500 Best Albums Ever: Data Collection and Preparation

Eunice Amorim, Henrique Sousa, Henrique Nunes

I. Abstract

This report focuses on data collection and preparation phase of a information processing and retrieval system about the 500 best music albums ever produced according to *Rolling Stone* magazine. Thanks to *Wikipedia*, *Spotify* and *Genius* it was possible to assemble the underlying data set with relevant attributes to identify and search for a song. To achieve this, a pipeline was built considering 3 main steps as follows: collecting, processing and analyzing. As a result of this work, a final data set of 9.21 MiB was gathered.

II. Introduction

Over the years, many tried to discover the best music albums ever released. One of these initiatives is the annual listing of the magazine *Rolling Stone*: the 500 best albums ever released [Mag20]. One reading this list may want to know more about each album, discover how many songs they have, what songs are included, and how good they are. For this reason, this project intends to build a search system which allows the user to search over this ranking with different metrics (album, artist, lyrics, year, ...) and thus satisfying their curiosity. Since there are many editions of this list, the focus was on only one: the 2020 version.

III. Data Sources

After some search, an already assembled data set with all information wanted (list of albums, artist, songs of each album) was not found. For this reason, it was decided to build a customized data set using the following sources:

A. Wikipedia

Wikipedia is a platform of shared knowledge [Wik22]. It was hard to extract all albums from the original magazine article because all albums had a different URL. So, since *Wikipedia* provides the complete list in a easier way to extract, it was decided to use it. In order to be sure about the validity of the list, coherence between *Wikipedia* and *Rolling Stone* was manually checked being confirmed as a reliable source. Regarding the licensing, the list is property of *Rolling Stones* magazine and is publicly available.

B. Spotify

Spotify is a music streaming website used by millions all over the world. It contains 80 million tracks being able to discover the information of all albums (7112 songs). The data access is publicly accessible [Spo22].

C. Genius

One of the main lyrics websites is *Genius*. It has a community of 2 million contributors [Gen09] including verified artists [Gen22b]. As a result, this was a natural choice to gather this information. It proved to be effective, only missing 738 song lyrics (check the Collecting lyrics section collecting lyrics for more details). The song lyrics are publicly accessible but belong to the authors of the song.

D. Alternatives

Other options were considered and tested. The most relevant one was *Musixmatch*. Due to the limitations of the *Musixmatch* free package, it was not used.

IV. Data Collection and Preparation

With the data sources chosen, the next step was gathering the data from said albums in a way that would fit the vision for the search engine. To assemble the data set, the name of the song, the album, the artist, and the lyrics would be imperative to be obtained. Additional metrics would also end up being collected, like the duration of each song, its release year and its album ranking.

A. Pipeline

During the development processes several hurdles had to be overcome, mainly when it comes to the quality of the data. After several iterations it was settled for the pipeline observed in A.1. Each main pipeline component is explained in the following sections.

B. Collecting albums

The entry point input is the *Wikipedia* list of the 500 best albums according to *Rolling Stone* [Wik20], which replicates the one in the magazine's website [Mag20]. However, since there is a single table, the *Wikipedia* page was found to be easier to scrape. The webpage is first stored locally. Afterwards, a python scraper collects the artists and the albums, sorted by ranking order, and stores them in a file.

C. Collecting songs

Next, the file obtained is iterated in the command line, taking the album and artist from each line, which is given as input to a python script that will collect every song of the given album. To do that, the *Spotify API* is used through the *Spotipy* library [Lam14]. The two tools will be addressed as a unit called either *Spotify* or *Spotify API* henceforth. The *Spotify API* allows to search for an album. This process was not trivial

and metrics had to be developed to obtain the closest match (by Levenshtein distance). For example, remastered or deluxe edition might be the only versions available in *Spotify* for a given album, but there might have been an album with the same name from a different artist. After the album is chosen, its tracks are collected, alongside the song duration and the album release date. The script will append all of the relevant information to a single file that will keep the songs featured in the data set. Additionally, the album ranking is also present for each song.

