**Programming Assignment-5 Decision Tree- Part II**

**First Name1: \_\_Richu\_\_ Last Name1: \_\_\_\_Mathew\_\_\_**

**First Name2:\_\_\_Andrew\_\_\_ Last Name 2: \_\_\_King\_\_\_\_\_\_\_\_**

**Introduction**

Implement the ID3 decision tree learning algorithm that we discussed in class. To simplify the implementation, your system only needs to handle binary classification tasks (i.e. each instance will have a class value of 0 or 1). In addition, you may assume that all attributes are binary-valued (i.e. the only possible attribute values are 0 and 1) and that there are no missing values in the training or test data.

A sample training file (train.csv) and test file (test.csv) are provided. In these files, only lines containing non-space characters are relevant. The first relevant line holds the attribute names. Each following relevant line defines a single example. Each column holds this example’s value for the attribute named at the head of the column. The last unlabeled column holds the class label for the examples. In all of the following experiments, you should use this last class attribute to help train the tree and to determine whether a tree classifies an example correctly.

When building a decision tree, if you reach a leaf node but still have examples that belong to different classes, then choose the most frequent class (i.e., the majority class among the instances at the leaf node). If you reach a leaf node in the decision tree and have no examples left or the examples are equally split among multiple classes, then choose the class that is most frequent in the ***entire*** training set. You **DO NOT** need to implement pruning.

**IMPORTANT**: Don’t forget to use logarithm base 2 when computing entropy and set 0\*log2 0 to 0.

1. You have already implemented “entropy” and “information gain” functions in Part I. Now, implement the ID3 algorithm and build a decision tree using the training instances.
2. Use the learned decision tree to classify the training instances. What is the accuracy of the tree? (In this case, the tree has been trained and tested on the same data set.) The accuracy should be computed as the percentage of examples that were correctly classified. For example, if 86 of 90 examples are classified correctly, then the accuracy of the decision tree would be 95.6%.

[Answer]

**Ans:**

85.375 %

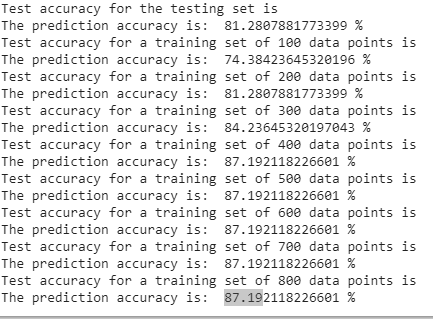


1. Use the learned decision tree to classify the test instances. What is the accuracy of the tree? (In this case, the decision tree has been trained and tested on different data sets.)

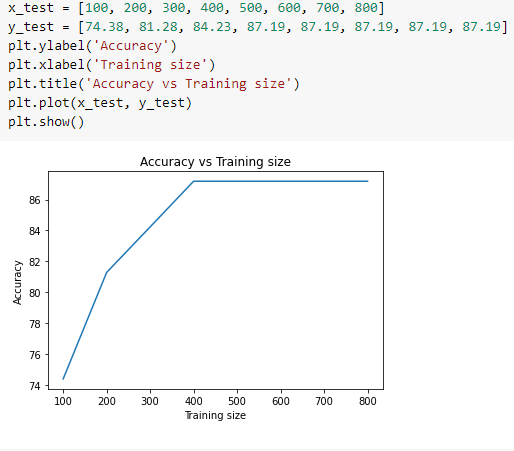
[Answer]

**Ans:**

87.19



1. Now, we want to investigate how the amount of training data affects the accuracy of the resulting decision tree. Plot a learning curve (i.e., a graph of the accuracy of your algorithm on the test set against different training set sizes) by re-training your learning algorithm using training set sizes of 100, 200, 300, . . ., 800.
2. Please paste your learning curve image below.



1. Briefly comment on the shape of the curve. Does it exhibit the usual properties of a learning curve?

It reflects properties of previous learning curves, the only difference is the representation of accuracy trending up opposed to the decline of loss.

**Additional Notes**

When reporting accuracy, two decimal places are sufficient. When making graphs, remember to label each axis and to provide a title that indicates what the graph is depicting;

**Submission:**

* **Rule1:**
  + If you work with a partner, please name your zipped file as follows:

PA5\_Part\_II\_LNAME1\_LNAME2.Zip for folder and PA5\_Part\_II\_\_LNAME1\_LNAME2.docx for a word document, i.e., the file names should include both LAST NAMEs.

* + If you work on your own, the format should be

PA5\_Part\_II\_\_LNAME.Zip for folder and PA5\_Part\_II\_\_LNAME.docx for a word document.

* **Rule2:**
  + Put your FULL names whether working in a group or individual in the word document that answers all the questions.
* **Rule3:**
  + **EVERYONE** in the class should submit this Assignment, which should provide all files (like jupyter notebook, test excel files etc.. ) that are necessary for the execution of code in the submission folder.