Foundations of Intelligent Systems - Project 2

Chinmay Jain
Tolga Cerrah

Algorithms

Multi Layer Perceptron (MLP) learning uses logistic regression for calculating the output. That being said it produces a numeric value which we threshold to perform classification. Decision Tree (DT) learning uses a binary tree to classify the data. It classifies the data using the split value for a attribute in the node. Hence we can use both type of models for classification but intuitively MLP is not a classification algorithm whereas DT is a classification algorithm.

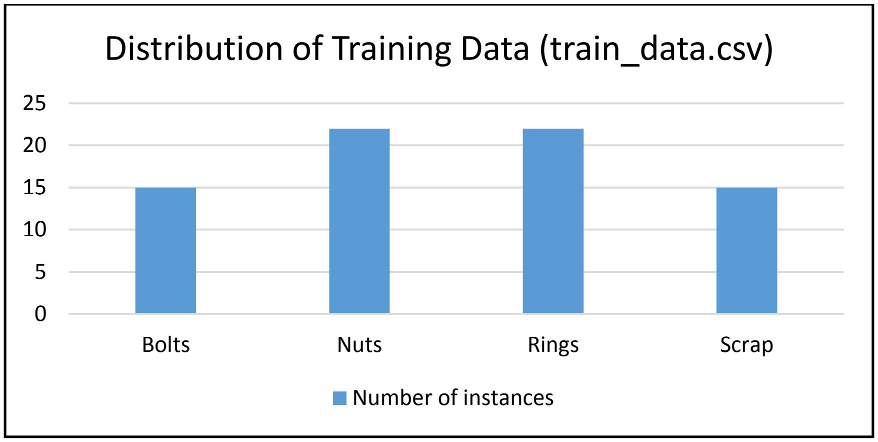
In terms of similarities both the algorithms are highly complementary to each other rather than in competition. Both models are used in classification and prediction problems and are statistic supervised models i.e. they use linear algebraic operations for their functioning and use a supervisor for their performance tuning. Both algorithms differ in terms of computational resources used, MLP algorithm employees extensive use of the arithmetic processor whereas DT does not. However the space required to store the classification tree is more as compared to store the array of weights used in MLP. Another facet of difference between the two algorithms is in terms of flexibility. MLP implementations are highly flexible as we can change the number of hidden layers or the number of nodes used in each layer. Increase in number of nodes or layers does not necessarily improve the model but in some cases it does give a plug and play option to tune the model. Due to its regressive nature, MLP models are less prone to overfitting as compared to the DT models. Hence DT models are used in hand with pruning methods like Chi Square Pruning to avoid overfitting. In terms of training time, DT models are faster to train than their MLP counterparts.

Despite all the similarities and differences, we expect MLP to perform better than the DT algorithm for the following reason:

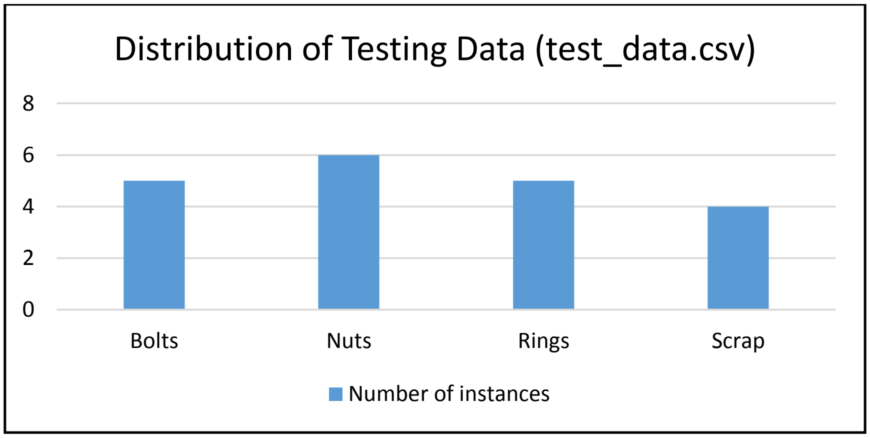
1. MLP considers the entire input space for its operations in an epoch whereas DT uses only one attribute at a time for deciding the split of a node. Although it doesn't make a huge difference in clearly separable data, small decision boundaries in closely related data may behave abnormally in DT.
2. Secondly, MLP is a regressor i.e. it calculates the probability of each output class for a instance. Whereas DT uses leaf nodes for its classification. Although we get only one output class from MLP, if we take the second best output class from the MLP for wrongly classified data, we would notice an increase in classification accuracy. DT do not provide provision for a second best options and the classification has no wiggle room.
3. MLP is a generalized algorithm i.e. it performs equally better with unseen test data as the development data. DT can be over fitted to training data, hence their performance with unseen test data is unpredictable.

