University of Copenhagen

MSc. IT & Cognition

Cognitive Science 2

What is a good rock album?

Data science exploration of rock albums audio features & lyrics sentiment analysis

Author:

Cristian Mitroi

04/06/2018

Contents

Introduction

Music is an essentially human enterprise. It has been with us since the earliest of societal gatherings and it is still an essential part of contemporary culture. In 2017 the music industry of the US generated 18.3 billion dollars (Statista 2018a). Out of this, rock music is the second most popular genre (Statista 2018b). While it is indeed a matter of subjective taste and culture, I am interested in whether we can see any patterns that make a rock album *better*.

The goal of this project is two-fold: first, I gather data about the rock albums in the "Top 500 Albums of All Time" (Rolling Stone 2012); secondly, I apply different visualization and analysis techniques to this dataset with the objective of uncovering any substantial differences between the albums at the top of the chart and those at the bottom.

My hypothesis is that there *is* a difference between top-ranking albums and the general mass of albums.

Background and Relevant Research

One of the most relevant pieces of research in the matter is a blog post written by one of the engineers that worked at Spotify (Dieleman n.d.). He documents the architecture and meaning behind the Spotify API's audio features. Apparently, they used a convolutional neural network in order to obtain "filters" that correspond to different types of music or sound characteristics. For example, there was a filter that was activated by vibrato singing. This article was relevant for my project because it allowed me to understand the significance of the features. It also showed that the features themselves are calculated as the means of said feature computed over multiple audio windows. This greatly minimizes the impact of the time dimension of music.

There have been attempts at determining which features could predict a song's popularity (Pham et al. 2016). In this they discovered that features such as "artist familiarity, loudness, year, and a number of genre tags" were the most apt. The project seems to have been quite thorough, in that they used multiple approaches (LDA, Linear Regression, SVM etc.).

By contrast, I am not working on predicting a specific continuous value (popularity, rating etc.). Rather, I use these numbers (popularity, rating) in order to separate albums in different groups and see what makes a salient quality for the albums at the top. Also, I am working on the album level, not the song-level. I believe this is a worthy avenue of questioning, since rock albums have been associated with a holistic artistic vision - the 'complete album approach' (Martin 2015, p.41).

It is also worth mentioning research into how music is an ultimately subjective experience, where the quality judgment of a certain piece is not dictated by just mere *audio features*. In fact, it has been shown that social pressure can influence the assessment of a song (Salganik et al. 2006).

Data acquisition

As mentioned in the introduction, one of the main efforts of this project is to acquire the relevant dataset. This means web scraping the APIs for the relevant information. In this section I will discuss this, with a focus on the decisions and several issues encountered along the way.

Firstly, I acquired the list of "500 Greatest Albums of All Time" (Rolling Stone 2012) from Kaggle (Gibs 2017). This included tags for genre that had been scraped from MusicBrainz (MusicBrainz n.d.). I then filtered this list to only include rock albums, which left me with 332 albums.

Then I needed to collect rating information. From my research, there does not seem to be one single accepted source of information for this (Quora 2018). I opted for the Discogs API (Discogs 2018), since it seemed like the easiest to access and use overall. I thus acquired the ratings for each of the albums.

Also using the Discogs API I obtained the list of tracks for each of the albums. This was necessary because most of the releases on the Spotify API (Spotify 2018) include "extra" (or bonus) tracks, such as remixes or demo versions. I needed to make sure that I only look at the tracks from the original release.

Then I turned towards the Spotify API. I first searched for the album ID using the artist and album name strings. In some cases this returned a single album entry; in other cases it returned multiple versions of the same (such as 're-masters' and 'digital re-releases'). This proved problematic, e.g. 'Velvet Underground' by Velvet Underground has multiple entries, each with different popularity scores. In this case, I opted for the entry with the highest popularity. It simply means that it is the most 'active' entry.

From the Spotify API I also collected the 'popularity' score of each of the albums. Because my hypothesis is about albums, I only collected this at the album level, where it's based on the average of the popularity of the tracks themselves.

One hurdle I faced was matching the tracks from Discogs to the ones provided by the Spotify API. One problem was making sure that I only got the tracks from the original release and nothing more. In some cases Spotify did not provide the full tracklist. In order to make sure that the dataset is meaningful I imposed a minimum of 70% of tracks to be present for each album.

For each of the tracks I then collected the audio features provided by the API:

- danceability

- energy
- instrumentalness
- key
- liveness
- loudness
- mode
- speechiness
- tempo
- time_signature
- valence

Finally, I collected the lyrics for each of the songs from two APIs: LyricWikia (LyricWikia 2018) and Genius (Genius 2018). These were used for text sentiment analysis, which constitutes another feature in the dataset. Thus, the dataset becomes multi-modal: audio and text.

3.1 Features

In this section I will provide an overview of the different features collected. Where the source is not mentioned, the Spotify API should be assumed.

1. danceability

   "Danceability 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." (Spotify 2018)

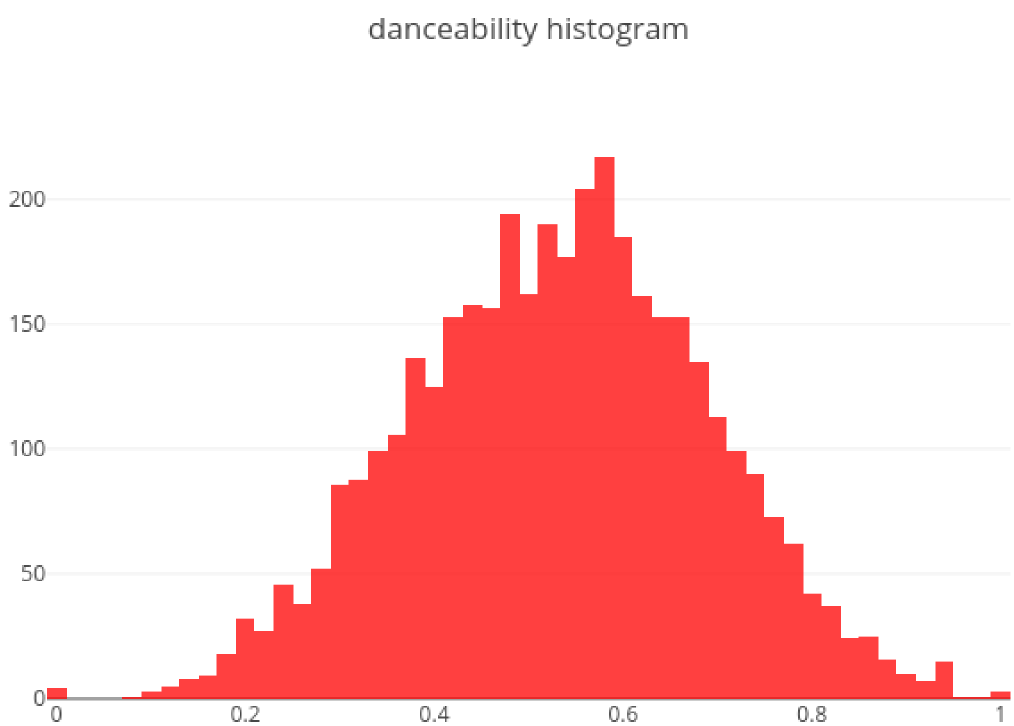

Figure 3.1: danceability

```
count    3899.000000
mean        0.531763
std         0.156972
min         0.000000
25%         0.423118
50%         0.536585
75%         0.641039
max         1.000000
```

2. energy

   “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.” (Spotify 2018)

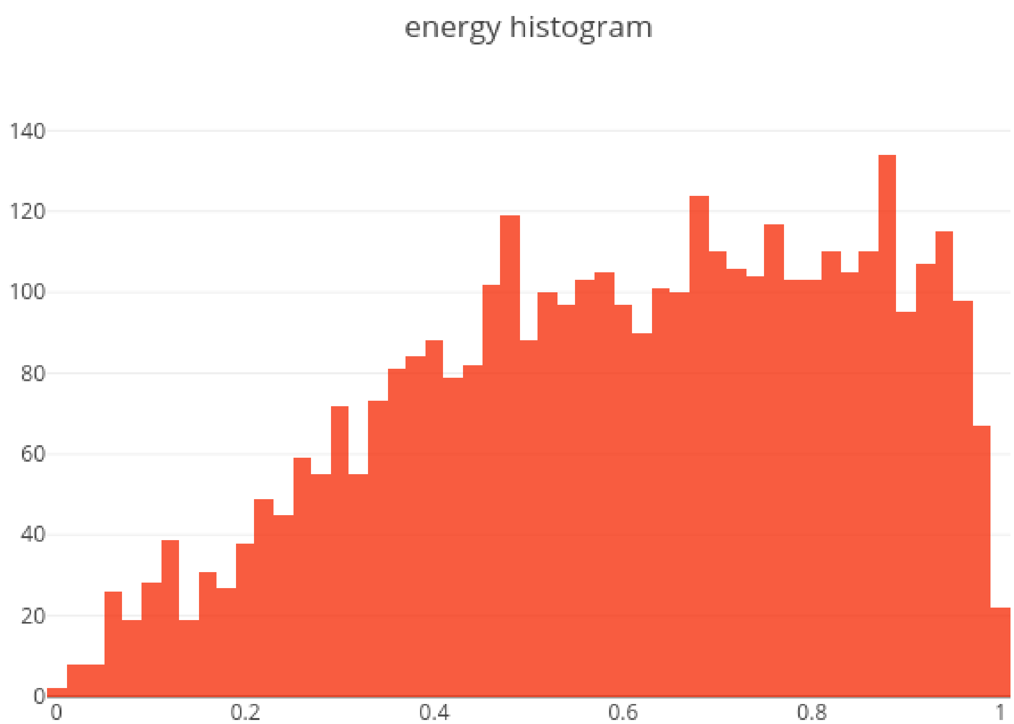

Figure 3.2: energy

```
count    3899.000000
mean        0.605178
std         0.239385
min         0.000000
25%         0.427136
50%         0.625126
75%         0.807035
max         1.000000
```

