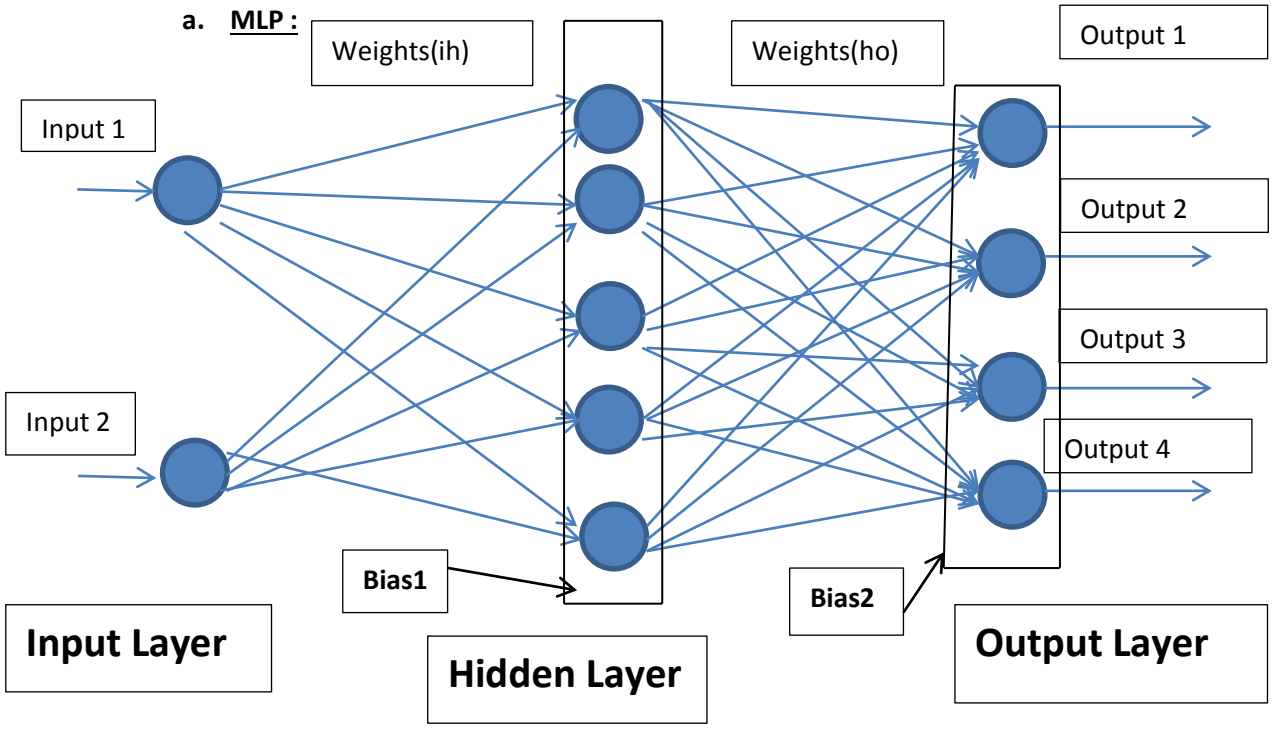
PROJECT 2

1. **Classifier Design :**
   a. **MLP :**



Where,

Weights(ih) = weights from input to hidden layer.

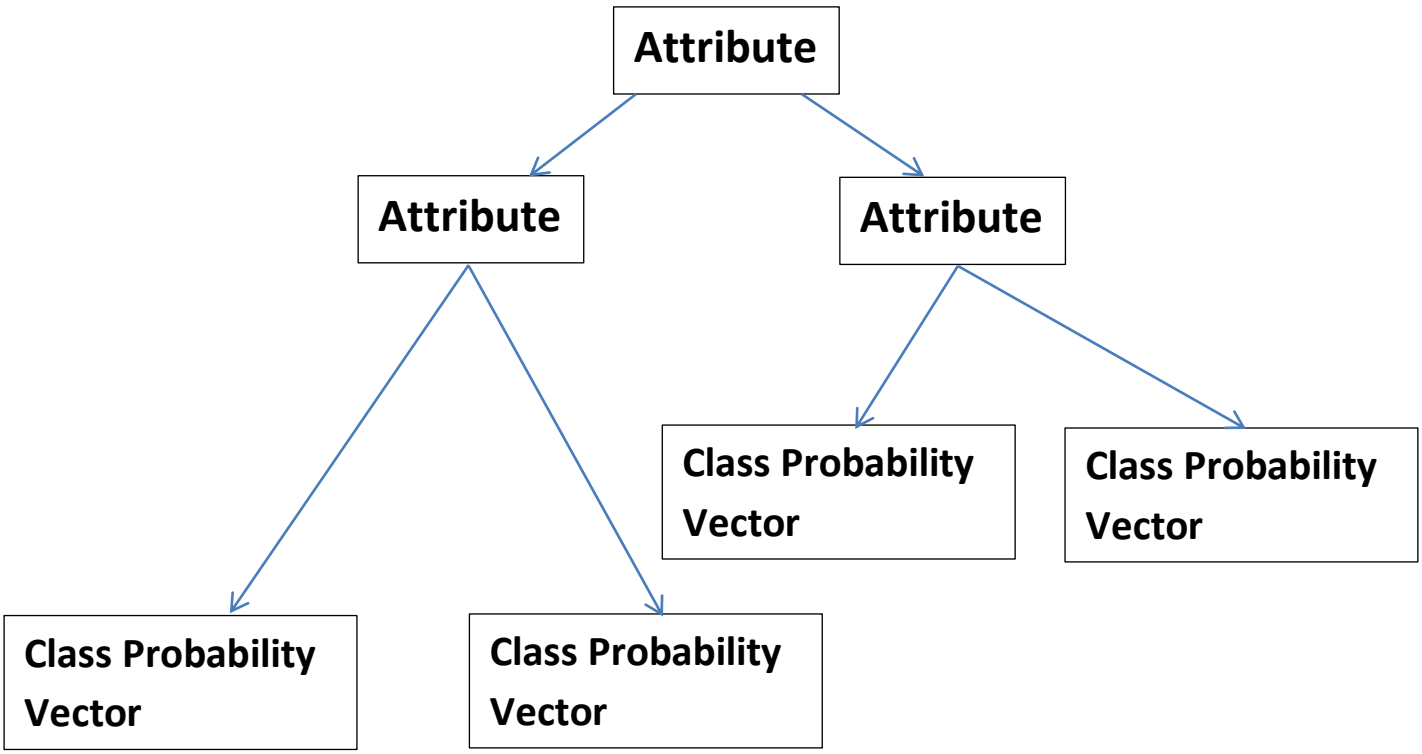
Weights(ho) = weights from hidden to output layer.

Bias1 = bias to all hidden layer.

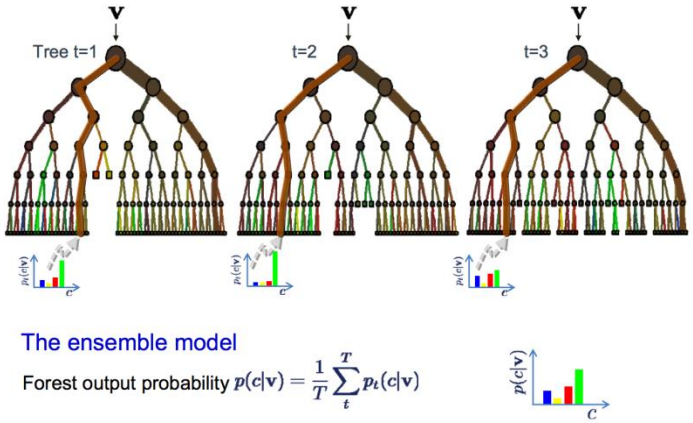
Bias2 = bias to all output layer

The architecture used in this example is a 2-5-4 architecture. There are two input, five hidden and 4 output nodes. The success of any architecture lies in its ability to successfully classify the elements into its correct classes. In our example it is bolts , nuts , scrap ,etc. Four output layer are for four classes of classification. 5 hidden layers selected are irrespective of any constraint. It is selected based on performance and can be found out by trial and error method. More hidden will only help the weight to converge faster since more weights will be used.

b. Decision Tree :



This is a simple decision tree of depth 2. We select the root based on the best attribute obtained from the information gain. We then create left and right node with splitting the data based on the split feature and split value. The leaf node are the class label.
The random forest would look something like



Both MLP and decision tree are classifiers they both classify data. They start at random weights.

