Below shows the data collection and plotting of the data set after letting it go through N-Fold Cross Validation, when 'N' was taken to be 10.
Below, the graphs have been plotted for Node Purity values, Depth Level of the recursion, thus the tree expansion and the Data record of a node, that is the size of the data set entering the decision tree algorithm for cross validation.

The values for each attribute were as follows:
Purity = [70 %, 75 %, 80 %, 85 %, 90 %, 95 %, 96 %, 98 %]
Data Records = [30, 25, 20, 15, 10, 8, 6, 5, 4, 3, 2]
Depth Level= [2, 3, 4, 5, 6, 7, 8, 9, 10]

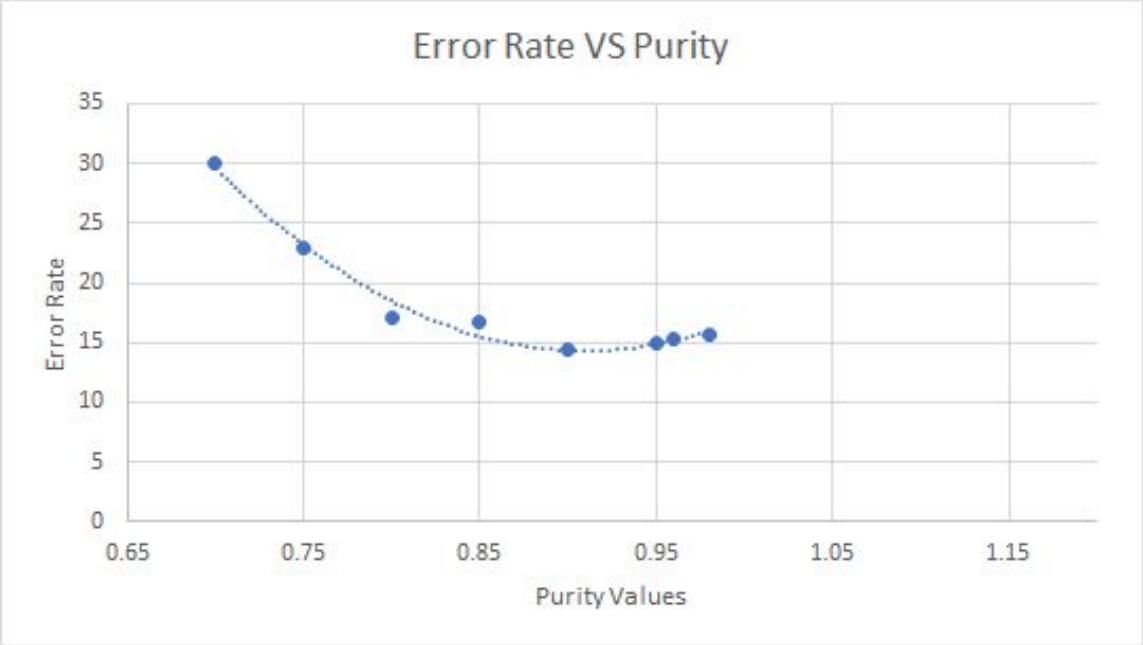

Figure 1

Purity Values	Error Rates
0.7	30.0625
0.75	22.8671875
0.8	17.1484375
0.85	16.796875
0.9	14.4375
0.95	14.921875
0.96	15.3515625
0.98	15.671875

Table 1

Figure 2

Data Records	Error Rates
3	11.94375
4	11.7375
5	11.8625
6	11.84375
8	12.025
10	11.9375
15	11.85625
20	11.86875
25	12.0875
30	12.04375