3. instrumentalness

"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." (Spotify 2018)

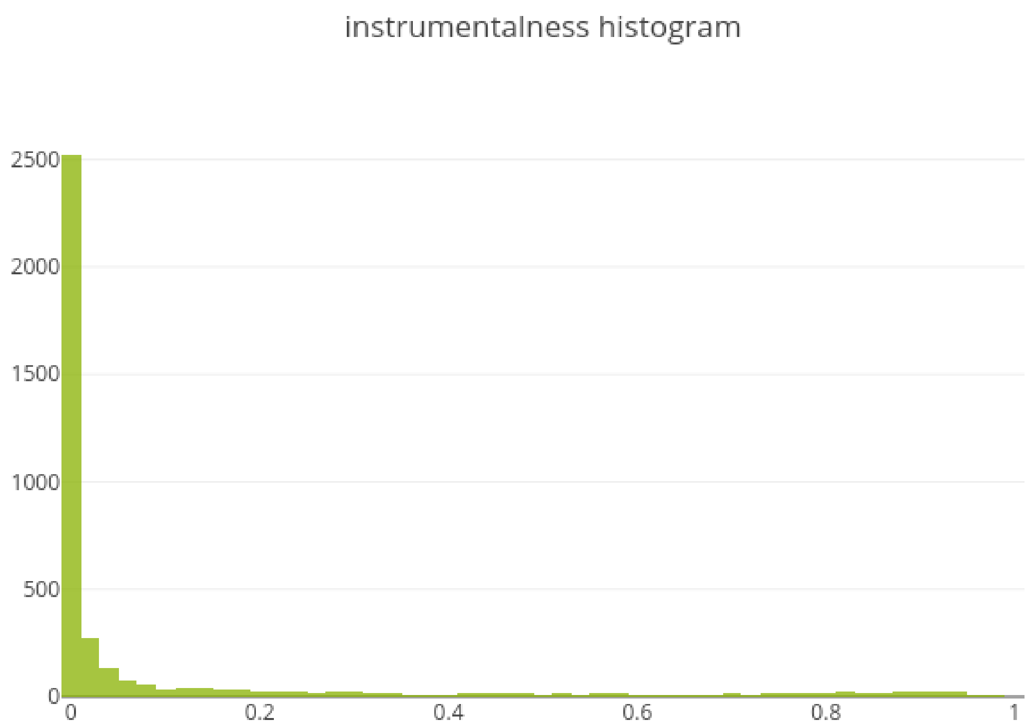

Figure 3.3: instrumentalness

```
count    3899.000000
mean        0.110764
std         0.240307
min         0.000000
25%         0.000000
50%         0.001014
75%         0.049696
```

```
max          1.000000
```

4. key

   “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.” (Spotify 2018)

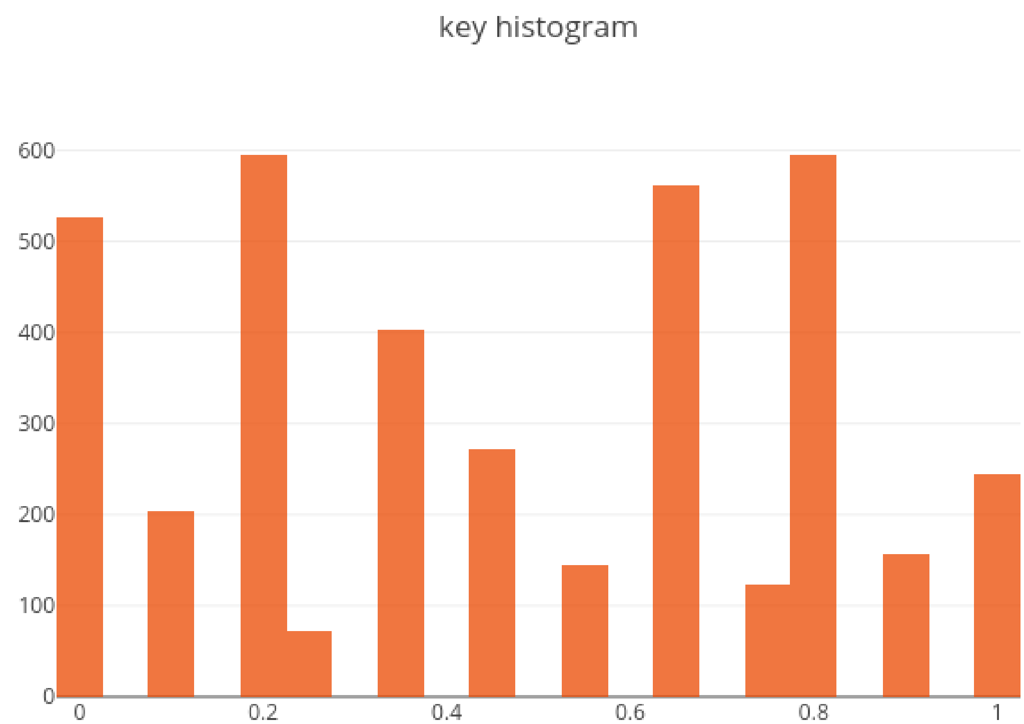

Figure 3.4: key

```
count    3899.000000
mean        0.466017
std         0.319563
min         0.000000
25%         0.181818
50%         0.454545
75%         0.818182
```

```
max          1.000000
```

NOTE: This feature is not important as such. This is because it is a matter of convention, not of subjective impression. This only becomes relevant as a matter of variance (or standard deviation) - “how often does an artist switch key during an album?”

5. liveness

   “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.” (Spotify 2018)

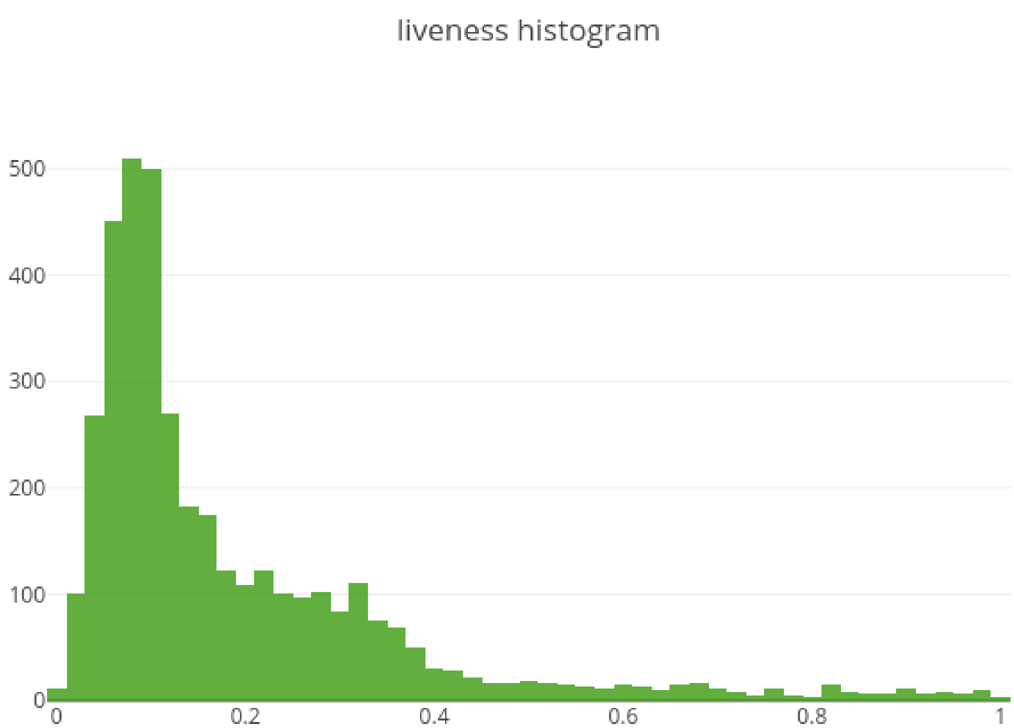

Figure 3.5: liveness

```
count     3899.000000
mean         0.188860
std          0.178912
```

```
min          0.000000
25%          0.075820
50%          0.115779
75%          0.248975
max          1.000000
```

6. loudness

   "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." (Spotify 2018)

