**Analyzing the Million Song Dataset**

CIS 602: Project Presentation  
Instructor: Dr. David Koop

Kevin D’Cruz   
ID: 01656193  
[kdcruz@umassd.edu](mailto:kdcruz@umassd.edu)  
  
**Overview:** An overview of the Project Report is as follows:

* Goal
* Dataset Information
  + Sample Dataset
  + Complete Dataset
* Technology Used
* Analysis (performed on Sample subset and Complete Dataset)
* Scalability Challenges
* Learning Outcomes
* Conclusion
* References

**Goal:** My goal of this project was to analyze the Million Song Dataset, available publicly, and answer questions/create visualizations relating to the dataset while focusing on the scalability factor.

**Dataset Information:**  
  
Title: Million Song Dataset

URL: <https://labrosa.ee.columbia.edu/millionsong/>

This Dataset is made available by the Echo Nest & LabROSA (Columbia University). It consists of audio derived features & metadata for a million contemporary popular music tracks. It consists of 54 attributes, and is available in two versions. The sample subset of the dataset of 10,000 Records (1.8 GB), consists of 10,000 records. It is available here: <https://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset>.   
  
The Complete Dataset made available is of 280 GB and consists of 1 million records. Each record consists of 1 song, it’s metadata and features. The dataset is in an HDF5 format, which is the Hierarchical Data Format. This Format is that of a Tree Structure, where data was stored in Nested folders. This page shows an example Track Description: <https://labrosa.ee.columbia.edu/millionsong/pages/example-track-description>.

This Dataset is also available on AWS Public Datasets where the size is of 500GB.

**Technology Used:**

* Jupyter Notebook
* Python
  + Pandas
  + Numpy/Pytables
  + Pyspark
* Amazon Web Services (AWS)

**Analysis:**  
  
The Initial Analysis I carried out, was on the sample subset of 1.8 GB, of 10,000 songs. Since this data was available in HDF5 format(.h5), the first step was to convert it to a csv format. I used the following GitHub link(script) for conversion of the files to csv format: <https://github.com/amgreenstreet/Million-Song-Dataset-HDF5-to-CSV>. This conversion also needed another file called hdf5\_getters.py which was available from the dataset’s website. Upon conversion, I first used Pandas (only on the subset), followed by Pyspark (subset+complete) to answer the following questions:

Question1: Which location most of the artists came from?

A close up of a map

Description generated with high confidence  
 Map provided from Google Maps API  
  
I plotted the Artist Locations for only 500 artists, since they were overlapping. Most of the artists came from the United States and Europe as seen in the map

Question2: What years had the highest number of album/song releases?

![A screenshot of a map

Description generated with high confidence]()  
Line Graph showing highest number of song releases as per years. The graph peaked over the years 2006-07 for both Subset and complete data

![A screenshot of a cell phone

Description generated with high confidence]()

Question 3: What artists had the highest mean Duration of songs (in minutes)?  
![A screenshot of a cell phone

Description generated with high confidence]()Most songs had a mean duration of 4-5 minutes for both subset & complete data

**Challenges/Problems Faced:**  
The first challenge I faced was handling this large amount of data in HDF5 format. I initially planned working on loading this data directly in a Dataframe format without conversion to csv. I was unable to go about this and hence, converted it to a csv format, from the code referenced [3]. Another challenge faced was configuring AWS and loading the complete data (500 GB) for conversion.  
  
**Scalability Challenges:**  
The focus of this project was handling the scalability focus. The first scalability challenge was: Large data for Crunching. Handling 500 GB of data on a local computer would have scalability issues, in comparison with that of an AWS Cluster. I started this conversion on Amazon EMR by creating an EC2 Instance of m4.large. This provided me 2 Cores and 8 GB of RAM. I used Mobax for accessing the cluster from my local machine. Upon loading the data using this instance, it was taking a lot of time for conversion. The full data would roughly take about 12+ hours for conversion. Hence, I upgraded the setting by choosing a higher EC2 instance. This was the c4.2xlarge which provided 8 Cores and 15GB RAM. Another setting done was that I manually tweaked the loading of files by creating 8 folders(p1-p8), which replicated the 8 cores. I assigned 3-4 folders to each replicated core and begin loading the data, thus accelerating the process 8 times faster. This file conversion took the roughly 2 hours for obtaining the 1 million records in a CSV format. Following 3 screenshots show this process:

![A screenshot of a cell phone

Description generated with very high confidence]()

![A screenshot of a computer

Description generated with very high confidence]()   
A screenshot of a computer

Description generated with very high confidence  
  
For focusing on scalability, I used Pyspark on Jupyter Notebook which by default enables parallel processing.

**Learning Outcomes:**  
This project helped me learn Pyspark and more importantly working and configuring AWS. I had never worked on such technologies before and hence got to learn a lot from it.   
  
**Conclusion:**  
I converted the sample subset of data on my local computer and the complete data on AWS EMR. Analysis was carried out on Jupyter Notebook using Pandas for the subset data and then Pyspark for both subset and complete data. Pandas when tried on the complete dataset was slightly slower than Pyspark and hence, I didn’t emphasize on it much.

**References:**  
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>

3. Conversion (HDF5 to CSV) reference   
<https://github.com/amgreenstreet/Million-Song-Dataset-HDF5-to-CSV>   
  
4. Pyspark/AWS/Pandas Documentation