Lyrics Entropy through the decades

?Annonymous?

December 16, 2021

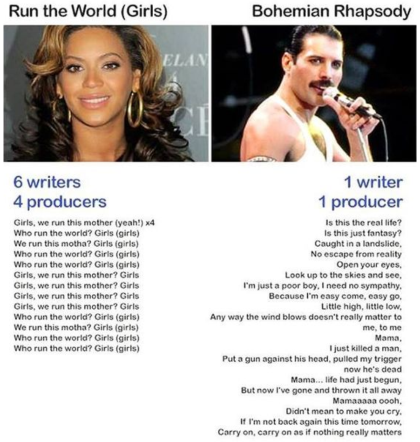
1 Introduction

1.1 Motivation

Music is my life, and that is not an understatement. My day starts and ends with music. I listen to Dylan while I'm preparing my breakfast and then switch to *Reggaeton* to make sure that I have something to dance to while in the shower. When I grab my phone for lunch, I'm probably going to comment with my dad about a recently released album or I will share one of my playlists with a friend that has just asked me for music recommendations.

In the afternoons, I try to always take 30-minute breaks to play guitar, which recently are marked by failed attempts of me playing the Stairway to Heaven acoustic intro; or finally nailing Blackbird or Wish You Were Here (after months of practice). If it is a weekend, I will most likely finish my day by going to a techno club to listen to some spacial techno beats for a while, even if it is just by myself.

Since I listen to music all the time, I also think about it frequently. Even though it is "just" music, sometimes my thoughts get very complex and philosophical. I like to think of music as a tool to understand culture, history and emotions. While having these melodic thoughts, I once came across a meme that made me wonder about the evolution of lyrics in the recent decades (see Fig. 1)



Girls, we run this mother (yeah!) x4
Who run the world? Girls (girls)
We run this motha? Girls (girls)
Who run the world? Girls (girls)
Who run the world? Girls (girls)
Girls, we run this mother? Girls
Girls, we run this mother? Girls
Girls, we run this mother? Girls
Girls, we run this mother? Girls
Who run the world? Girls (girls)
We run this motha? Girls (girls)
Who run the world? Girls (girls)
Who run the world? Girls (girls)

Is this the real life?
Is this just fantasy?
Caught in a landslide,
No escape from reality
Open your eyes,
Look up to the skies and see,
I'm just a poor boy, I need no sympathy,
Because I'm easy come, easy go,
Little high, little low,
Any way the wind blows doesn't really matter to
me, to me
Mama,
I just killed a man,
Put a gun against his head, pulled my trigger
now he's dead
Mama... life had just begun,
But now I've gone and thrown it all away
Mamaaaaa oooh,
Didn't mean to make you cry,
If I'm not back again this time tomorrow,
Carry on, carry on as if nothing really matters

Figure 1: Number of writers and producers of Beyonce's Run the World compared with Queen's Bohemian Rhapsody

Obviously, we've learned that this meme is not statistically significant since it is just doing selective sampling to prove a point; but still, it planted an idea in my analytical-music-lover head. Since then, I have done small approaches to the idea by using Spotify's API in the past but could not find the time to arrive to relevant conclusions of my analysis, until now...

My project, therefore, will focus on multivariate statistics applied to song features (available in Spotify's API) with the important addition of lyrics information (retrieved with Genius API [1]). Specially, I'm interested in finding a way to assign a "lyrics diversity" score to each song in my dataset, and then

[1] Genius is the most popular website to find song lyrics and annotations. They provide an API to easily access song lyrics from programming environments (https://pypi.org/project/lyricsgenius/)

use this diversity variable as the main feature to analyze and describe throughout the project. My work won't be too focused on supervised learning but more in visualization and unsupervised learning techniques, as the methodology map shows (see Fig. 2).

