Build a Decision Tree

Description:
In this assignment you will build a fully functional decision tree based on the ID3 algorithm. The decision tree will read a set of data specified by the user, calculate the optimal decision tree, then output the performance metrics of the decision tree based on 10-fold cross validation.

Requirements:

- The implementation should be written in Java. You can do it in Python, but it must adhere to object oriented principles.
- Implementation should be based on the Iterative Dichotomizer 3 (ID3).
- The tree should grow to maximum size given the data.
- The tree should be pruned back from its full size (using reduced error pruning) - *No need to implement, but you can just play with it to understand more.*
- The final pruned tree should be evaluated with 10-fold cross validation.
- The performance metrics output needs to include the confusion matrix and accuracy (1-misclassification rate).
- The code must be commented so that each implemented component of the decision tree algorithm is properly labeled.
- Use object-oriented techniques (do not make the whole program one monolithic procedural implementation).
    - Separate the entropy calculations, information gain calculations, the data structure, the tree structure, and the cross validation components so that each of them can be developed and reviewed distinctly - *Output these metrics to the tree structure - more details below.*

Data Set:
For this assignment you will use the well-known and manageable Iris flower data set. It can be downloaded at:
http://en.wikipedia.org/wiki/Iris_flower_data_set
or
http://archive.ics.uci.edu/ml/datasets/Iris

Resources:

- http://en.wikipedia.org/wiki/ID3_algorithm
- http://en.wikipedia.org/wiki/C4.5_algorithm
- http://en.wikipedia.org/wiki/Pruning_(decision_trees)
- An Analysis of Reduced Error Pruning (PDF)
- http://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation
- http://www.cs.cmu.edu/~cga/ai-course/overfit.pdf
- http://stackoverflow.com/questions/7619700/10-fold-cross-validation

Build a Decision Tree

Output Tree:

Your final tree would be something like the figure shown below. Note: It may not be exactly the same for your dataset. (Source)

