Project Proposal: Dataset Analysis

1. **Standard Metadata: Title, Authors, Date**
   The Dataset I plan on working on is:

   Title: Million Song Dataset
   URL: https://labrosa.ee.columbia.edu/millionsong/

   The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The core of the dataset is the feature analysis and metadata for one million songs, provided by The Echo Nest. The dataset does not include any audio, only the derived features. It's a cluster of complementary datasets contributed by:

- SecondHandSongs dataset
- musiXmatch dataset
- Last.fm dataset
- Taste Profile subset
- thisismyjam-to-MSD mapping
- tagtraum genre annotations
- Top MAGD dataset

   It is a Collaborative project between The Echo Nest & LabROSA, supported in part by the NSF

   Authors: The original data was contributed by The Echo Nest, with Subsequent donations from SecondHandSongs.com, musiXmatch.com, and last.fm

   Date: This Dataset was released on the 8th of February 2011

2. Introduction: An overview of the project and its goals

Dataset Attributes:

280 GB of data
- 1, 000, 000 songs/files
- 44, 745 unique artists
- 7, 643 unique terms (Echo Nest tags)
- 2, 321 unique musicbrainz tags
- 43, 943 artists with at least one term
- 2, 201, 916 asymmetric similarity relationships
- 515, 576 dated tracks starting from 1922

The website also provides a MillionSongSubset file, which consists of 10,000 songs which is 1% of the entire dataset (1.8GB). I plan on working on this sample dataset first and then gradually extending the same functionalities on the larger dataset.
Also, this same dataset exists on aws.amazon.com in two sizes. AWS Public Datasets: Million Song Dataset, one of 5 GB in size (sample) and another of 500 GB (entire dataset)

Goal: My main goal for this project is to perform Scalable Data Analysis on this large MSD dataset (280 GB) and answer a few questions which I have mentioned below. I plan on using Amazon Web Services (AWS) where I will attach the dataset to an Amazon EC2 VM, followed by setting up an EBS disk instance and run the analysis on the cloud. I have never worked on AWS before and this will be my first project using cloud resources.

I have plans on running some Machine Learning model on this Dataset for prediction (preferably Linear Regression) but I may have a time crunch.

3. **Background**:

Background Information:

Data is stored using HDF5 format to efficiently handle the heterogeneous types of information.

The Attributes for every single song, along with the description of the field is mentioned below:

Sr No.	Field name	Type	Description	Link
1	analysis sample rate	float	sample rate of the audio used	url
2	artist 7digitalid	int	ID from 7digital.com or -1	url
3	artist familiarity	float	algorithmic estimation	url
4	artist hotttnesss	float	algorithmic estimation	url
5	artist id	string	Echo Nest ID	url
6	artist latitude	float	latitude	
7	artist location	string	location name	
8	artist longitude	float	longitude	
9	artist mbid	string	ID from musicbrainz.org	url
10	artist mbtags	array string	tags from musicbrainz.org	url
11	artist mbtags count	array int	tag counts for musicbrainz tags	url
12	artist name	string	artist name	url
13	artist playmeid	int	ID from playme.com, or -1	url
14	artist terms	array string	Echo Nest tags	url
15	artist terms freq	array float	Echo Nest tags freqs	url
16	artist terms weight	array float	Echo Nest tags weight	url
17	audio md5	string	audio hash code	
18	bars confidence	array float	confidence measure	url
19	bars start	array float	beginning of bars, usually on a beat	url
20	beats confidence	array float	confidence measure	url
21	beats start	array float	result of beat tracking	url
22	danceability	float	algorithmic estimation	
23	duration	float	in seconds	
24	end of fade in	float	seconds at the beginning of the song	url
25	energy	float	energy from listener point of view	
26	key	int	key the song is in	url
27	key confidence	float	confidence measure	url
28	loudness	float	overall loudness in dB	url
29	mode	int	major or minor	url