Table 2

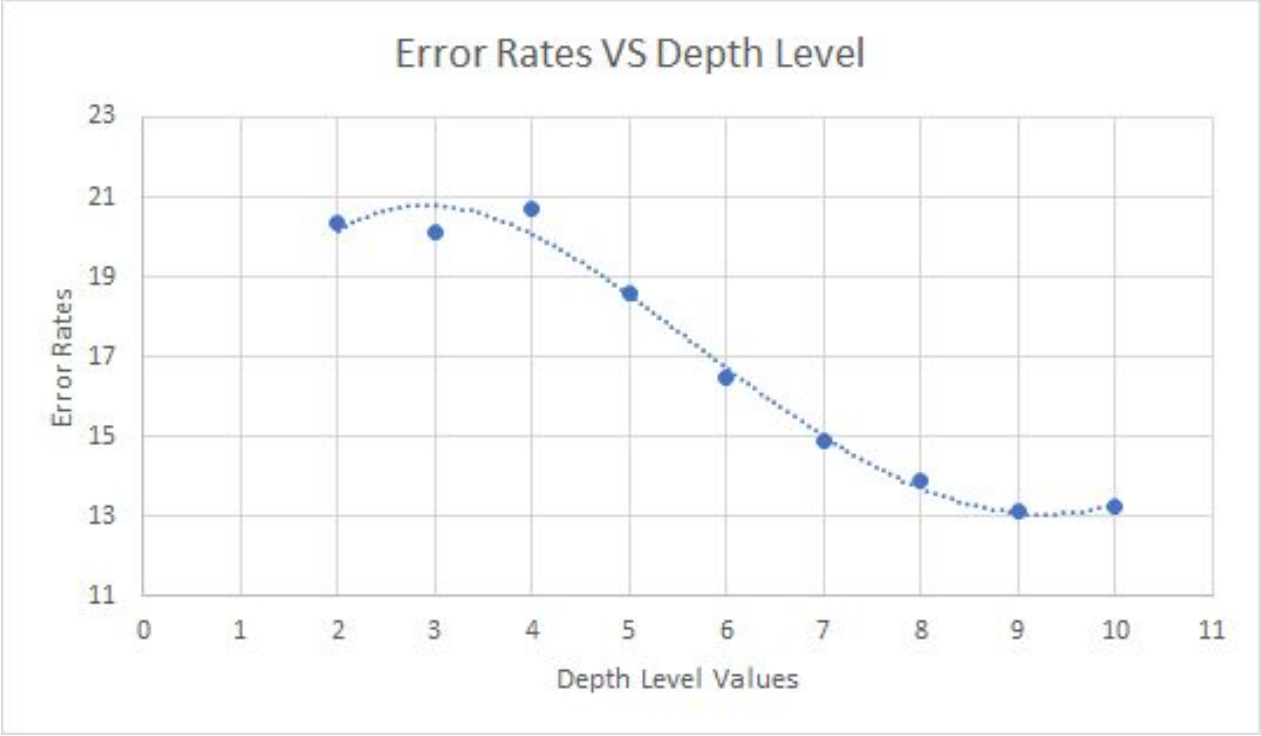

Figure 3

Depth Levels	Error Rates
2	20.32638889
3	20.125
4	20.73611111
5	18.60416667
6	16.45833333
7	14.86805556
8	13.86805556
9	13.13888889
10	13.25694444

Table 3

From the above graphs, it can be said that the experiment was a success. The graphs on Figure 1 and 3 behaved as planned, where it started from a high value, then shot down. After reaching the bottom, it then slightly rose. However, there was a slight difference in the Figure 2 graph. The graph did behave as planned, however, after slightly rising for the second time, it dipped again. From the above experiments, it can be deduced that the best possible attributes for the classification of this type of data are mentioned below:

Purity = 0.9 (14.4375 % error)
Data Records = 4 (11.7375 % error)
Depth Level = 9 (13.1389 % error)

By adding in these values and running the 10 fold cross validation on it, not only we are getting a new finalised classifier, but we also get a new list of error rates. They are shown below. From these error rates, the average is found to be 87.88125 %. This indicates that the chances of an input being Assam is 87.88125 % based on the Decision Tree that printed out the final classifier.

87.9375
88.5
89.125
86.5625
89.125
89.125
87.5
86.3125
86.9375
87.6875

Error Rates from the Final Classifier.

This particular assignment was one of the hardest but intellectually rewarding assignments. Even though we used the core part of HW-05, we need to automate the part of testing the decision tree with different parameters for depth level, data record and purity. Even though logically we implemented the whole thing by Sunday, the thing we struggled with most is explained below:

- In HW-05 our mentor file spits out a training file. What we did in HW-07 is we, after the decision tree spits out the classifier file, we run the 9 sets (1 set was used for training) for testing and see how much accuracy we got. Now, because our classifier is getting rewritten in every iteration, we need to import the classifier file every time as a brand new one. This is where the issue started, our classifier file after the first iteration, was getting cached (even though we explicitly used __import__. We tried many methods, but because it was getting cached, after the first iteration, even though classifier was being built in every run, the training file was not using it. On Wednesday night, after countless hours of trying different methods, we used python importlib's reload module which solved the issue of the file getting cached in the memory, thus, our newly built decision tree was used in every run.