BRETT ASHLEY MCCOUBREY

ML MIX LAB

13 December 2018 | Galvanize DSI

TL;DR

Creating great mixes is an art, and musicians have been using sampling to improve their songs for years. Finding songs that mix well together can be difficult and time consuming, and can yield unsatisfying results. Using one seed song, DJ's can use ML Mix Lab to find a list of songs that they can use together in a mix. I have used collaborative filtering to determine what makes songs mix well together, according to a database of production level mixing and sampling. Technologies used include Python, PostgreSQL, PySpark, Selenium, AWS, and Psycopg2.

Business Understanding

Searching for music takes time, and understanding the intricacies of successful mixing takes years of experience. This tool will cut down on this time and help DJ's create better music.The idea and structure of this project can be expanded to using audio files, finding songs that can be mixed into or sampled in songs that are still in the ideation and production process.

Data Understanding and Preparation

Steps for Data Collection from WhoSampled:

1. Start at at the main page for a particular genre, such as https://www.whosampled.com/genre/Soul-Funk-Disco/.
2. Collect most influential and most popular artists in the genre.
3. Scrape top artists' songs and the songs they are connected to
4. Add all new artists from collected songs to the database
5. Continue steps 3-4, prioritizing based on artists' song frequency.

Data Preparation Methods:

1. Use song URL as unique identifier in postgres database.
2. Resolve inconsistency with URL parsing with psycopg2.
3. Read database in with pandas and convert to Spark dataframe for use with ALS.
4. Search Spotify API for artists that are in database.
5. Match spotify data based on fuzzy match of song name.

Concept

Equating song sampling to creative mixing allows us to use this dataset to train a model to recommend songs to mix together. We can think of each song as a 'user' and 'item' in a recommendation system, with each song being a 'user' when it samples another 'item' song.

Model

Create an implicit recommendation system, using each song in the dataset as both a user and item, assigning a '1' to connected songs and '0' to all others. To the right is a user-item matrix for three songs where 'Song 1' and 'Song 3' are connected. As a next step, I will incorporate ratings based on song features, such as upweighting a connection if two songs have a bpm within +/- 4. For example, the connection 'Song 1' and 'Song 4' gets assigned a '2' because they have the same BPM.

Song ID	Song 1	Song 2	Song 3	Song 4
Song 1	1	0	1	2
Song 2	0	1	0	0
Song 3	1	0	1	0
Song 4	2	0	0	1

Evaluation

Baseline models have been compared using rmse (root mean squared error) to start tuning hyperparameters such as matrix rank. Further tuning is in progress, comparing professional DJ's rankings versus the model using a hypothesis test such as the Wilcoxon signed-rank test.

Deployment

Visit http://mlmixlab.com/ to start mixing! Visit http://github.com/brettashley/ml-mix-lab for more info.