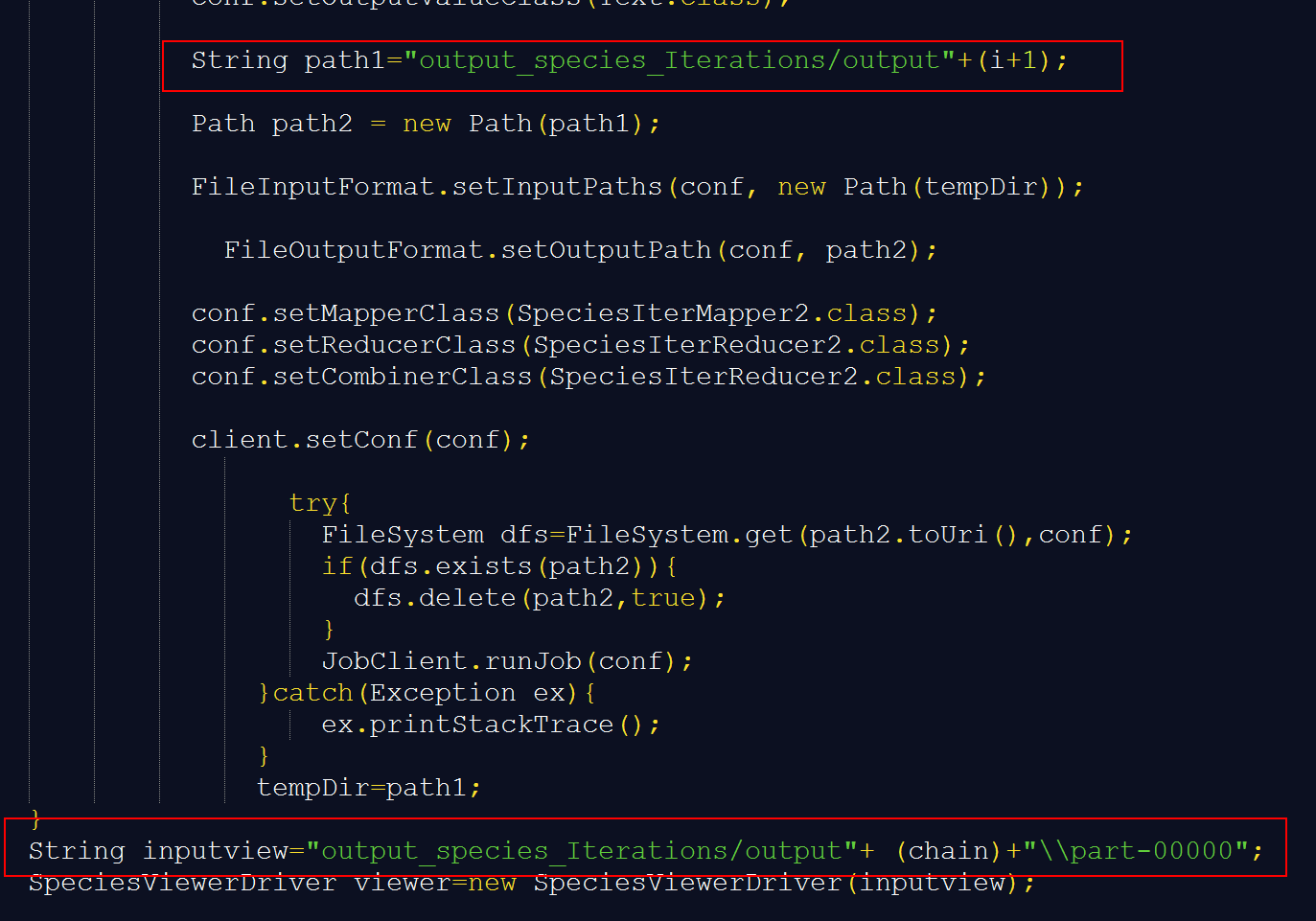
**Summary**

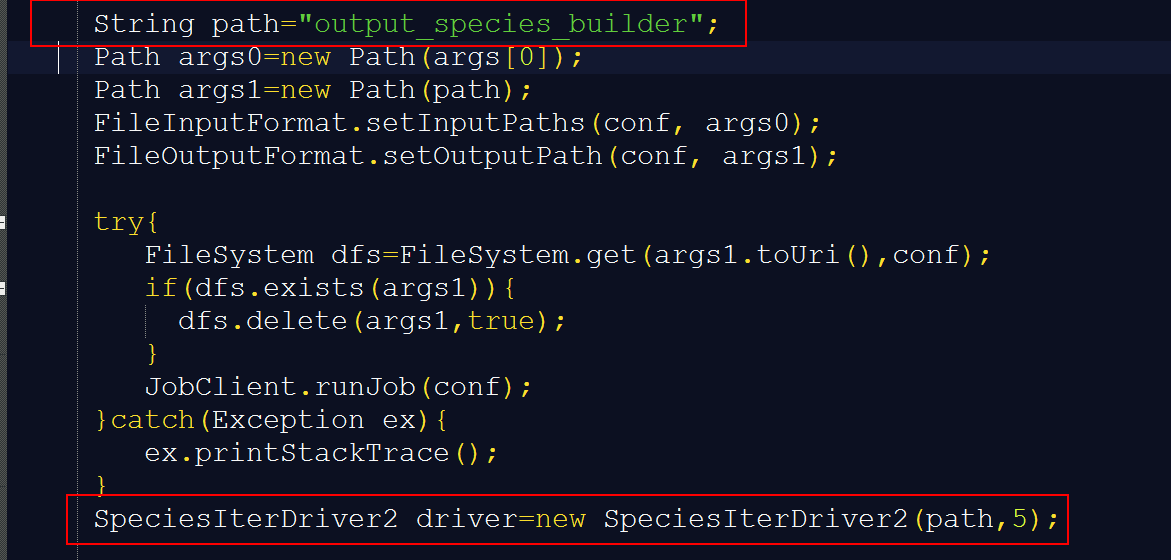
Part 1

First, follow the step on the Species Lab Slide and make 9 code file for the Page Rank,

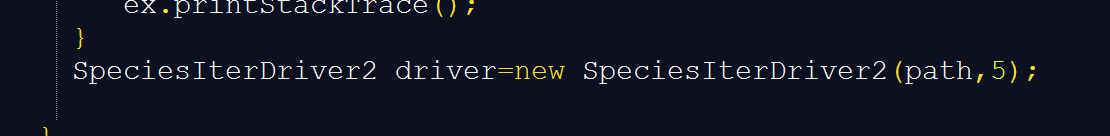
Separate them in 9 different java file.