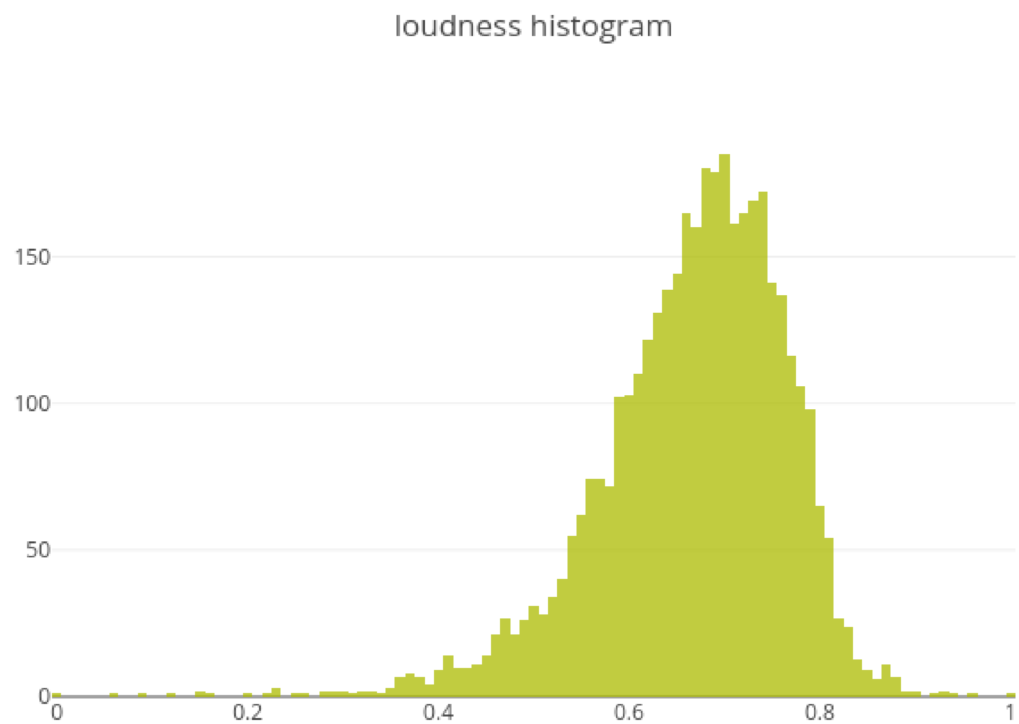

Figure 3.6: loudness

```
count    3899.000000
```

```
mean         0.666821
std          0.099733
min          0.000000
25%          0.612565
50%          0.679963
75%          0.736031
max          1.000000
```

7. mode

   "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." (Spotify 2018)

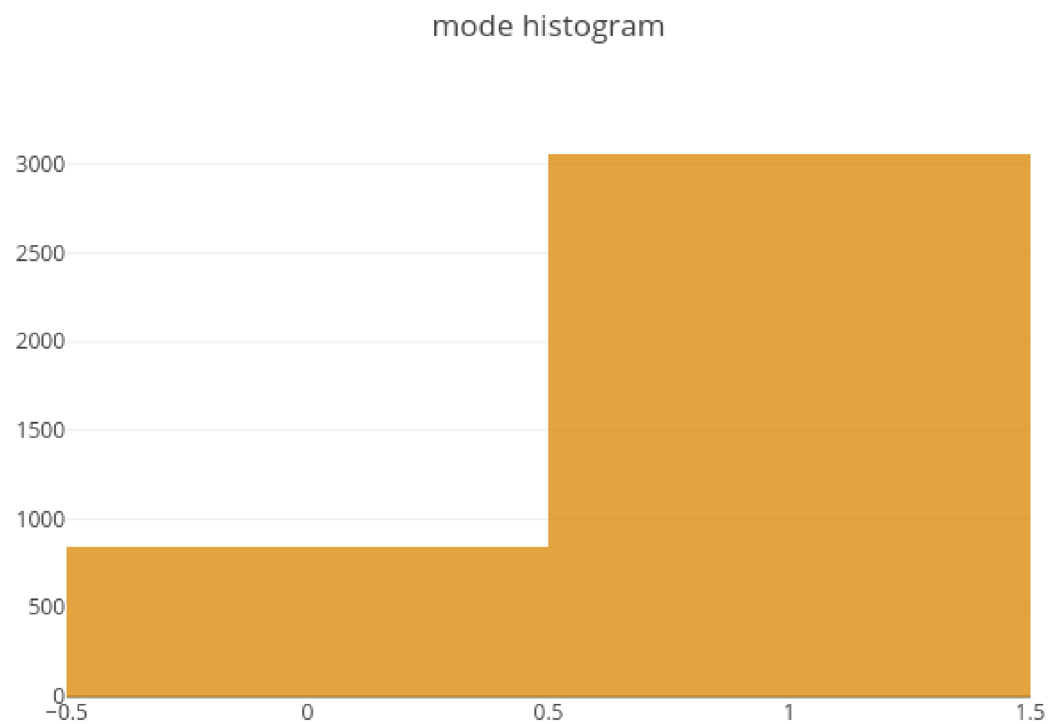

Figure 3.7: mode

```
count     3899.000000
```

```
mean         0.784047
std          0.411534
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          1.000000
```

8. speechiness

   "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." (Spotify 2018)

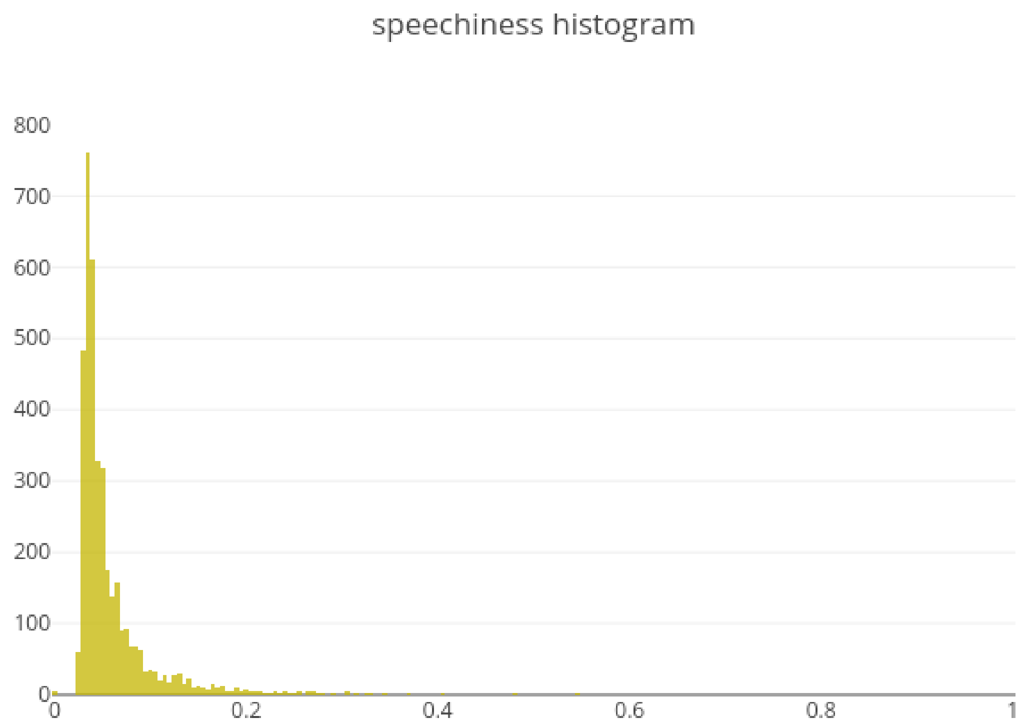

Figure 3.8: speechiness

```
count    3899.000000
mean        0.062362
std         0.062430
min         0.000000
25%         0.034745
50%         0.043431
75%         0.064061
max         1.000000
```

9. tempo

   “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.” (Spotify 2018)

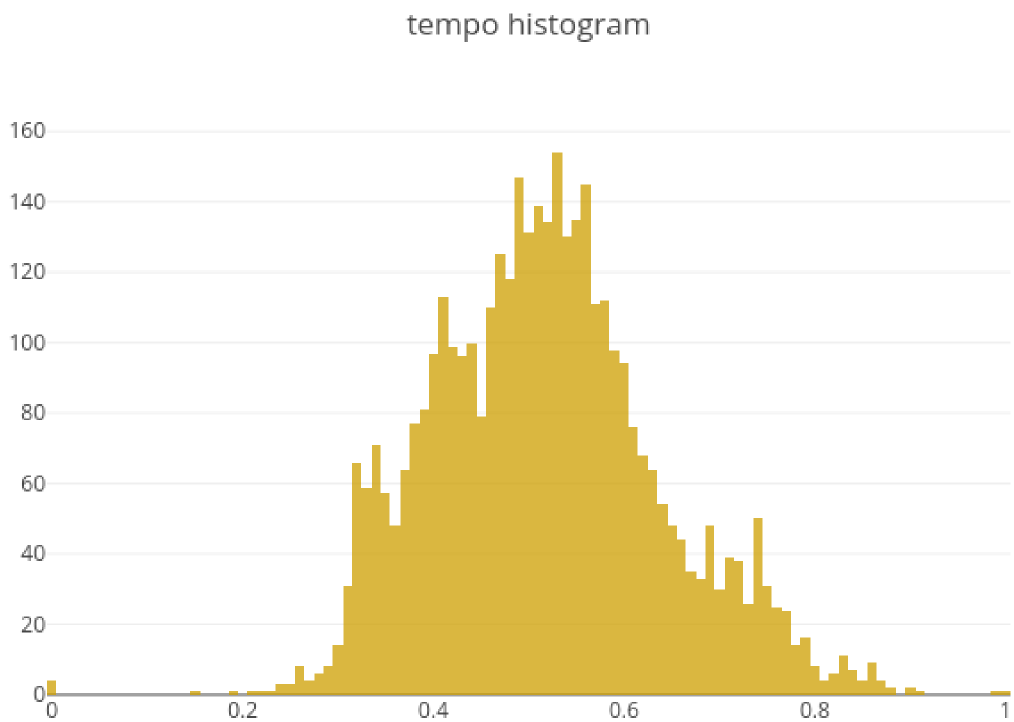

Figure 3.9: tempo

```
count    3899.000000
mean        0.518314
std         0.120759
min         0.000000
25%         0.431049
50%         0.513962
75%         0.589135
max         1.000000
```