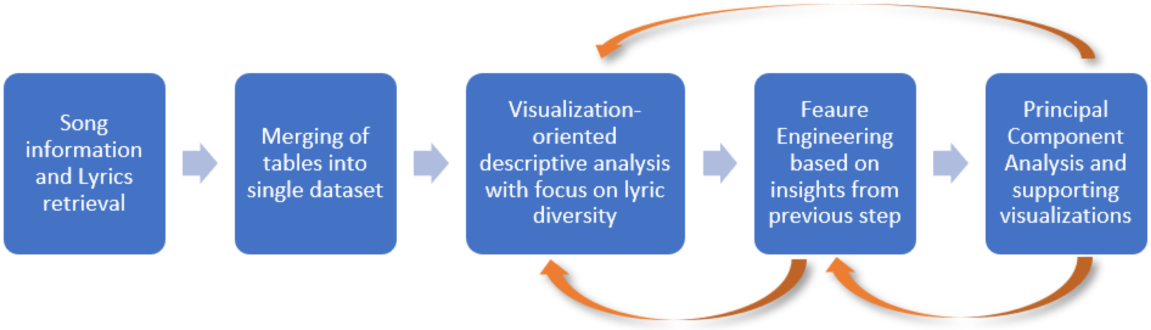

Song information and Lyrics retrieval
Merging of tables into single dataset
Visualization-oriented descriptive analysis with focus on lyric diversity
Feaure Engineering based on insights from previous step
Principal Component Analysis and supporting visualizations

Figure 2: Project Methodology

I want to provide the reader with plenty of visualizations that help to answer questions such as "How have lyrics diversity changed since the 70s?", "Have all genres followed the same patterns or not necessarily?", "Does song diversity have an influence in a song's success, and if so, does it remain constant through decades and genres?".

As a wrap-up, I want to switch my attention to the artists; and, by running a PCA in a artist-grouped dataset, I intend to decipher the interactions that lyric diversity has with other musical features to be able to generate some recommendations for future success and catchier songs (apologies in advance).

1.2 A Very Brief Literature Review

In 1975, amid a merger between Warner Records and Polygon Records, sociologists Peterson and Berger [1] argued that an increase in market concentration would lead to a decrease in the diversity of products offered in the music recording industry.

In 1996, Alexander [2] revisited this idea from an entropy standpoint, as he argued and proved that entropy had increased since the 1970s, meaning a decrease in music diversity (measured in terms of musical elements detected in music sheets). Because Alexander's data was limited, he only analyzed each year's top 40 songs according to the charts, which amounts for no more than 1000 songs analyzed in total. Also because of this, he could not carry his research forwards into genre-dependent insights.

More recently, in 2017, Stanford University researcher, Tsaptsinos [3] performed genre classification by using song lyrics as inputs and Hierarchical Attention Networks (HAN) as the selected deep learning technique. He argued that HAN have the potential to learn the structures and hierarchies present in song lyrics. However, I find that his paper is lacking in discussing implications for the industry instead of just focusing on presenting the accuracy results and the neural network's weights.

This year, in Nature, Bello & Garcia [4] revisited the idea of cultural divergence in popular music, arguing that since 2017, with the popularization of streaming platforms such as Spotify, an upward trend in music consumption diversity has been recorded worldwide.

My work will attempt to cover the issue of data availability that Alexander faced in 1996, allowing me to reach more robust and valid conclusions, and also elaborate more relevant insights compared to Traptsinos approach. Additionally, I'm interested in seeing whether my findings validate or cast doubts to the recent conclusions reported by Bello & Garcia in their 2021 paper.

2 Data

2.1 Data Retrieval and Preparation

The first step of the project was making the dataset a reality. It is true that it is possible to find Spotify datasets in Kaggle but I noticed that these sets don't have the column "genre" in them. "Genre", in Spotify's database, is not associated to a song but to an artist, so to paste the genre one has to do a bit of data engineering. I could have simplified my project by removing genre from the analysis, but my musical gut instinct told me that it would be a relevant variable to group the data, as it will become evident in the next sections. I also think that building the dataset from scratch allowed me to generate a larger collection with better representativeness of music sources.

I wrote a Python code to query Spotify's database using the library SpotiPy[2] by scraping a list of 23 public playlists. I paid attention to select diverse playlists in terms of decades and genres so the dataset would be as representative as possible. See Appendix A to see the names and sizes of the playlists consulted. Finally, I had $24,036$ tracks with their information. I then used the API to collect the information about the list of artists found in those songs, including the "genre" associated to each artists. To avoid contradictions in my data, I decided to keep songs that only had 1 artist (solo or band) as author.

