

Spotify Recommendation System

Animesh Sharma (as3592), Yaniv Bronshtein(yb262), Wanying Mo(wm318), Venkata Krishnam Raju Dalta(vkd20), Vipul Gharde(vig4), Aditya Maheshwari(am2971), Toshitt Ahuja(ta498), Fan Shen(fs470)

Motive:

With the rise of streaming platforms, the amount of music available to consumers at their fingertips or using their favorite AI(Google, Alexa, Siri) has skyrocketed. At first glance, users seemingly benefit from this plethora of available music. The opposite is true or was valid without the help of robust, state-of-the-art recommender systems.

Recommender systems thus face a powerful dilemma: on the one hand, they must observe user preferences and cater albums or individual songs to those preferences. On the other hand, they must be careful to introduce a degree of novelty so that struggling artists can have exposure. In our final project, we aim to create a helpful recommendation system that hopefully addresses this dilemma.

Dataset description:

We joined data from Spotify from dataset 1 containing genres and audio features and dataset 2, which included user_ids, to build a robust recommendation system. The two datasets alone did not provide enough observations and/or interesting predictors. The datasets are joined on a key containing the combined artist and track name that was lower-cased and combined into one word.

As of now, the genre is the target variable for our Machine Learning Models. We plan on investigating the contributions of the following features: popularity, acoustics, danceability, duration_ms, energy, instrumentals, key, liveness, loudness, mode, speechiness, tempo, time_signature, and valence.

Proposed Model Algorithm:

Our model will learn the genre of music the user likes to listen to over a period of time. The model will then create a rating scale, determining the level of interest a particular user has in that genre of music. Once a scale has been generated, the model will start recommending playlists or songs based on the rating for the genres the user likes. An example: The model will initially set the equal probability of playing music from any genre. Suppose a user likes to listen to hip hop, rock, and electronic music, with the probability of them playing those genres of music, given a set of all music, being 0.22, 0.18, and 0.14, respectively. The model would learn this probability trend for music selection by the user over a period of time. The model will keep adjusting the probabilities for genres till it reaches a point where there is a minuscular change. It will then start recommending the songs/playlist of a genre, the frequencies of which are based on the probabilities it has learned. That means it will suggest hip-hop songs/playlists more frequently than electronic music.

If time permits, we plan on trying other algorithms, including neural networks.