Data:

The data consist of two input features. The first feature is symmetry and the second feature is the eccentricity of the metal piece. Both the features are continuous in nature and have fractional values in the range 0-1. The training data consist of 74 instances in total. The distribution of data is fairly equal, however we do have more rings and nuts as compared to bolts and scrap. Following the graph that shows the distribution of the data.



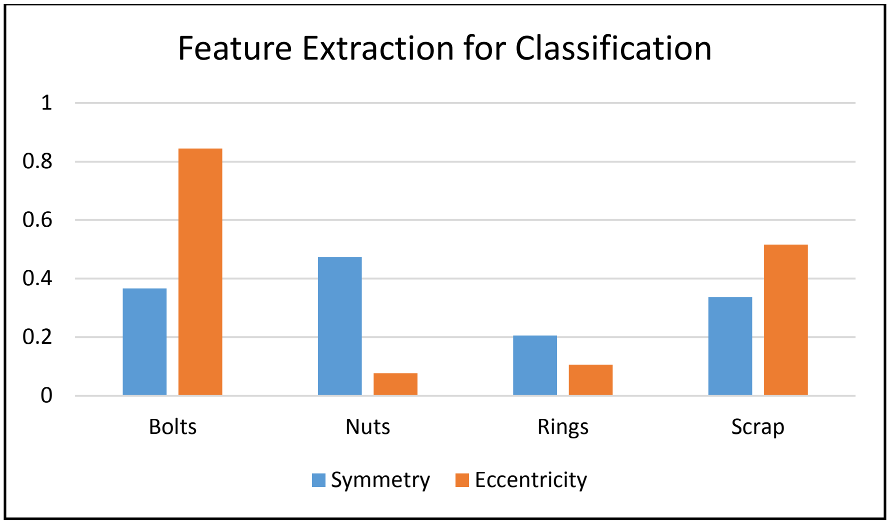
The testing data consist of 20 instances in total. The distribution of this data is more equal as compared to training data. The distribution of testing data is as shown in the graph below



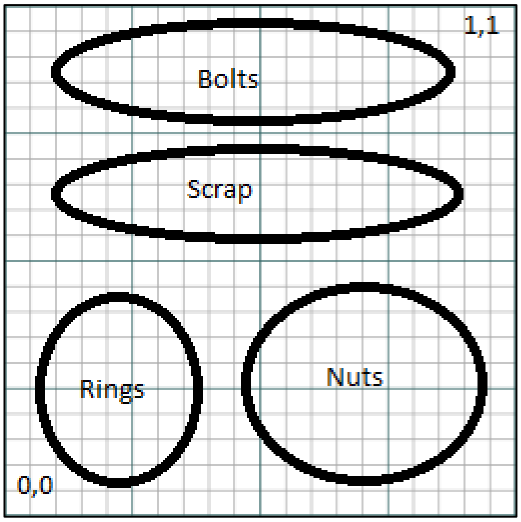
The following graph shows the average of a feature for each class. By looking at the graph we can make the following assumptions:

- Bolts would have a high eccentricity value.
- Nuts and Rings have lower eccentricity but both differ in terms of symmetry.
- Scrap and bolts are similar in terms but symmetry but scrap have a lower eccentricity value.

Although these assumptions are rough estimates rather than the classification boundaries themselves.



By these assumptions we can get a rough view of classification regions as, with symmetry on X axis and eccentricity on Y axis.