As per the lyrics, I used the columns of song name and artist already in my table to perform a search in Genius's database using a similar querying API[3]. In fact, with a single line of code one can retrieve the lyrics to a song, which I considered very convenient. After both of the scrapings were done, I wrote the tables as CSVs and loaded them from R to do the rest of the project.

Once in R, after merging the songs, lyrics and artists datasets, I shifted my attention towards the lyrics preprocessing. NLP in R was a new world for me but after doing some research and exploring some libraries I found the way to perform all the tasks that I had in mind in terms of text normalization. Overall, I'm pretty satisfied with the resulting NLP preprocessing pipeline which is simple but robust for this project's goals. It includes removal of line jumps, extra spaces, punctuation and stop words; also perform replacement of contractions and lower case mutation. Also relevant to comment is that running this pipeline through the whole dataset takes no more than 10 minutes on a regular personal laptop.

With the lyrics processed, I proceeded to create basic features based on said variable: number of words in a song (n-words), number of unique words in a song (n-unique) and then both of those 2 new metrics divided by the minutes of a song to have both words-per-minute (WPM) and uniques-per-minute (UPM). All of these derived features will later provide the foundation to calculate more complex ones to describe and compare the repetitiveness of lyrics in a tune.

Please, find in Appendix B, a sample of 5 rows of the resulting dataset before modeling.

2.2 Data Description

2.2.1 Univariate Description

We start our graphical based analysis by reviewing some univariate distributions. First, I wanted to get a sense of the distribution of words and unique words in the dataset. As Fig. 3 shows, the distributions are very similar in terms of deviation and *kurtosis*. This gave me a hint that these 2 metrics were potentially safe to combine into a single one.

I took the ratio of words and unique words by minute and again saw the distributions were very close to a normal one (see Fig. 4. This metric felt more "standarized" for me since it adjusts for very long or very short songs where a lot or very few lyrics can be found but we still want to analyze somehow in the context of lyric diversity.

I also wanted to understand the distribution of song duration that I had in the dataset. Did I have mostly short songs? Long songs? Did I have a single peak or perhaps 2 or more peaks in my distributions? Since I am taking logarithm, the most popular duration seems to be around 1.2 shown in Fig. 5, which taken back to the true number gives us $e^{1.2} = 3.3$ which is pretty close to the duration of songs we are all used to.

[2] https://spotipy.readthedocs.io/en/2.19.0/

[3] https://pypi.org/project/lyricsgenius/

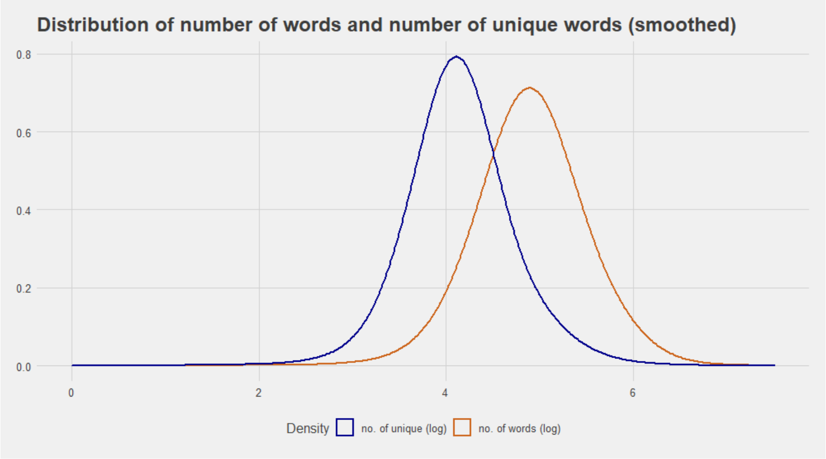


Figure 3: Distribution of number of words and number of unique words (smoothed)