30	mode confidence	float	confidence measure	url
31	release	string	album name	
32	release 7digitalid	int	ID from 7digital.com or -1	url
33	sections confidence	array float	confidence measure	url
34	sections start	array float	largest grouping in a song, e.g. verse	url
35	segments confidence	array float	confidence measure	url
36	segments loudness max	array float	max dB value	url
37	segments loudness max time	array float	time of max dB value, i.e. end of attack	url
38	segments loudness max start	array float	dB value at onset	url
39	segments pitches	2D array float	chroma feature, one value per note	url
40	segments start	array float	musical events, ~ note onsets	url
41	segments timbre	2D array float	texture features (MFCC+PCA-like)	url
42	similar artists	array string	Echo Nest Artist IDs (sim. algo. unpublished)	url
43	song hotttnesss	float	algorithmic estimation	
44	song id	string	Echo Nest song ID	
45	start of fade out	float	time in sec	url
46	tatums confidence	array float	confidence measure	url
47	tatums start	array float	smallest rhythmic element	url
48	tempo	float	estimated tempo in BPM	url
49	time signature	int	estimate of number of beats per bar, e.g. 4	url
50	time signature confidence	float	confidence measure	url
51	title	string	song title	
52	track id	string	Echo Nest track ID	
53	track 7digitalid	int	ID from 7digital.com or -1	url
54	year	int	song release year from MusicBrainz or 0	

Related Papers:

1. **The Million Song Dataset**, T. Bertin-Mahieux, D. Ellis, B. Whitman and P. Lamere, *ISMIR '11* http://ismir2011.ismir.net/papers/OS6-1.pdf

2. **The Million Song Dataset Challenge**, B. McFee, T. Bertin-Mahieux, D. Ellis and G. Lanckriet, *AdMIRe* https://pdfs.semanticscholar.org/73d2/16afb813f53134a3e635c3dcbecb612cb23e.pdf

4. **Analysis & Design**:

A few questions and visualizations I plan on answering/showing using the dataset are as follows (Tentative List):

1. Which location do most of the artists come from?
➔Displaying artists using location based on latitude and longitude. I will use Bubble map/heat map to visualize this analysis

2. What Genre was the most popular as the years passed by?
➔Using artist hotness to calculate top artists for every year and then visualizing them by grouping them as per their Genre. I plan on using line graph for this analysis

3. What years had the highest number of album/song releases? Show top 10 years
➔Use Songs to search album names and output the album names corresponding to highest number of songs released that year OR show top 10 years with highest number of song releases

4. From the dataset, what are the average durations of songs, as years passed by?
➔ Using duration and year attribute, determine the average for every year and I tend to use line graph in displaying this analysis

5. **Scalability Challenges**:

   The main Scalability challenge will be Pre-Processing the Data. 280 GB of data is huge and will have to use special techniques for processing such data. Large data Processing on a local machine will take a lot of time to complete and hence will have to perform techniques such as using local CPU cores by task distribution. This will be faster than the previous approach. However, this would work on the sample datasets (1.8GB - %5 GB). However, managing 280GB could be challenging.

   Converting the given data which is in HDF5 format to CSV format could be a challenge because of the data size.

   Performing Data Cleaning will be another challenge because of error/missing values in the dataset. Cleaning large Data on a local machine may give an Out of memory error. Hence by dividing the data in small chunks and feeding them on different CPU cores could be faster.

6. **Implementation**:
   I will be using Wrappers in Python which use NumPy, PyTables for converting the HDF5 files to CSV format, matplotlib for Visualizing the data and PySpark (Faster Big Data real time processing).
   I haven't looked much into these technologies apart from the Python packages mentioned above, but I am keen to learn and work on Apache Hadoop MapReduce and Apache PIG which are more inclined to Big Data Processing.
   Amazon EC2 and EBS for storage and Processing for the entire dataset. However, I need to to overlook on the estimated cost for storing my dataset on AWS.

7. **Project Plan**:

   1. November 21st - 25th: Initial Understanding of the Dataset and Configuring the Dataset in AWS EC2 and creating EBS instance
   2. Nov 26th – Dec 1st: Converting from HDF5 to CSV followed by Data Cleaning/Wrangling
   3. Dec 2nd – Dec 8th: Complete all the analysis on the sample dataset (1.8 GB)
   4. Dec 9th – Dec 14th: Work on the scalability factor by trying to analyze the entire dataset
   5. Dec 15th: Integration and making Final Presentation and Report