10. time_signature

    “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).” (Spotify 2018)

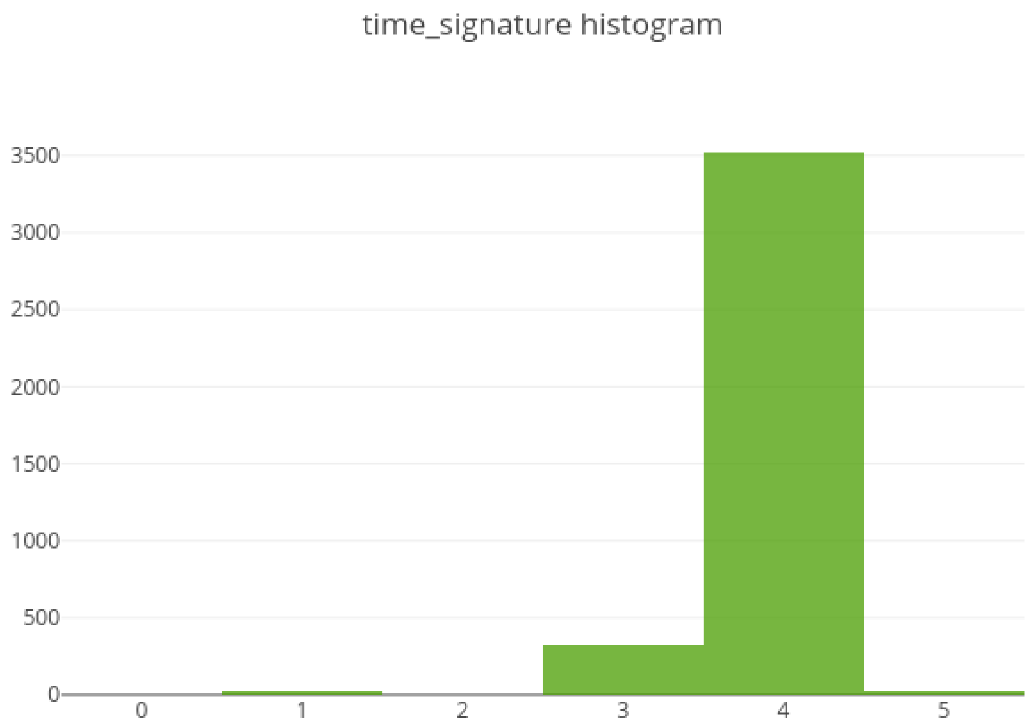

Figure 3.10: time_signature

```
count    3899.000000
mean        0.780303
std         0.078478
min         0.000000
25%         0.800000
50%         0.800000
75%         0.800000
max         1.000000
```

NOTE: This feature, like the *key* feature, is not useful as such. Rather, it is interesting to see the variance (or standard deviation) of this feature. That can tell us how often the time signature is changed across an album.

I have mostly chosen to ignore this feature, as there are reports of invalid detection from the Spotify engineers (mortenhjort GitHub user n.d.).

11. valence

    “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).” (Spotify 2018)

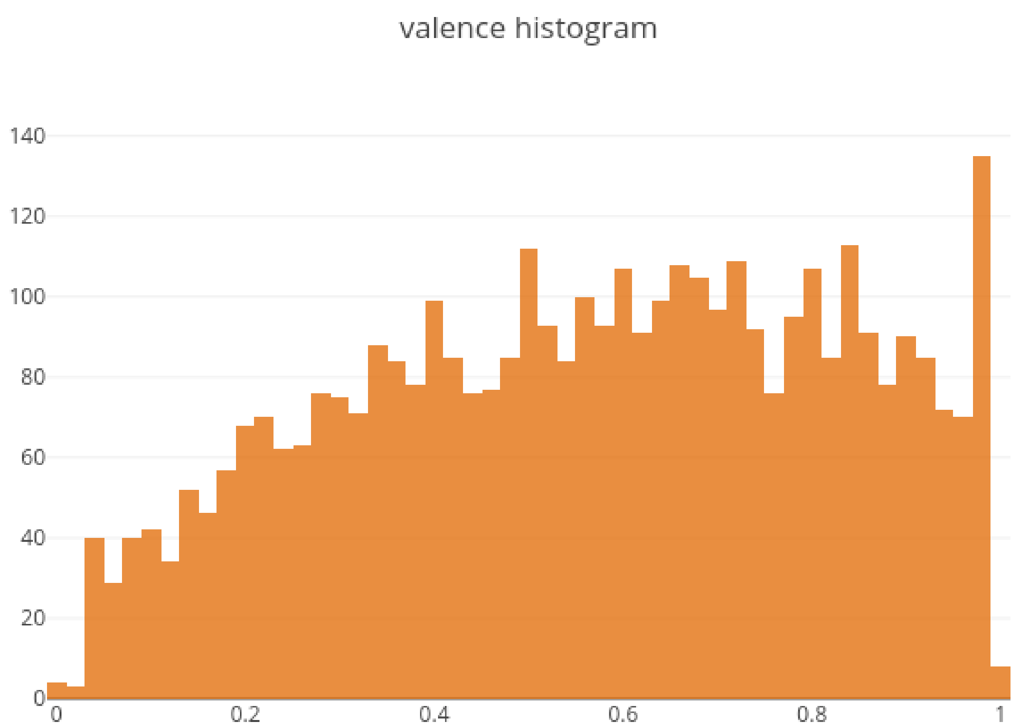

Figure 3.11: valence

```
count    3899.000000
mean        0.565662
std         0.255829
min         0.000000
25%         0.364467
```

```
50%           0.582741
75%           0.782741
max           1.000000
```

12. popularity (at the album level)

    "The popularity of the album. The value will be between 0 and 100, with 100 being the most popular. The popularity is calculated from the popularity of the album's individual tracks." (Spotify 2018)

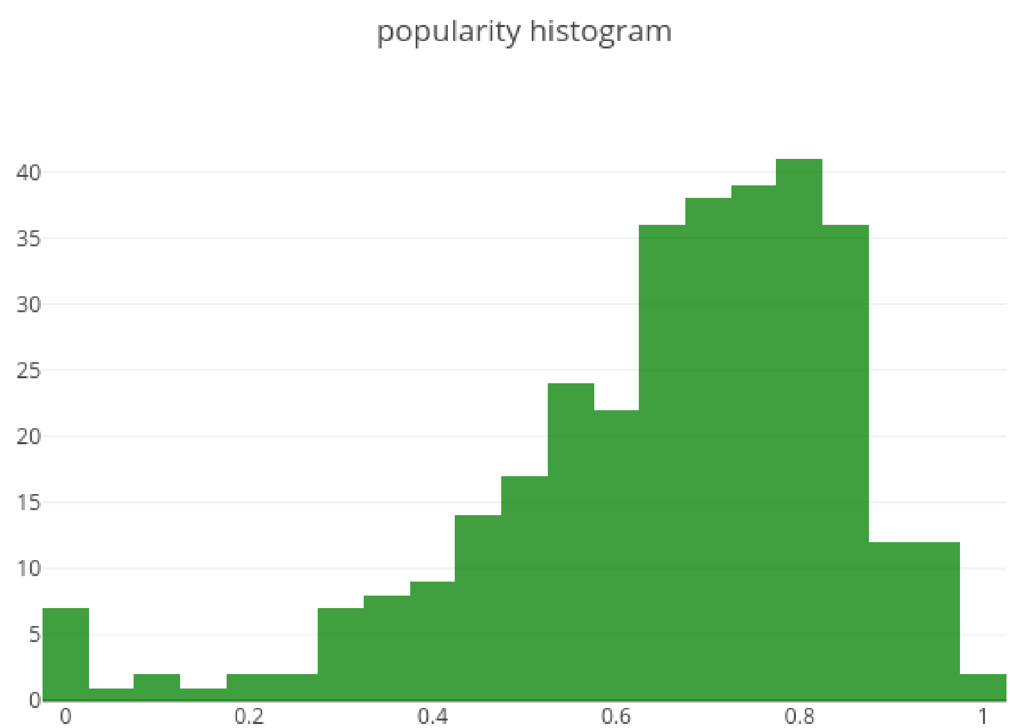

Figure 3.12: popularity

```
count     332.000000
mean        0.659372
std         0.201916
min         0.000000
```

```
25%        0.544304
50%        0.696203
75%        0.810127
max        1.000000
```

NOTE: This will be one of the features - along with “rating” below - used to measure how *good* an album is.

13. rating

    From Discogs API. It is the average of the ratings provided by the users of the Discogs website. Like the “popularity” score from the Spotify API, it is a measure of what a *good* album is.