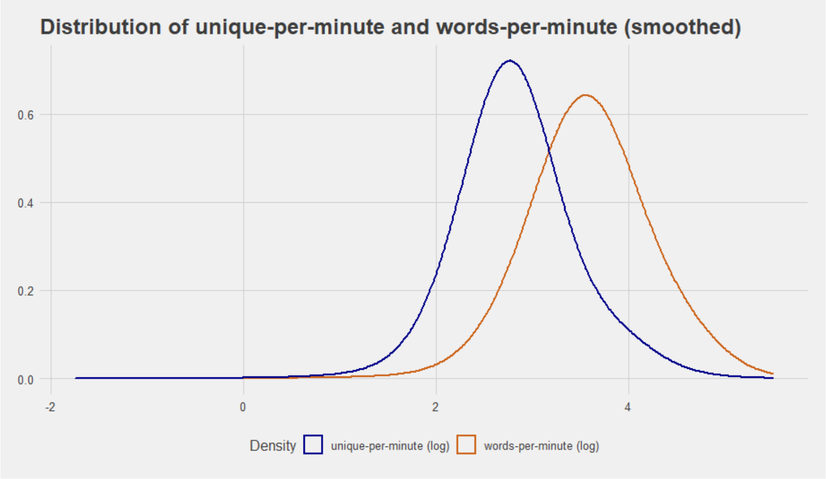


Figure 4: Distribution of unique-per-minute and words-per-minute (smoothed)

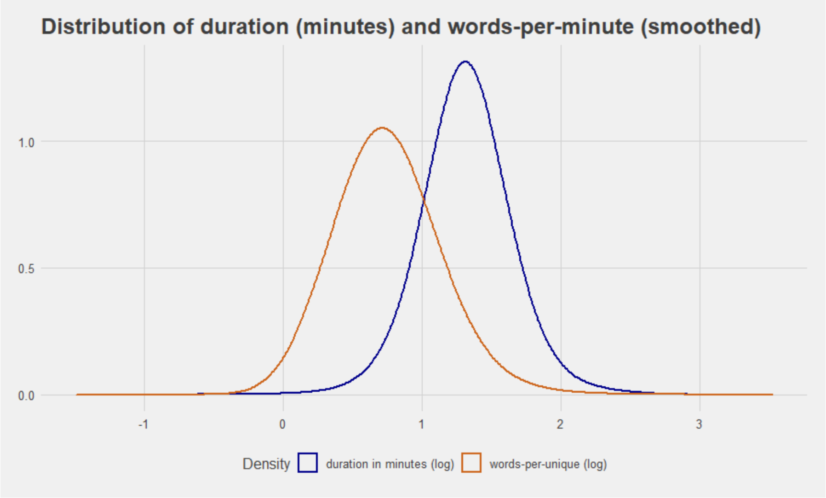


Figure 5: Distribution of unique-per-minute and words-per-minute (smoothed)

Finally, since my analysis will mostly be based on decades and genres, I want to make sure that I have similar amounts of each of this cuts, or at least a minimum amount. In Fig. 6 and Fig. 7 I plotted histograms for decades and genres in the dataset. As one can see, we have 1000 or more samples for each decade or for each genre, which ensures that our analysis will be robust.