Results:

Color Map for all the graph plots:

Class1- Bolt	Purple
Class2 - Nut	Blue
Class3 - Ring	Yellow
Class4 - Scrap	Red

<u>Multi Layer Perceptron:</u>

Plots showing the test samples and classification regions produced by different numbers of training epochs

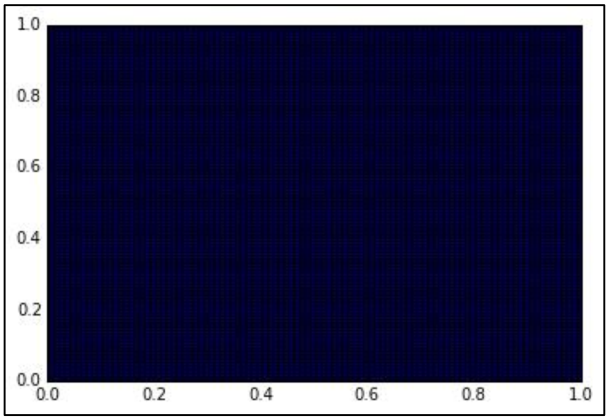
<u>After Epoch 10:</u>

Training accuracy after 10 epochs is 36.49 %

The confusion matrix is as follow

	Bolt	Nut	Ring	Scrap
Bolt	4	0	0	0
Nut	0	14	13	0
Ring	11	8	9	15
Scrap	0	0	0	0

The classification regions can be given as



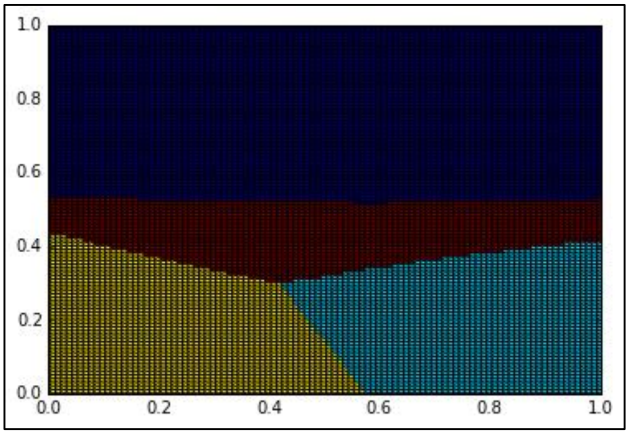
<u>After Epoch 100:</u>

Training accuracy after 100 epochs is 64.86 %

The confusion matrix is as follow

	Bolt	Nut	Ring	Scrap
Bolt	15	0	0	11
Nut	0	17	6	0
Ring	0	5	16	4
Scrap	0	0	0	0

The classification region can be given as



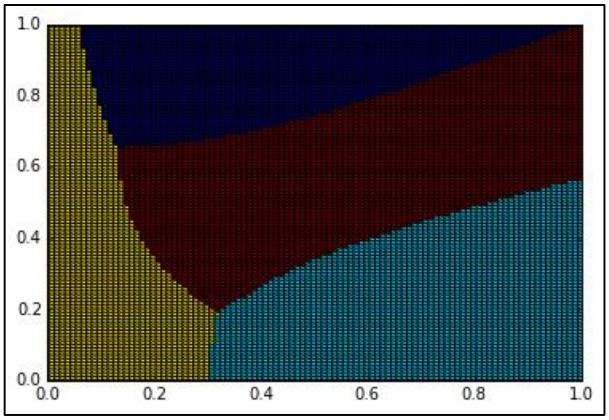
<u>After 1000 epochs:</u>

Training accuracy after 1000 epochs is 87.84 %

