Big Data Analytics Assignment 3

Implement and train a simplified random forest

**Task:**

Using the decision tree and data matrix that you have created, you will now create a simplified random forest model to learn how to predict the labels you have given your data. The long-term goal is to create a MapReduce random forest implementation, but the next step in learning how to accomplish that goal is to aggregate the decision tree weak learner you already have made into a non-map reduce random forest implementation. This provides the code-template that you will use to create the fully distributed ensemble method.

This random forest need only take a single input parameter N. N specifies the number of trees to grow. None of the other traditional random forest/bag of trees input parameters need to be supported at this point.

**Minimum deliverables:**

Source code that creates the random forest

README.txt if more than one source file

Summary statistics that shows the performance of your random on a sample of your data

**Due Date:**

One week from assignment (10/8/15)

**Requirements:**

The random forest should be implemented in Java, so that it can later be easily translated into the MapReduce paradigm. Your code must be commented, and if there is more than one source code file, you must include a README.txt file that explains the relationship of all source code files.

You must split your data into training and testing sets for verifying your random forest functionality. The split must be 80% train, 20% test. The sample should be no less than 10,000 instances, but can be larger. You can reuse the training/test split you created for the decision tree weak learner assignment.

**Hints**:

* Remember that random forest makes its final decision by a vote of all the weak learners
  + Classification decision is the majority weak-learner classifications
  + Regression decision is the mean of the weak-learner regressions
* Again, keep your data structures clean! Adhering to good object oriented principles and taking a little extra effort to comment your code well can go a long way.
* Don’t forget that you train each tree of the forest on only a subset of the data
  + Randomly selected subset of about 2/3 of the instances (rows)
  + Randomly selected subset of about .
* There are tutorials for random forest methods readily available online.
* The basic algorithm for growing a random forest is:
  + Grow your tree and add it to the forest
  + Send the 1/3 of data not used in training down the forest and get the performance statistics. Persist these in a data structure.
  + If N > 1, iterate until N trees are grown.
  + Persist the grown forest for later use.

**Readings and Resources:**

*Bootstrap Aggregation:*

https://www.youtube.com/watch?v=5Lu1eTiX7qM

*Random Forest:*

https://en.wikipedia.org/wiki/Random\_forest

http://www.bios.unc.edu/~dzeng/BIOS740/randomforest.pdf

https://www.youtube.com/watch?v=5Lu1eTiX7qM

https://www.youtube.com/watch?v=loNcrMjYh64

*Random Subspace:*

https://en.wikipedia.org/wiki/Random\_subspace\_method

*Trees, Bagging, Boosting in General:*

http://jessica2.msri.org/attachments/10778/10778-boost.pdf