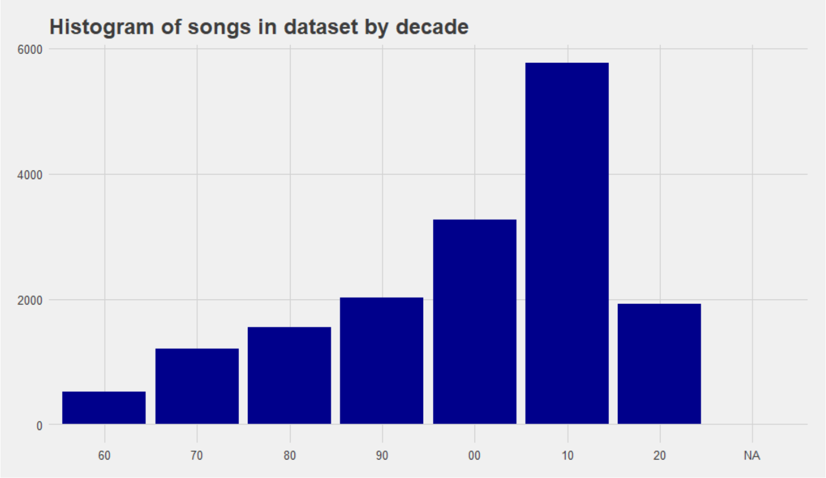


Figure 6: Histogram of songs by decade of release

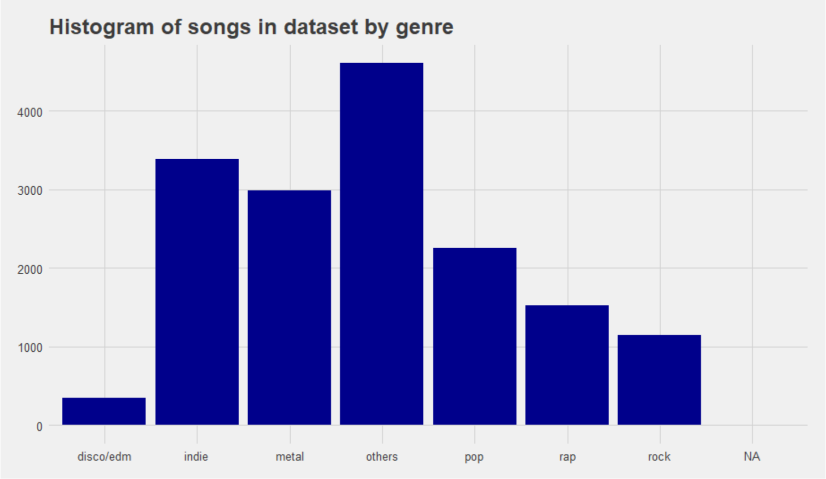


Figure 7: Histogram of songs by main genre

After guaranteeing that I had a diverse and representative enough set, I proceeded to start analyzing correlations and patterns in the next Section 2.2.2

2.2.2 Multivariate Description

Starting our reasoning using a simple metric as song duration, one can notice in Fig. 8 that there is a positive trend, as one would expect, between song popularity and minutes but this only occurs until around the 3 and a half minutes mark. Afterwards, the popularity seems to drop a little and then hit a plateau. More generally, the graph tells us that it is difficult to tell anything about a song's popularity based on its duration after leaving the 3m30s region; therefore, this is not a metric I would really use to compare genres and decades.

Since we saw in section 2.2.1 that the metrics WPM and UPM were normally distributed, we will plot their relationship with song popularity in Fig. 10. Here, we start detecting a very interesting trend. Both metrics have a similar behavior against song popularity. If we focus on the orange curve, we can interpret it by thinking that as a song becomes excessively "speechy", it is less likely for people to listen to it. What caught my attention is that the UPM metric, in the blue curve follows the same pattern, and it becomes detrimental for a song's commercial success even "earlier". In other words,