The confusion matrix is as follow

	Bolt	Nut	Ring	Scrap
Bolt	15	0	0	7
Nut	0	22	0	1
Ring	0	0	22	1
Scrap	0	0	0	6

The classification region can be given as



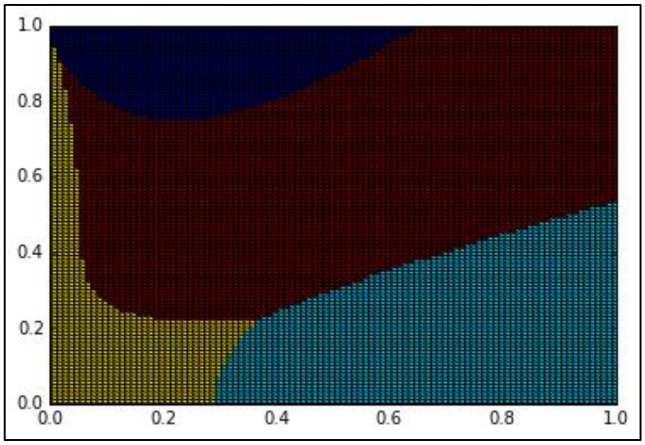
<u>After 10000 epochs:</u>

Training accuracy after 10000 epochs is 94.59 %

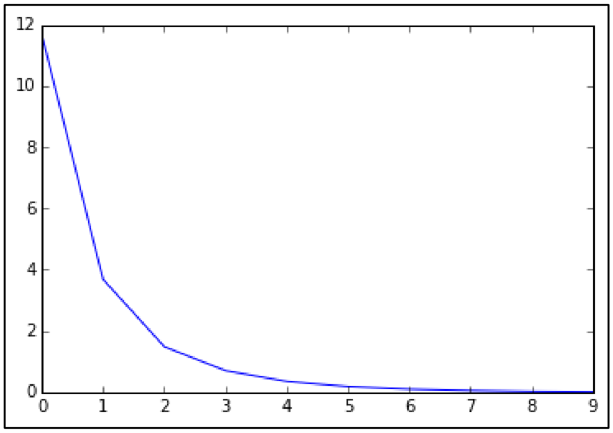
The confusion matrix is as follow

	Bolt	Nut	Ring	Scrap
Bolt	15	0	0	1
Nut	0	22	0	1
Ring	0	0	22	1
Scrap	1	0	0	12

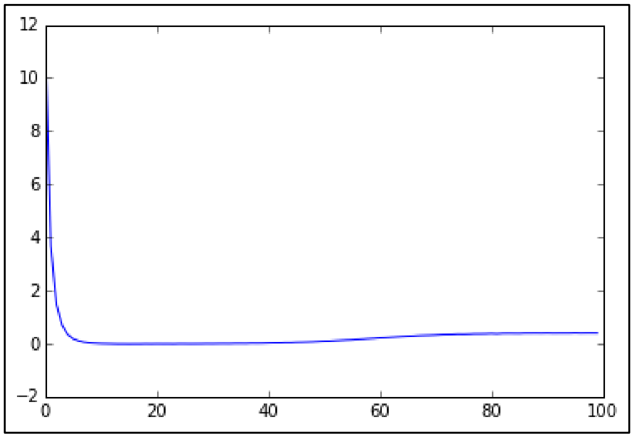
The classification region can be given as



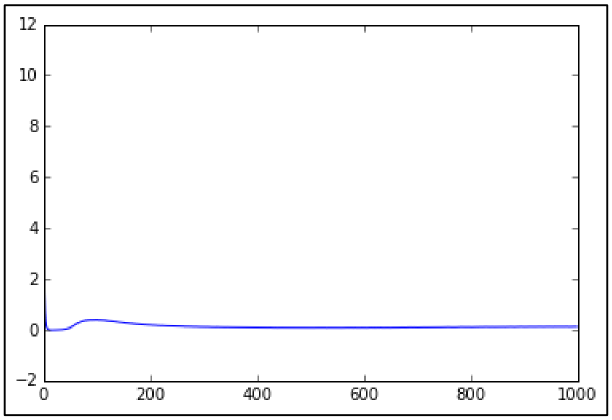
The learning curve after 10, 100, 1000 and 10000 epochs are shown below.



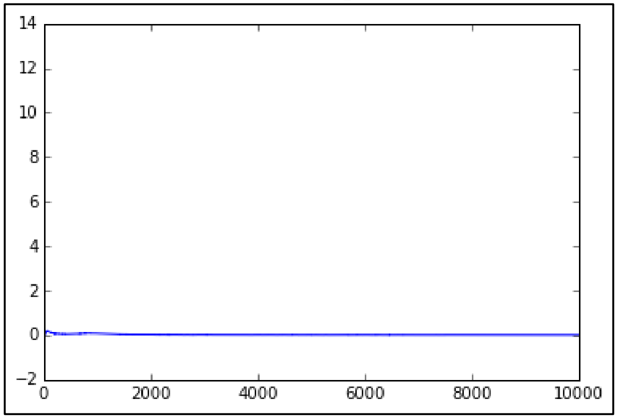
After 10 epochs



After 100 epochs



After 1000 epochs



After 10000 epochs

Recognition rate and Profit

	Accuracy	Profit
After 10 epochs	25%	-80
After 100 epochs	60%	99
After 1000 epochs	95%	199
After 10000 epochs	100%	203