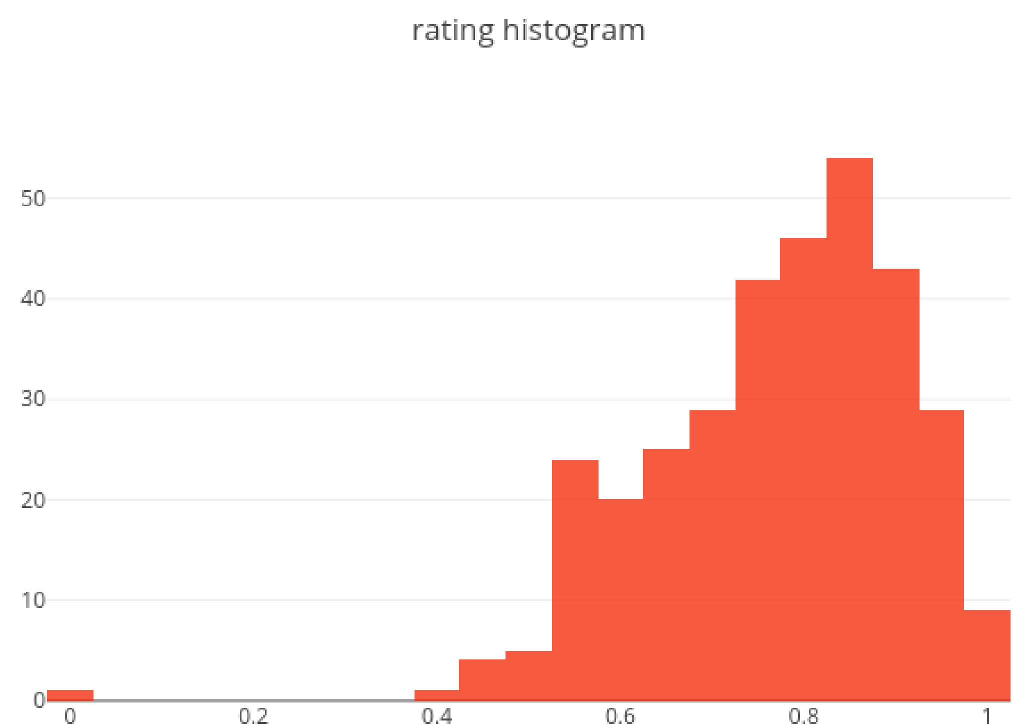

Figure 3.13: rating

```
count    332.000000
```

```
mean        0.771050
std         0.136597
min         0.000000
25%         0.684286
50%         0.788571
75%         0.874286
max         1.000000
```

14. sent

    "Ranges between -1.0 (negative) and 1.0 (positive) and corresponds to the overall emotional leaning of the text" (Google n.d.). I have used the Google Cloud Language API for obtaining this feature from the lyrics of each song.

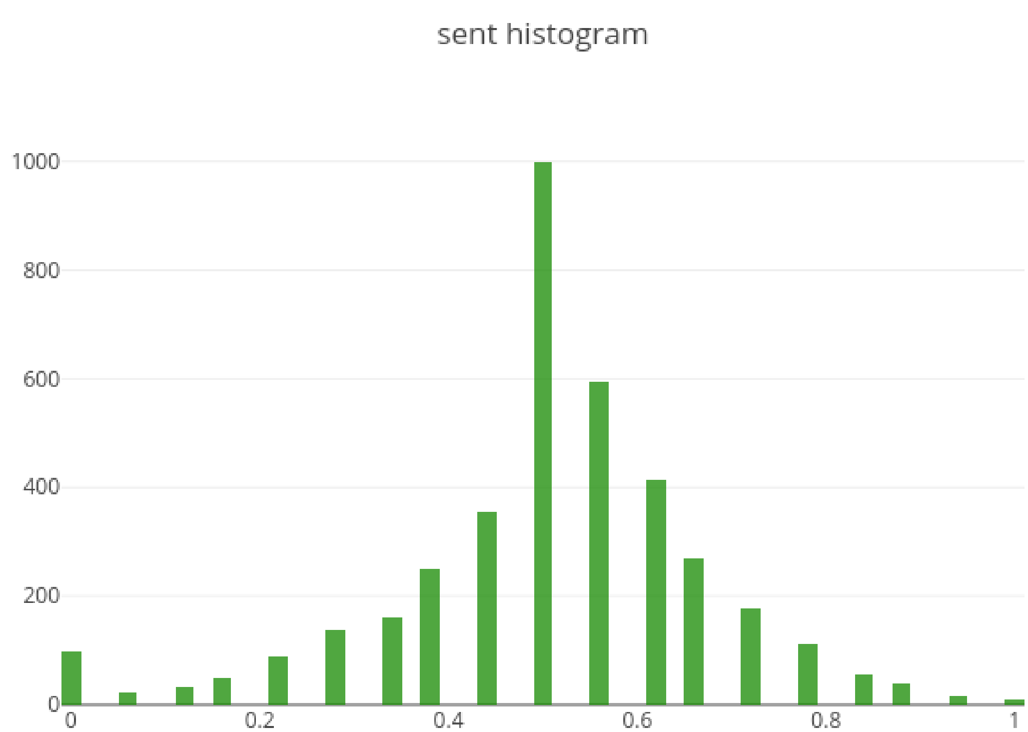

Figure 3.14: sent

```
count    3892.000000
mean        0.506312
std         0.169508
min         0.000000
25%         0.444444
50%         0.500000
75%         0.611111
max         1.000000
```

I decided to use the Google Cloud Language API because of a lack of time. Also, the goal of the project was not to develop a sentiment analysis system on my own.

15. sent_magn

    "Indicates the overall strength of emotion (both positive and negative) within the given text, between 0.0 and +inf. Unlike score, magnitude is not normalized; each expression of emotion within the text (both positive and negative) contributes to the text's magnitude (so longer text blocks may have greater magnitudes)" (Google n.d.).

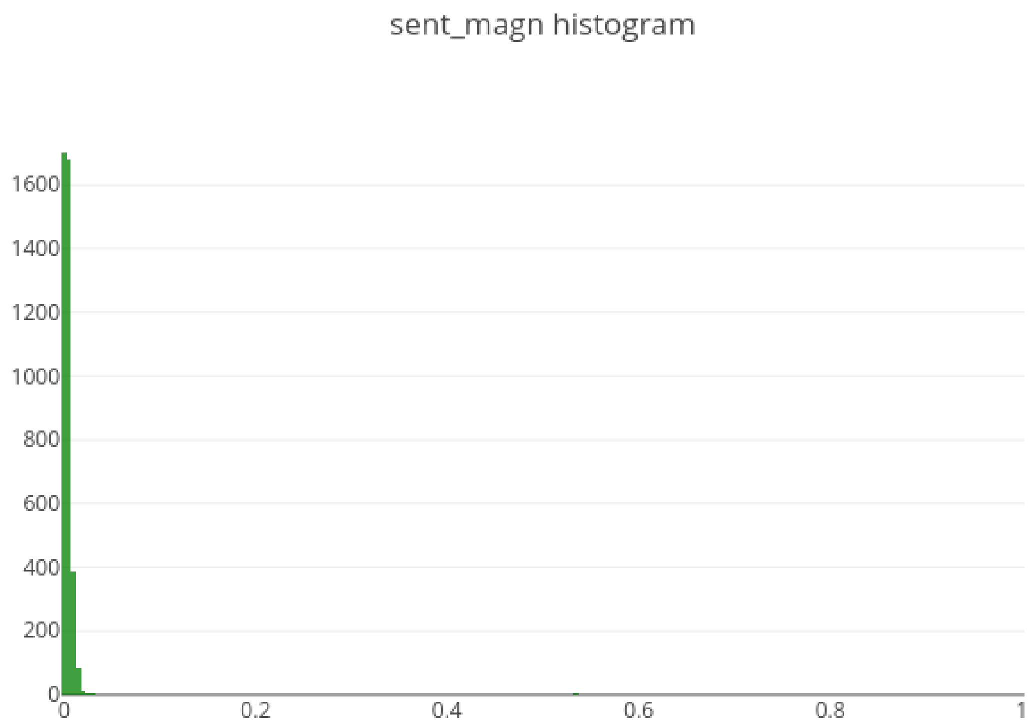

Figure 3.15: sent-magn

```
count    3892.000000
mean        0.006033
std         0.035806
min         0.000000
25%         0.001654
50%         0.002856
75%         0.004885
max         1.000000
```

3.1.1 Note on the features

I would like to provide a note about the features being *means* of the different audio segments. This unfortunately eliminates a lot of the information essential to music: *time*.

Music is a temporal experience, heavily reliant on chord progressions, melody and other shifts. Working with just the *average* of these elements excludes a lot of the essential qualities of the subjective perception of music. In this sense, my project attempts to test whether we can detect any patterns that separate the top albums from the bottom, *within* the confines of such abstractions.

3.2 Data preprocessing

I scaled all the features of the tracks. Most of them were already scaled (in the 0.0 - 1.0 range). I also scaled the popularity and rating of the albums. For this I used the `MinMaxScaler` from the `sklearn` library. The data shown in the "Features" section is the data after the scaling has been done.

Methods and results

In this section I will discuss the various techniques I used in order to visualize and ascertain whether there *is* any difference between top and bottom albums.

4.1 Description of an album

My hypothesis is about the rating / popularity at the album level. Thus I had to devise a way of 'describing' an album based on the features of its tracks.

I decided to use the standard deviation and mean of the features. Thus, for each album I collected all its tracks, and then computed the mean and standard deviation for their features. I believe this captures enough of the information to provide meaningful results for the sake of this project.

I have also decided to use all the features available from the tracks, with the exception of sentiment magnitude.

4.2 Histograms

Firstly, I used one of the classic methods of data science, the histogram. I first split the 332 albums into a top percentile and a bottom percentile. At first this was a 50-50%. In the end I decided on 30-30% since this would include less of the middle area of the albums, and would hopefully emphasize the differences.

Thus I plotted the histograms for each of the features, for the two groups (top and bottom) and compared them for any salient differences. I did this with sorting both by rating and by popularity.

Unfortunately this did not yield any significant or striking results for most of the comparisons.

I will note here the differences that did seem intriguing:[1]

[1]See histograms in appendix.

4.2.1 Sorted by rating

danceability mean

We can see that on average top albums have a higher danceability, with a big chunk of the top albums being in the 0.53 - 0.61 range.

energy std

We notice that top albums tend to have more variety (higher standard deviation) in their energy.

