Meta Musical Memes

Fall 2015 W205 Project Proposal

Megan Jasek, James King, Sean Underwood

Overview

The Meta Musical Memes project will create desired musical statistics for musicians and producers. These statistics will be derived from song attributes and can be used to understand which types of songs might be popular in the future or which types of songs will have a unique sound that might not appeal to everyone. The input to the project will be the Million Songs Database which is hosted on Amazon Web Services (AWS) and other music data from sources TBD. The output of the project will be a straightforward way to view useful musical statistics.

Project Goals

The goals of this project are as follows:

- Ingest the Million Song Database into a "big data" architecture.
- Incorporate one or two other music datasets into the architecture
- Extract, transform and load the Million Song Database (MSD) into a datastore where further analysis will be done.
- Analyze data from the MSD and generate useful statistics for the music industry.
- Answer questions that would be important to musicians, songwriters, producers and others in the music business.
- Get experience using AWS, Spark and other data science tools.

Data Details

The Million Song Dataset is a freely-available collection of audio features and metadata for one million contemporary popular music songs (tracks) provided by The Echo Nest (a music intelligence platform, see References below). The dataset does not include any audio, only the derived features. The purpose of this dataset is as follows:

- To encourage research on algorithms that scale to commercial sizes
- To provide a reference dataset for evaluating research
- As a shortcut alternative to creating a large dataset with APIs (e.g. The Echo Nest's)
- To help new researchers get started in the MIR field

That data is stored in HDF5 format which is a file format for storing and managing data that supports an unlimited variety of datatypes. There is one HDF5 file per song. Each song has approximately 50 attributes associated with it including: title, album name, artist name, key, loudness, pitch, timbre, tempo and time signature. Each song does not have every attribute populated. All of the songs from the database will be loaded and analyzed. The output will include some or all of the following items:

  - Music theory components of a song
  - Table showing characteristics of songs that make money in the music industry
  - Table showing characteristics of songs that don't make money in the music industry
  - Histogram of keys or tempo that successful songs are written in
  - Interesting graphical description of successful songs
  - What songs were popular in a particular season (like summer)
  - Table of Key/Time Signature combinations
  - "Hotness" prediction ("hotness" is a measure of song popularity) based on a general linear model of the MSD fields
  - Predictive model of a song making the Billboard Top 40 based on historical Billboard chart data (which would be scraped and merged)

Other potential data sources
These are potential other data sources that we could merge with the MSD to create some of the desired musical output.

- Kaggle data to support generating a profile of what music people who like a particular song or artist looks like.
- Scrape and merge Billboard chart data. Data is available starting in the 1950s.

Potential Issues

The following items have been identified as potential issues for the project

- HDF5 format. The MSD stores the song information in HDF5 format. Code will need to be written to convert the data into a more useable format. There are numerous informational sites on how to do this conversion, so this is not expected to be a big risk.
- Number of files. There is a large number of small files as each of the one million songs is stored as a file. This might be difficult for a big data architecture to deal with as they are usually designed to deal with very large files. To address this issue, when the HDF5 data conversion is done, the songs will be grouped into folders and imported in larger chunks.