D. Collecting lyrics

Finally, it is retrieved the lyrics from the songs obtained. The lyrics are collected from the *Genius* website [Gen22a]. The file with the tracks will be iterated line by line in the command line to obtain a single tracks' name and artist. The two are put together into a URL which is used to directly access the webpage of the song. If no result is found, the most likely culprit is the name obtained from the *Spotify's API* which is not always canonical. In that case, the name is sanitized and another attempt is performed. If one of the processes is successful, the program will scrape the HTML for the lyrics and the string will become the name of the file where each of the lyrics is stored. This string will also be added to the file which agglomerates the whole collection of songs so that referencing is possible.

Despite a *Genius API* being available, it wasn't reliable, leading to more missing information than the scraping did.

E. Processing

The data set at the end of the collecting process didn't need much processing since such processing was done while collecting. It would be impossible to gather a significant part of the data without sanitizing the song name for example.

However, a separated processing routine was viable: the normalization of the release date. The file obtained previously will be fed to a program whose function is to normalize the song's release date, since the format the *Spotify API* yields is not homogeneous. All of the dates are, therefore, standardized to include just the year of the release.

F. Makefile

To automate the process, a makefile was created to generate the different files. It is structured between the collecting, the processing and the analysis phases. The main file is structured in CSV format and each lyrics is contained in a TXT format.

V. Data Characterization

The assembled data set has been tailored to the needs of the project. As the sources used are reliable and reputed, it's reasonable to assume the correctness of it. Nevertheless, there are still conclusions that can be made.

A. Domain model

Data is made of two parts: the song data and the lyrics data. The song data is a single file with all of the songs available in the data set. It contains the name of the song, the album, the artist, the release year, the duration and the ranking of the album in which it is featured. The lyrics data is composed of one file per song and contains the lyrics for that song.

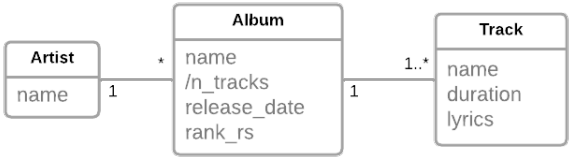

Fig. 1: Domain Model

1) Data fields: The name and lyrics of the track, the name of the album and the name of the artist are all string values. The duration of the track is an integer value with the number of seconds of the song. The date of the album is a integer value with the year of the release date of the album. The rank_rs is the *Rolling Stone* ranking of the album. The n_tracks is a derived attribute with the number of songs in the album.

2) Missing Values: The only missing value are the lyrics for some of the songs. Due to some technical difficulties, gathering lyrics for all tracks was not possible.

3) Size and Volume: The total data has a volume of approximately 9.21 MB. It contains a total of 7112 tracks, 6374 have lyrics. However, some of these lyrics are present in multiple albums and share the same lyrics' file.

B. Descriptive and exploratory statistics

With large data sets like this one, it is important to know the data in order to take advantage of its properties by the time to use it. Therefore, an exploratory analysis was performed. Plots were used to explore its contents and discover some patterns that allows a better understanding of it.

The album duration and the album mean track duration follow a similar distribution (A.2 and A.3). The outliers, especially, are similar. However, it cannot be concluded that this is due to the same album, since a long mean song duration could just imply that an album is made of a single long song, for example. It is possible to conclude that most albums are about 50 minutes in length and tracks are mostly under 4 minutes long which is consistent to the recommended 3-5 min duration.

The majority of the albums have less than 20 songs. There are a few with more than that and some outliers with even more than 40 songs (A.4). It could be the case that the albums with this many songs are a result of a extended version of the album found by *Spotify*.