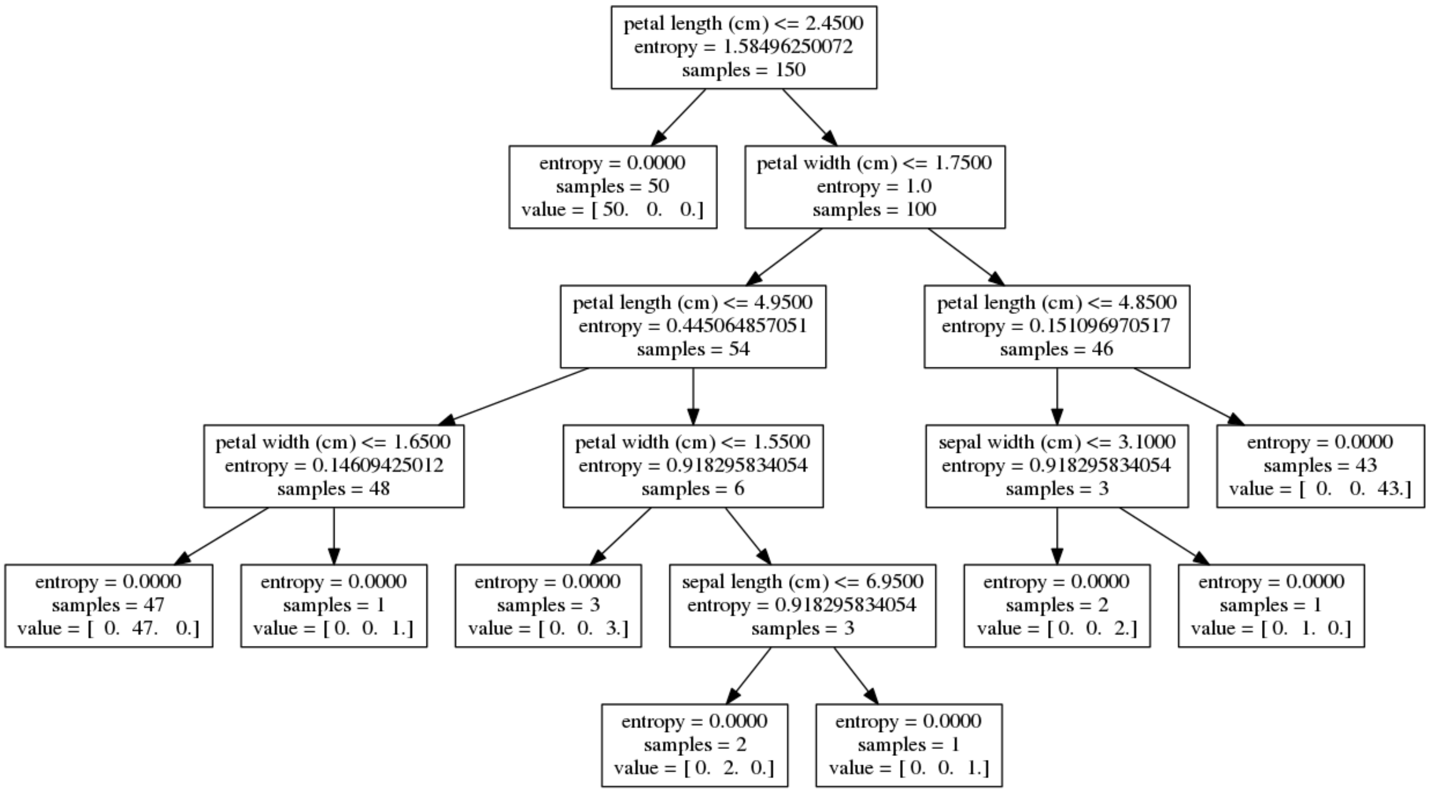
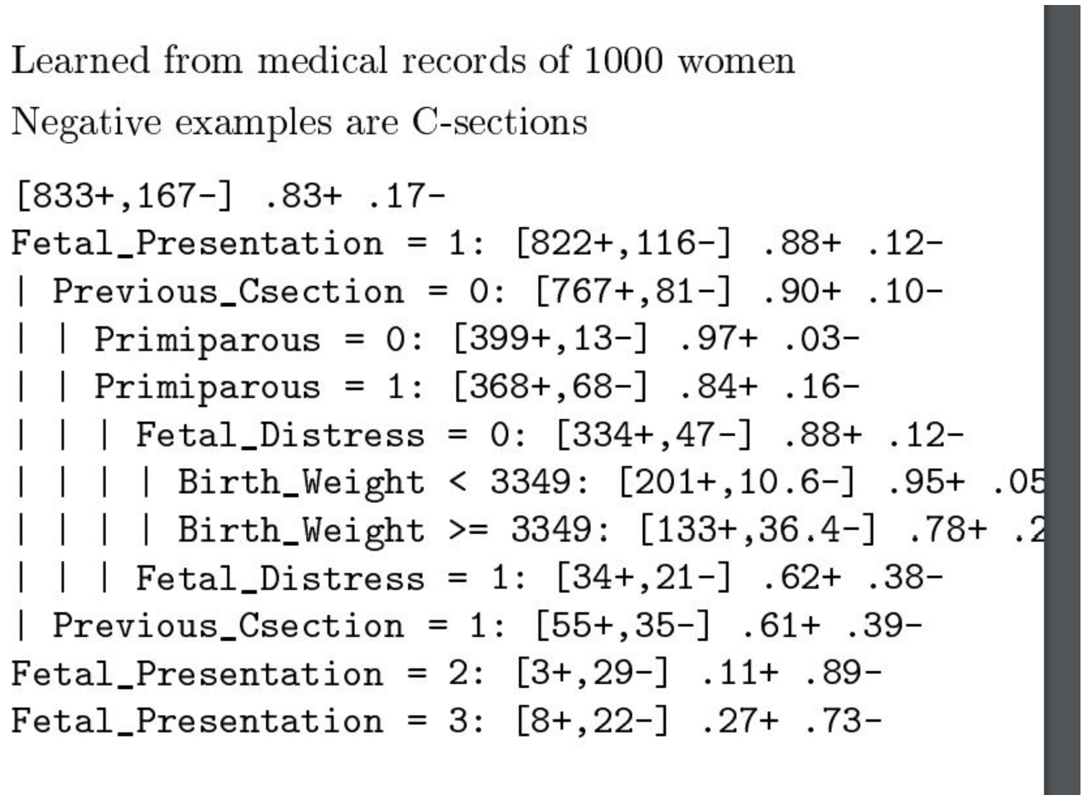


Figure 1.

You should build your decision tree and output the results like the tree shown below. The below example (figure 2) is from Tom Mitchell's slides.

Build a Decision Tree



Learned from medical records of 1000 women

Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .2
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Figure 2

What the above diagram means - The first line shows the total number of positive and negative examples. In our case, positive and negative examples would the the data greater than or less than some value with respect to a feature. Next you print the feature, along with the decision (like =, <, >, etc) and then print the positive and negative examples beneath that node in the tree. Then you also have to print the entropy values that are calculated for each node, next to it.

You need to print the above structure (Figure 2, and not Figure 1) in a text file and submit it along with your code. It is fine if your tree doesn’t match exactly to the one shown in Figure 1 (it is just an example), we will check your implementation and grade you on the accuracy with respect to the tree generated by TAs.