# **Data Extraction: Spotify Datasets**

## **Description and Relevance to Project**

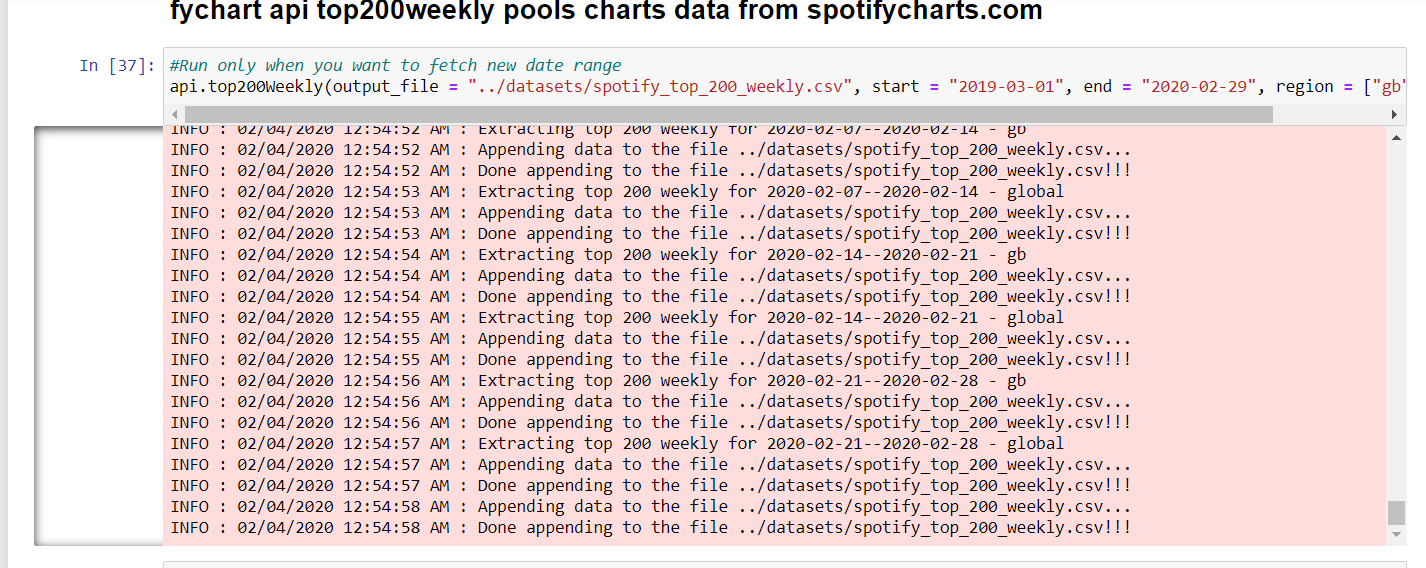
The spotify chart is available as downloadable CSV file at <https://spotifycharts.com/regional>. It comprises of top music tracks which can be filtered by region (global and country) and period (weekly and daily) starting from January 2017 to date. The top weekly chart is very fundamental to our scenario because it rates tracks by number of streams by listeners weekly. In terms of granularity of data, weekly is preferred because the data can be easily re-dimension into monthly. Regarding popularity, we adopted the stream count as a measure of track popularity which aggregates to artist popularity. Spotify Streams are counted for Artists when a song is streamed for 30 seconds or more.

We chose the spotifycharts because according to musically.com, as at February 2020, spotify accounts for more than 124m premium subscribers more than double the 2nd placed Apple music with 60m subscribers. Spotify also confirmed on their website that their chart figures are formula-driven and protected against artificial distortion. Our analysis and subsequent recommendation rely heavily on the charts as provided by spotify.

## **Extraction Process**

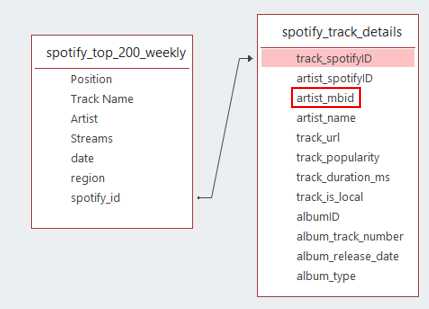
The extraction process was automated using fychart - a full fledge python library for extracting charts from spotifycharts.com. Top-200 weekly tracks were extracted between March 2019 and February 2020.

*Figure 1.2.1: Spotify Charts Extraction*



The keywords for the search are the Start date (‘2019-03-01’) and End date (‘2020-02-29’). For the regions, keywords include ‘global’ and Great Britain (‘gb’). Lastly, for period, the keyword is ‘weekly’. In all, we fetched 20,800 records (10,400 each for Great Britain and Global charts respectively). The spotify\_top\_200\_weekly dataset does not come with artist music\_brainz id (mbid) a unique identifier that can be used to merge different music/artist data sources. Since we will be joining this dataset with the songkick dataset via the mbid, it becomes important we explore a new library that can map this. The mbspotify, a python library that aims to provide mapping between [MusicBrainz Identifiers](https://musicbrainz.org/doc/MusicBrainz_Identifier) and [Spotify URIs](https://developer.spotify.com/web-api/user-guide/#spotify-uris-and-ids), in conjunction with the Spotify API were employed to generate the artist\_mbid in the spotify\_track\_details table as illustrated in 1.2 below.

*Figure 1.2.2: Entity Relationship between the Spotify Datasets*



As shown in figure 1.2 above, the spotify\_id (track id) from the spotify\_top\_200\_weekly dataset has a one-to-many relationship with the track\_spotifyID in the spotify\_track\_details dataset, which by design has an attribute artist mbid that can be linked to the artists mbid in our songkick event and venue data. The script for extracting the two datasets can be found in the source\_code folder of the project artefact (spotipy.ipynb).

References

<https://musically.com/2020/02/19/spotify-apple-how-many-users-big-music-streaming-services/>

<https://artists.spotify.com/faq/stats>

<https://github.com/kelvingakuo/fycharts>

<https://github.com/metabrainz/mbspotify>

<https://developer.spotify.com/documentation/web-api/>