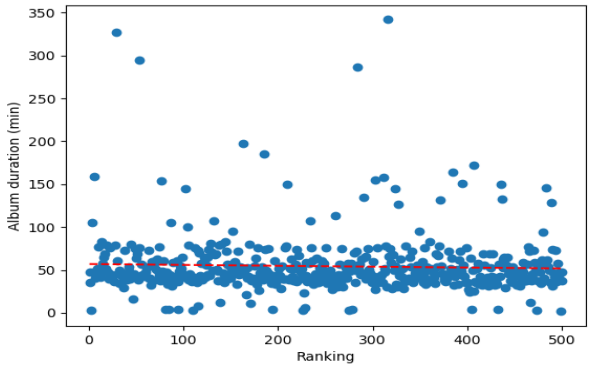

Fig. 2: Album duration by ranking

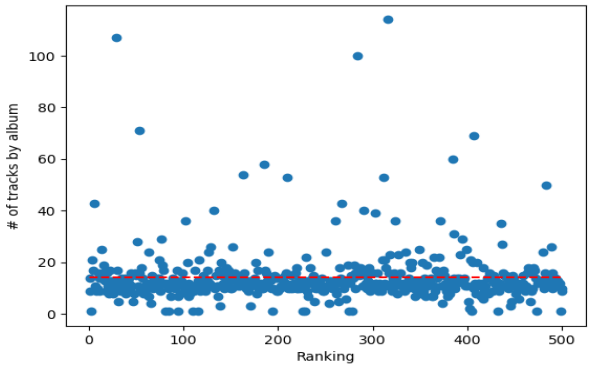

Fig. 3: Album number of songs by ranking

By plotting the linear regression that expresses the correlation between the above metrics and the ranking of the respective album, there are some conclusions to be made. Longer albums and albums with an higher track mean duration have a better ranking (Fig. 2 and A.5). It is impossible to determine if this is a causal relationship and, even if it was so, it would only make the album perform marginally better. Additionally, the number of tracks of an album appears to be mostly irrelevant to its ranking (Fig. 3).

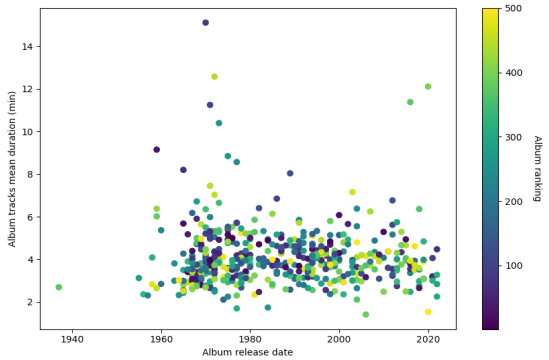

Fig. 4: Album ranking distributed by release date and mean track duration

The graphic in Fig. 4 shows the distribution of albums according to its release date and the mean time duration of its tracks. Also, the markers in the plot are colored following a scale given from album ranking. Analysing the plot, it is possible to conclude that the best albums of the 500 are more agglomerated between the 60's and 90's and between 3 and 5 minutes per track, which was expected.

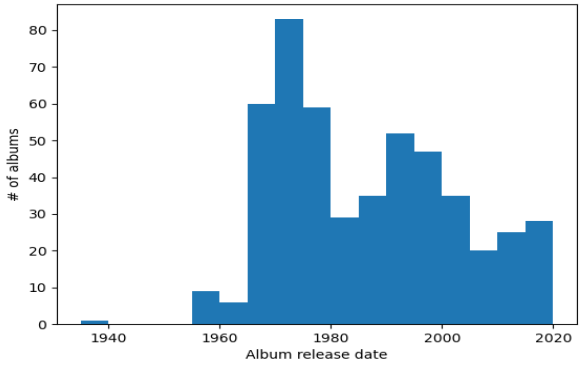

Fig. 5: Album distribution by year

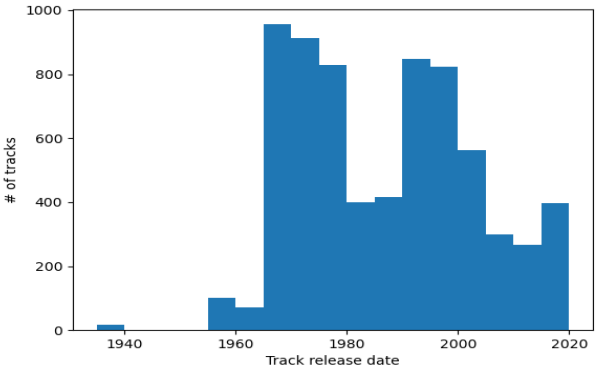

Fig. 6: Song distribution by year

The distributions for the album release date (Fig. 5) and for the track release date (Fig. 6) follow a similar distribution. This is to be expected, as more albums in a given year means, generally, more songs for that same year.

