TO RUN THE DECISION TREE BUILDER:

First do:

subl ~/.bashrc

copy the contents of bashrc in the instructions directory to your .bashrc

then do:

source ~/.bashrc

Getting the files:

git clone <https://github.com/harrisonzhao/distributedDecisionTree>

cd into treeBuilder

Create required directories:

You need to create the following directories in hdfs:

1. /tmp
2. /tmp/input
3. /tmp/output

Try doing:

hfs –ls (directory name)

to check if the directory is already made

If a directory does not exist yet, do:

hfs –mkdir (directory name)

add input files to hdfs’ /tmp/input

for this example, I’ll use the chesstrain.csv dataset found in treeBuilder’s directory

first do

hfs –ls /tmp/input

to check if chesstrain.csv is already on the hdfs

if chesstrain.csv is not there and /tmp/input is not empty, do:

hfs –rm /tmp/input/\*

to remove all other files that would be considered as input, then do:

hfs –copyFromLocal chesstrain.csv /tmp/input

to upload chesstrain.csv from the local machine to hdfs

AFTER SETUP IS DONE:

BEFORE RUNNING: check pom.xml

Make sure the command line arguments are correct (there should be a multiline comment right before where one can edit the line). For chesstrain.csv, a line of input has 37 comma-separated values per training line. The output class is the 37th value in the line. If one tokenizes the line and loads it into an array (which is 0 indexed), the output class will be at the 36th index (36 is what the command line argument should be set as).

In treeBuilder directory:

TO CLEAN:

mvn clean

TO BUILD:

mvn compile package

TO EXECUTE:

mvn exec:exec

Once the job completes, the output tree xml file will be in the local directory of treeBuilder named tree.xml

It should also be in /tmp/output/[job id]/tree/tree.xml

You can copy files from hdfs to local using:

hfs –copyToLocal [path to directory or file in hdfs]

to clean up the output directory do

hfs –rm –r /tmp/output/\*

TO RUN THE DECISION TREE TESTER:

Move the outputted tree.xml file into the treeTester directory.

Rename in appropriately. Make sure there’s a test file corresponding to the data set. Edit the pom.xml command line arguments to accept the output tree file and the test file.

In treeTester directory:

TO CLEAN:

mvn clean

TO BUILD:

mvn clean compile package

TO EXECUTE:

mvn exec:exec

there should be a stdout message similar to the following:

Result: correct: 733/ total: 779 - accuracy: 94.09499358151476