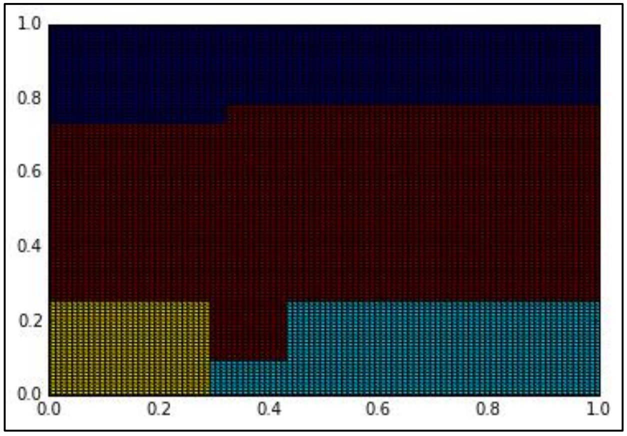
Decision Trees:

The decision tree details are shown in the table below

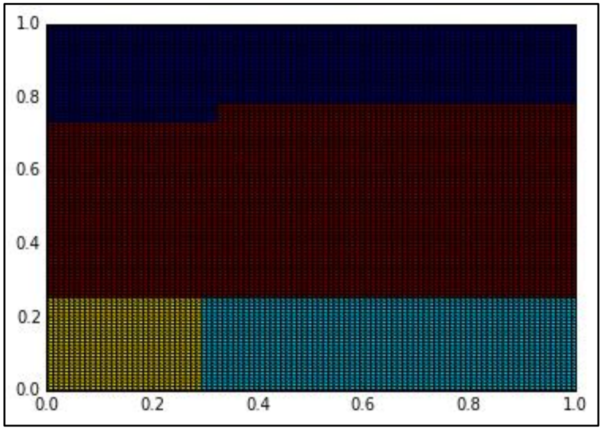
	Without pruning	With pruning
Number of Nodes	19	15
Number of Leaves	10	8
Maximum Depth	5	5
Minimum Depth	3	3
Average Depth	4.5	4.125

