Homework 3: Programming Component

USING MAPREDUCE TO ANALYZE THE MILLION SONG DATASET
VERSION 1.0

DUE DATE: Wednesday April 17th, 2019 @ 5:00 pm

OBJECTIVE

The objective of this assignment is to gain experience in developing MapReduce programs. As part of this assignment, you will be working with the Million Song dataset. You will be developing MapReduce programs that parse and process song analyses and metadata to answer questions about the songs.

You will be using Apache Hadoop (version 3.1.2) to implement this assignment. Instructions for accessing datasets and setting up Hadoop clusters are available on the course website.

This assignment may be modified to clarify any questions (and the version number incremented), but the crux of the assignment and the distribution of points will not change.

1 Cluster setup

As part of this assignment you are responsible for setting up your own Hadoop cluster with HDFS running on every node. We will be staging datasets on a *read-only* cluster. You should use your ***own*** cluster to write outputs produced by your MapReduce programs. MapReduce clients will be able to access namespaces of both clusters through Hadoop ViewFS federation. Your programs will process the staged datasets; data locality will be preserved by the MapReduce runtime.

2 Million Song Dataset

The dataset contains metadata and analysis results for one million songs provided by the Echo Nest. Records are stored in separate CSV files. The file name consists of the type of data and an index. For example, the first file of metadata is stored in metadata1.csv, and the first file of analysis data is stored in analysis1.csv. Each line in a file corresponds to a single song, with each field separated by commas. There are approximately 1 million records in the entire dataset at approximately 280 GB.

The table below summarizes the fields. Fields in the following table are appearing in the same order as in a record. The complete documentation including the data dictionary for the dataset is available at https://labrosa.ee.columbia.edu/millionsong/. Please note that we are using a reduced dataset. This dataset contains only a subset of the fields from the list of fields described in the data dictionary provided in the above link.

The datasets are available under the directories `/data/analysis` and `/data/metadata` in the shared HDFS.

2.1 Analysis File

Index	Field Name	Description
1	song_id	Unique string identifier
2	song_hotttnesss	Real value between 0-1 (0 means this song was not rated)
3	analysis_sample_rate	Positive integer
4	danceability	Real value between 0-1 (0 means this song was not rated)
5	duration	Positive real value representing seconds
6	end_of_fade_in	Positive real value representing seconds
7	energy	Real value between 0-1 (0 means this song was not rated)
8	key	Positive integer
9	key_confidence	Real value between 0-1 - confidence that key is correct
10	loudness	Real value – higher value Is louder
11	mode	Positive integer
12	mode_confidence	Real value between 0-1 - confidence that mode is correct
13	start_of_fade_out	Positive real value representing seconds
14	tempo	Positive real value
15	time_signature	Positive integer
16	time_signature_confidence	Real value between 0-1 - confidence that time_signature is correct
17	track_id	Unique string identifier
18	segments_start	Space-separated string of real values representing seconds
19	segments_confidence	Space-separated string of real values between 0-1 – confidence that each segment_start time is correct
20	segments_pitches	Flattened (in row-major order) space-separated string of real values - pitch of each segment
21	segments_timbre	Flattened (in row-major order) space-separated string of real values - timbre of each segment
22	segments_loudness_max	Space-separated string of real values representing maximum loudness of each segment
23	segments_loudness_max_time	Space-separated string of real values representing the time of maximum loudness in each segment
24	segments_loudness_start	Space-separated string of real values representing the loudness of the start of each segment
25	sections_start	Space-separated string of real values representing seconds
26	sections_confidence	Space-separated string of real values between 0-1 – confidence that each section_start time is correct
27	beats_start	Space-separated string of real values representing seconds
28	beats_confidence	Space-separated string of real values between 0-1 – confidence that each beats_start time is correct
29	bars_start	Space-separated string of real values representing seconds
30	bars_confidence	Space-separated string of real values between 0-1 – confidence that each bars_start time is correct
31	tatums_start	Space-separated string of real values representing

		seconds
32	tatums_confidence	Space-separated string of real values between 0-1 – confidence that each tatums_start time is correct

2.2 Metadata File

Index	Field Name	Description
1	artist_familiarity	Real value between 0-1 (0 means this artist was not rated)
2	artist_hotttnesss	Real value between 0-1 (0 means this artist was not rated)
3	artist_id	Unique string identifier
4	artist_latitude	Real value
5	artist_longitude	Real value
6	artist_location	String of variable format
7	artist_name	String, not guaranteed to be unique
8	song_id	Unique string identifier
9	title	String name of song
10	similar_artists	Space separated list of artist_ids
11	artist_terms	Space separated list of terms describing this artist
12	artist_terms_freq	Space separated list of real values 0-1 representing frequency of artist terms
13	artist_terms_weight	Space separated list of real values 0-1 representing weight of artist terms, sorted in descending order
14	year	Positive integer

3 Analysis of the Million Song Data Set

You should develop MapReduce programs that process the Million Song dataset to answer the following questions.

