

Spotify Recommendation Engine Homework

Sreevatsa Nukala, Kirti Magam, Devyanshi Chandra

DS 43000

Process & Description:

With the given data from Spotify, we chose to obtain a random sample of 1000 songs and additionally add the songs of the artist that someone is basing their recommendation request off of. By choosing a random sample of data, we could ensure a wide range of songs that are not limited to specific genres or artists. This also means that with random sampling, the recommendation system does not become specific to one particular artist someone is searching based off of.

Within Neo4j, our graph model consisted of the nodes which were referred to by the songs names. We selected the most relevant song qualities to include in the attributes of the node because the other factors were already considered in the similarity calculation between the songs in the sample. The properties we kept for each song node were the unique song index, artist, album name, track name, popularity, and genre. Relationships between nodes were labeled in terms of being similar to another song and were labeled as “similar to”.

The recommendation algorithm we implemented was focused on finding songs similar to the one’s the user submitted, but less popular so that a user was likely receiving recommendations for songs and artists they haven’t heard of yet. We aimed to have the user discover lesser known artists within the same genres and subgenres they were already interested in. The first step in this algorithm was calculating the similarity scores between all the songs within the random sample. We used features such as danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, and tempo to calculate the cosine similarity. We then only kept the relationships of songs that had a similarity score greater than 0.8 so that we would ensure that songs we would end up recommending were mostly similar to the one the user submitted. After loading the nodes and relationships in the Neo4j, the last step in the

recommendation was filtering for the five songs with the highest similarity scores to the inputted songs that were less popular or equal in popularity.