Creating a Socially-Powered Song Recommender Using the Spotify API

Capstone Presentation | 06.09.2020

Hannah Reed | Data Science Immersive

The problem: can I create a recommender using Spotify's API?

What is Spotify® ?

- Music streaming service based in Sweden
- Founded in 2006, launched 2008, went public in 2018
- Known for accuracy of personal recommendations

Why Spotify?

- Robust API for developers
- Spotipy library to interact with API using Python
- One of the largest market shares of music streaming in the U.S.

Why this problem?

Generally: for many, music discovery remains a social activity, yet it has lost its personal touch in the digital age

With Spotify: Social activity and interaction is currently an untapped method for user engagement. Many users love the results of Spotify’s advanced algorithms, but they wish there were integrated ways to interact with friends or users with similar tastes.

Ultimate Goal

Playlists that are sort of like “Discover Weekly” (add definition/description), but for discovering music you might like from friends’ libraries - could be one-way discovery or two-way that’s sourced from both libraries (or “preferred songs/public tastes” playlists specifically designated for use by this feature

Use Cases

When could it be helpful to have a playlist personalized to two or more users, instead of one?

- Finding new music you might like
- Sparking conversation with friends
- Getting to know the tastes of someone new
- Shared playlist for a car ride

Challenges

For its various playlist generators, Spotify relies on multiple types of information. According to a Medium article by Sophia Ciocca, Spotify employs three main types of recommendation models:

“1. **Collaborative Filtering** models … which analyze both your behavior and others’ behaviors.

2. **Natural Language Processing** (NLP) models, which analyze text.

3. **Audio models**, which analyze the raw audio tracks themselves.”

I was really interested in Spotify’s machine learning-driven audio analysis, so I started by exploring how this analysis played out for different genres

Audio Features: Basic

Tempo: overall estimated tempo of a track in beats per minute (BPM)

Loudness: overall loudness of a track in decibels (dB)

Length: length of track, in seconds

Audio Features: Spotify-Generated

Liveness: Detects the presence of an audience in the recording

Acousticness: whether the track is acoustic.

Speechiness: detects the presence of spoken words in a track.

Energy: represents a perceptual measure of intensity and activity

Danceability: how suitable a track is for dancing

Valence: the musical positiveness conveyed by a track

Popularity: a value between 0 and 100, with 100 being the most popular

Spotify API + Spotipy

- Reading my own library
- Reading others' public playlists
- Combining data from different endpoints (ex. Song info from library, then audio features, then genre information from artist pages)

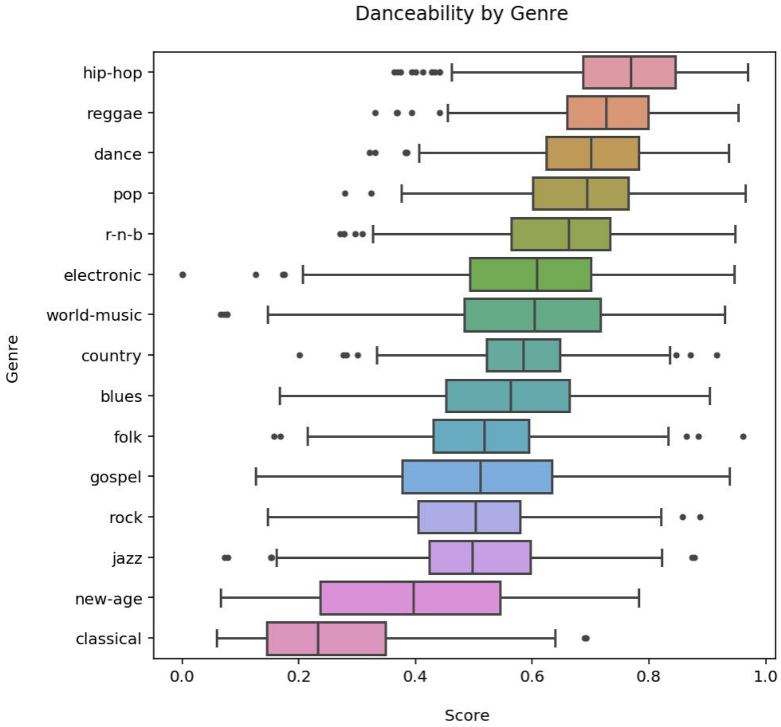
First problem attempted - can genre be predicted based on audio features?