I was expecting MLP to perform better because it is updating at every input and this gives better accuracy as compared to the probabilistic approach that split at random points in decision tree. It

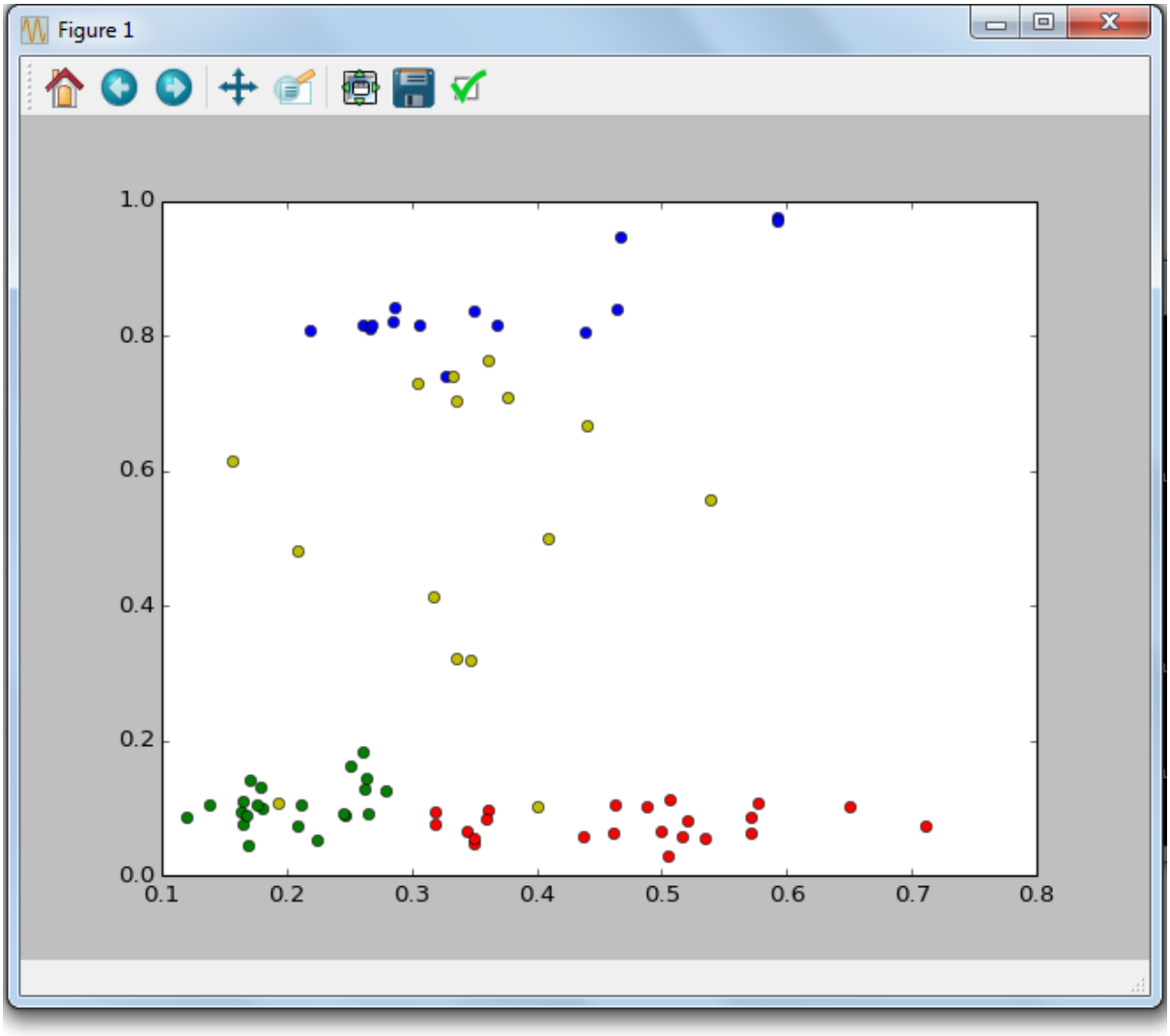
can be seen that decision tree has performed poorly and cannot be used but it was unable to classify one of the inputs.

2. **Data Sets :**

Train Data :

The colors used throughout the project are as follows :

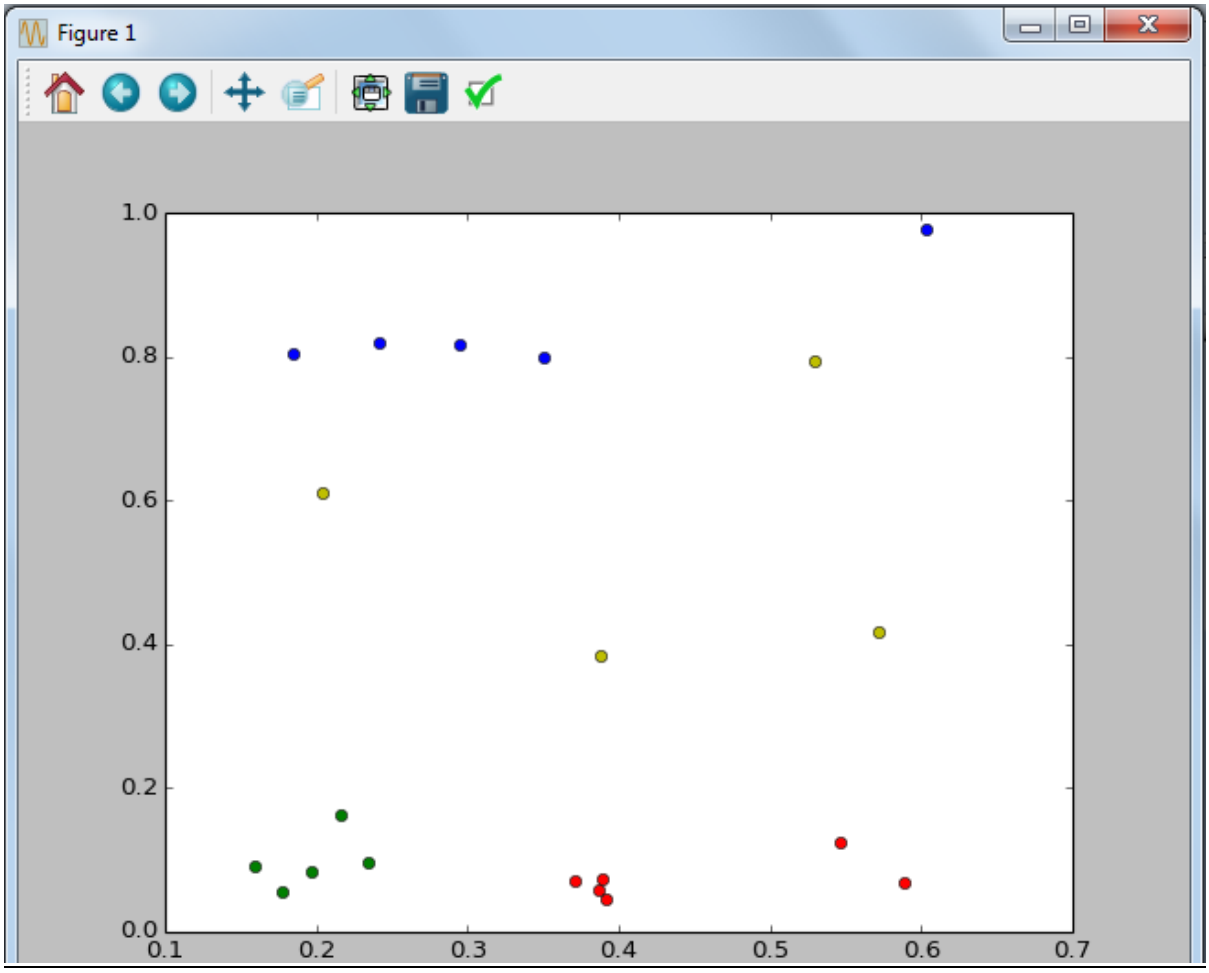
1. Bolts is blue
2. Nuts are red
3. Ring is green
4. Scrap is yellow



It can be seen from the training data that all the bolts have a higher eccentricity. Scrap has equal amounts of both eccentricity as well as six fold rotational symmetry. Ring has very low eccentricity which is justified because of its circular structure. Nuts have fairly same eccentricity as that of rings but have higher six fold rotational symmetry. It appears to be a

well scattered data set. This seems like a good training set as it specifies the region clearly. It does not have all samples clustered in the same sample space.