4.2.2 Sorted by popularity

danceability std

We can see that there is a large number (~35) of top albums with the mean of the standard deviation for danceability, as compared with bottom albums. This can indeed be evidence for a 'sweet spot' in terms of what makes a 'popular' rock album.

sentiment mean

Another interesting observation can be made about the lyrics sentiment. While the top and bottom popularity albums have almost the same exact mean (0.502 and 0.519), the distribution is quite different. For top albums, there are clearly more albums in the 0.44 - 0.53 range (the middle area), while for bottom albums, there are more in the right margin (0.53 - 0.59).

Overall there does not seem to be any obvious differentiating qualities between top and bottom albums.

4.3 tSNE

T-distributed Stochastic Neighbour Embedding is a machine learning algorithm for visualization. It is used for projecting a dataset in a higher dimensionality in a lower dimensionality space.

It has been already been used for music analysis in (Eck & Montréal) in order to cluster music audio features.

My approach is to project the dataset of the albums (with 28 features) into a lower dimensionality space (2 or 3), with each of the datapoints being colored depending on its position in the ranking. My hypothesis would suggest that there would be some clusters based on the ranking of the album.

Unfortunately, this did not prove to be the case, as there was no visible separation of between the albums. No clusters were formed.

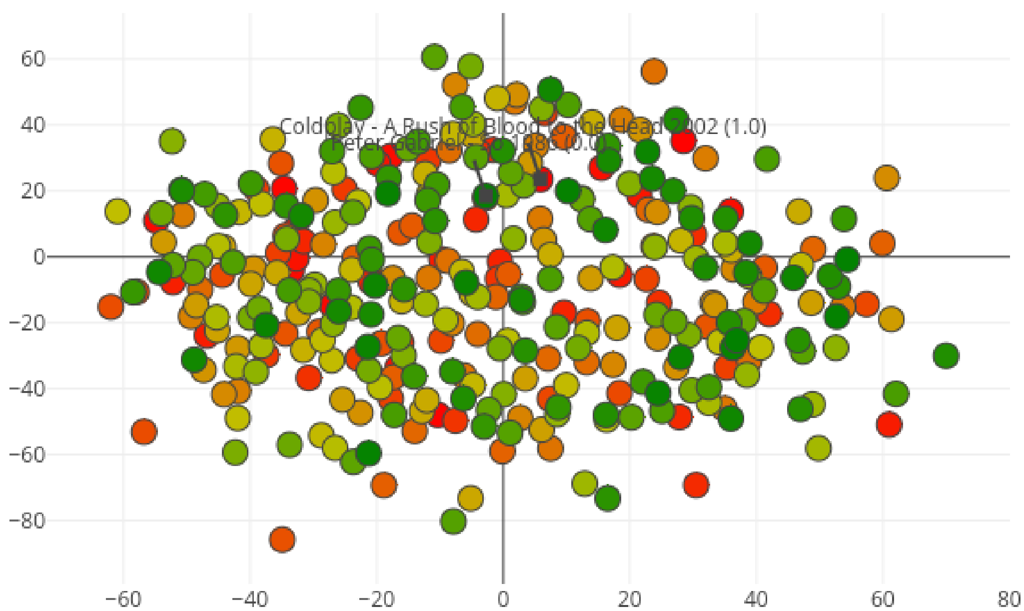

Figure 4.1: tsne by popularity

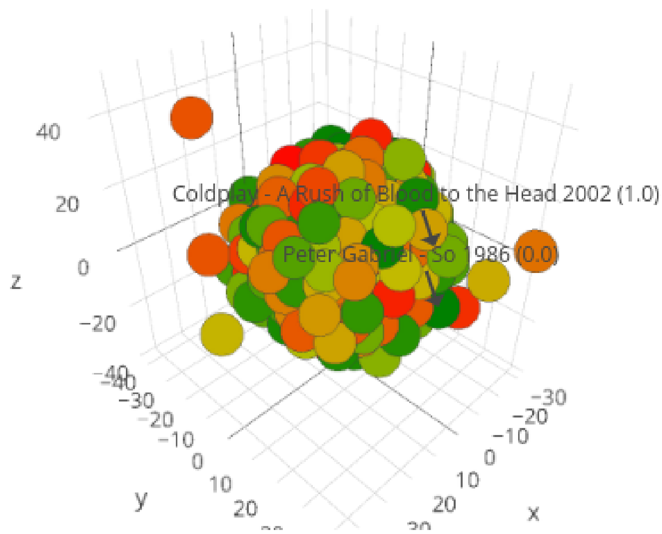

Figure 4.2: tsne3d by popularity

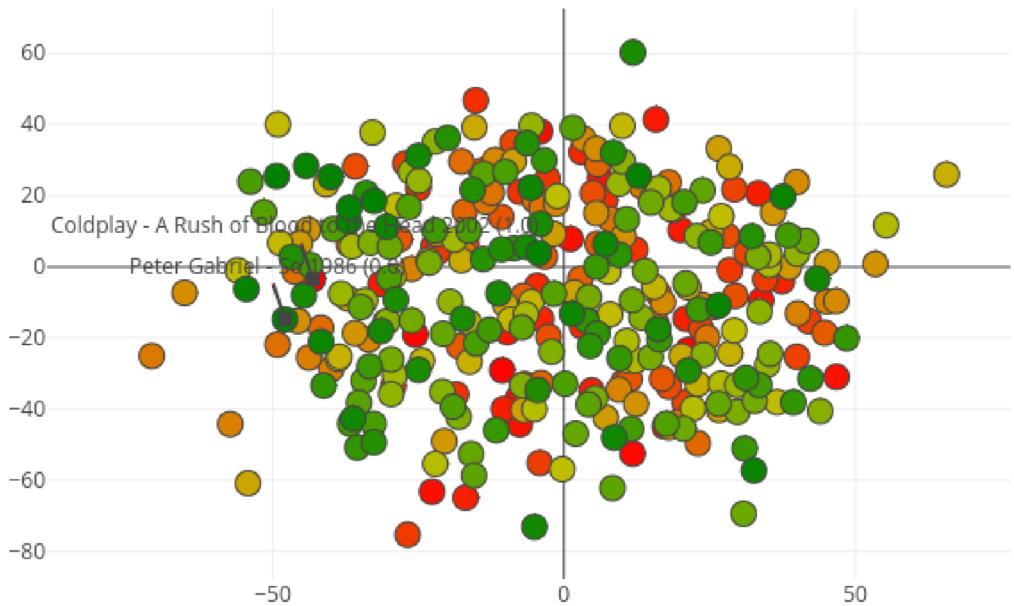

Figure 4.3: tsne by rating

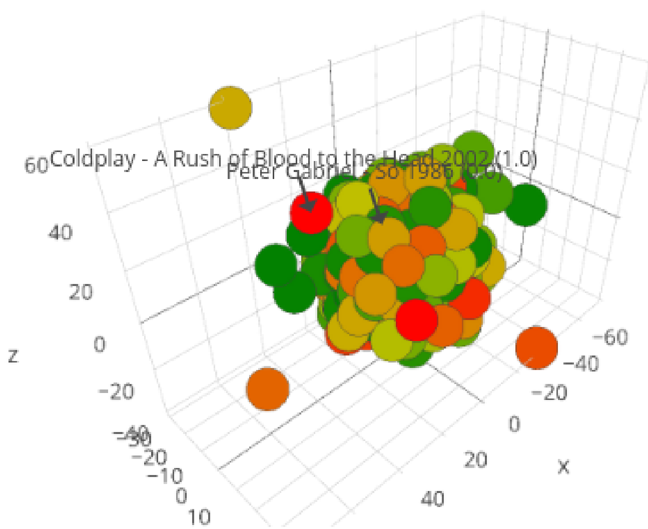

Figure 4.4: tsne3d by rating

We can see that the green (low rank), red (high rank), and all the colours in between overlap each other, with no clear separation.

4.3.1 PCA

I have also attempted to first reduce the dimensions of the dataset with Principal Component Analysis before applying tSNE. With eigenvalue decomposition I managed to capture 96% of the variance in 16 Principal Components. Even so, the tSNE plot did not seem to be any different. No separation into clusters occurs.

Results

In this section I will attempt to interpret the meaning of my results.

Overall, there does not seem to be any discernible difference between the albums at the top of the list and the albums at the bottom of the list. This is indeed contrary to my initial hypothesis. There were however some noticeable findings in the histogram section.

I would like to mention that two of the observations are about the same underlying audio feature of a track, danceability, even though they arise out of sorting by two different ranking systems. This can perhaps be interpreted as 'successful' albums being at the same time more *danceable* but also having more variety in terms of *danceability*. The second fact is backed by the observation that top albums also have more variety in terms of energy (which could be interpreted as making a song more *danceable*).

In this sense, it is surprising to see that top albums (in terms of Spotify popularity) have, in general, more emotionally negative lyrics. This, coupled with the previous idea, creates the image of a successful album being both danceable, yet also dark in its poetic content.

The very few contrasting characteristics I noticed in the histogram analysis are interesting, I think. However, I do not think they are enough to support my hypothesis that there is a *strong* separation between top and bottom albums.

I consider this to be further proof that music is a very complex individual experience. It is composed of not just the audio elements, but also personal taste and social influence [salganik2006experimental].

5.1 Further research

For further research, I would like to look at various other approaches to exploring this hypothesis or this dataset:

One approach would be to utilize feature engineering to select those audio features that capture more variance, and hopefully more meaning. Indeed, it might be the case that a lot of these features are muddled by noise.

