EE 660 Final Project Proposal Fall 2015

# Predictive Analysis for Music using Real Noisy Dataset

**Your name and email address**

**N/A (Team Project)**

**If a team project, state how many team members (including yourself) and list them**

3 member team
Nishant Nath | nnath@usc.edu
Arnav Mendiratta | amendira@usc.edu
Ahmet Ozbek | ozbek@usc.edu

***Dataset Used**

Million Song Database (referred as MSD here-on).
Compiled by : Lab ROSA (Columbia University – NSF grant)
Compiled using : The Echo Nest API (+ several independent datasets)
Source : http://labrosa.ee.columbia.edu/millionsong/
Availability : InfoChimps , AWS Public Data Set
Size : 273GB compressed (hdf5 format) [130kB hdf5 file ~ 2.2 MB csv file]
Files : 1,000,000 (covering 1 million songs, 1 song = 1 file)
Other Statistics : About 515,576 dated music tracks, tracks from 44,745 unique artists
Data range : ~100 years [1922-2007] (~ 10 Music decades)

***Software packages, language, and code**

All code development will be tracked & managed through Git.

**PHASE 1 : Pre-training & Preliminary Tests (0.3%, 3000 songs, size variable)**
**Prototyping : MATLAB (+Statistics & Machine Learning Toolbox)**
**Languages: Python v3.x (Anaconda distribution)**
**Softwares Packages: h5py, Numpy, SciPy, Scikit Learn, matplotlib, pytables, etc.**

**PHASE 2 : Test using subset (1% , 10000 songs / 1.8 GB compressed)**
**Languages: Python v3.x (Anaconda distribution)**
**Softwares Packages: h5py, Numpy, SciPy, Scikit Learn, matplotlib, pytables, etc.**

***PHASE 3 : Test using Whole Data Set (100%, 1 million songs / 273 GB compressed)**
**Languages: Python v3.x**
**Softwares Packages: h5py, Numpy, SciPy, Scikit Learn, matplotlib, pytables, etc.**
**Environment : AWS (EC2,EMR,EBS,RDS), Google Big Query**

***Clear statement of the problem and/or goals.**

PHASE 1 & PHASE 2 are primary objectives and time permitting; we would undertake the PHASE 3.

**PHASE 1 : Pre-training & Preliminary Tests (0.3%, 3000 songs, size variable)**
**PHASE 2 : Test using subset (1% , 10000 songs / 1.8 GB compressed)**
***PHASE 3 : Test using Whole Data Set (100%, 1 million songs / 280 GB compressed)**

We would like to start off with a case-study in point. Soundcloud – Inarguably the worlds largest music collection on cloud claims to have 40 million registered users with 175 million unique monthly listeners. An independent survey puts the percentage of amateur music producers at nearly 90%. With the music industry growing in exponential rate, the motivation for the topic comes from two key areas – MIR (Music Information Retrieval) and Personalized Music Recommendation Systems. There is a constant need for automatic tagging of music mainly for its metadata but also for user-predictions and success factors. As our topic suggests, we would be performing predictive analysis on music encompassing both regression as well as classification problems.

As with any good machine learning application, the question to be answered determines the veracity of the application. For our problem, we have come up with 5 basic questions which give us deep insights regarding music. They are: (*C = classification problem, *R = Regression Problem)

1. ***Automatic Genre Tagging (*C)***
   Given a song & its features (TBD) – predict which Genre it belongs to?
2. ***Automatic Artist Prediction (*C)***
   Given a song without its acoustical features – predict the artist of the song?
3. ***Year/Music-Decade of Release (*C)***
   Given a song & its features (TBD) – predict which music decade/year does it belong to?
4. ***Predict the popularity of the song for discotheques (*R)***
   Given a song & its features (TBD) – predict how danceable the music turns out to be?
5. ***Predict the popularity of the song among listeners (*R)***
   Given a song & its features (TBD) – predict how hot/viral the music turns out to be?

Apart from these basic questions, we will attempt to answer more such challenging & insightful questions (time permitting).

## *A plan of preprocessing and feature extraction

As mentioned in the topic, we would be working with a real-life dataset which by its inherent nature is highly noisy & has missing information. Noise comes into picture given the dataset's dependency on user-ratings & tags on websites such as musicbrainz, 7digital, lastfm and many other websites. This unreliable source also means missing data or unintelligible data which needs pre-processing before it is usable.

Major areas where we foresee significant amount of pre-processing:

- ✓ Reading, Storing & Using hdf5 files in computationally efficient way. (avg. 93% compression)
- ✓ Unintelligible data (Unicode-UTF8 encoding errors + human induced errors like improper tagging).
- ✓ Redundant information (mostly metadata but lots of ids & tags which cannot be used apart from identification purposes).
- ✓ Features reduction/normalization (features are highly random and inconsistent in sizes – so needs to be pre-processed for easier comparison & usage).

Significant effort would also be devoted into modeling the hypothesis regarding which features are important and which are not. Being an iterative process, the hypothesis may change slightly between each iteration of tests.

## *A plan of your approach

As discussed earlier, the work is comprised of 3 PHASES. Each phase will deal with all the objectives (questions to be answered). Each phase will have a SCRUM-styled cyclic process of:

- Develop Hypothesis
- Implement Algorithm
- Test Algorithm (Compute error & other metrics)
- Cross Validate & Document Results

We will be using the data randomized in the following proportion in each stage: (data = 100%)
60% training
10% tuning
30% testing

**Rough time-line for the project: (tentative dates only, may change)**
Wk 0 [11/04 – 11/08] : Learning the tools, Exploring Dataset, Brainstorming & Prelimnary Tests
Wk 1 [11/09 – 11/15] : STAGE 1 – Algorithms tried & implemented, results obtained & compared
Wk 2 [11/16 – 11/22] : STAGE 2 – Verification of Algorithm Results from STAGE 1 on 1% subset
Wk 3 [11/23 – 11/29] : Cross Validation + Algorithm improvement + New algorithms to be tried
Wk 4 [11/30 – 12/07] : STAGE 3 – Implementation on AWS & Final tuning

As a final test and to give our learning algorithms more credibility, we would use tracks that are NOT in the dataset, (these tracks would be taken from DRM-free sources). The features would be extracted partially through The Echo Nest API to provide a real-life simulation of new music.

Coming down to the details of the initial plan of approach, we would start with the classification problem of Genre Tagging. We are going to use a combination of artist tags, artists terms, their frequency of occurrence, their weights along with some temporal and acoustic features like beats/bars/segments and loudness to classify into 10 broad genres (classical, metal, hiphop, dance, jazz, folk, soul, rock, pop, blues) which by eyeballing some data, we found to be most representative. Iteratively we are going to optimize the features used and induce prior information like the music decades most prominent for a particular genre to improve the overall results (source : Google Music Research Big Picture).

We are going to break down the artist classification into a subtask after Genre Tagging and using Music Decade information. Very rarely artists play music from more than 1 genres, having this assumption we can reduce the possible labels for artists from 50k+ to few hundreds which is a more feasible task. We would perform this classification based on features like predominant keys for an artists, the general energy in a song, information of beats, bars and tempo which to a seasoned musician are characteristics of particular artist.

Year or music decade prediction is an ambitious challenge and is based on the basic assumption that music undergoes a paradigm shift in its general tendency every decade (more or less 10-15 years)

A similar process will be followed for the other 2 objectives and need more brain storming before deciding on the approach.

***A description of any prior or parallel work of yours**

Although the current project is new and no work has been done by either of the members directly relating to the topic, our background provides leverage towards the work on the said topic.

Nishant has worked on Music Information Retrieval earlier and was involved in an early-stage start-up in the field and brings on board knowledge about the domain as well as the industry.
Arnav has worked extensively on music in production & post-production side and brings on board great insights about music industry.
Ahmet has a deep passion for music and brings on board domain knowledge of music & multimedia.

***If yours is a team project, roughly describe how work will be divided**

To ensure fair distribution of responsibilities & learning scope, we have decided to pick one objective (question to be answered) at a time and divide the work based on algorithm being implemented. With each new objective, the algorithm each member was working on will be rotated which provides an opportunity of working with all algorithms as well as all objectives.

**Other Comments**

Most prominent algorithms under consideration for implementing are:

- Linear Regression
- Logistic Regression
- Ridge Regression
- LOESS
- LASSO
- Bayes Classifier (Naïve, Gaussian, Multinomial)
- kNN / k-Means /k-Medians
- Decision Trees
- Ensemble learning (bagging/boosting)

The PHASE 3 objectives are dependent on progress of project over the next few weeks. More objectives & goals may be added in due course.