Test Data :



The train data is also very unique in its appearance. The tests are scattered throughout and do not appear clustered. This helps to understand whether the program can properly classify the dataset or not .The regions of bolts nuts screws and scrap appear in the same regions as that in the training examples.

3. **Results :**

a. MLP:

TABLE FOR RECOGNITION RATE AND PROFIT

EPOCH	RECOGNITION RATE	PROFIT
0	25.0	-0.64
10	25.0	-0.64
100	80.0	1.81
1000	100.0	2.03
10000	100.0	2.03

CONFUSION MATRIX:

For n =0 ,

0 0 5 0
0 0 6 0
0 0 5 0
0 0 4 0

For n = 10

0 0 5 0
0 0 6 0
0 0 5 0
0 0 4 0

For n = 100

5 0 0 0
0 5 1 0
0 0 5 0
2 1 0 1

For n = 1000

5 0 0 0
0 6 0 0
0 0 5 0
0 0 0 4

For n = 10,000

5 0 0 0
0 6 0 0
0 0 5 0
0 0 0 4

b. Random Forest :

TABLE FOR RECOGNITION RATE AND PROFIT

EPOCH	RECOGNITION RATE	PROFIT
0	55.0	1.27
10	95.0	1.99
100	95.0	1.99
1000	95.0	1.99

CONFUSION MATRIX:

For n =0 ,

5, 0, 0, 2
0, 6, 5, 2
0, 0, 0, 0
0, 0, 0, 0

For n = 10

5, 0, 0, 1
0, 6, 0, 0
0, 0, 5, 0
0, 0, 0, 3

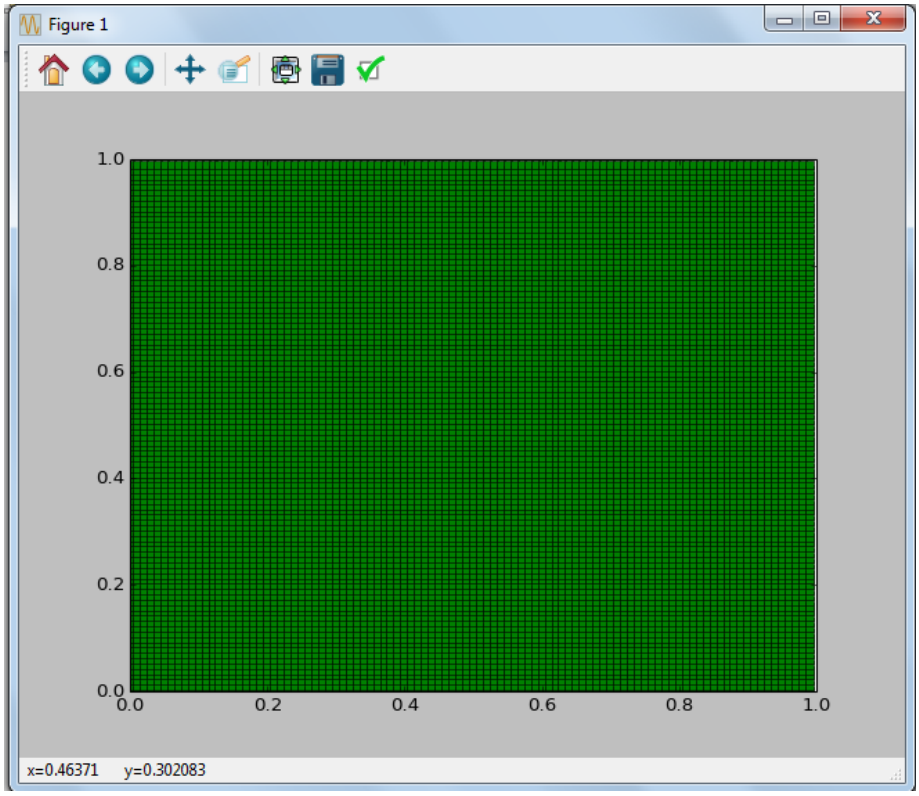
For n = 100

5, 0, 0, 0
0, 6, 0, 0
0, 0, 5, 0
0, 0, 0, 3

c. <u>For MLP :</u>

For n = 0

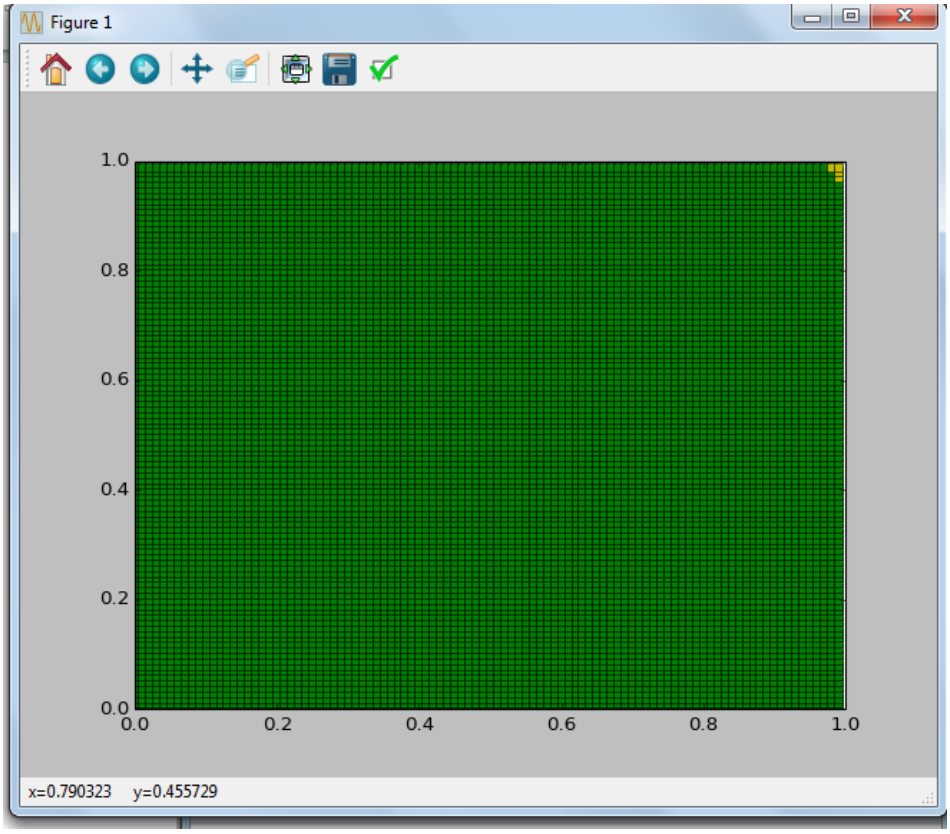
Decision Boundary



