Checkpoint 4
Due November 2, 2016

This checkpoint will add our first advance machine learning algorithm, a decision tree. This will allow us to practice generating trees.

Objectives:

1. Understand our plans for moving forward on GUI
2. Build a tree with recursion
3. Use a tree to compute some decision
4. Provide experience doing tree walks

Interface:
At this point, we need to start to build a more complex GUI interface. As referenced in checkpoint 2, I recommend an interface similar to that of JGAAP. However, the choice of how you define your interface is up to you. In this checkpoint, you will need to provide me a plan on moving forward for your interface. This will be a document with pictures of what your interface will look like (This can be done with word art or ms paint), and a verbal description to go with it. Your interface must be able to do the follow:

1. Allow the user to upload multiple documents and keep track of them.
2. Allow the user to add characteristic information to a document that might not be found in the file, such as genre or year
3. Allow the user to apply multiple textfilters to the document after uploaded
4. Allow the user to select a statistical method, then select which documents and attributes (author, year, word count, etc) of the document the statistical method will be trained.
5. Use a trained statistical method to try to predict.

Decision Tree:
We will be using a decision tree built using the ID3 Algorithm. A decision tree is a way to provide a hierarchy of partitions to a set of data. We will need to build a class for our decision tree. The class has a method called train that will take in a 2Dlist of data where each row is a new document's information and each column is an attribute, a 1D list of the column titles (These are classifiers for which the decision tree will be built from and will assume the first row is the classification key), and an integer that will be the maximum depth of the tree. The class will also have a method called eval, which will take a 2Dlist of data where each row is a new document's information and the first column are all None as they will be assigned by the tree. Also we will need a decisiontree Node. Each node in the tree may have multiple children. Each node will have to keep track of a data attribute that stores the document attribute used for the decision. It will also have to keep track of number of documents that have this attribute. Please see class notes of the big ideas of decision tree.

Testing:
Testing does not have to be done through the GUI interface at this time. You may just provide a script called checkpoint4.py that will do the following.
Here our data will be 10 training text examples given for this checkpoint. These text examples are from a number of sources and vary in genre, topics, and year. But some are by the same person! We will build our decision tree in order to classify based on author (our classifier key = author). You will also be given 5 examples with unknown authorship to test our tree after (This is the fun part!). Build decision trees for both authors. Write pre, post, and infix walks for the tree and print the walks for both decision trees into a file marked with the author's name. Make sure to turn in these files too.

ID3 Algorithm Pseudo Code

```
Function ID3 (data, input attributes)

If (data is empty): return failure

If (if all rows of data have the save value for an author):
    Return node with that value

If (input attributes are empty):
    Return a single node with the value of the most frequent value of author

Compute the gain for each attribute:

Let X be the attribute with the largest Gain(X,T) of all attributes
Let {x_j | j =1,2,…., m} be the values of X
Let {T_j | j = 1,2,…, m} be the subset of T when T is partitioned
    Make a node labeled with X, have the node store the labels of edges to children and counts.
    Call ID3(T_1, input attributes – X), …., ID3(T_m, input attributes – X)
```

Computing the Gain(X,T): where p is the probability of independent event.

$$H(P) = -\sum_{i=1}^{n} p_i \lg\ p_i$$

$$H(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|}\ H(T_i)$$

$$Gain(X,T) = H(T) - H(X,T)$$