Run the file in the Cygwin and get the first output.

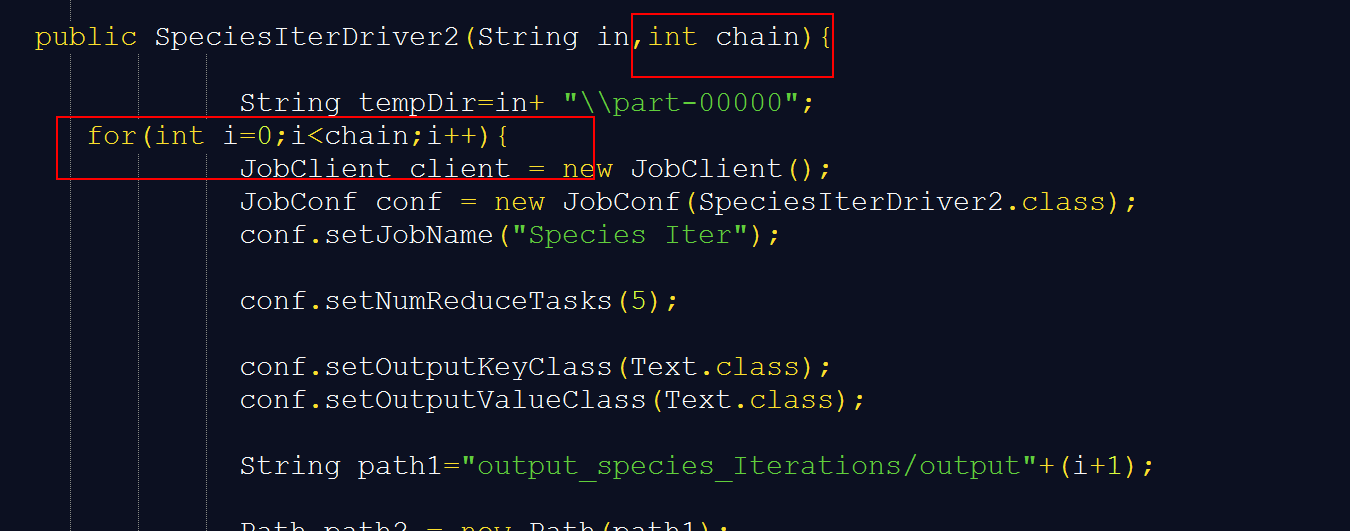
In my code, I have already set the input file and output file ‘s directions and every driver or runner can read other’s output file automatically :



and my iterations run for 5 times:



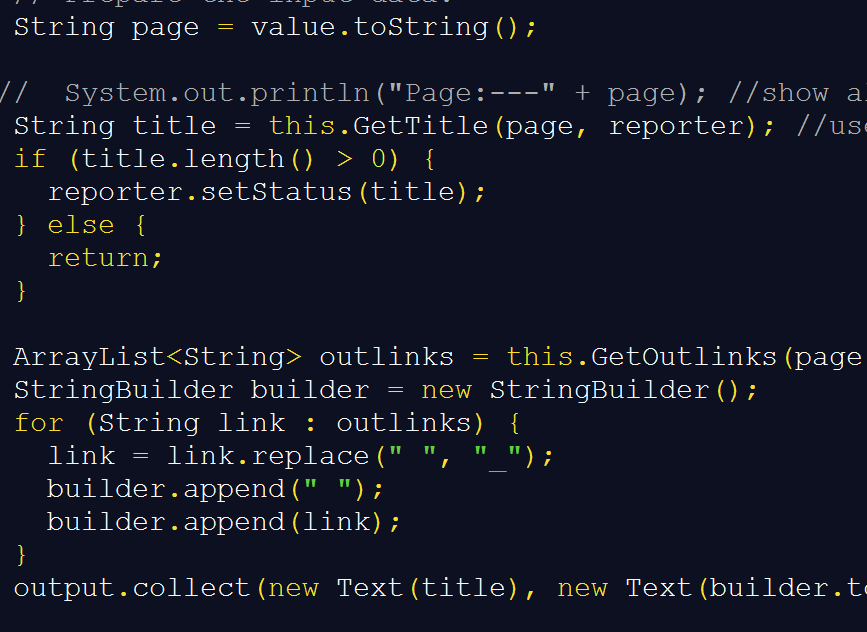
5 is the parameter for the function SpeciesIterDriver class because I have made a for loop for the iterations:



In the Mapper part：

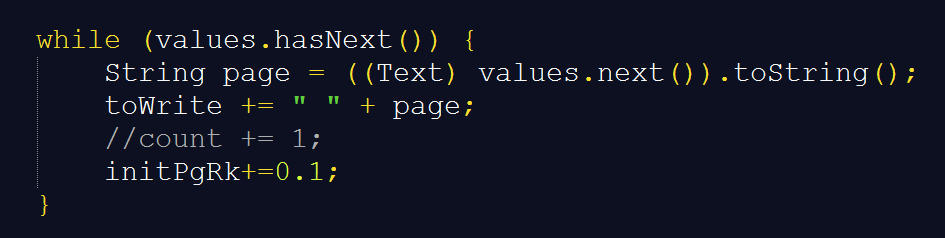
We can see that in the code we can recognized about the xml files and split it than transferred it into the output file.

In the Mapper part, we can get the data from every line, and get the position of “:”, then split the data to the split[0] and split[1].



