An Open Exploration of Movie Soundtracks Using IMDb and Spotify

by Xuechun Wang, Yao Huang,
Yuhui Tang, Yan Liu

Oct. 07, 2018

1. Data Science Problem

Soundtrack is one of the key parts in the movie, which effectively conveys the theme and emotion of the whole storyline. Good soundtracks could add popularity to a movie while celebrated movies might correspondingly bring attention to its soundtracks. In addition, the soundtrack itself is an individual piece of art that could be described by multiple music characteristics such as tempo and loudness. While in previous studies, researchers have investigated how the soundtracks would affect audience's emotion (Hoeckner et al.) and the potential relationship between the search volume of movie soundtracks and the movie revenue (Xu & Goonawardene), there are limited researches which study the quantitative features of movie soundtracks. In this project, we will investigate the relationship between popularity of our targeted films and characteristics of their soundtracks as to see how they affect each other. Besides, we will investigate the music attributes of the soundtracks and try to find some trends in the highly rated movies within different genres.

2. Dataset Collection Procedure

1) IMDb Data Scraping using HTML

IMDb is "the world's most popular and authoritative source for movie... content"[1]. For the IMDb dataset, we use the advanced title search provided by the IMDb website[2] and set up our filters as follows: "Title Type: Feature Film", "User Rating: 7.0 to 10.0", "Number of Votes: minimum 20000", "Title Data: Soundtracks". For the "Display Options", we choose "Detailed" mode, show "250 per page" and sort the results by "Popularity Ascending".

From the search results, we scraped titles, years, runtimes, genres, ratings, votes and the grosses for all 2122 movies, which consist of our first part of dataset.

Then we intended to use movie titles to query the corresponding soundtracks and their music attributes using the powerful Spotify API. Yet, it turned out to be troublesome. While Spotify API could accurately return the soundtrack(s) for most movies, for some of them, it failed. We detected several characteristics of failed movie names by trial and error and decided to pre-clean our movie titles to facilitate our query in Spotify.

The cleaning procedures are listed in *Table 1: Undesired Patterns in Movie Titles and Clean Procedures*. After cleaning, all the punctuations and special characters were formatted with space or normal letters in English. This method improved our query results. The total number of failed movie names decreases almost one thirds, from 290 to 202; the total number of collected tracks increases 4.8%, from 25098 to 26292.

1 Source: https://www.imdb.com/

2 https://www.imdb.com/search/title

Table 1: Undesired Patterns in Movie Titles and Clean Procedures

Undesired Patterns	Before Cleaning	After Cleaning	Examples
Scrape Error	"&"	"and"	From "Pride & Prejudice" to "Pride and Prejudice"
Special Letters with accent	"ô", "é", ...	"o", "e",...	From "Léon: The Professional" to "Leon: The Professional"
Titles including Roman Numbers	"Episode VIII", "Part II"	" " (space)	From "The Godfather: Part II" to "The Godfather: "
Special Characters	"!, ', ?, ,, *, /,), :,], [, ., (, -"	" " (space)	From "Qu'est-ce qu'on a fait au Bon Dieu?" to "Qu est ce qu on a fait au Bon Dieu"

2) Spotify Data Scraping Using API

Spotify is a digital music service that gives its user access to millions of songs. We use movies titles collected in IMDb datasets and Spotify API to scrape album data using Spotify search.

In *getTracklist.py*: we used the movie title in *cleanMVData.csv* as a searching criterion to find the corresponding soundtrack albums. The search will return multiple results in a *json* file. We used the topmost related result and collected the track lists and the corresponding track id of each track in the album. We combined the track name and id with the movie information collected in the last step to create an aggregate data frame and saved it as *tracklist.csv*.

In *getFeaturelist.py*: we used the track ids in *tracklist.csv* to scrape feature data of each soundtrack. Our collected information includes acousticness index, danceability index, duration of the soundtrack, energy level, instrumentalness index, key signature, liveness index, loudness measure, mode, speechiness index, tempo, time signature, valence level, and popularity. Then we saved the data in *feature_list.csv* file.

Finally, we integrated all the data sets into the *full_dataset.csv*.

3) Dataset and Variable Descriptions

We have two datasets scraped from IMDb and Spotify separately. In the IMDb dataset (cleanMVData.csv), we have movie genre, movie revenue, movie rate, runtime, movie title, number of people vote for the rate and year of release from IMDb website. We conducted web scraping on the most popular feature films with soundtracks. The rate of each movie is between 7.0 and 10.0 and has at least 20,000 user votes. We will use the rate to evaluate the popularity of

the movie. In terms of the Spotify dataset, we have detailed information on the attributes of each soundtrack as well as the popularity of these music individually.

4) IMDb Dataset

Movie Genre (string): Type of the movie.

Gross (float): Movie revenue of the movie, in millions of dollars.

Movie Rate (float): The rate of the movie by viewers' voting, with a range between 0 to 10. A larger number means higher rating.

Runtime (integer): The length or duration of the movie, expressed in minutes.