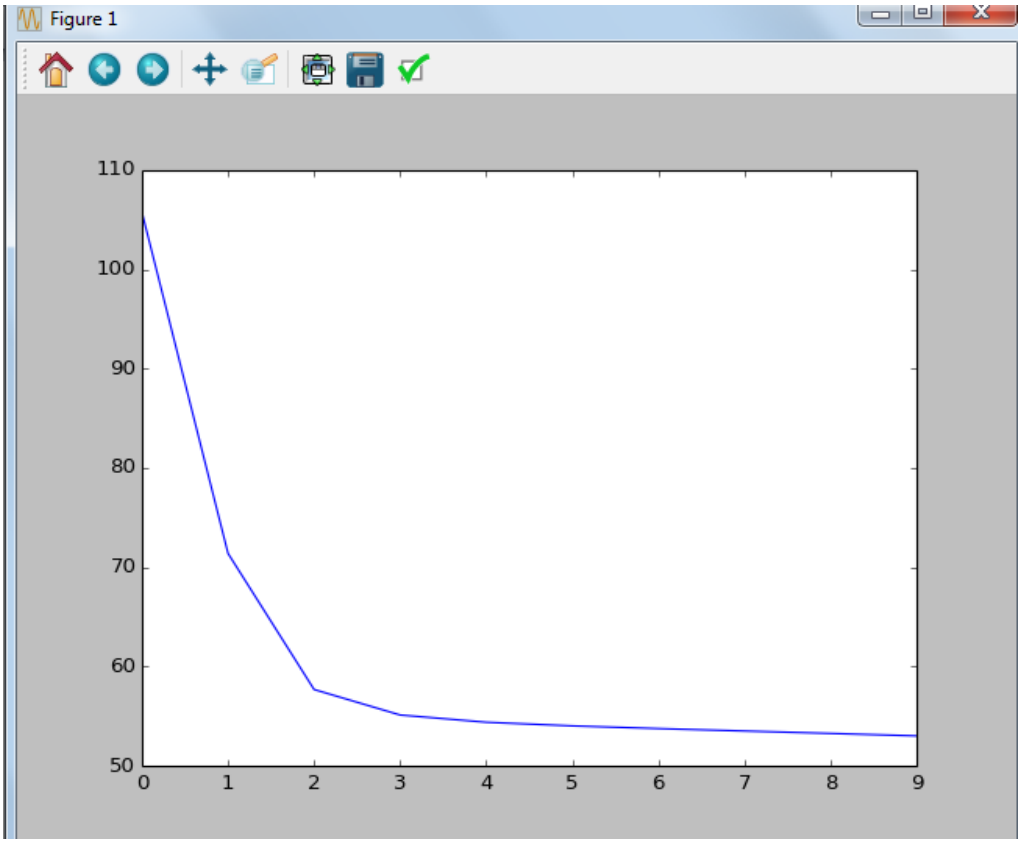
Learning Curve

For n = 10

Decision Boundary

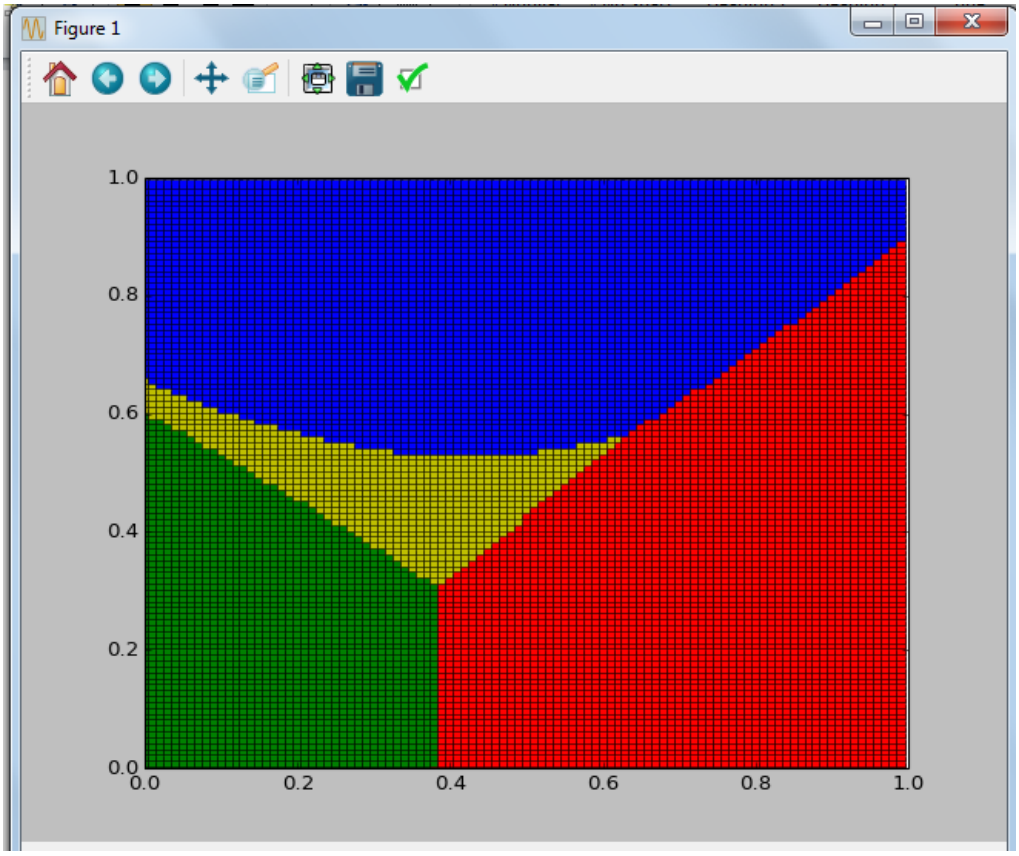


Learning Curve

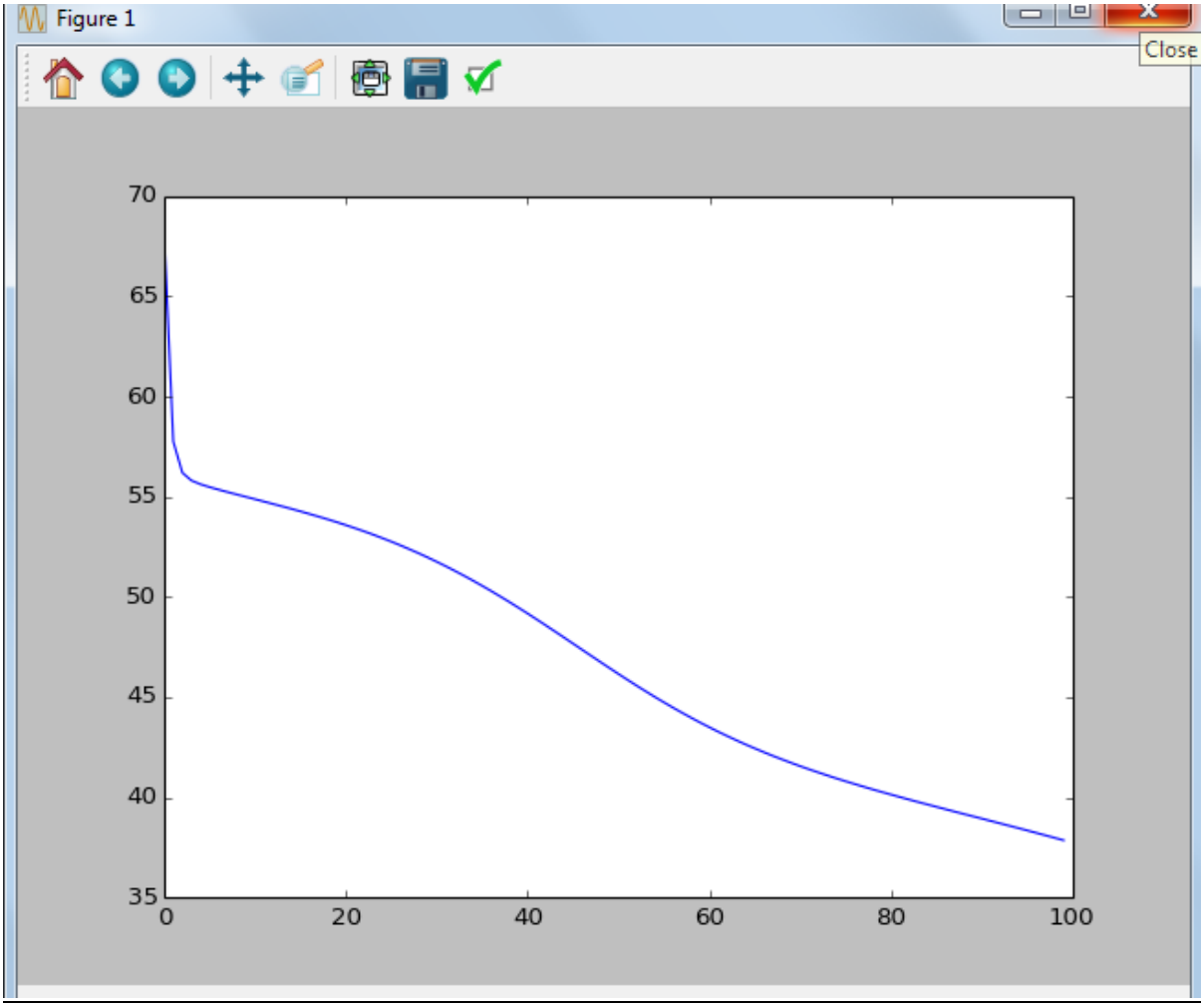


For n = 100

Decision Boundary

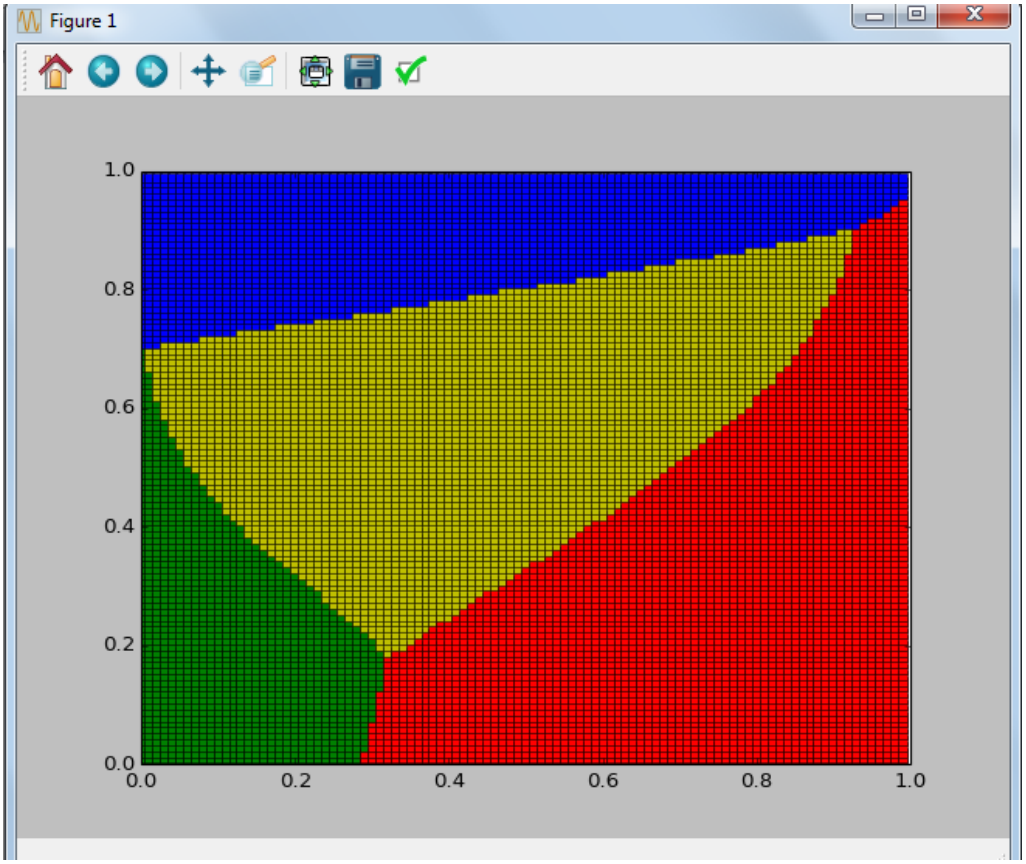


Learning Curve

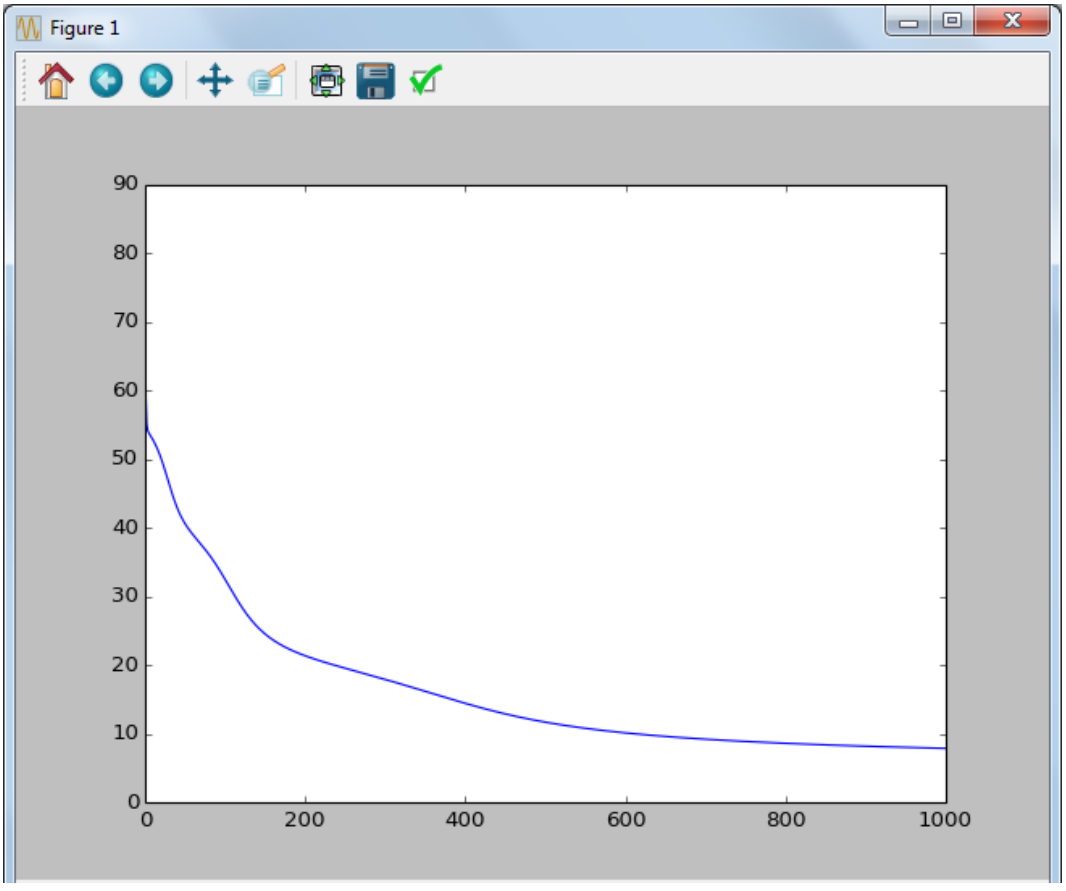