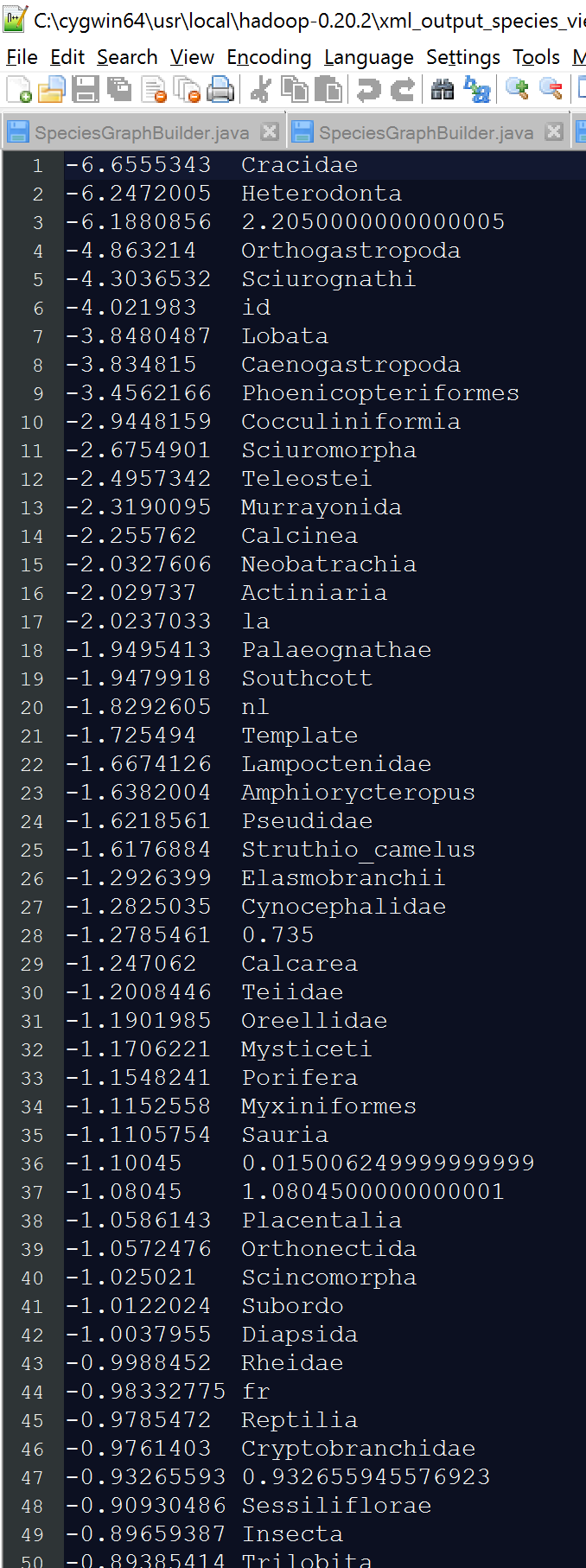
In the Reducer, I make every value follow with 0.5.

And In the Iterator Mapper part, we calculate the numbers of every title and print the outlink which followed with previous title and then calculate how many titles are direct to this outlink. In the reducer, we get the values and the numbers we calculated before, then we judge from the “:”.



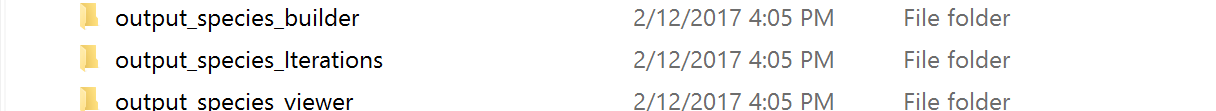
And change the initial page rank to 0.1 instead of 1;

In the viewer, we get the page and split it into title and PR. And get the final results.

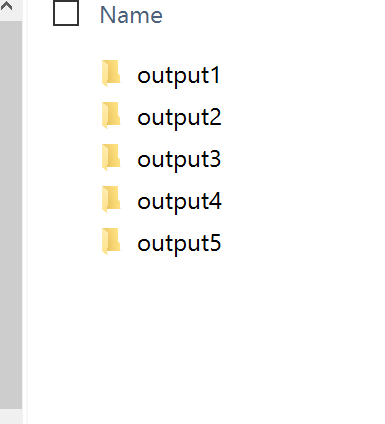


After if run the iteration 5 times, it converges.

Than we got the output files for every steps:



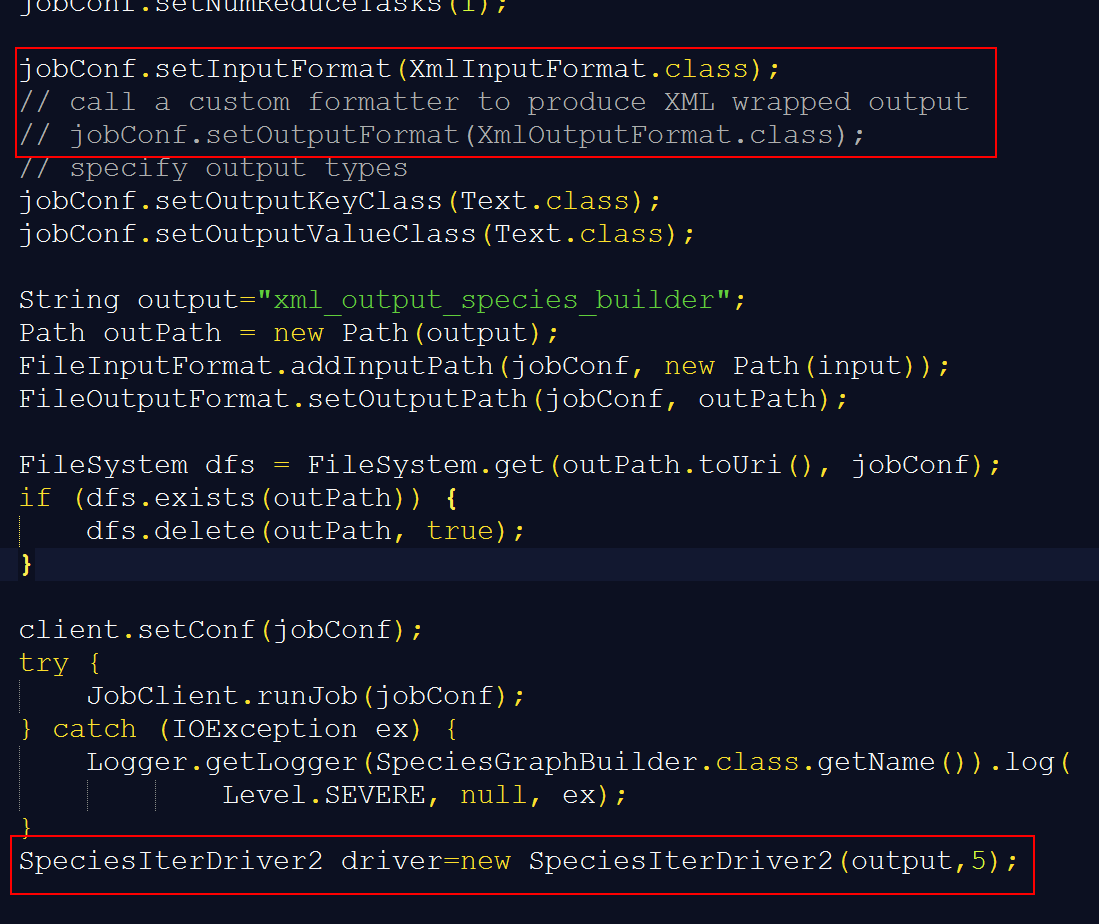
Iterations:

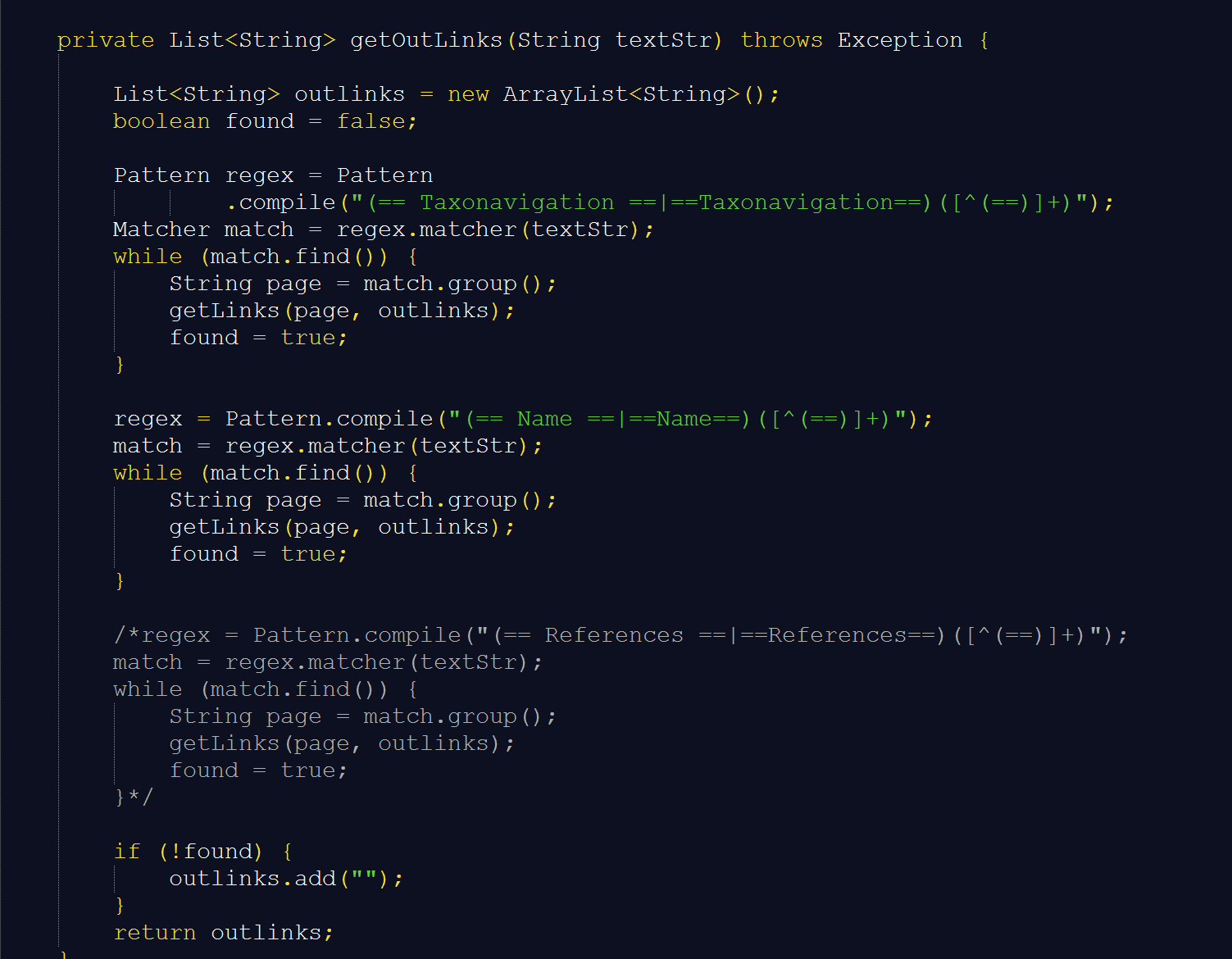


***Change input file to XML***

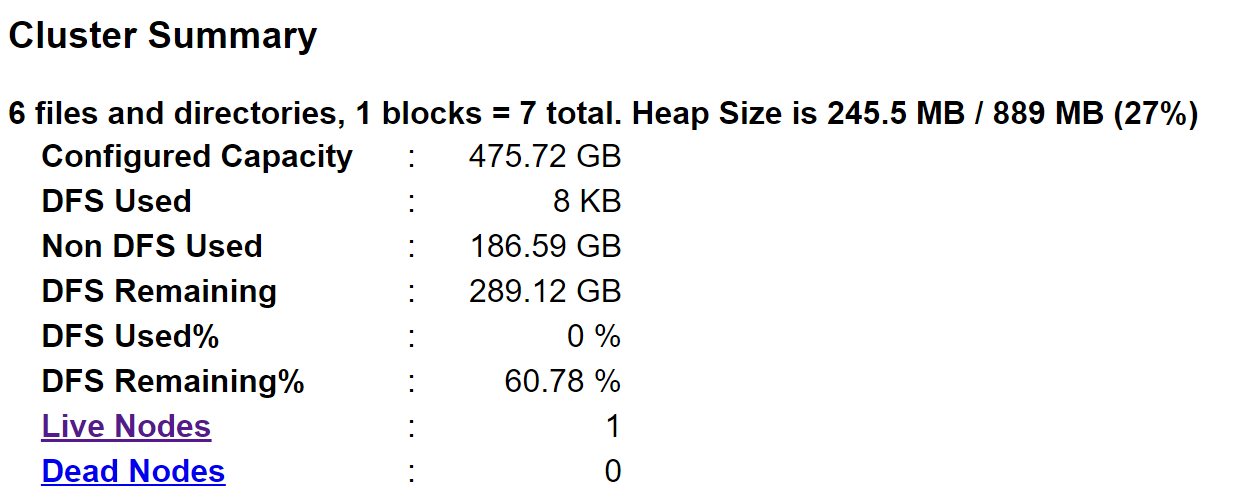
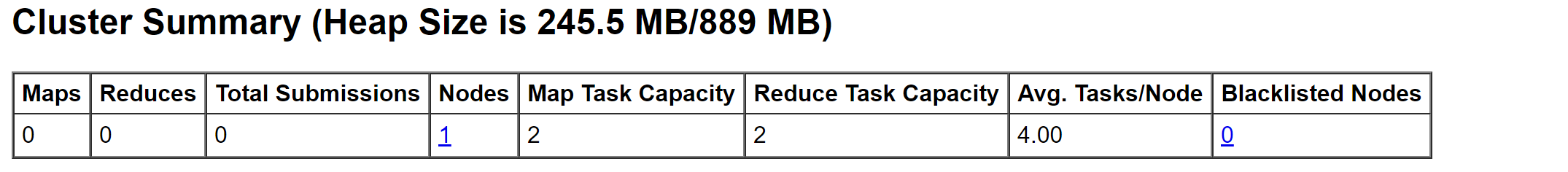
In this week, I change the input file from txt format to xml format, and the run mode is changed from single mode to pseudo distribute mode:

In the code, I change the code in the SpeciesGraphBuilder:

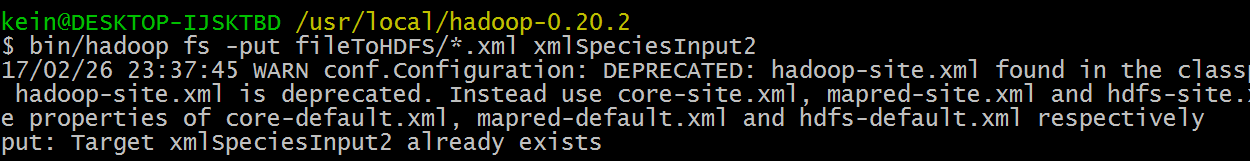


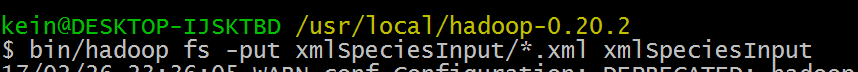


***Run in Pseudo Distributed Mode:***

Config pseudo distributed mode: 

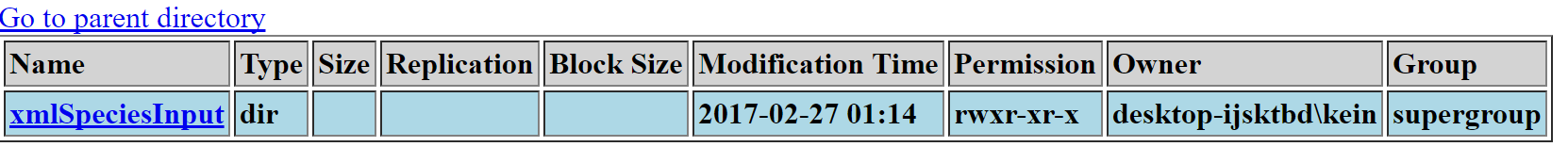
First , configure the mode until it have the 5 jps results, then set the input files to the HDFS:





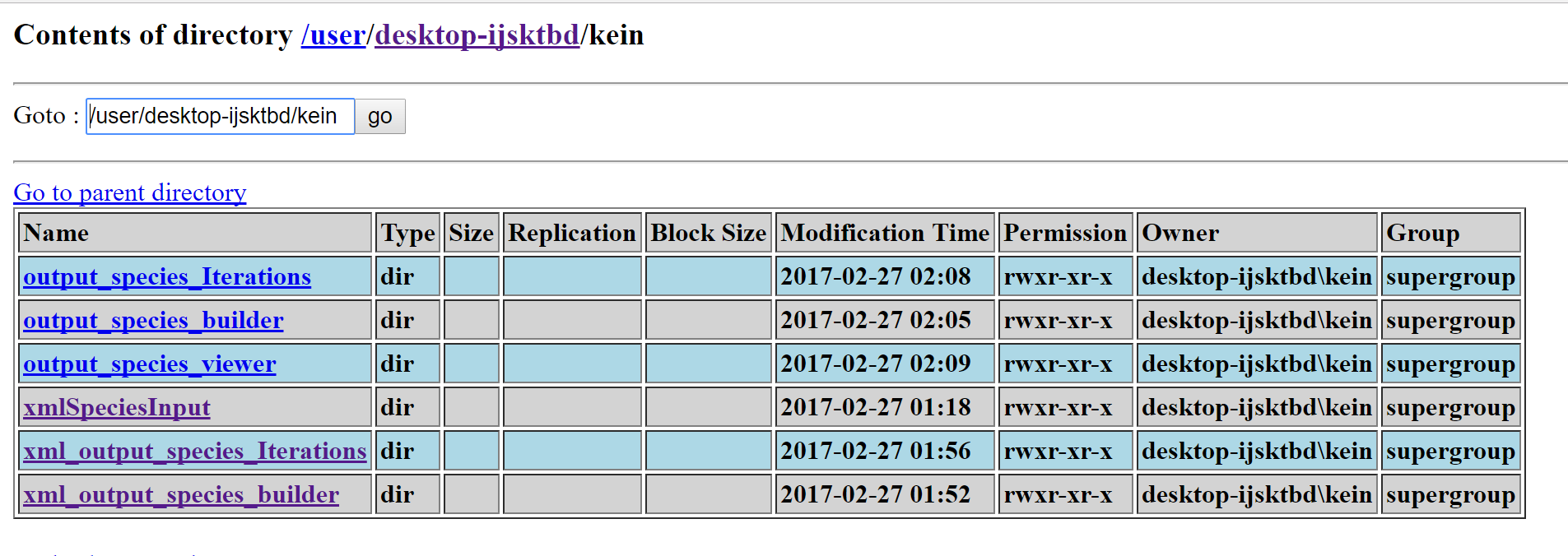
Run it in Cygwin:

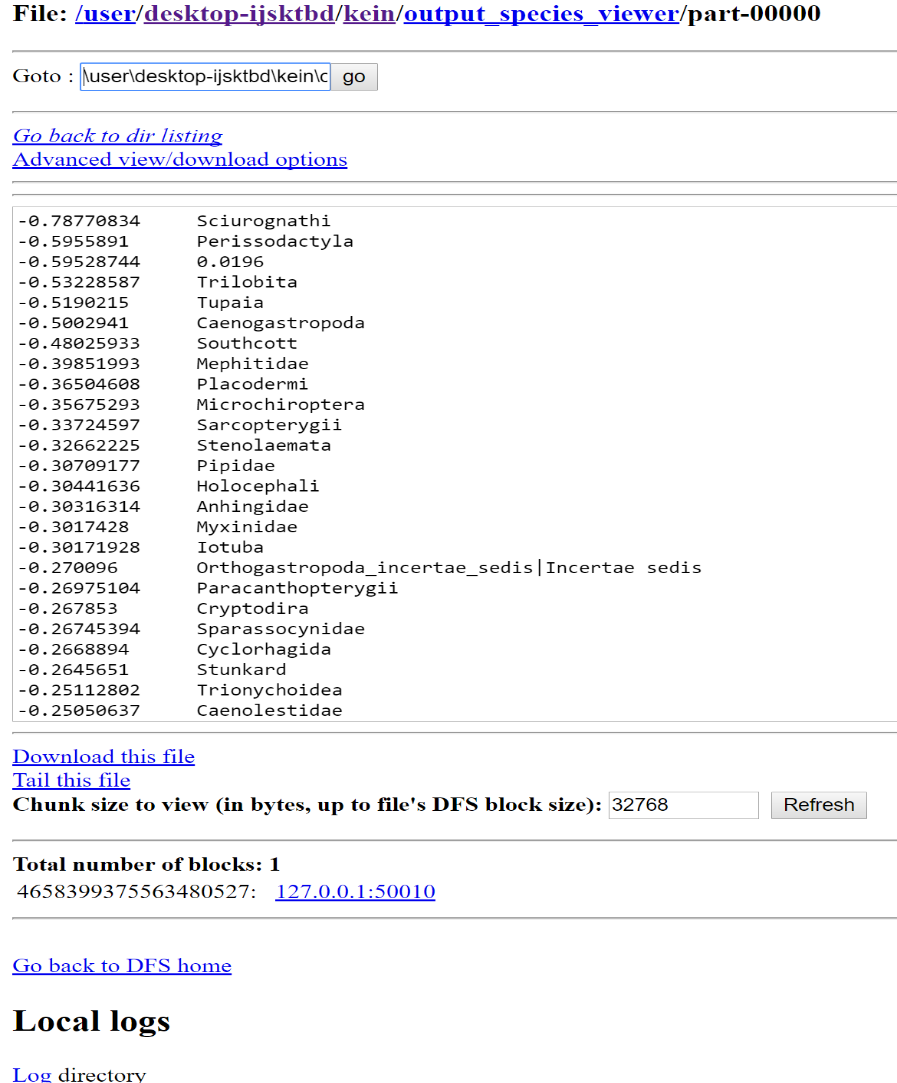




***GET Result:***

Output file:

***Final Result:***



P.S. Because I have tried many times so there are lots of meaningless folder in my HDFS , so the outputfile’s name are different in the single mode and the HDFS. Except that, all the same.

# Part 2

We just need the Mahout jar to run with Hadoop. Follow the step in the slide. And then run it in local and pseudo mode. Use the u.data and change it to the txt mode. After that, use the command line:

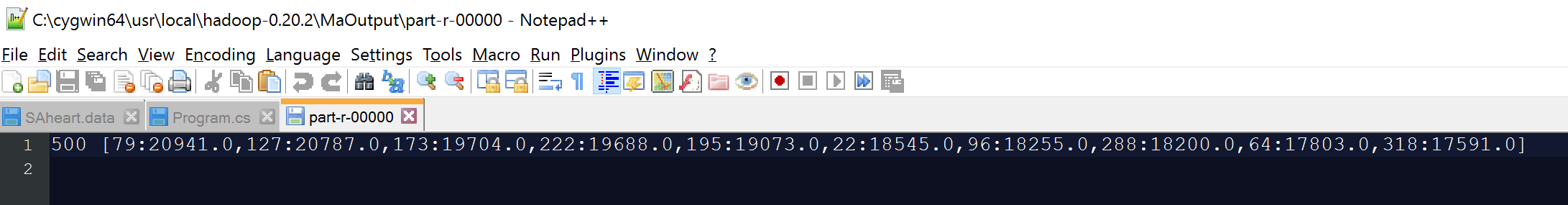
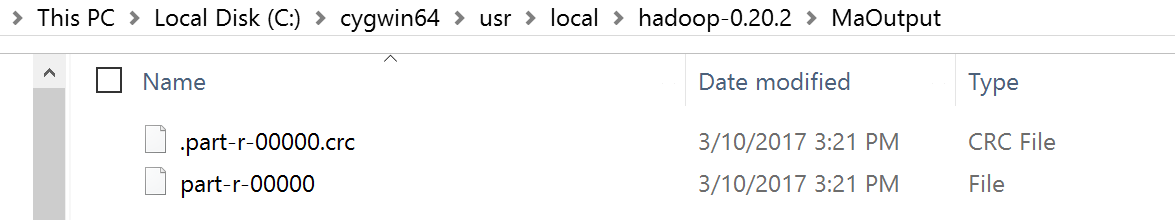
*bin/hadoop jar mahout-core-0.7-job.jar org.apache.mahout.cf.taste.hadoop.item.RecommenderJob -Dmapred.input.dir=input/input.txt -Dmapred.output.dir=MahoutTmpOutput --usersFile input/users.txt --booleanData true --similarityClassname SIMILARITY\_COOCCURRENCE*