Architecture

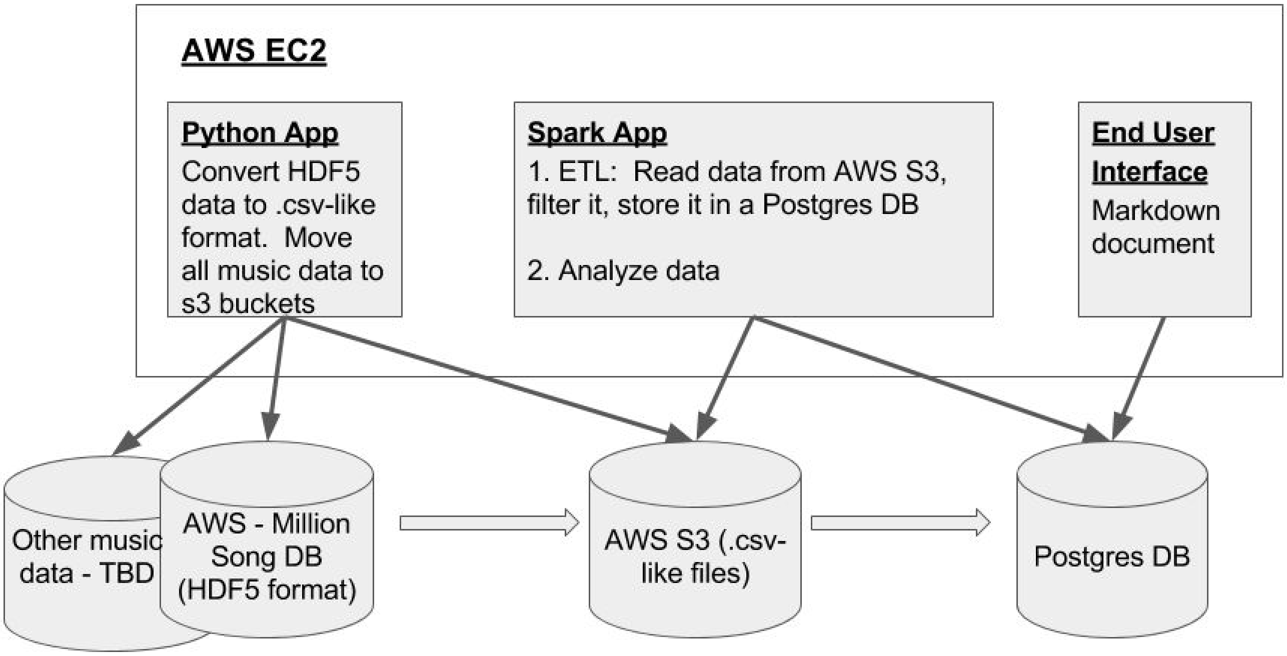
The architecture for this project will use a Netflix-like architecture in order to keep costs down. Specifically, AWS S3 will be used to store the data instead of something like HDFS. S3 is cheaper and the data is immutable so it's appropriate for S3. S3 is slightly slower, but it is worth the small slow down in performance for a lower cost. An AWS EC2 instance will be used as the server to drive the data acquisition, conversion and analysis. See architecture diagram. Data will flow through the system as follows.

1. MSD data is sourced on AWS in an EBS snapshot. The data is stored in HDF5 format. This data needs to be converted to a format that is more easily manipulated. A python program running from the AWS EC2 instance will convert the data from HDF5 format to a .csv-like format. The data will be batched up into groups of songs and stored in AWS S3 buckets. Data from other data sources will also be loaded into S3 buckets.
2. Once the data is in AWS S3, a pySpark program will read it and filter out only the data that is required for the end-user analysis and then load it into a relational database (Postgres) for further analysis. The Postgres database was chosen because it was available in the working environment and team

members were familiar with using it. Upon loading into the relational database, the data will be transformed into a relational schema that will facilitate analysis.

3. Once the data is in the Postgres DB, analysis will be done on the song attributes to generate useful song statistics.
4. Once the data is analyzed an aesthetic markdown document will be generated to report results to the end user. Other possibilities of outputs that could be created:
    a. A REST API which enables anyone to query the derived metadata
    b. Enable genre-specific or artist-specific recalculations
    c. If time permits, buttons will be added so the user can do their own analysis on the data.

Meta Musical Memes Architecture Diagram



AWS EC2
Python App
Convert HDF5 data to .csv-like format. Move all music data to s3 buckets
Spark App
1. ETL: Read data from AWS S3, filter it, store it in a Postgres DB
2. Analyze data
End User Interface
Markdown document
Other music data - TBD
AWS - Million Song DB (HDF5 format)
AWS S3 (.csv-like files)
Postgres DB

Acknowledgements

In using the million song dataset, we acknowledge the following paper [pdf] [bib]:

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

References

https://www.hdfgroup.org/pubs/papers/Big_HDF_FAQs.pdf
http://top40charts.net/
https://www.ee.columbia.edu/~dpwe/pubs/Ellis07-timbrechroma.pdf
http://labrosa.ee.columbia.edu/millionsong/faq
http://the.echonest.com/