Classification regions

Without Pruning

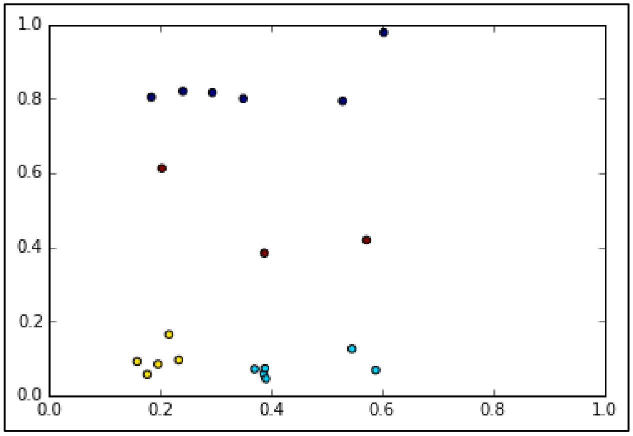


With Pruning

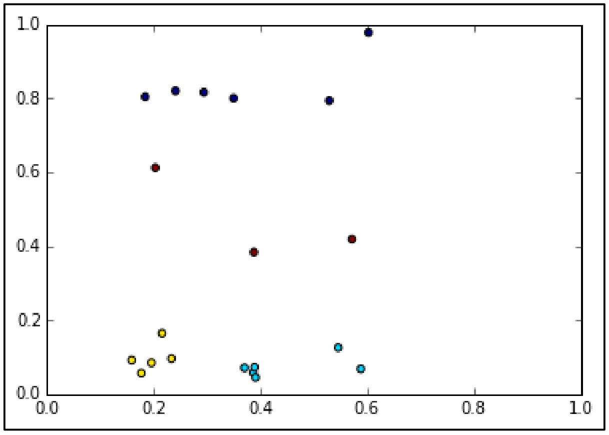


Classification of Test samples

Without Pruning



With Pruning



Recognition rate and Profit

	Accuracy	Profit
Without Pruning	95%	199
With Pruning	95%	199

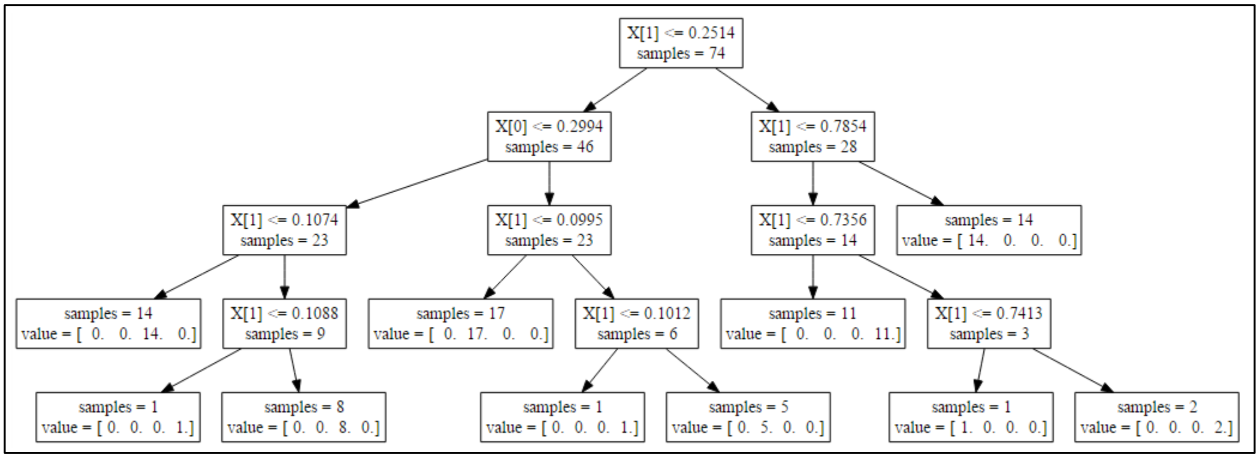
Discussion:

As mentioned in the earlier section, we expect MLP algorithm to perform better than the DT algorithm. And as seen by results of MLP (after 10000 epochs) with the DT (without pruning) we can infer that MLP does perform better than the DT algorithm. This because the MLP generalizes the feature space for its classification. As seen from the classification regions plots for the two algorithms, MLP uses quadratic lines for defining the classification boundaries whereas the decision boundaries in DT are liner in nature. This makes MLP powerful than the DT algorithm. However, in terms of training time, DT model was trained faster than the MLP model. On an average DT models were trained at least 10 seconds faster than the MLP model for 10000 epochs. There is a direct dependency between accuracy and the profit obtained for the classification. The maximum accuracy obtained for testing data is 100% with a profit of 203 units.