When try it once, we need to delete the file it automatic create in order to run it again:

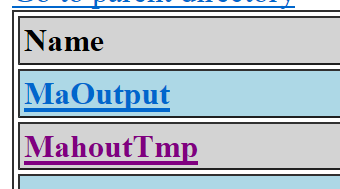
*bin/hadoop fs -rmr temp*

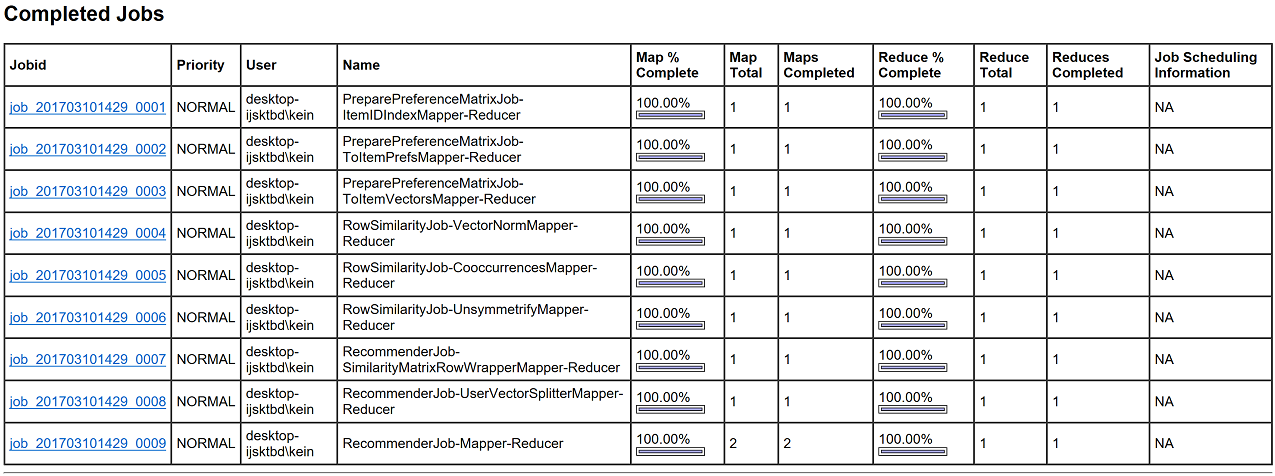
create the input file with users.txt and input.txt. run the Mahout recommender for 100K GroupLens.Finally, get the results of the user whose user id is 500:

