Music Recommendation Algorithm for Spotify Dataset 1921-2020

Purdue SigAI Spring 2021

Daniel Park, Trung Bui, Maximilian

Table of content

Introduction

Data

EDA

Feature selection

Model Building

Model evaluation

Primary**:**

* - id (Id of track generated by Spotify)

Numerical**:**

* - acousticness (Ranges from 0 to 1)
* - danceability (Ranges from 0 to 1)
  + describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
* - energy (Ranges from 0 to 1)
  + represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
* - duration\_ms (Integer typically ranging from 200k to 300k)
* - instrumentalness (Ranges from 0 to 1)
  + whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
* - valence (Ranges from 0 to 1)
  + the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
* - popularity (Ranges from 0 to 100)
  + calculated from the popularity of the album’s individual tracks.
* - tempo (Float typically ranging from 50 to 150)
* - liveness (Ranges from 0 to 1)
  + Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
* - loudness (Float typically ranging from -60 to 0)
* - speechiness (Ranges from 0 to 1)
  + presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
* - year (Ranges from 1921 to 2020)

Dummy**:**

* - mode (0 = Minor, 1 = Major)
* - explicit (0 = No explicit content, 1 = Explicit content)

Categorical**:**

* - key (All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on…)
* - artists (List of artists mentioned)
* - release\_date (Date of release mostly in yyyy-mm-dd format, however precision of date may vary)
* - name (Name of the song)

[Recommendation algorithm examples](https://www.bluepiit.com/blog/classifying-recommender-systems/)

[Extensive thesis on music recommendation algorithms](https://hpac.cs.umu.se/teaching/sem-mus-17/Final-slides/Madathil.pdf)

1. Collaborative filtering (KNN)
   1. PRO: completely independent of any machine-readable representation of the objects being recommended and work well for complex objects where variations in taste are responsible for much of the variation in preferences
   2. CON: Relies on previous feedback data

based on the assumption that people who agreed in the past will agree in the future and that they will like similar kind of objects as they liked in the past

cold-start problem

Diagram

Description automatically generated

1. **Content-based (almost all ML)**
   1. PRO: no need for big dataset / expensive server for training
   2. CON:
2. Hybrid models and deep learning
   * Combines content-based and collaborative filtering together
   * Non-linear, not prone to oversimplification
   * Weighted, switching, mixed hybrid

b. CON: need extensive hyperparameter tuning

Different ways to get user input

* Simple list of songs listened
* Playlists, favorites

<https://researchportal.port.ac.uk/portal/files/14416496/User_Based_Hybrid_Algorithms_for_Music_Recommendation_Systems_Thesis_Murtadha_Al_Maliki.pdf>

<https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639>

<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>

<https://www.amazon.science/the-history-of-amazons-recommendation-algorithm>

<https://www.kaggle.com/arthurcerveira/spotify-recomendation-engine>