For n = 1000

Decision Boundary

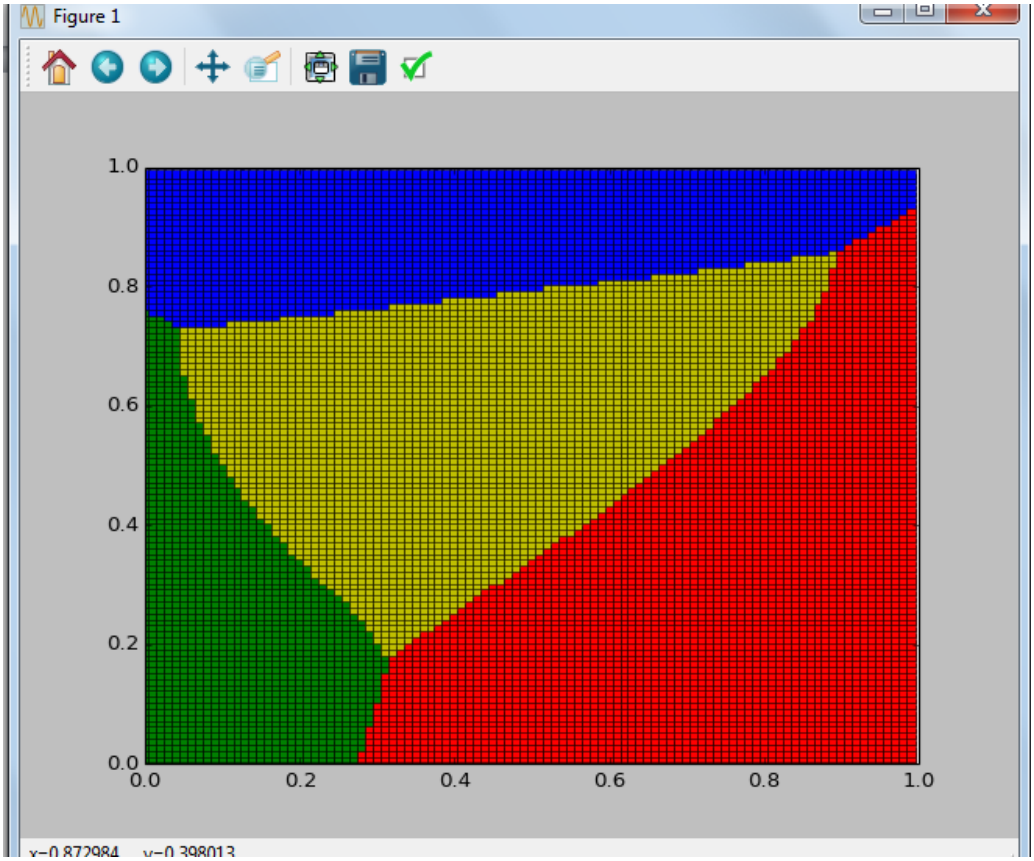


Learning Curve

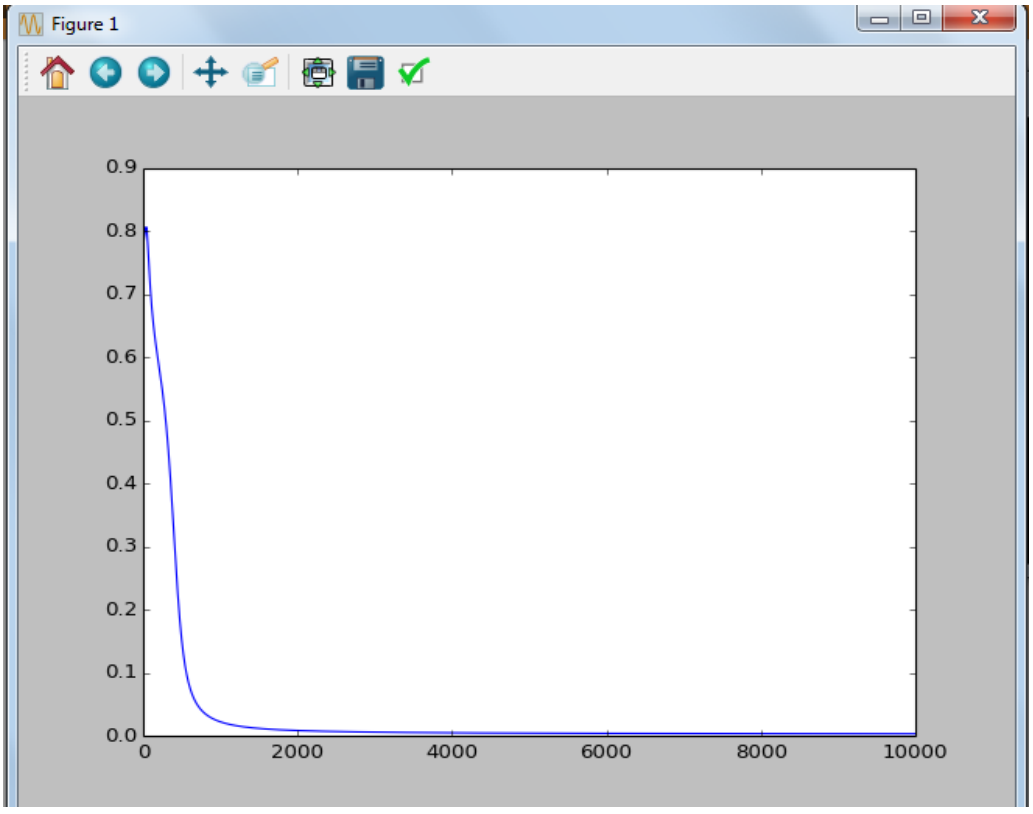


For n = 10,000

Decision Boundary



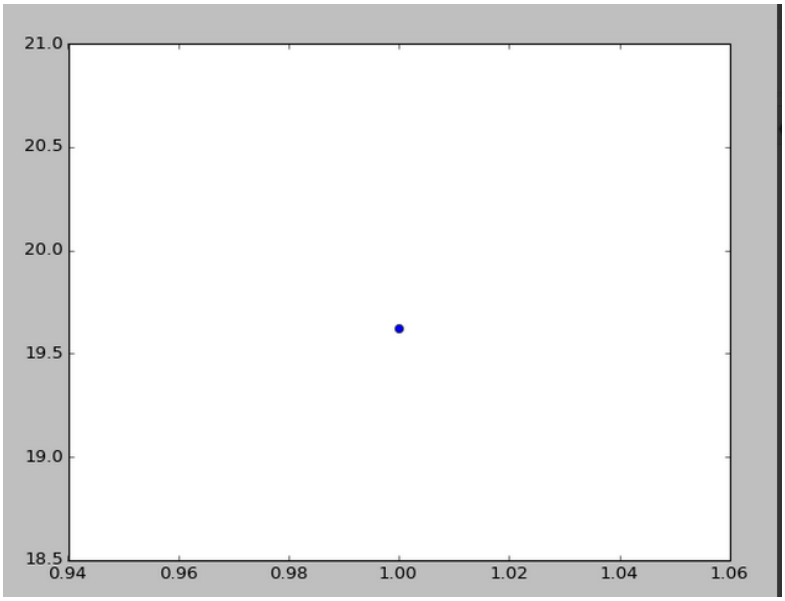
Learning Curve



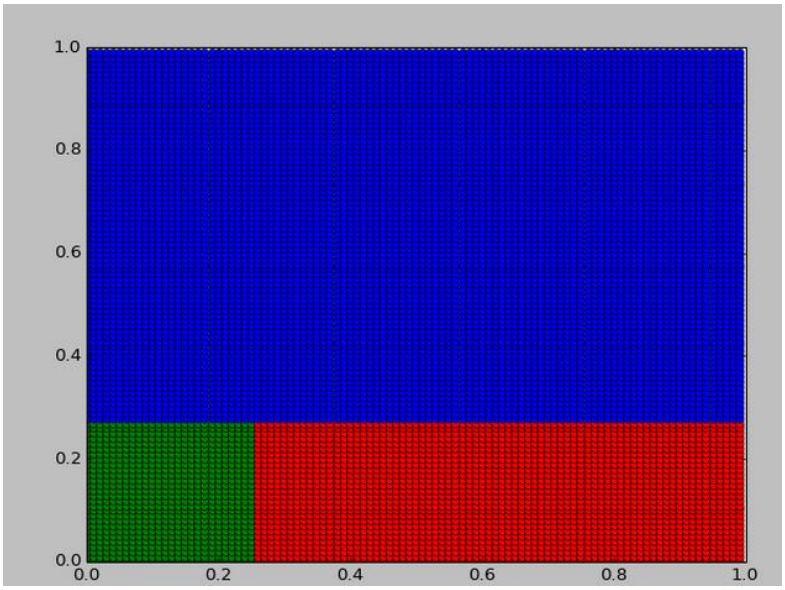
For Random Forest :