Another avenue would be to obtain a more larger dataset, including rock albums that are not in the "Rolling Stone Top 500". It might be the case that the hypothesis holds when we compare top albums from this list to *mediocre* albums. I would need to answer the question of what a mediocre rock album is.

Even more, I would like to analyze trends per artist / band. Is there a set of features that makes a *Pink Floyd* album a *Pink Floyd* album? In this project I would perhaps need more specific features for songs, or more albums for every artist. It would also be a matter of comparing an artist to the numerically average rock album.

Since music is a temporal experience, it would also make a lot of sense to scrutinize the timeline of each of these features per album. This would be a matter of visualization mostly. Are there trends per artist? Are there trends per subgenre? How about time periods?

Conclusion

In conclusion, I believe this project was quite fruitful. While I did not prove my original hypothesis, I did explore this idea and proved the opposite.

I started this project with the idea that there *is* a strong differentiation between rock albums at the top of the chart and those at the bottom. I developed approaches to visualizing the dataset in a revelatory manner (histograms, tSNE, PCA with tSNE). Of these, only the histograms presented anything interesting.

A great deal of effort was also spent in acquiring the data itself. I had to scrape different APIs, all of them with their own limitations and issues. One of the most challenging steps was collecting the lyrics for the songs, as the lyrics websites had very different approaches to searching by title and artist.

I have also presented several opportunities for furthering this project's question and several other questions relevant to the dataset.