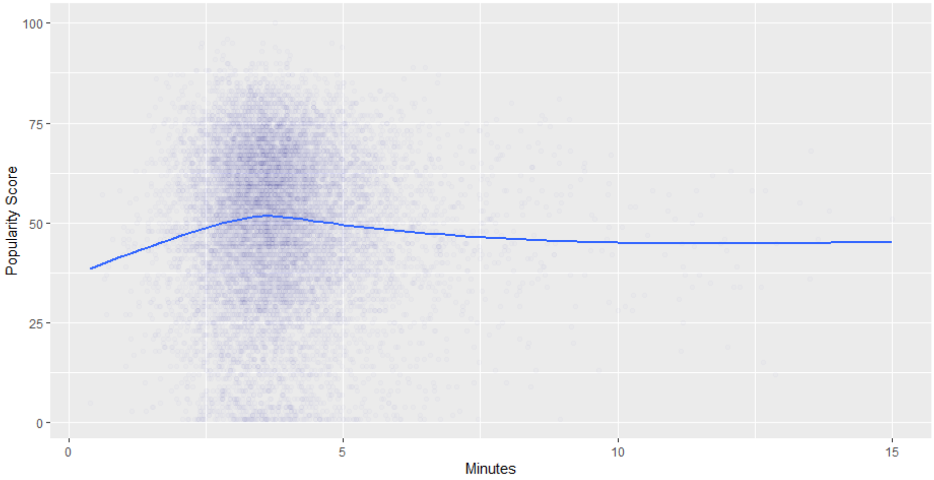


Figure 8: Correlation betweem song minutes and popularity

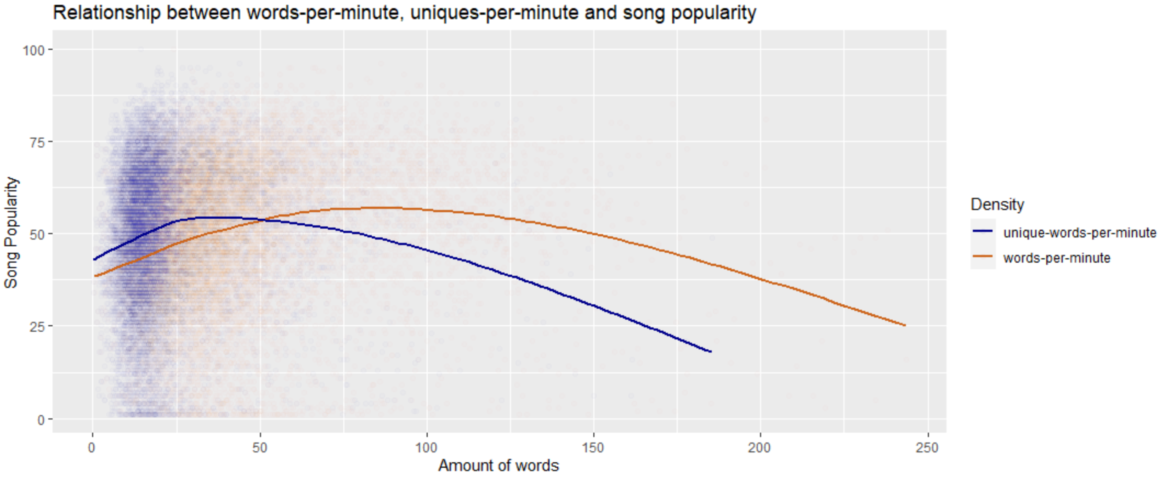


Figure 9: Correlation between words-per-minute, unique-per-minute and Popularity Score

people can tolerate wordy songs as long as some of the vocabulary repeats, but they grow tired of the song faster if each word is different than the previously sang ones. It seems like repetition is something our brains prefer, either consciously or unconsciously.

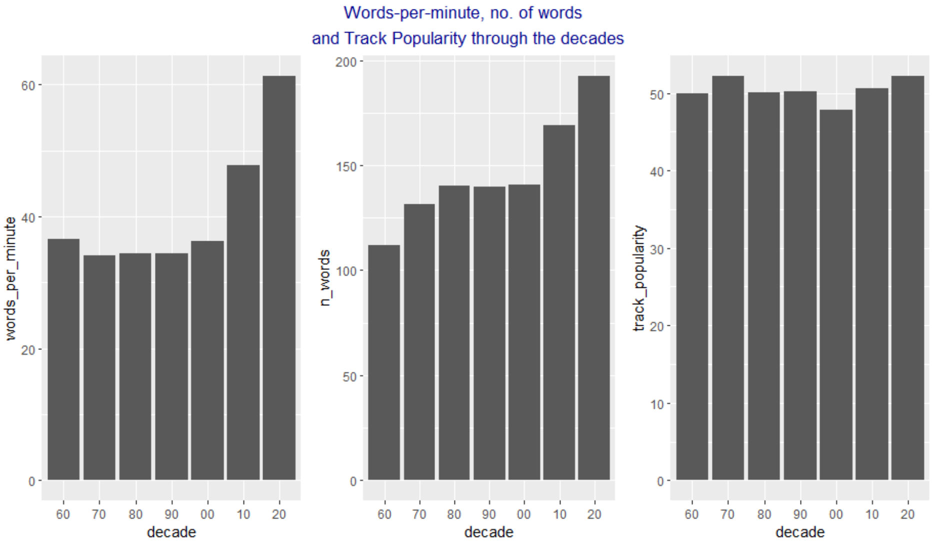


Figure 10: WPM, No. of words and track popularity

Now that we know that WPM has certain predictive power of a song's popularity, let's understand if it has always been like that. First I wanted to understand how this metric has been evolving in the last decades (from the 60s to the 2020s). I noticed that WPM is actually following an upward trend, driven by an increase in total number of words in the last 2 decades. However, track popularity has reached similar average levels each decade for the past 60 years, so WPM is still not the ideal metric that allows us to understand the trends of each decade.