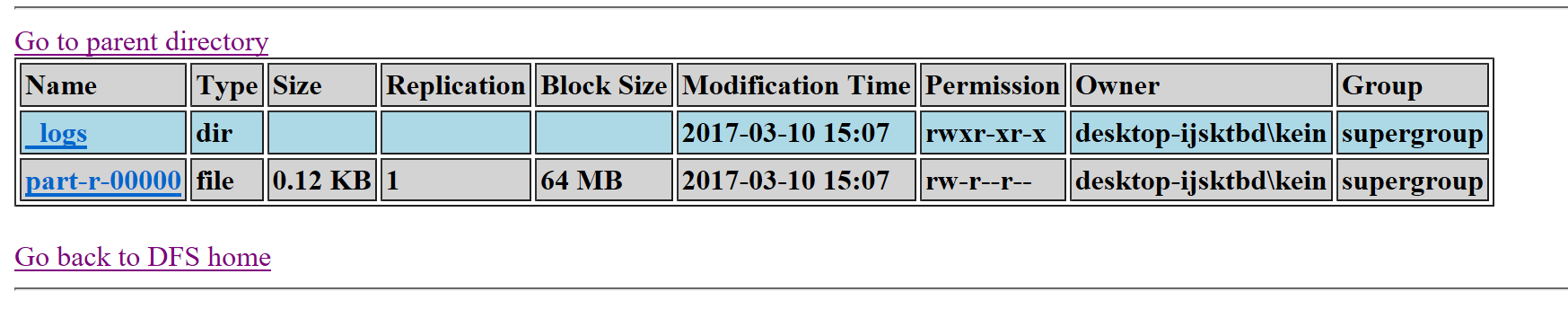
**Standalone mode:**

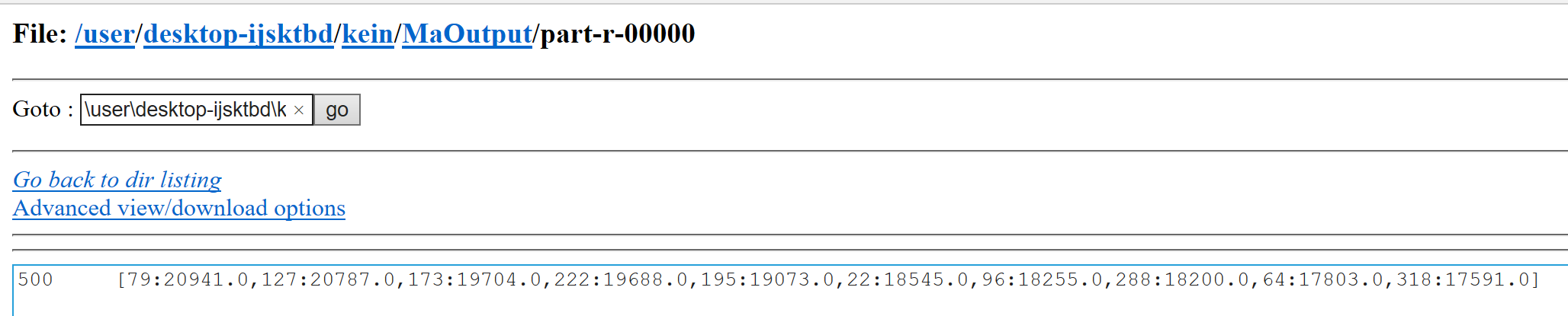


**HDFS distributed mode:**



****



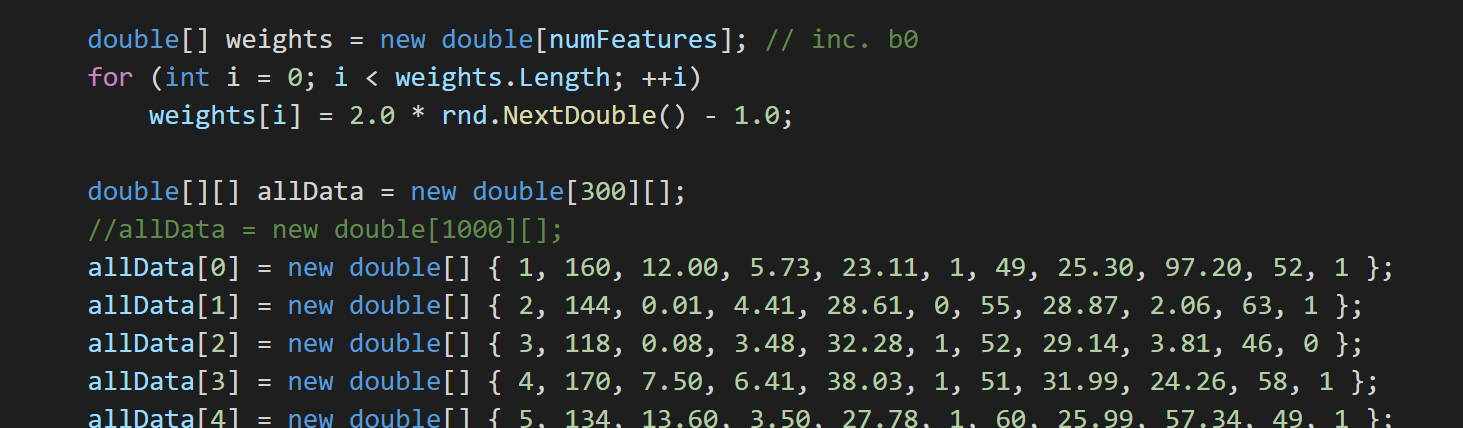


# Part 3:

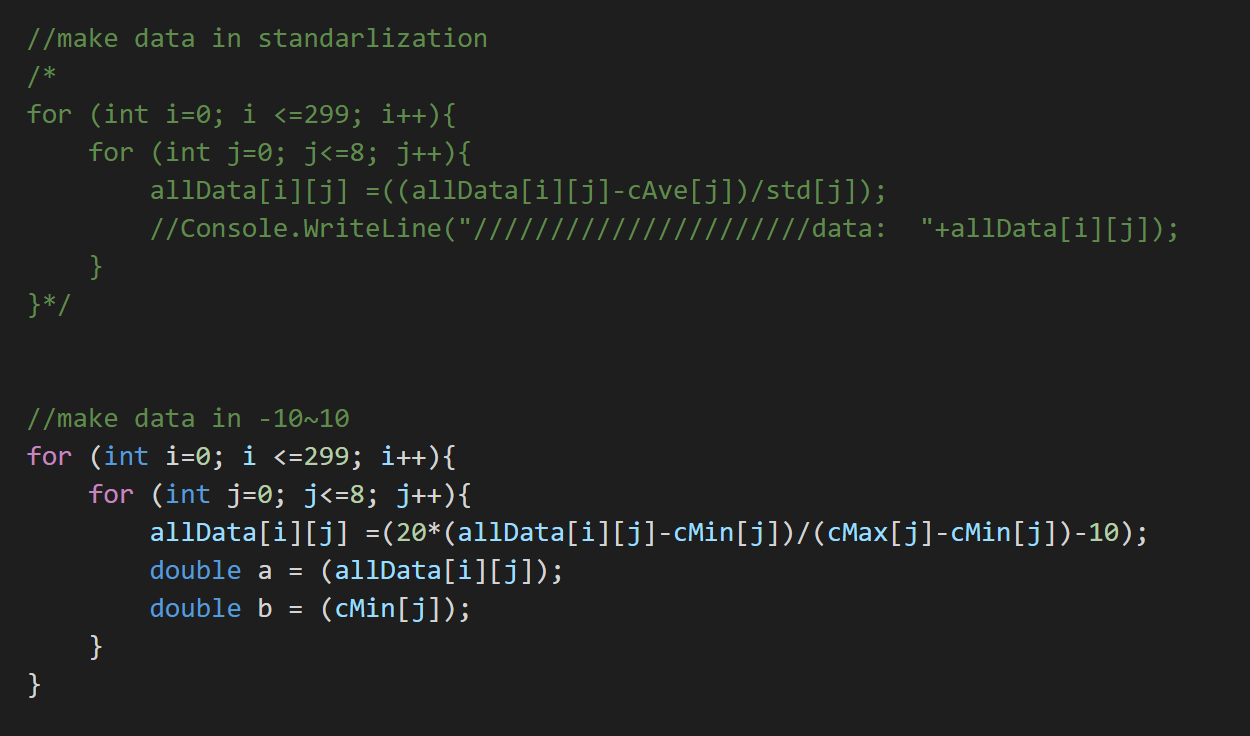
Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

In the code from professor, the dataset is automatic created and its range is from -10~10 which make the dataset more standard and easy to get the high accuracy. But from my dataset, there are lots of different value and its ranges for every column are different so it makes me to normalize and standardize my dataset to improve the accuracy.

Input dataset:



Standardize dataset and normalize the dataset to make the range all between -10 to 10:





Get the result and the accuracy:

