**Overview**

* We have studied the use of content- based approaches to form playlists from a given seed song. Our major approach would be to follow audio similarity measure. This measure compares songs according to the novelty of their frequency spectrum and the below mentioned audio parameters. In this approach, we investigate extensions to simply choosing the N closest songs to a seed as a playlist. The playlist will be keep on updating as the user will like/ dislike the currently played song in the mobile application and playlist will be formed by trajectories through the distance space using user relevance feedback.

**Backend/ Algorithm**

* We imagine a graph of all songs in our database. Each song is a node and links between songs describe how closely the songs are related. The simplest graph uses our distance measure for the link strength. A playlist can be formed using this graph by choosing the shortest path of length N emanating from the seed song. However, this prior work assumes that relevant attributes have already been determined for each song rather than extracting them from the audio as in our case.
* The million songs 2.3GB data set from Echonest, contains all the audio features. We will be using libraries such as hdf5, numexpr, cython, blosc, tables and numy. Our data set is in HDF5 format, we have to use h5py python library to process this data. The h5py package is a Python interface to the HDF5 binary data format. It lets you store huge amounts of numerical data, and easily manipulate that data from NumPy. For example, you can slice into multi-terabyte datasets stored on disk, as if they were real NumPy arrays. Thousands of datasets can be stored in a single file, categorized and tagged however you want.
* MFCC ( Mel-frequency cepstrum coefficient) of our data has short spectrum information and there are lot of segments in the data which contains timbre information, so we will consider timbre as the major comparison feature to match songs and we can also use other audio features like tempo, pitch and loudness as well. The following features will be taken into consideration for comparison purpose and to find the minimum distance to represent the closely matched songs in graph representation:
* ‣ **key**: the estimated overall key of a track. The key identifies the tonic triad, the chord, major or minor, which represents the final point of rest of a piece.
* **tempo**: the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
* **loudness**: the overall loudness of a track in decibels (dB). Loudness values in the Analyzer are averaged across an entire track and are useful for comparing relative loudness of segments and tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude).
* **timbre** is the quality of a musical note or sound that distinguishes different types of musical instruments, or voices. It is a complex notion also referred to as sound color, texture, or tone quality, and is derived from the shape of a segment’s spectro-temporal surface, independently of pitch and loudness. The Echo Nest Analyzer’s timbre feature is a vector that includes 12 unbounded values roughly centered around 0. Those values are high level abstractions of the spectral surface, ordered by degree of importance. For completeness however, the first dimension represents the average loudness of the segment; second emphasizes brightness; third is more closely correlated to the flatness of a sound; fourth to sounds with a stronger attack; etc. See an image below representing the 12 basis functions (i.e. template segments). The actual timbre of the segment is best described as a linear combination of these 12 basis functions weighted by the coefficient values: timbre = c1 x b1 + c2 x b2 + ... + c12 x b12, where c1 to c12 represent the 12 coefficients and b1 to b12 the 12 basis functions as displayed below. Timbre vectors are best used in comparison with each other.

**Frontend Aspects**

* We could not get 10,000 songs legally, that’s why we are using data set of audio features only. Our data set will return the unique id of the song which we will be playing and we can play the desired songs by accessing it using its UID in the format (Spotify:track:**6kNuZA5SeXupyMiO0Ka9Az)**  on Spotify via Spotify API controls. Similarly, we can access the playlists and albums using the URI for the playlists and albums.
* In addition to this, we are going to access our data set locally, as it will be quite expensive to host 2.3GB of data on any server.

Just go through the company website, what are the products ?

Just go through the profile.

1. **What is my weakness ?**

Well, I have a weakness related to my work. I get carried away, sometimes I loose track of the work which I am suppose to do by going into extra depth of the task. Suppose, someone has given me a oiece of code or a function, I always tend to start studying or scrutinize the stuff into deep. But I feel that has helped me a lot in long run or a bigger picture as we should make our basics/ fundamentals clear.

1. **Tell me about a time when you showed initiative in a work situation.**

This has happened many times in my previous experience. Suppose when we used to have finalize the design or any functionalities in any application, I have always been curious to give inputs. I don’t remember the project, but yes we were looking for a CSS design to follow, then I had proposed bootsrap framework to my team lead, he wasn’t quite impressed with my proposal but later on I had convinced him and that framework has been followed now in every project.

**What would you do if a client asked you for a service that you were unable to provide?**

I would try to convince him with an alternative solution to his requirement, if that solution doesn’t get fit in, then I will ask for some time to do any research or trouble shooting on that problem and meanwhile I ll discuss the issue with my team lead or any supervisor and then will respond to him. Basically, I don’t want to refuse at the first place, by doing this he will loose some interest in us.

**What would you do if you were approaching a deadline that you knew you**

**were unable to meet?**

Suppose we have a deadline for an application which we are unable to meet. I will start prioritizing the functionalities, will consider the front end and important functionalities to be the top most priority and will try to complete those tasks. In case of go-live or any deployment, the functions which wont be working/ or pending, I will ask to delay the deadline, keeping in mind that the important functionalities have been covered already.

**What did you like least about your last job?**

Well, the last firm in which I had worked was a start up and there wasn’t that big hierarchy just like we have in CMM level 5 companies, for example you have a PM, TL and other designated positions. So it used to be a one man show, others were not given equal opportunities to propose any new idea or technology on which they can work in future. But later on, as I gained experience in that firm, I got some opportunities later to propose new ideas.

Monotonous technologies, same technologies.

**Tell me about a time you had a disagreement with a co-worker**

Disagreement with the technology being used.