initially but quickly raises to 95% even with 10 trees.

So, it can be said that MLP classified better than Decision trees. Because MLP could achieve 100% accuracy and highest accuracy achieved for all executions of Decision Tree was 95 %. But MLP took a minimum of 1000 epochs for good accuracy and whereas decision trees achieved almost accurate results even with only 10 trees of depth 2.