Movie Title (string): The name of the movie.

Vote (integer): Number of people vote for the rate.

Year (integer): The year when the film is released.

5) Soundtrack Dataset

We then combine the IMDb dataset with the Track Feature Dataset collected on Spotify to get our final Soundtrack Dataset. The desired description of attributes of Soundtrack Dataset after cleaning will be as follows:

Acousticness (float): A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

Danceability (float): It describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Duration_ms (integer): The duration of the track in milliseconds.

Energy (float): Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

Instrument (float): Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

Key (integer): The key the track is in. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.

Liveness (float): Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

Loudness (float): The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.

Mode (integer): Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

Speechness (float): Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

Tempo (float): The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Time_signature (float): An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

Valence (float): A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Popularity (integer): The popularity of the track. The value will be between 0 and 100, with 100 being the most popular.

Track_name (object): The name of the track

Track_ID (object): The Spotify ID for the track

Album_name (object): The album on which the track appears.

Album_ID (object): The Spotify ID for the album

Movie_name (object): The name of the movie

Movie_genre (object): The genre of the movie

Movie_gross (object): The movie revenue

Movie_rate (float): The rate of the movie by viewers' voting, with a range between 0 to 10. A larger number means higher rating.

Movie_runtime (object): The length or duration of the movie, expressed in minutes

Movie_yr (object): The year when the film is released.

3. Potential Analysis

The IMDb dataset provides general information on popular movies. The Spotify dataset contains detailed information of the musical features of the soundtracks. This enables us to conduct comprehensive analysis from various perspectives. Besides, the popularity data, which represents how often the music is played by the Spotify users, is an interesting attribute worth investigating.

Here are some potential questions we would like to study:

- Does the soundtrack popularity have a positive correlation with the movie popularity?
- What type of movie genre, in general, has the most popular soundtracks?
- For each genre of movies, what are the musical characteristics their soundtracks share?
- What is the most popular time signature and key signature for each genre of movies?
- Do soundtracks in crime and thriller genre usually have negative valence?
- What are the similarities of all the tracks in a single movie or a film series?
- How do the genre and attributes of the soundtracks change over time?
- Do movies with longer duration require more soundtracks?

4. Data Cleaning

1) Data Cleanliness Check

The following table shows the data types in our raw data.

Table 2: Data Types of Raw Data

Attribute	Data Type	Attribute	Data Type
acousticness	*float64*	valence	*float64*
danceability	*float64*	popularity	*float64*
duration_ms	*float64*	Track_name	*object*
energy	*float64*	Track_ID	*object*

instrumentalness	*float64*	Album_name	*object*
key	*float64*	Album_ID	*object*
liveness	*float64*	Movie_name	*object*
loudness	*float64*	Movie_genre	*object*
mode	*float64*	Movie_gross	*object*
speechiness	*float64*	Movie_rate	*float64*
tempo	*float64*	Movie_runtime	*object*
time_signature	*float64*	Movie_yr	*object*

Considering the potential problems which we will investigate, we decide to transfer object data type of Movie_gross, Movie_runtime and Movie_yr to *int* or *float* type.

2) Data Cleaning

By checking the raw datasets of tracklist.csv and featurelist.csv, we noticed that there existed many null values, duplicated rows, and unmatched records. Besides, there might be invalid values for each attributes. To measure the quality of the raw datasets, we provided the quality check based on rate of null values and rate of records that are out of range. The quality table of the two raw datasets are as follows:

Table 3: Track Quality Check Table

Attribute	Track Name	Track ID	Album Name	Album ID	Movie Name	Movie Genre
Null Value Rate	0	0	0	0	0	0
Out-of- range Rate	0	0	0	0	0	0
Attribute	Movie Gross	Movie Rate	Movie Runtime	Movie Rate	Movie Year	
Null Value Rate	0	0	0	0	0	
Out- of- range Rate	0.02931	0	0	0	0	

Table 4: Feature Quality Check Table

Attribute	Acoustic-ness	Danceability	Duration _ms	Energy	Instrumen-talness	Key	Liveness
Null Value Rate	0.001545	0.001545	0.001545	0.001545	0.001545	0.001545	0.001545
Attribute	Loudness	Mode	Speechi-ness	Tempo	Time _signature	Valence	Popularity
Null Value Rate	0.001545	0.001545	0.001545	0.001545	0.001545	0.001545	0
Attribute	Analysis _url	id	track_href	type	uri	track_id	
Null Value Rate	0.001545	0.001545	0.001545	0.001545	0.001545	0	

The normal range of each attributes were defined by the standard of the "API Endpoint Reference: Tracks[3]" published on Spotify Official Website.

The script "Cleaniness_Check.py" is to check the data quality.

The file "Feature_Movie_Clean.py" is to clean the raw data based on the criterions we mentioned above, including:

- Remove attributes that are useless
- Remove records with unmatched track IDs
- Remove records that have null values in track feature
- Remove duplicate records
- Correct other potential errors, including null values, wrongly formatted values, etc.