Q1.	Which artist has the most songs in the data set?
Q2.	Which artist's songs are the loudest on average?
Q3.	What is the song with the highest hotttnesss (popularity) score?
Q4.	Which artist has the highest total time spent fading in their songs?
Q5.	What is the longest song(s)? The shortest song(s)? The song(s) of median length?
Q6.	What are the 10 most energetic *and* danceable songs? List them in descending order.
Q7.	Create segment data for the average song. Include start time, pitch, timbre, max loudness, max loudness time, and start loudness.
Q8.	Which artist is the most generic? Which artist is the most unique?
Q9.	Imagine a song with a higher hotttnesss score than the song in your answer to Q3. List this song's tempo, time signature, danceability, duration, mode, energy, key, loudness, when it stops fading in, when it starts fading out, and which terms describe the artist who made it. Give both the song and the artist who made it unique names.
Q10.	Come up with an interesting question of your own to answer. This question should be more complex than Q7, Q8 or Q9. Answer it. For this component, think of yourself as the lead data scientist at a start-up firm. What would do with this dataset that is cool? You are allowed to: (1) combine your analysis with other datasets, (2) use other frameworks such as Mahout for performing your analyses, and/or (3) perform visual analytics. **Restrictions:** Note that there should be NO DISCUSSIONS about Q10 on Piazza or Canvas. Your analysis must be something that you have come up with on your own. Q10 is quite open-ended and you have a lot of freedom. That freedom comes with the responsibility that you manage your own problems and don't except someone else (be it the Professor, GTA, or your peers) to solve your problems for you. You have to iron out all problems that you are facing on your own.

4 Additional Requirements

Some data may be missing or improperly formatted. It is up to you to handle such cases in your program in the manner you consider appropriate.

The last four questions are deliberately vague and have no set answer. It is up to you to determine how best to answer them. Grading will be conducted by interview, and it is important that you are able to explain the method you used to get your answer and why you believe that method accurately answers the question asked.

Try to design your MapReduce jobs as elegantly as possible. This means minimizing the number of jobs and the amount of data transferred between each job. Minimizing the amount of data transferred between the mapper and reducer within each job is also important as it significantly impacts the amount of time the job will take to run. Submissions that feature many jobs or contain jobs that take an unreasonably long time to complete may be penalized.

5 Deductions

There will be a **16-point deduction** if any of the restrictions below are violated.

1. You should not implement this assignment as a stand-alone program.
2. You should not implement this assignment using anything other than Hadoop MapReduce. Implementing your own framework or using a 3rd party framework (that is not Hadoop) to implement this assignment is not allowed.

6 Grading

Homework 3 accounts for 20 points towards your final course grade. The programming component accounts for 80% of these points with the written element (to be posted later) accounting for the remaining 20%. This programming assignment will be graded for 16 points. The point distribution for this assignment is listed below.

Point Breakdown:

1 point:	Correctly configured Hadoop cluster
1 point each:	Correct answer for questions Q1-Q6
1 point:	Provided answer to question Q7
1 point:	Explained methodology for question Q7
1 point:	Provided answer to question Q8
1 point:	Explained methodology for question Q8
1 point:	Provided answer to question Q9
1 point:	Explained methodology for question Q9
1 point:	Created interesting question to answer for Q10
2 points:	Answered question Q10 and explained methodology

7 Milestones:

You have 4 weeks to complete this assignment. The weekly milestones below correspond to what you should be able to complete at the end of every week.

Milestone 1: You should have your HDFS/MapReduce cluster configured. You should be able to read data from your HDFS cluster into a MapReduce program and write data from a MapReduce program back to the cluster. You should also be able to read the Million Song Data Set from the shared HDFS cluster.

Milestone 2: Use MapReduce to answer Q1-Q6 and write the answers into your HDFS cluster in a easy-to-read manner.

Milestone 3: Use MapReduce to answer Q7, Q8, and Q9. Be able to explain how you got those answers and why you chose to get them in the way you did. You should also have brainstormed several ideas for Q10.

Milestone 4: Finalize your idea for Q10 and answer it.

8 What to Submit

Use the CS455 checkin program to submit a single .tar file that contains:

- All the Java files related to the assignment (please document your code)
- the build.gradle file you use to build your assignment
- A README.txt file containing a description of each file and any information you feel the GTA needs to grade your program.

The folder set aside for this assignment's submission using checkin is **HW3-PC**

Filename Convention: All classes should reside in a package called **`cs455.hadoop`**. The archive file should be named as <FirstName>-<LastName>-HW<x>-PC.tar. For example, if you are Cameron Doe then the tar file should be named Cameron-Doe-HW2-PC.tar.

9 Version Change History

This section will reflect the change history for the assignment. It will list the version number, the date it was released, and the changes that were made to the preceding version. Changes to the first public release are made to clarify the assignment; the spirit or the crux of the assignment will not change.

Version	Date	Comments
1.0	3/13/2019	First public release of the assignment.