**Getting the data**

In this project our team had two data sources: spotifycharts.com and official spotify API. Spotifycharts.com provided track\_position, track\_name, track\_id of Weekly Spotify Top 200 tracks from 2017 to September 2019. Spotify API provides several audio features for each track such as: instrumentalness, tempo, key, duration, streams, danceability etc. which can be pulled from Spotify API with track\_id.

Data was clean since all the data from the sources were well structured and there were no missing values.

Our final data consists of weekly lists from 2017 to 2019. Essential columns in DataFrame are: position (1-200), track\_name, week, streams and all the audio features. Some of values were continuous such as tempo, duration, streams and some were categorical such as key. This was perfect starting point to start our predictions and visualizations.

**Instructions for getting the data**

Spotifycharts.com has unofficial Top200 lists of Spotify. Python library called “fycharts” was a great and easy-to-use tool for pulling the data.

***\_spotifycharts.com\_***

**Install fycharts**

> pip install fycharts

**Download data**

from fycharts import SpotifyCharts

api = SpotifyCharts.SpotifyCharts()

api.top200Weekly(output\_file = “name of the file”, start=“YYYY-MM-DD”,end=”YYYY-MM-DD”)

More details about pycharts library:

[**https://pypi.org/project/fycharts/**](https://pypi.org/project/fycharts/)

***\_Spotify API\_***

In order to use spotify web api a user must have Spotify Premium account. Premium account gives an access to developer’s “client\_id” and “secret\_id” and “username” which are required to log in to API.

Spotfy API has several libraries and we used spotipy which is a library for python.

**Install spotipy:**

pip install spotipy

**Connecting to Api**

***try:***

***token = util.prompt\_for\_user\_token(username, scope, client\_id=client\_id, secret\_id = secret\_id, redirect\_uri = redirect\_uri)***

***except:***

***os.remove(f”.cache-{username}”)***

***token = util.promp\_for\_user\_token(username,scope)***

Scope:

* defines a need of user while operating in API. Different options for scope can be found from Spotify’s developers site.

More details about Spotify API: <https://developer.spotify.com/documentation/web-api/>

Audio features:

* duration\_ms:
  + continuous value
  + duration of track in milliseconds
* key:
  + Categorical value: 0-12 defines all keys.
  + C = 1, Db = 1, D=2… ...A=10, Bb=11, B=12.
  + no key = 0
  + Estimation of overall key
* mode:
  + categorical value
  + major = 0 or minor = 1
* time\_signature
  + beats per bar
* acousticness
  + a confidence measure from 0.0 to 1.0
  + 1.0 = a high confidence that a track has acoustic elements
* danceability
  + a confidence measure from 0.0 to 1.0
  + Describes how danceable a track is
* energy
  + a confidence measure from 0.0 to 1.0
  + Describes how energetic a track is
  + If there’s high energy level, the song is likely to be fast, noisy and loud.
* instrumentalness
  + a confidence measure from 0.0 to 1.0
  + the greater the value is the less it has vocals
* liveness
  + a confidence measure from 0.0 to 1.0
  + Describes if a track is a live or studio performance
* loudness
  + a confidence measure from -60dB to 0dB
  + The overall loudness of a track in decibels
* speechiness
  + a confidence measure from 0.0 to 1.0
  + Speechiness detects the presence of spoken words in a track.
  + For example audiobooks are more likely to have a higher value
* valence
  + a confidence measure from 0.0 to 1.0
  + Describes a positivity of the song.
  + 1.0 = positive, 0.0 = negative
* tempo
  + Continuous value
  + Beats per minute
* id
  + track\_id in Spotify
* uri
  + track\_uri in Spotify
* track\_href
  + a link to web API that provides all details of track
* analysis\_url
  + a http url to access full audio analysis of a track
* type
  + object type: “audio\_features”