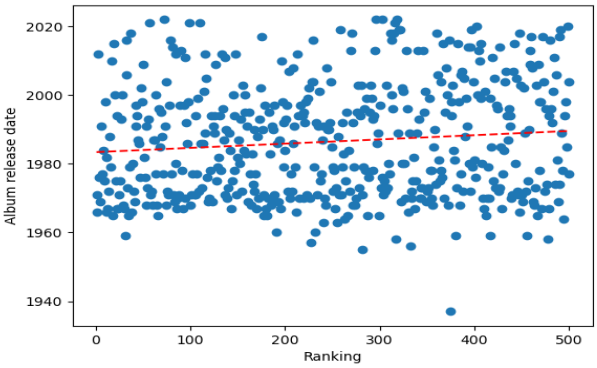

Fig. 7: Album release date by ranking

The graph in Fig. 7 plots the linear regression correlating the ranking of an album and its release date. Though small, it can be seen that an older album is associated with a better ranking. One hypothesis that explains this behavior is the fact that more recent music is, in general, worse. Of course, this fails to account for many other factors, for example, the very fact that ranking albums can be a very subjective task, and that the people making such ranking have inherent bias.

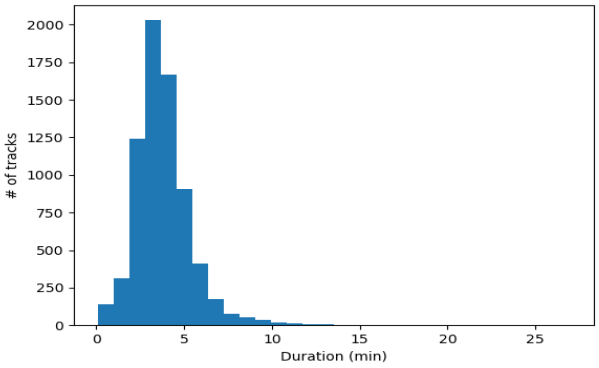

Fig. 8: Song distribution by duration

The song distribution by duration (Fig. 8) is similar to the album distribution by mean track duration. It is not surprising that the grouped data replicates the pattern of the individual data. As it can be seen, most songs have a duration of less than 5 minutes.

The graphic in A.6 presents an estimate of the number of tracks for which lyrics have been found. Despite being quite large in absolute number, the number of tracks for which the lyrics were not obtained is significant, but not crippling in terms of the effectiveness of the data.

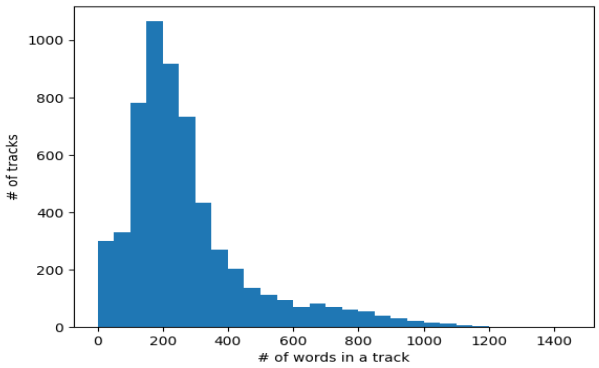

Fig. 9: Number of words by track

Most tracks tend to have a number of words smaller than 400 (Fig. 9). In Fig. 10, it is also possible to see that a considerable number of tracks, about 200, have no words at all. Those are instrumental only tracks.