For n = 0

Learning Curve

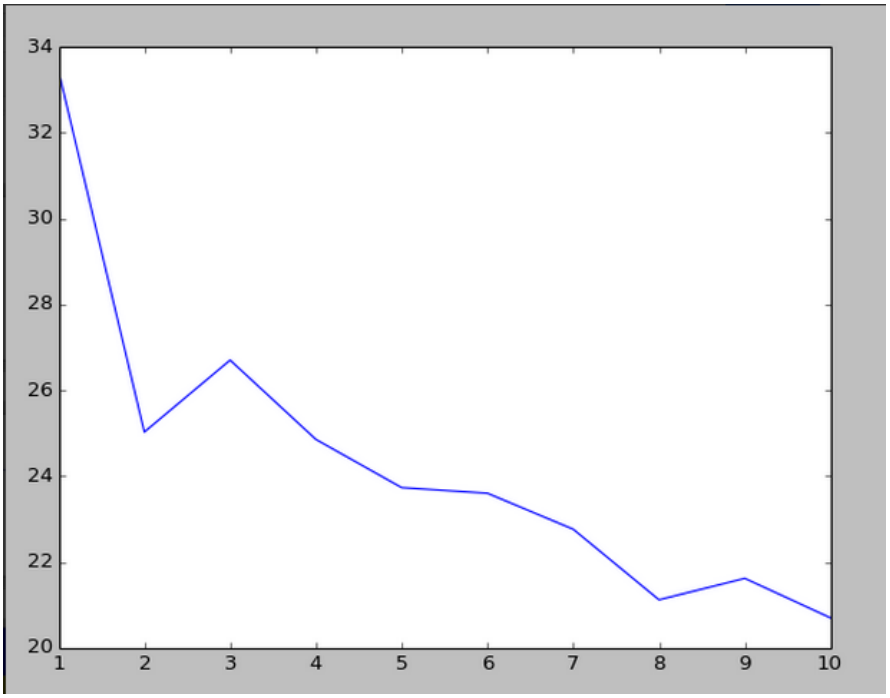


Decision Boundary

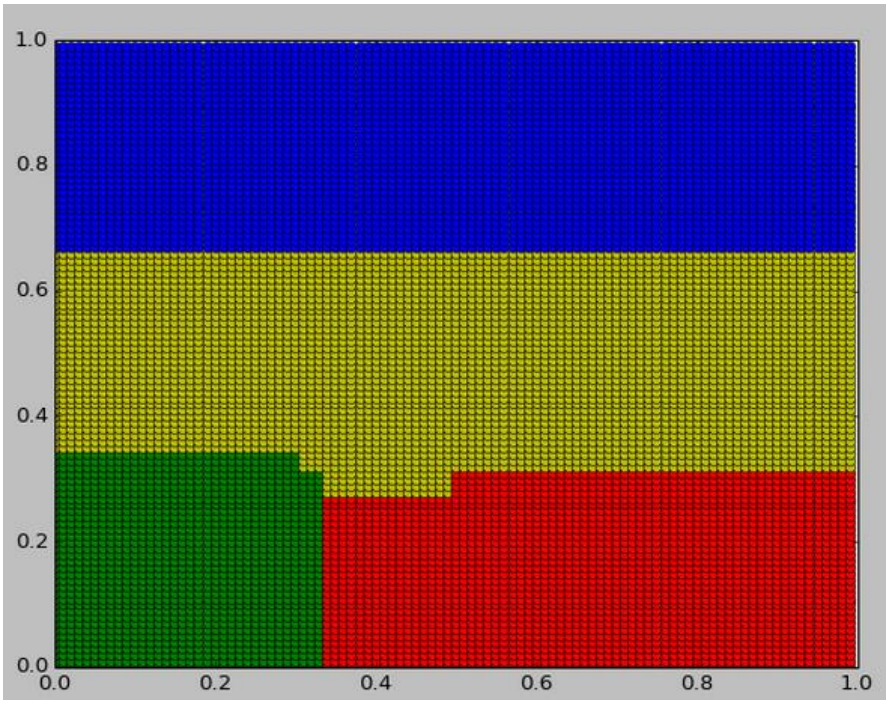


For n = 10

Learning Curve

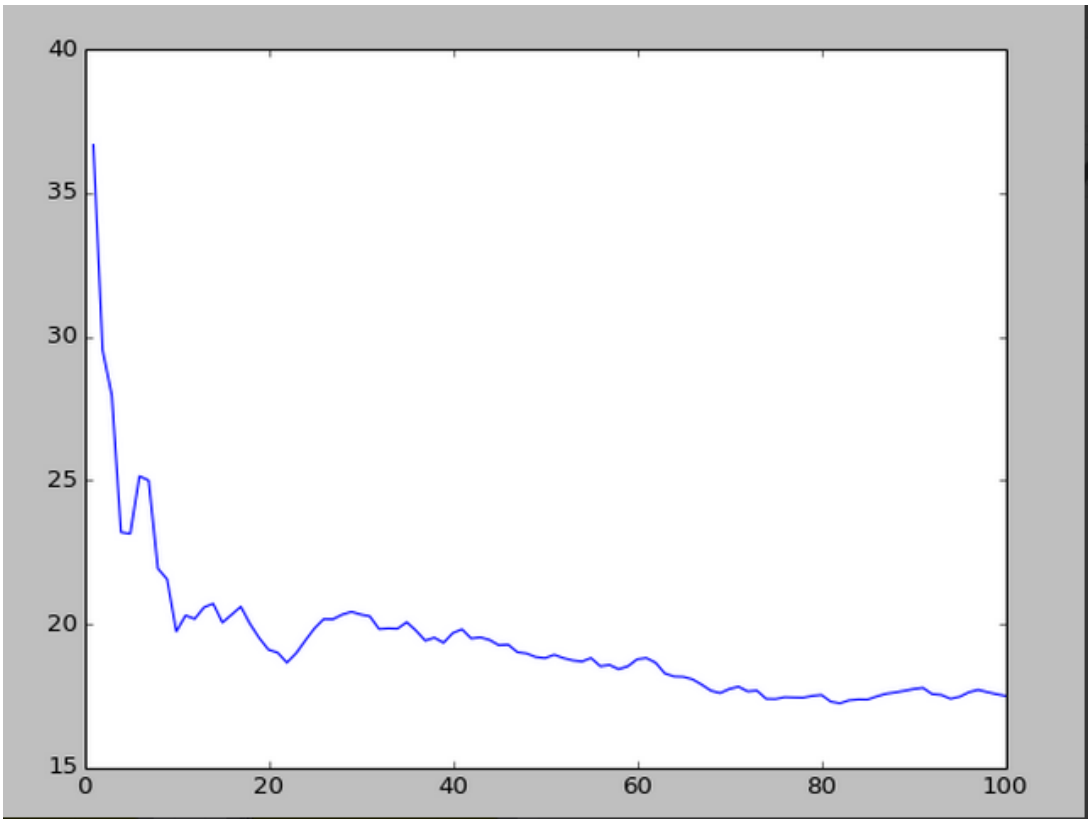


Decision Boundary

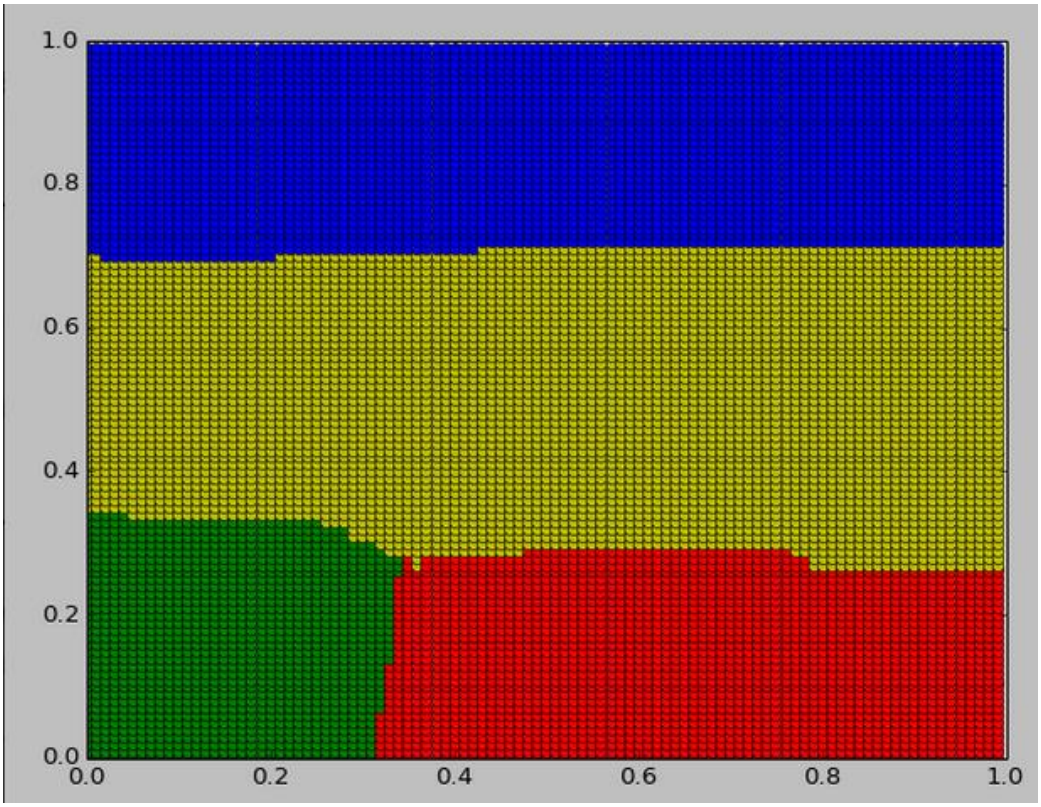


For n = 100

Learning Curve

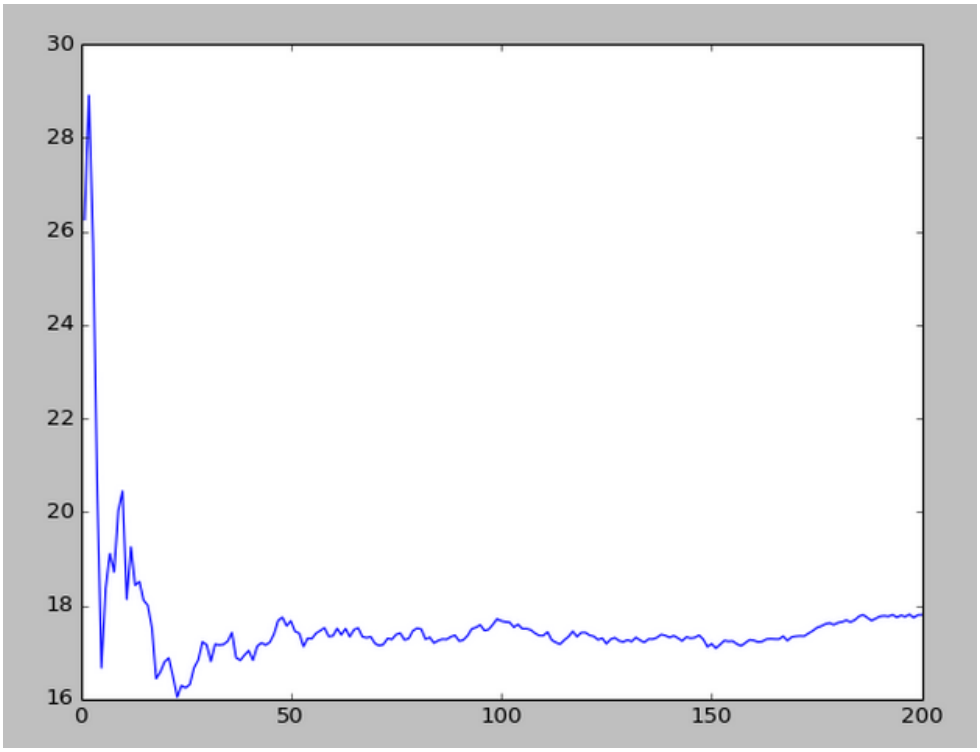


Decision Boundary

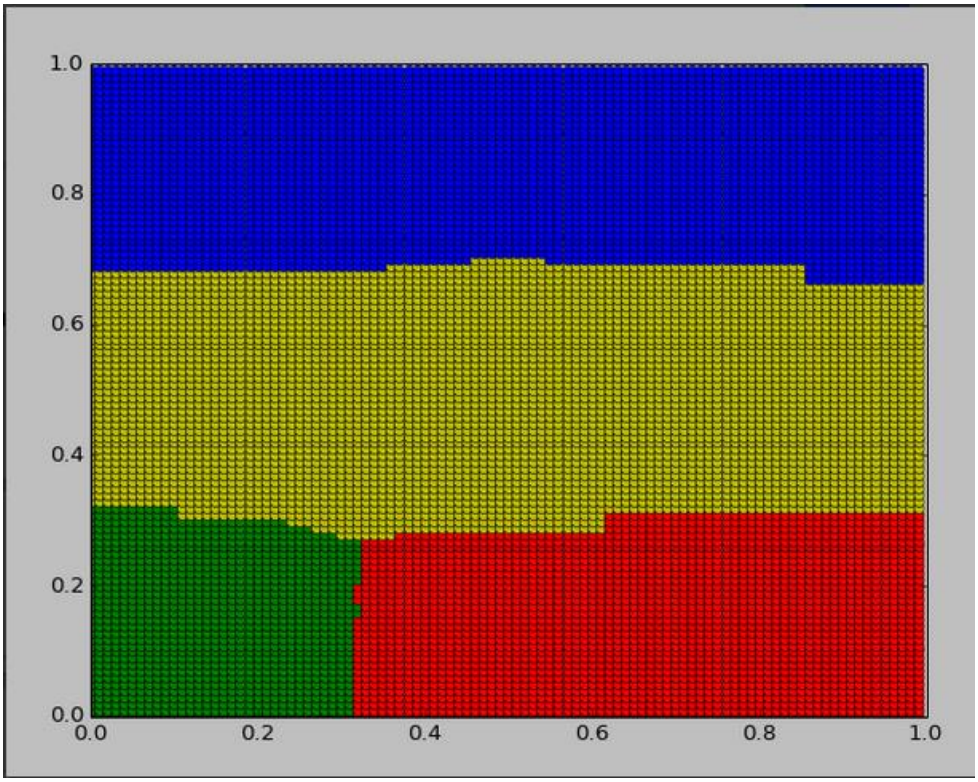


For n = 200

Learning Curve



Decision Boundary



4. Discussion

The results for the MLP are more accurate as compared to that for the decision tree. The MLP classified 100% of the test data whereas decision tree failed to classify one of the dataset. The class boundaries for both the MLP and decision tree are different. In decision tree they appear more horizontal due to the nature of the algorithm that randomly decides split points. So it splits data and forms straight lines. For MLP the boundaries are more flexible. They are more curved at the edges. This gives flexibility in deciding regions. MLP gives more accurate regions of decisions. For 0 we see it only classifies for ring In MLP whereas at decision tree it is able to make some decision which are better and can classify three classes. At it is trained for more iterations improvements in decision tree stop but for MLP weights keep updating and it keeps on getting better and better. Regular updates to weights gives are a very accurate selection of region which is not possible for decision tree. For MLP all classifier were good. In Decision tree it was unable to identify bolt. It wrongly classified it as scrap. This was wrong and it was not able to correct even in further iterations. It was unable to calculate the correct output because of the probability calculations. It was not able to accurately calculate the probability of it occurrence. Because of the probability it was not able to completely distinguish between classes. The nature of the graph is the most interesting aspect. The classification made appears completely different. Looking at this I was expecting more wrongly classified attributed but there was only one attribute that appeared to be wrong. In my opinion the test data was not very good. The performance metric increases with increase in number of epoch. With the weights updating it tends to learn more and give more accurate results. Similar is the case with decision tree. But at one point it is no longer further able to classify. The best test data would be to test data at the boundaries. As I have seen most of the data at the center would be classified correctly but boundaries become very difficult. That's the region where we want maximum accuracy. For decision tree the range of probability is very low so the chances that the attribute will correctly lie in the range is also very less as a result of this we get one of the output as wrong.