Spotify 1: Creating the Perfect Playlist

Content-Based Generation of Spotify Playlists

Laura Ware, Omar Abboud, Sonu Mehta
Agenda

Overview
- Goal and motivation of the project

Datasets
- Overview of datasets used in modelling

Lit Review
- Literature Review of the previous work

Models
- Description of models used

Demo
- Demo of playlist generation engine

Conclusions
- Future work and take away from the project.
Introduction to Spotify

Spotify is a music, podcast, and video streaming service.

“Spotify brings you the right music for every moment - on computers, mobiles, tablets, home entertainment systems, cars, gaming consoles, and more. Just search for music you love, or let Spotify play you something great.”

- Over 40 million subscribers, 100 million active users
- 30 million songs and 2 billion playlists
Spotify Playlists

- A **playlist** is a collection of tracks that can be compiled by individual users or by Spotify itself.

- Spotify-curated playlists are followed by millions of Spotify users. These playlists are created by a combination of algorithmic and human-driven processes.

- Spotify playlists are both personalized (e.g., Discover Weekly) as well as intended for general browsing based on genres, moods, or current events.
Project Goal

1. Explore methods for **predicting** the **success** of a playlist using *acoustic* & *non-acoustic* features of tracks present in the playlist.

2. Use these predictive models to develop novel processes for **curating successful playlists**.
Motivation

Q1. What makes a **playlist successful**?

Q2. Can we predict the **success** of a playlist using **acoustic** features of tracks present in the playlist?

Q3. Does the **sequencing** of tracks contribute to a playlist’s success?
Motivation

Q4. Can less popular but acoustically similar songs be sequenced to make a successful playlist?

Prior Methods
- Genre-based Recommendation
- User-based Collaborative Filtering
- Metadata-based Recommendation

Our Approach
- Acoustic Similarity Filtering
Data Sources

Spotify API

Raw Audio
Data sources: Spotify API

1. Playlist Data
   - Total tracks
   - Sequence of tracks
   - No. of followers

2. Tracks Data
   - Acoustic features (danceability, loudness, energy, liveness, etc.),
   - Duration
   - Popularity
Data sources: Raw-audio

- 30 seconds raw-audio samples from Spotify API
- 34 features extracted at each timestamp (1 sec)
  - Energy, Entropy of energy
  - MFCCs (Mel-frequency Cepstrum Coefficients) - 13
  - Chroma Coefficients - 12
Literature Review


Relationship between track popularity and playlist followers

![Graph showing the relationship between track popularity and playlist followers. The graph plots the mean popularity score of a playlist's tracks against the number of followers, with a trendline indicating a positive correlation. The R-Squared value is 0.47, and the P-value is <0.001.]
Modelling Approach:

Part 1: **Playlist popularity prediction**

Part 2: **Track popularity prediction**

Part 3: **Acoustic Similarity**

Part 4: **Playlist Generation Engine**
Baseline Model

**Response variable:** Quantile of popularity (score of 1-5)

<table>
<thead>
<tr>
<th>Predictor Set</th>
<th>Misclassification Rate (RFC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Popularity Only</td>
<td>65%</td>
</tr>
<tr>
<td>Spotify Acoustic Features Only</td>
<td>33%</td>
</tr>
<tr>
<td>Mean Popularity + Acoustic Features</td>
<td>22%</td>
</tr>
</tbody>
</table>

**Conclusion:** Acoustic features carry predictive significance
Playlist Popularity Prediction

Predictors:
- Acoustic features of tracks (danceability, loudness, energy, liveness, etc.), duration, popularity of tracks
- Divide features by sequencing (median across first 25%, second 25%,...)
- 78 total predictors

Response: No. of followers divided into 5 bins, with 5 being the most popular

Model: Random forest classifier

Accuracy: >95% training accuracy, 50% testing accuracy
Track Popularity Prediction

**Predictors:** Acoustic features of tracks (MFCCs, Chroma coefficients, Energy at each second)

**Response:** Popularity divided into 6 bins.

**Model:** Random forest classifier

**Performance:**
- 76% cross-validation accuracy
- F-1 score = 1 for the most popular class
Methods to determine Acoustic Similarity

1. **Earth Mover’s Distance:**
   
   Using acoustic features extracted from raw-audio.

2. **Manhattan Distance:**
   
   Using audio features from the Spotify API.
Earth Mover’s Distance

1. Obtain Spectral signature

   - Divide audio into frames
   - Convert each frame to a spectral representation
   - Cluster frames

   Signature for song

2. Calculate Acoustic similarity

   - KL-Divergence based distance used to calculate distance between 2 clusters
   - Linear programming approach to minimize cost (flow multiplied with KL divergence distance)

Modeling
Vantage point approach to optimize search

![Vantage point approach to optimize search](image-url)
Playlist Generation Algorithm

Input: seed song

Candidate tracks compiled from:
1. Top tracks from related artists
2. Acoustically similar tracks from related artists
3. Acoustically similar tracks from vantage point database

Simulated annealing in parallel starting with 4 random sample of 30 tracks

Return the playlist that optimizes the no. of followers

Overview  Data Sets  Lit Review  Modeling  Demo  Conclusions
Simulated Annealing

Cost in this model: weighted average of top class probabilities.

At each step: **swap out a track** and run prediction.
Results

Simulated Annealing increases optimization performance significantly

```
Class Probability

Quantile of Predicted # of Followers

Similar tracks - Random Sampling
Similar tracks - Simulated Annealing
```
Results

**Weighting of Top Tracks vs. Performance:**
Using a higher percentage of top tracks improves performance but decreases the acoustic similarity of the resulting playlist.

![Graph showing Simulated Annealing Energy (loss) vs. Proportion of Candidate Songs that are Top Tracks](image)
Conclusions

1. Playlists can be optimized for popularity using acoustic & non-acoustic features
2. Simulated annealing is an effective method to generate combinations of songs that maximize the predicted number of followers
3. Comparing pools of potential songs to include in a playlist reveals that non-popular songs can be effectively incorporated into successful playlists
Challenges faced

1. Overfitting: our final random forest classifier had a low classification accuracy on the testing set relative to the training set
   a. Tuning the model parameters affected results only slightly
   b. Potential confounding that can be resolved with more metadata

2. Small sample size: to date, Spotify has only self-curated about 2,000 playlists.
Challenges faced

3. Skewed dataset: playlists rarely correspond to high-popularity classes
Future Work

1. Reduce the effects of model overfitting

2. Collect more metadata

3. Further raw audio analysis & feature engineering
Thank you!