# EDI Calculator Documentations

Components(MapReduce jobs):

averageCalc: calculating the average of recent *n* years of data for a country, irrelevant datas(e.g. overall data for regions) would be thrown out during this stage.

totalAverage: calculating the average value of the whole dataset, output to a file

minmaxCalc: finding the min and max value for the datasets

singleNormalizer and twoWayNormalizer: using linear regression to calculate score based on the value. singleNormalizer is used min value, max value in the dataset, and the overall average of the dataset as references for linear regression; while the twoWayNormalizer would have a optimal point which gives a full score, left and right base point which give 0 in score; anything beyond the base point would result in negative value, and I also set cut off points on both left and right side of the optimal point which a flat maximum deduction would taken for values beyond the cut off points.

EDISynthesizer: assemble normalized scores together for all indicators to form final EDI score, countries with missing indicators would be thrown out in this stage.

Ranker: rank the data in ascending order

So the overall overflow would be first call averageCalc on all indicators' input files, then call totalAverage, minmaxCalc on the results of averageCalc to get min, max and average value; then use the results of all these three jobs to call normalizer(if using twoWayNomalizer, min, max, average values are not needed), after these steps are completed for all indicators, call EDISynthesizer to sum normalized scores for all indicators, then call Ranker to sort the result.

EDI equation and configurations:

Total GDP: weight 10%(10 points), considered recent 10 years of data, country with maximum value receive 10 points, country with minimum value receives 0; set average value receives 5 points, and used two segments linear fitting to calculate scores of the values below average and above average.

Total GNI: weight 5%, used 10 years of data, max gets 5, min gets 0, average gets 2.5.

GDP-growth: weight 5%, used 10 years of data, max gets 5, min gets -5 (there are countries GDP gets decreasing), set 0 as middle point, country gets 0 at 0.

Gdp-per-capita: weight 17.5% used 10 years of data, max gets 17.5, min gets 0, average gets 9
GNI-per-capital, weight 5, max gets 5, min gets 0, average gets 2.5

Inflation-consumer-price, weight 12.5%, used 10 years of data, set optimal point at 2, which receives 12.5 points, set left base point at 0, right base point at 3, country with value at base points getting 0 points, country with values beyond base points getting negative scores, set left cut off at -7, and right cut-off at 10, any countries beyond cut-offs receive get a flat -12.5.

Current account balance: weight 5, max gets 5, 0 or smaller gets 0, no hurt.

Total-reserve, weight 10, used 10 years, max gets 10, min gets 0, average gets 5

Industry-value-added: weight 3, max gets 3, min gets 0, didn't use average

Agriculture-value-added: weight 1, max gets 1, min gets 0,

Services-value-added: weight 6, max gets 6, min gets 0

FDI: weight 6, max gets 6, min gets 0

Trade-in-services weight 4, max gets 4, min gets 0

Gross-saving, weight 5, max gets 5, min gets 0

Ease of doing business, weights 5, min gets 5, max gets 0

## Run Instruction for EDI Calculator:

At the path containing the input file folder and the jar file, run the program with two parameters, first is the path to the folder containing all input files, second is path to the configuration file(do not need to worry about the output path, output will be automatically put in a directory called EDIoutput under the root directory of HDFS). Also, you need to put the file called "inrelevants.txt" (sorry for the typo, but it spelled like this) under the root directory of HDFS, it contains all of the regions need to be filtered out. If for some reason, the program failed in the middle, it does not need to be re-run from the scratch, you can just see if the failing step already created any output folder or incomplete output file yet, if yes, just delete

that output folder for the failing step(sorry maybe I should make the program delete the incomplete outputs if it fails), then correct whatever the possible is, then re-run the program, the program will be able to run start at the failed point, and do not need to run the jobs already finished. If you successfully finished one run successfully, later on you just happened that want to change some weights of the indicators, you can modify weights in the configuration file, then run the program normally with 1 additional argument ‘true’, the program will re-calculate the normalized value with the new weight, but not re-calculate the whole thing: average, min, max…they are never changed. If you want to add/drop indicators, but did not modify the weight of other indicators, u can modify the configuration file accordingly, run the program normally WITHOUT supplying that ‘true’ argument, then the program will not re-normalization all indicators(which is not necessary) but only do the calculation on the new indicators and re-assemble EDI values.