After first stage of cleaning, we got our original cleaned dataset: full_dataset.csv. As to check whether the dataset has been absolutely cleaned, we provided the description table of each attribute as follows:

[3] https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

Table 5: Descriptions of all attributes

Attribute	Acoustic-ness	Dance-ability	Duration _ms	Energy	Instrumen-talness	Key	Liveness	Loudness
Count	26321	26321	26321	26321	26321	26321	26321	26321
Mean	0.500077	0.452396	201255.8	0.439615	0.386196	5.13229	0.193272	-13.5241
Std	0.366199	0.21768	115187.7	0.294995	0.412345	3.546603	0.170154	7.523836
Min	0	0	1733	2.01E-05	0	0	0	-47.999
25%	0.104	0.259	130493	0.17	5.1E-06	2	0.097	-18.299
50%	0.532	0.47	192726	0.414	0.109	5	0.12	-12.122
75%	0.865	0.625	248773	0.696	0.861	8	0.226	-7.367
Max	0.996	0.983	3664274	1	1	11	0.994	0.626

Attribute	Mode	Speechi-ness	Tempo	Time _signature	Valence	Popularity	Track _name	Track_ID
Count	26321	26321	26321	26321	26321	26321	26321	26321
Mean	0.651799	0.097097	113.2979	3.782113	0.351143	22.69279	Null	Null
Std	0.476409	0.155745	32.81185	0.672162	0.283949	17.65572	Null	Null
Min	0	0	0	0	0	0	Null	Null
25%	0	0.0362	87.836	4	0.0738	7	Null	Null
50%	1	0.0436	111.772	4	0.298	21	Null	Null
75%	1	0.071	134.822	4	0.577	35	Null	Null
Max	1	0.966	244.965	5	0.992	98	Null	Null

Attribute	Album _name	Album_ID	Movie _name	Movie _genre	Movie _gross	Movie _rate	Movie _runtime	Movie _yr
Count	26321	26321	26321	26321	26321	26321	26321	26321
Mean	Null	Null	Null	Null	64.98233	7.567121	117.6857	1997.21
Std	Null	Null	Null	Null	100.2947	0.422295	23.89315	19.32738
Min	Null	Null	Null	Null	0	7	63	1926
25%	Null	Null	Null	Null	3.77	7.2	101	1989
50%	Null	Null	Null	Null	25.97	7.5	115	2004
75%	Null	Null	Null	Null	82.42	7.9	129	2012
Max	Null	Null	Null	Null	936.66	9.4	271	2018

By further cleaning the dataset, we got an “almost clean” dataset. Its cleanness quality is shown as follows:

Table 6: Data Cleanliness of All Attributes

Attribute	Acoustic-ness	Dance-ability	Duration_ms	Energy	Instrumen-talness	Key	Liveness	Loudness
Normal Range	[0,1]	[0,1]	[0,4000000]	[0,1]	[0,1]	[0,11]	[0,1]	[-60,1]
Out-of-range Rate	0	0	0	0	0	0	0	0
Null Value Rate	0	0	0	0	0	0	0	0

Attribute	Mode	Speechi-ness	Tempo	Time _signatur e	Valence	Popularity	Track _name	Track_ID
Normal Range	[0,1]	[0,1]	[0,1]	[0,1]	[0,8]	[0,100]	Null	Null

Out-of-range Rate	0	0	0	0	0	0	Null	Null
Null Value Rate	0	0	0	0	0	0	0	0

Attribute	Album _name	Album _ID	Movie _name	Movie _genre	Movie _gross	Movie _rate	Movie _runtime	Movie _yr
Normal Range	Null	Null	Null	Null	[0,10000]	[7,10]	[0,1000]	[1900, 2018]
Out-of-range Rate	Null	Null	Null	Null	0.095019	0	0	0
Null Value Rate	0	0	0	0	0	0	0	0

After cleaning the full dataset, we noticed that the only remaining problem was that the gross of some movies is 0. To resolve this issue, we further confirmed with the IMDb official website. We found that these are correct gross values because the movies with gross less than 0.01 billion would appear as “0” while scraping the information from IMDb. Considering the potential problems in our future analysis, we decided to exclude these movies from our dataset.

Finally, we got our cleaned datasets ready to be analyzed: cleaned_data.csv. Using the cleaned dataset, we would like to investigate the potential problems mentioned earlier on soundtracks of our targeted movies,

References

Xu, Haifeng and GooNullwardene, Nulldee, "DOES MOVIE SOUNDTRACK MATTER? THE ROLE OF SOUNDTRACK IN PREDICTING MOVIE REVENUE" [2014).PACIS 2014 Proceedings. 350. http://aisel.aisnet.org/pacis2014/350

Hoeckner, B., Wyatt, E. W., Decety, J., & Nusbaum, H. [2011). Film music influences how viewers relate to movie characters. Psychology of Aesthetics, Creativity, and the Arts, 5[2), 146-153.