A6 - High Fidelity

"Is it wrong, wanting to be at home with your record collection? It's not like collecting records is like collecting stamps, or beermats, or antique thimbles. There's a whole world in here (…)"

— Nick Hornby, High Fidelity

Objectives

Learning the dataset and the basics of Spark

Dataset

We are using a dataset based on the metadata in the Million Song Dataset (https://labrosa.ee.columbia.edu/millionsong/). The data can be found here:

- Million Songs Subset Metadata (CSV (http://violet.ele.fit.cvut.cz/~kondziu/pdpmr/MillionSongSubset.tar.gz))
- Million Songs Full Dataset Metadata (CSV (http://violet.ele.fit.cvut.cz/~kondziu/pdpmr/MillionSongDataset.tar.gz))

Use the subset for development and testing. Use the full dataset for your final submission.

Description of the dataset can be found here (https://labrosa.ee.columbia.edu/millionsong/pages/example-track-description). Watch out for dirty data!

Functional requirements

Write a Spark program that retrieves:

- number of distinct songs, artists, and albums
- top 5 loudest songs
- top 5 longest songs
- top 5 fastest songs
- top 5 most familiar artists
- top 5 hottest songs
- top 5 hottest artists
- top 5 hottest genres (mean artists hotness in `artist_term`)

- top 5 most popular keys (must have confidence > 0.7)
- top 5 most prolific artists (include ex-equo items, if any)
- top 5 most common words in song titles (excluding articles, prepositions, conjunctions)

Evaluate your solution using your a local machine. **Optionally** evaluate your solution using AWS.

Write a report discussing the results and the performance results. Pay attention to how the data and operations impact performance.

Non-functional requirements

Assignment is performed in groups of 2 (assigned by TA). Authors are clearly marked on all deliverables.

Required files:

- `src/` (directory containing the sources of the implementation)
- `README.md` (Markdown file containing the description of the implementations)
- `Makefile`
- `report.Rmd` (The report as described in the previous section)
- `report.pdf` (A PDF rendering of `report.Rmd`)
- `input/` (A directory containing sample input files; please `gzip` all text files to save space)

Provide a Makefile with the following rules:

- `build` builds all the implementations
- `run` runs the variants
- `report` generates the reports
- `all` build, run, and report

Prepare the report in R Markdown and generate a PDF.

Extra credit

It is possible to get extra credit for the assignment by providing a complete Hadoop implementation of this assignment and comparing the performance of the two solutions.