Appendix

6.1 Histograms

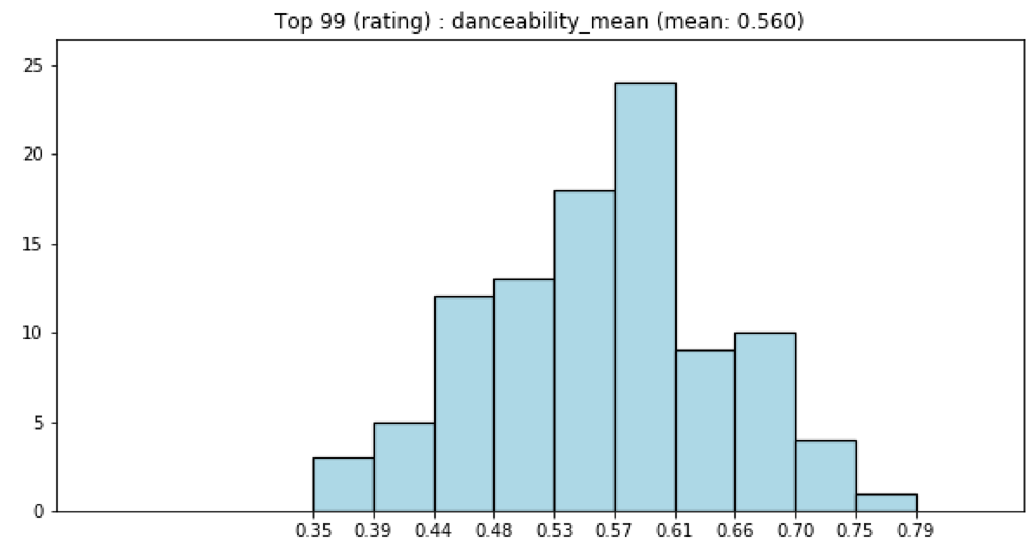

Figure 6.1: rating-top-danceability-mean

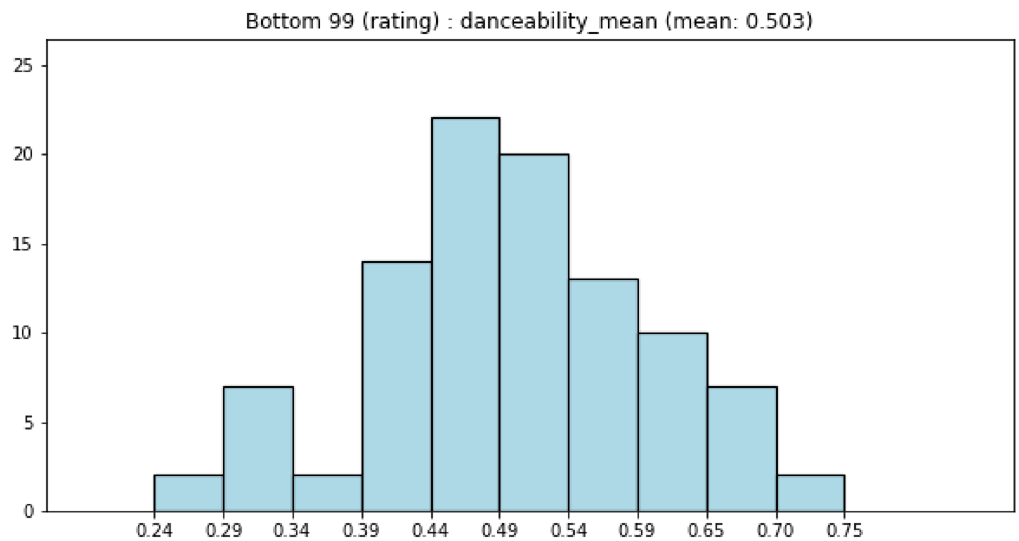

Figure 6.2: rating-bottom-danceability-mean

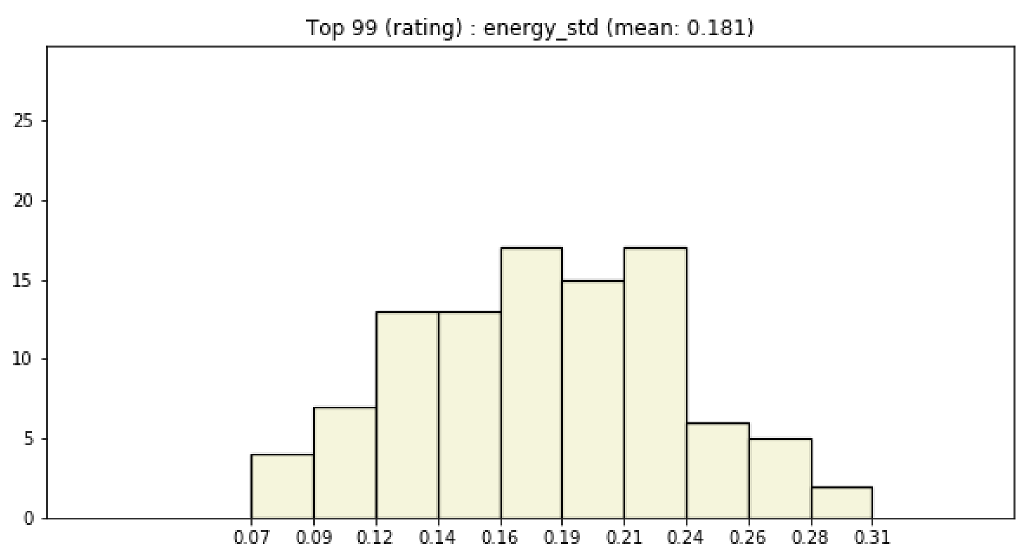

Figure 6.3: rating-top-energy-std

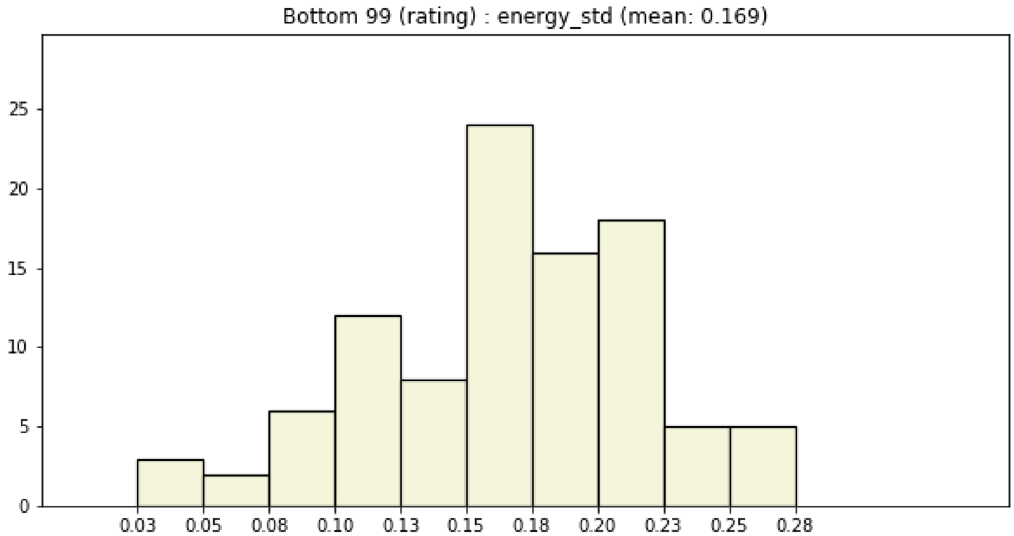

Figure 6.4: rating-bottom-energy-std

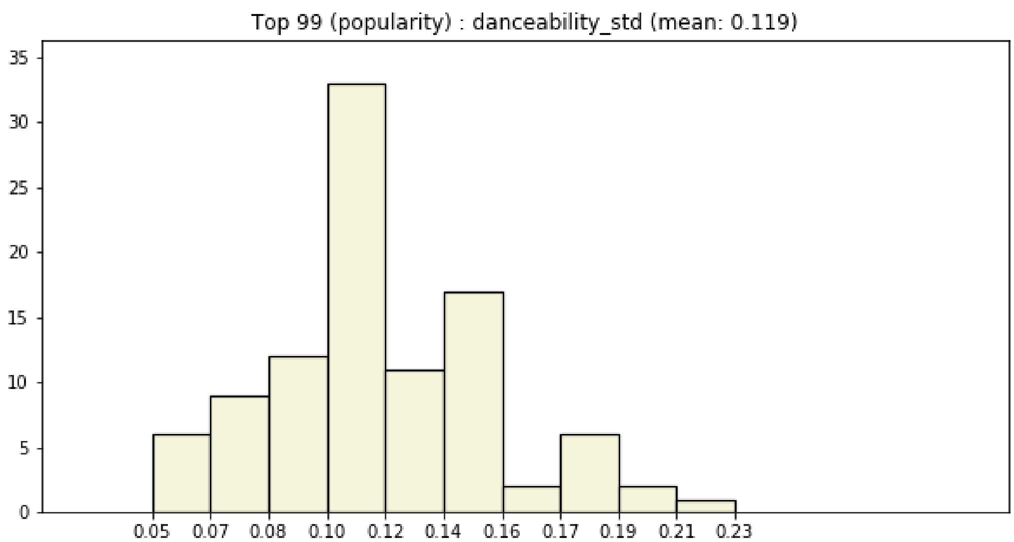

Figure 6.5: popularity-top-danceability-std

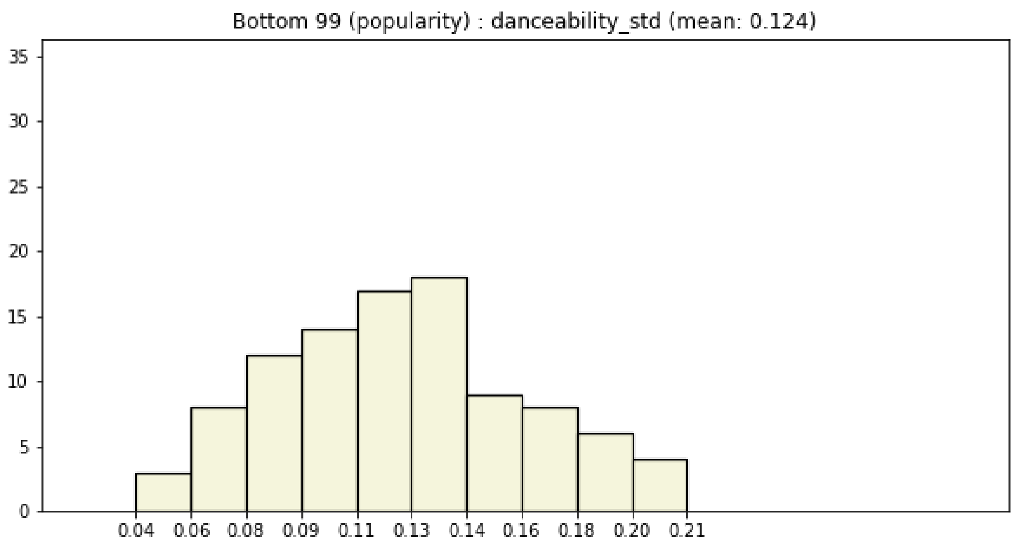

Figure 6.6: popularity-bottom-danceability-std

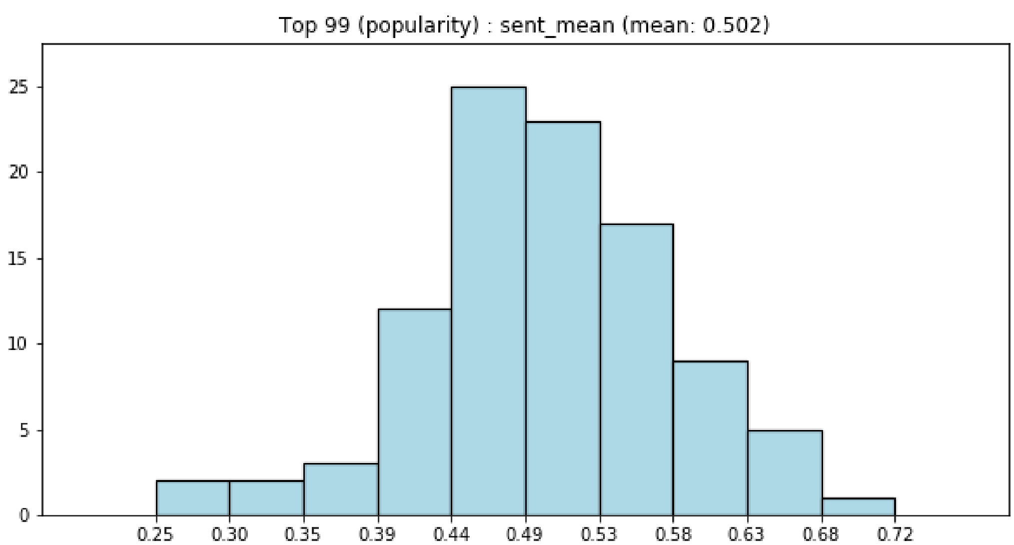

Figure 6.7: popularity-top-sent-mean

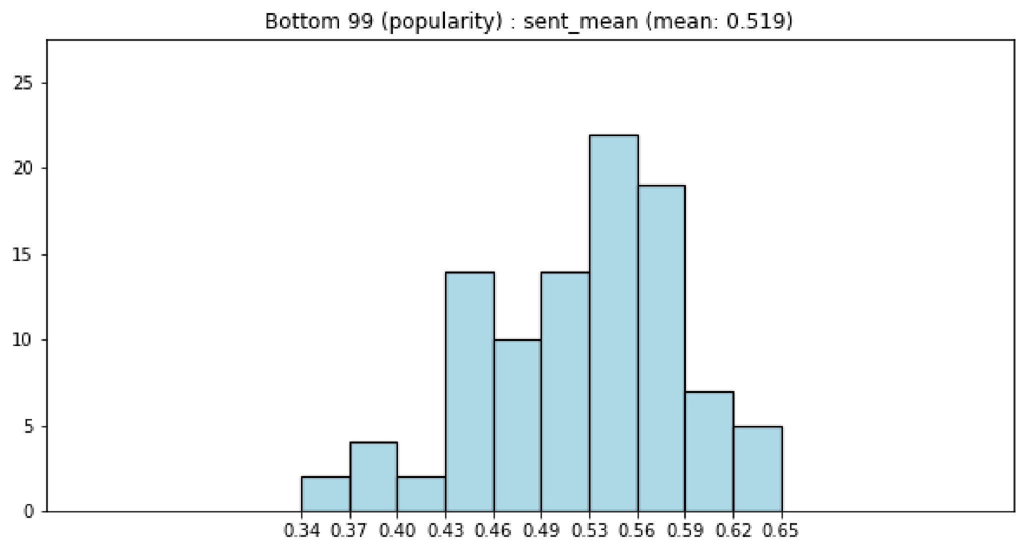

Figure 6.8: popularity-bottom-sent-mean

Bibliography

Dieleman, S., Recommending music on spotify with deep learning. *Sander Dieleman's blog*. Available at: http://benanne.github.io/2014/08/05/spotify-cnns.html.

Discogs, 2018. Discogs API. Available at: https://www.discogs.com/developers/ [Accessed May 7, 2018].

Eck, D. & Montréal, U.D., Learning features from music audio with deep belief networks.

Genius, 2018. Genius API. Available at: https://docs.genius.com/ [Accessed May 25, 2018].

Gibs, 2017. Rolling stone's 500 greatest albums of all time | kaggle. *Kaggle*. Available at: https://www.kaggle.com/notgibs/500-greatest-albums-of-all-time-rolling-stone?sortBy=null&group=dataset.

Google, Google Cloud Language Documentation. Available at: https://cloud.google.com/natural-language/docs/.

LyricWikia, 2018. LyricWikia API. Available at: http://lyrics.wikia.com/wiki/LyricWiki [Accessed May 25, 2018].

Martin, B., 2015. *Listening to the future: The time of progressive rock, 1968-1978*, Open Court. Available at: https://books.google.dk/books?id=UoT6CgAAQBAJ.

mortenhjort GitHub user, 'Time signature' issue on spotify api's github. Available at: https://github.com/spotify/web-api/issues/379.

MusicBrainz, MusicBrainz. *MusicBrainz*. Available at: https://musicbrainz.org/.

Pham, J., Kyauk, E. & Park, E., 2016. Predicting song popularity. *nd): n. pag. Web*, 26.

Quora, 2018. What is the best IMDB-style site for music. Available at: https://www.quora.com/What-is-the-best-IMDB-style-site-for-music-Why [Accessed May 7, 2018].

Rolling Stone, 2012. 500 greatest albums of all time. *Rolling Stone*. Available at: https://www.rollingstone.com/music/lists/500-greatest-albums-of-all-time-20120531.

Salganik, M.J., Dodds, P.S. & Watts, D.J., 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762), pp.854–856.

Spotify, 2018. Spotify API. Available at: https://beta.developer.spotify.com/ [Accessed May 7, 2018].

Statista, 2018a. Music industry revenue in the united states from 2012 to 2021 (in billion u.S. dollars). Available at: www.statista.com/statistics/259980/music-industry-revenue-in-

[Accessed May 29, 2018].

Statista, 2018b. Share of music song consumption* in the united states in 2017, by genre. Available at: www.statista.com/statistics/694862/music-song-consumption-genre/ [Accessed May 29, 2018].