spotify-data-science

Progress report

project repository

project website

For best viewing experience open this document here

About Data

1. We have collected our data from multiple sources: Spotify and RateYourMusic.
2. Spotify data consists of its metadata like name, genre, popularity and audio features like liveness, tempo, danceability, etc.
3. RateYourMusic is a public forum where users review albums and songs. Its data consists of critic rating, user rating, reviews, and number of reviews.
4. We have merged this data such that each entry has the song metadata, audio features, and ratings data. The ratings data will serve as a label and this is the reason we scraped data from multiple sites.
5. The data is stored in csv format with a ~ separator.
6. Link to our dataset.

In [ ]:
```
import pandas as pd
df = pd.read_csv('../data/merge-data-1.csv', sep='~')
columns = ['id', 'name', 'danceability', 'energy', 'key', 'loudness', 'mode',
df = df[columns]
df.describe()
```

Out[ ]:

	danceability	energy	key	loudness	mode	speechiness
count	14363.000000	14363.000000	14363.000000	14363.000000	14363.000000	14363.000000
mean	0.441265	0.490766	5.232055	-13.140224	0.653206	0.084234
std	0.183334	0.303645	3.518423	6.991007	0.475966	0.130424
min	0.000000	0.000000	0.000000	-60.000000	0.000000	0.000000
25%	0.302000	0.217000	2.000000	-17.153000	0.000000	0.036400
50%	0.435000	0.484000	5.000000	-11.518000	1.000000	0.045200
75%	0.573000	0.762000	8.000000	-7.987500	1.000000	0.070100
max	0.967000	1.000000	11.000000	3.744000	1.000000	0.964000

Data Spec

1. Audio features: 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'type', 'uri', 'duration_ms', 'time_signature',

2. Song metadata: 'id', 'name', 'album', 'date', 'genres', 'track_href', 'analysis_url', 'popularity'

2. Ratings from RYM 'rym_rating', 'ratings', 'reviews'

In [ ]:
```
import random
df.iloc[random.sample(range(4950, 5500), 15)]
```

Out[ ]:

	id	name	danceability	energy	key	loudness	mode	sp
5110	6XMHWGtaCQcEGMTy2QIXM5	My Love Is Growing	0.5420	0.675	9	-10.819	0	
5181	2fYxpSFhg95sBwaGp5JlFU	Your Sister Can't Twist (But She Can Rock'n' R...	0.4640	0.972	4	-6.887	1	
5251	1Hivv2cXFYPINeKTZLY6Kh	Milagre Dos Peixes - Ao Vivo	0.3470	0.458	9	-11.763	1	
5472	11C3sTCe07SHNDxi2IAZoI	Soon - 2003 Remaster	0.1450	0.361	9	-8.760	0	
5095	1UMoEbjcFhtSwZUNo2MK4y	Mrs Vandebilt - Remastered 2010	0.6960	0.641	9	-8.633	1	
5023	1HKbWgJm1kYJvuU1yl7Aoy	I Am The Sea	0.0839	0.691	1	-25.041	1	
4957	1PNk1Xno8t3spNEU78Hxsu	Future Days	0.7030	0.664	7	-17.789	0	
5188	35D8nUihmqRQaFHEQnOXXQ	What's The Buzz? / Strange Thing, Mystifying	0.5950	0.746	7	-8.972	1	
4996	4gUNTtPmsTzpTdJY3aZbLw	Fluff (2009 - Remaster)	0.4930	0.236	2	-17.116	1	
5101	39l1UORlhuHvUWfxG53tRZ	Let's Get It On	0.5390	0.599	3	-10.638	1	

	id	name	danceability	energy	key	loudness	mode	sp
5164	7pT6laXJKXDq85Ucu2fNUg	Ottagtha Katikko	0.9670	0.450	10	-10.763	0	
5113	1emvYBqAWFPpFUQyhBG9He	I'd Give My Life For You - Alternate Mix	0.2230	0.409	0	-8.256	1	
5294	1EHgtjRSiVPq9nC6AHonzt	Voso	0.5930	0.850	1	-6.566	1	
4988	5OBptQtIlGOG94la7qil6c	Firth Of Fifth - Remastered 2008	0.2470	0.594	4	-9.710	0	

Data Collection

Collection Process

1. Our approach was to first scrape the data from RYM as there was some uncertainity with it. Retrieving data from Spotify is straightforward with Python library `spotipy` which internally integrates with its well documented API.
2. We have stored our data in csv format

How did collect your data

1. We first explored libraries to scrape data: decided on spotipy for Spotify and rymparser for RateYourMusic.
2. We scraped through the charts of top 20 albums each year from 1960-2021 over RYM. Collected features like reviews, ratings, number of reviews, genres, album name.
3. Mapped the album name from RYM data to Spotify by using Spotify search APIs.
4. Added more features relating to audio, quality and song metadata.

How did you clean your data

1. One advantage of scraping your own data is we can make cleaning part of the scraping process which is what we did. Since the data was in abundance we filtered out the tracks that has missing values.
2. While combining the RYM with Spotify dataset we defined and used our own similarity metric wrt to the album titles when querying the Spotify APIs
3. To have a good distribution of genres we deployed multiple metrics like: limiting the number of songs by a particular artist, taking the most popular songs from the rated albums.

Mention any difficulties you faced in the beginning steps

1. I had initially missed to check the `robots.txt` file over RYM which resulted in my IP ban on that site. I used VPN to bypass this ban.
2. `beautifulsoup` also did not work over RYM due to the `robots.txt` configuration. We explored multiple projects over Github and finally found a repository that used `selenium` which emulates your browsing by actually maintaining session within your browser to the website

Exploratory Data Analysis

1. After creating a correlation matrix, we found there are only a couple variables that have a strong correlation between each other. For example, acousticness and energy had a higher correlation of -.8.

2. Our preliminary EDA shows there is a strong bias, on the top 20 genres from Rate Your Music's community. The top 5 genres include Baroque Music, Film Score, Romanticism, Pop Rock, and Modern Classical would not be the top genres for other websites as these ratings depend on the community of rateyourmusic.

3. Looking at Pop Rock specifically over the years, there is a peak of pop rock genre songs in 1968.

4. The dataset comes from Rateyourmusic and Spotify. The rym rating is from Rateyourmusic and popularity is from Spotify. When comparing those metrics, there is an almost 0 correlation. From this, we can also say there is a huge bias from the Rateyourmusic's community, as Spotify is more telling of music metrics that is more mainstream.

5. Some methods we used to explore the data were bar graphs, a correlation matrix, and line plots.

6. Find our initial EDA here. We plan to do more work on EDA since our most of our time was spent in data collection and cleaning.

Challenges

1. Formulating the problem statement was challening. We saw many datasets and their EDA over Kaggle for Spotify. All of them were some sort of unsupervised modelling. We thought
2. Scraping and cleaning: We ran into various problems of installation, IP banning, and combining & cleaning data.
3. We foresee a challenge to devise a metric that correlates the songs of different genres. User ratings for songs alone cannot serve as a metric for both recommending

the best song and that too from a different genre

Next Steps

1. We plan to do some more EDA along the lines for reducing dimensions/finding the most relevant features.
2. We plan to build our song recommender using a combination of supervised learning - based on user ratings and unsupervised learning - based on audio features.
3. In the end will compare our results with the plain unsupervised models (existing work on Spotify datasets).

Duties

1. Data scraping and cleaning - Aditya Kanthale
2. EDA - Amanuel Odicha, Vivek Patel