VI. Prospective Search Tasks

With all the information collected, it is important to understand which queries are relevant for the system. The final

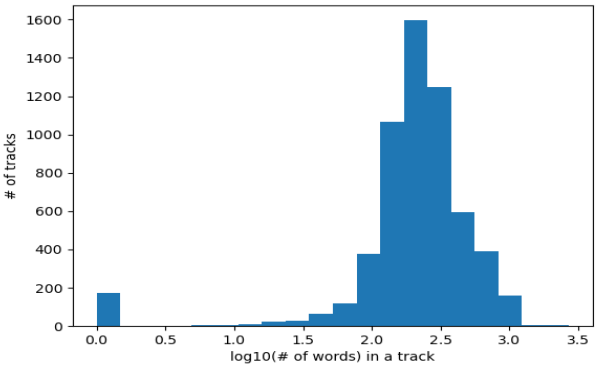

Fig. 10: Number of words by track (logarithmic scale)

search system should be able to help users fulfilling their information needs. Some of them may include:

- Search for an album of a given artist and year
- Search for albums considering its ranking
- Search for a song using its name or part of it
- Search for a song using part of its lyrics
- Search for songs in a range of duration

VII. Conclusion

During the course of the development process, it has become evident that the process of creating a data set from the very inception is not a trivial task. There are a lot of factors that determine the quality of the final data. Since the project relies on gathering data from outsides sources, as opposed to collecting points of data ourselves, the reliance in external entities makes the whole process more prone to inconsistencies as was the case. A good data collection pipeline is one that takes into account all of the susceptibilities of the composing parts and works around those. The pipeline that has been developed attempts to achieve such a performance after a progressive refinement of its data collection processes.

References

[Gen09] Genius. *About Genius*. 2009. URL: https://genius.com/Genius-about-genius-annotated (visited on 10/12/2022).

[Lam14] Paul Lamere. *Welcome to Spotipy!* 2014. URL: https://spotipy.readthedocs.io/en/master/# (visited on 10/12/2022).

[Mag20] Rolling Stone Magazine. *The 500 Greatest Albums of All Time*. 2020. URL: https://www.rollingstone.com/music/music-lists/best-albums-of-all-time-1062063/ (visited on 10/12/2022).

[Wik20] Wikipedia. *Wikipedia:WikiProject Albums/500*. 2020. URL: https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Albums/500 (visited on 10/12/2022).

[Gen22a] Genius. *Genius Home Page*. 2022. URL: https://genius.com (visited on 10/12/2022).

[Gen22b] Genius. *The Genius verified artist*. 2022. URL: https://genius.com/verified-artists (visited on 10/12/2022).

[Spo22] Spotify. *About Spotify*. 2022. URL: https://newsroom.spotify.com/company-info/ (visited on 10/12/2022).

[Wik22] Wikipedia. *Wikipedia:About*. 2022. URL: https://en.wikipedia.org/wiki/Wikipedia:About (visited on 10/12/2022).