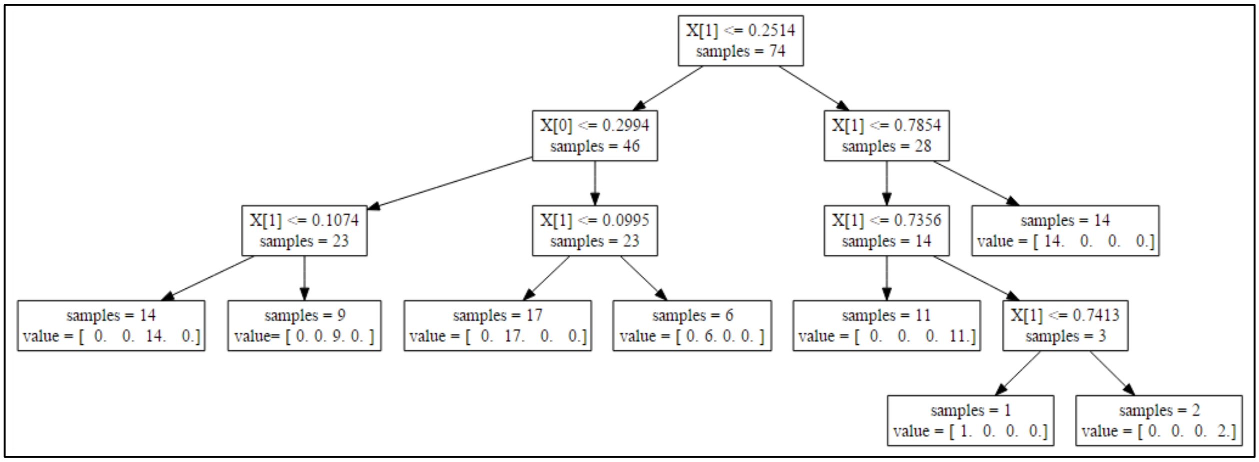
As seen from the classification regions plot for the MLP model, the model initially classifies the maximum of the data as class1 i.e. Bolt. Later on it learns to classify other classes and the region are more vivid in subsequent epochs. An interesting point to notice from the classification plots after 100, 1000 and 10000 epochs is the region for class 3 i.e. the yellow region. It increase in dimension from 100 epochs to 1000 epochs but it reduces again in the 10000 epochs. It shows how the algorithm tries to find the exact boundary of classification between the bolt and the nut region.

We tried Chi Square pruning with significance level at 1% and 5%. In total 4 nodes were pruned with 5% significance level and 2 nodes were pruned with 1% significance level. In terms of classification, tree with 1% pruning classified only one instance wrong and the tree with 5% classified two instances wrong. Since pruning with 5% significance gives a smaller tree and wrong classification of only one instance more than the 1% counterpart, we decided to go ahead with significance level of 5%.

The DT generated without pruning is as shown below



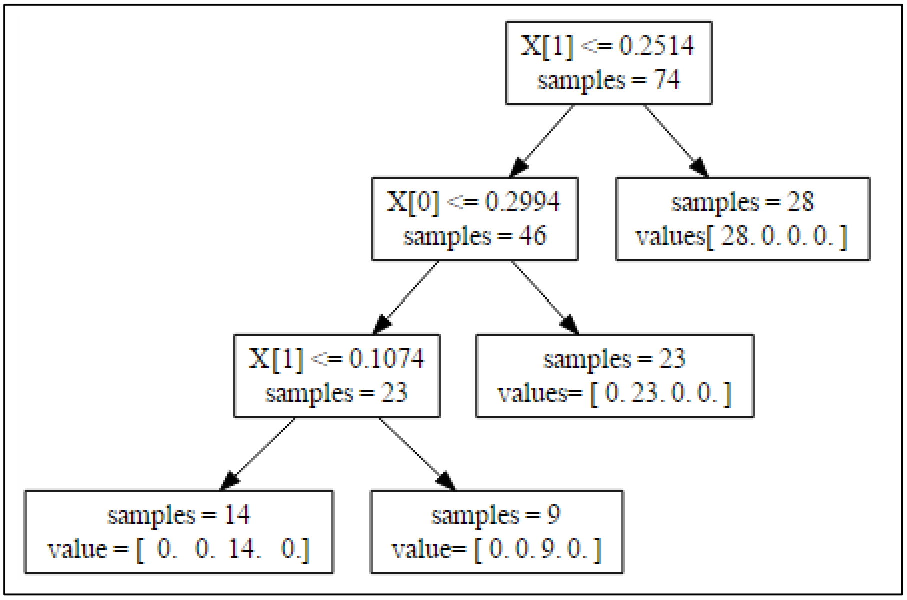
From initial look of the tree, the intuition is that all the leaf nodes at level 4 will be pruned. However, the leaf nodes at the extreme right do not get pruned. The number of instances at those nodes is less, but the ratio is higher as compared to other two pairs of leaf nodes. Since the ratio is higher, the significance level is higher and hence it does not get pruned. The pruned tree with significance level 5%, looks like this



Further we experimented with pruning significance level at 10%, we saw that more nodes were pruned. The total tree details are

Number of Nodes	7
Number of Leaves	4
Maximum Depth	4
Minimum Depth	2
Average Depth	3.25

The decision tree looks like this



The classification accuracy for this tree is 78.38%.