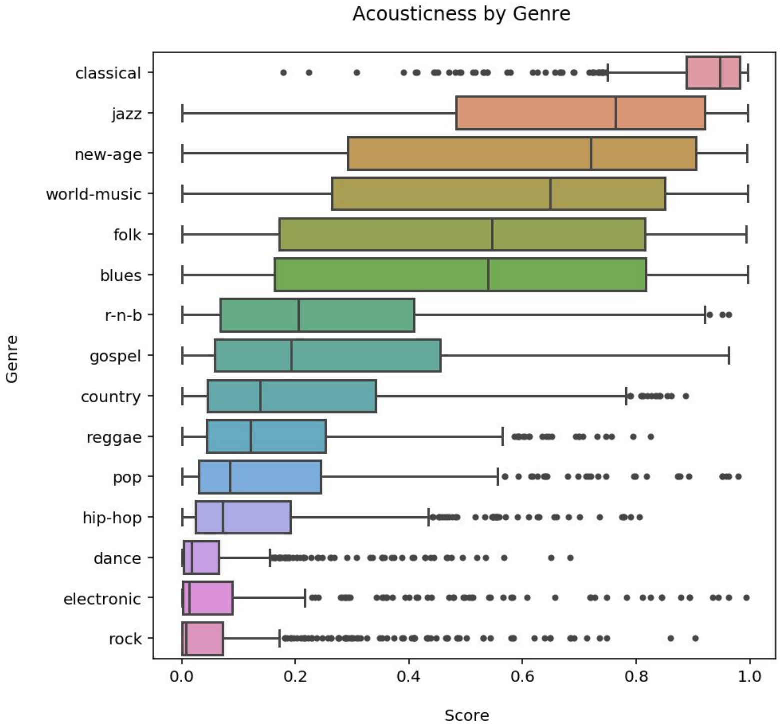
- Can genres be predicted from audio features?
- Could a more nuanced understanding of genres contribute to a more sophisticated recommender?

Challenges for Genre Analysis

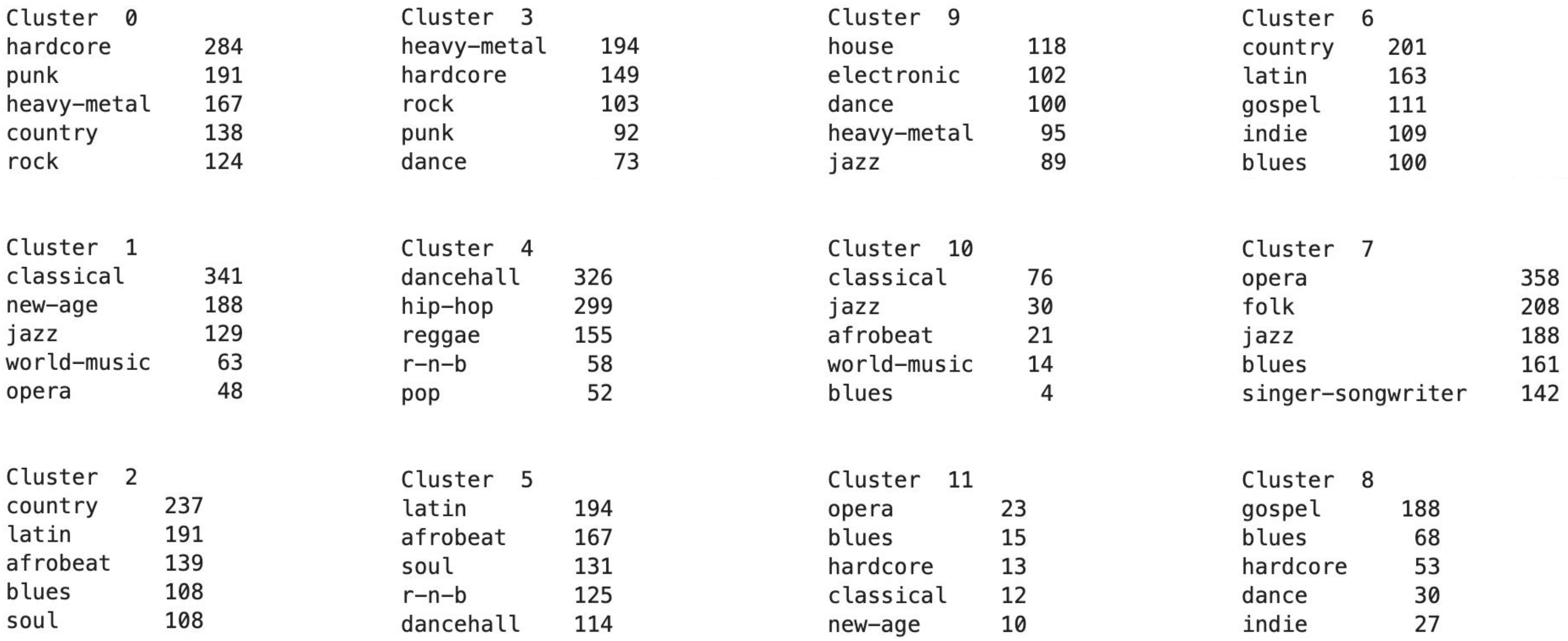
- Spotify doesn’t list genres for tracks, only for artists
- Artists aren’t limited to one genre - when analyzing my personal library, one song had 26 genres assigned to it
- Spotify allows for a LOT of sub-genres - “The Sound of Everything” by “The Sounds of Spotify” has one song for each genre currently tracked - 4291 genres!
- Decided to simplify to higher-level genres, but that presented its own problem of which ones to choose

EDA: Audio Features, by Genre





Clustering on Audio Features



```
Cluster  0
hardcore       284
punk           191
heavy-metal    167
country        138
rock           124
```

```
Cluster  1
classical      341
new-age        188
jazz           129
world-music     63
opera           48
```

```
Cluster  2
country     237
latin       191
afrobeat    139
blues       108
soul        108
```

```
Cluster  3
heavy-metal    194
hardcore       149
rock           103
punk            92
dance           73
```

```
Cluster  4
dancehall    326
hip-hop      299
reggae       155
r-n-b         58
pop           52
```

```
Cluster  5
latin        194
afrobeat     167
soul         131
r-n-b        125
dancehall    114
```

```
Cluster  9
house          118
electronic     102
dance          100
heavy-metal     95
jazz            89
```

```
Cluster  10
classical      76
jazz           30
afrobeat       21
world-music    14
blues           4
```

```
Cluster  11
opera        23
blues        15
hardcore     13
classical    12
new-age      10
```

```
Cluster  6
country    201
latin      163
gospel     111
indie      109
blues      100
```

```
Cluster  7
opera                 358
folk                  208
jazz                  188
blues                 161
singer-songwriter     142
```

```
Cluster  8
gospel      188
blues        68
hardcore     53
dance        30
indie        27
```

Predictive Models & Switching Gears

- I used various models to try to predict genre based on audio features, but they were all pretty terrible.
- I'd like to return to this problem and try other methods, but in the meantime I returned to my original question: can I use those audio features to build a recommender?

Example

Used audio features of tracks in my library and tracks in my friend's library to create a cosine similarity matrix, then generated a playlist based on these values.

To the Jupyter Notebook we go!

Possible Next Steps:

- Working out the details in basic song:song recommender mentioned above
- Creating generalized code that other people can explore
- Further exploring Spotify API in order to be able to allow users to authorize the app so I can read their libraries and create playlists
- Create a basic web dashboard to facilitate the above and possibly allow for users to see analysis of their own libraries?
- Further explore existing apps developed for Spotify
- Add user personalization endpoints to analysis