APPENDIX

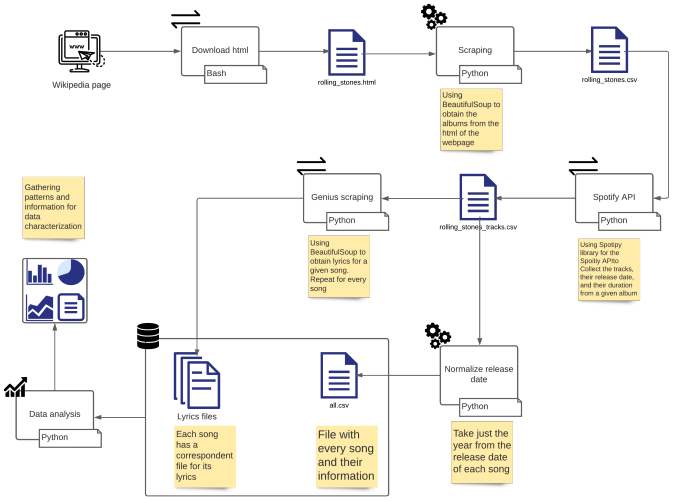

Fig. A.1: Pipeline to collect and process data

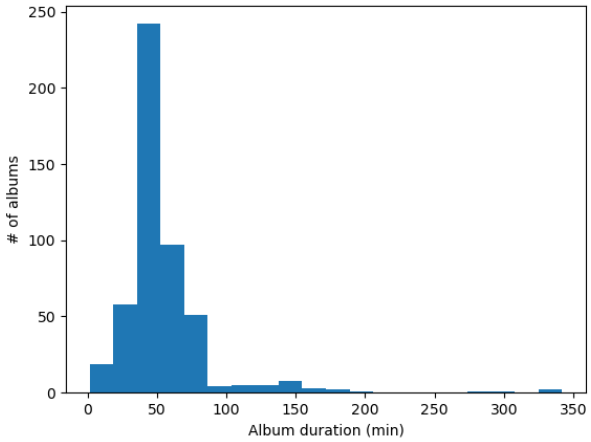

Fig. A.2: Album distribution by duration

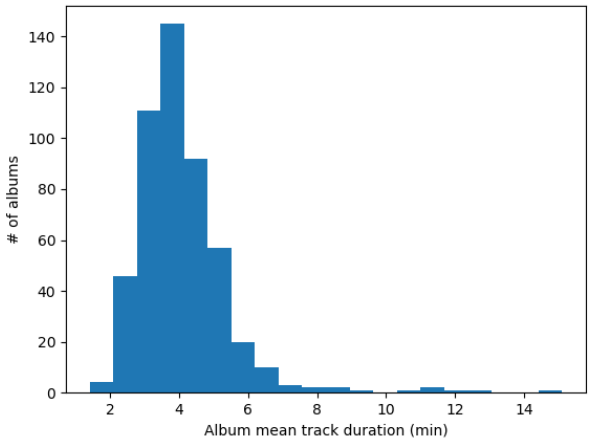

Fig. A.3: Album distribution by song duration

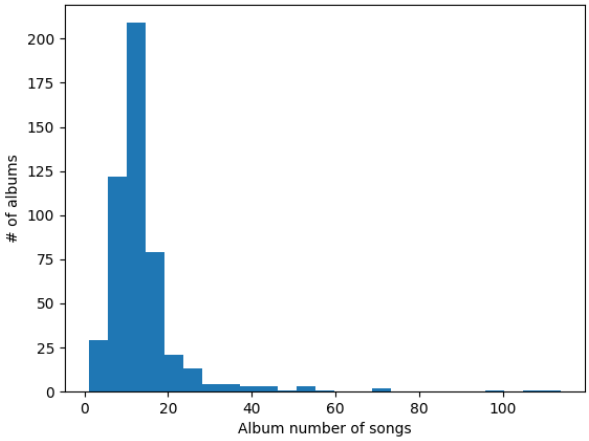

Fig. A.4: Album distribution by number of songs

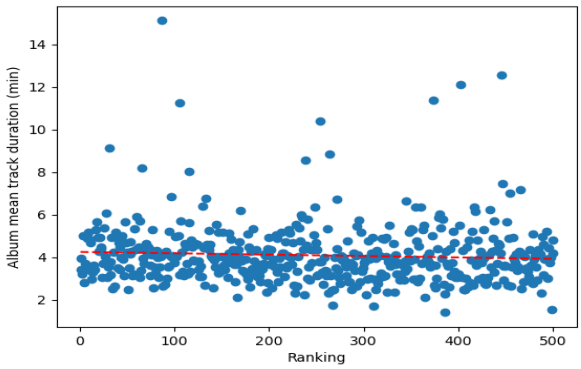

Fig. A.5: Album mean song duration by ranking

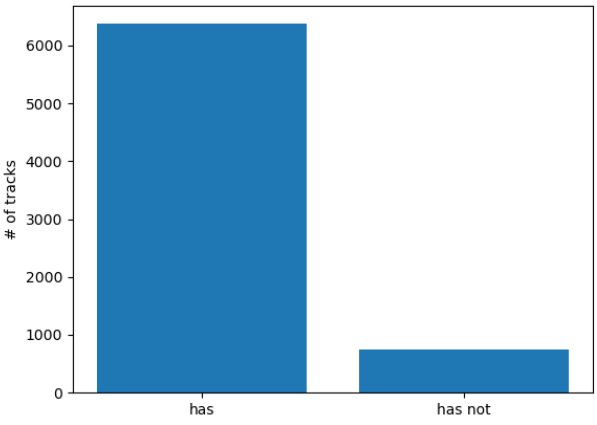

Fig. A.6: Lyrics existence