3 Analytics of Lyrics Entropy

At this stage of the project, we want to focus the analysis on the lyrics feature. We already introduced metrics such as n-words and n-unique but we are still lacking a key metric to give each song a score of lyrics "diversity". The most simple and straight-forward one is *words-per-unique (WPU)* which is a simple ratio between the number of words existing in a whole song and the unique words in that same piece. If we consider a song i, then its WPU would be calculated the following way.

$$WPU_i = \sum(words_i)/\sum(uniques_i)$$

For example, for a repetitive song such as Bon Jovi's "Bad Medicine", the WPU is of 2.8 whereas for a song like Pink Floyd's "Breathe" with barely repeating words, the WPU takes a value of 1.2. Therefore, one can think of this number as an indicator of how likely it is that the next word in a song will be a term that has already been mentioned before in that same track.

The previous idea resonated in my head with the concept of entropy in information theory, which states that the "entropy of a random variable is the average level of information, surprise, or uncertainty inherent in the variable's possible outcomes"[4]. In fact, many alternatives have been proposed to measure entropy in text documents. For instance, Shannon's Entropy [5] is a popular framework to summarize documents into entropy scores. However, after some testing, I realized that the WPU method gave foot for better and easier to interpret visualizations. Henceforward, the word entropy, or "lyric diversity"will be used to refer to the WPU metric indistinctly.

My main finding of this section, as will be reflected through the upcoming plots is that even though lyrics diversity has not changed a lot through the decades, the way people appreciate it has indeed changed each decade, even in a genre-dependent matter.

[4] https://en.wikipedia.org/wiki/Entropy_(information_theory)

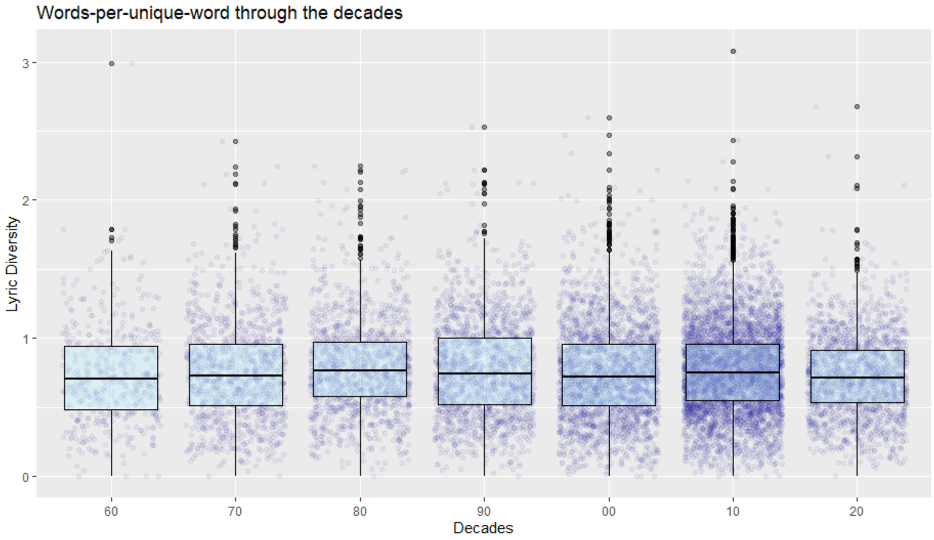


Figure 11: Words-per-unique-words (WPU) distribution each decade

In Fig. 11 we see that the WPU measurment remains relatively constant since the 60s. This means that the generic song of each decade has been, generally speaking, producing 2.2 words per each unique word in the lyrics. From the hockey-stick shape of the curve in Fig. 12 we can understand that users tend to appreciate very much as a song is approaching a WPU of 2.5, but from there, there is a steady decrease in a song's popularity if it keeps getting more and more repetitive. In my experience, this makes sense. I obviously enjoy songs that are easy to learn because of its lyrics consistency, but I also want to feel like the authors made a decent work in writing a poetic message in their lyrics. However, has this been the way listener behave the whole time?

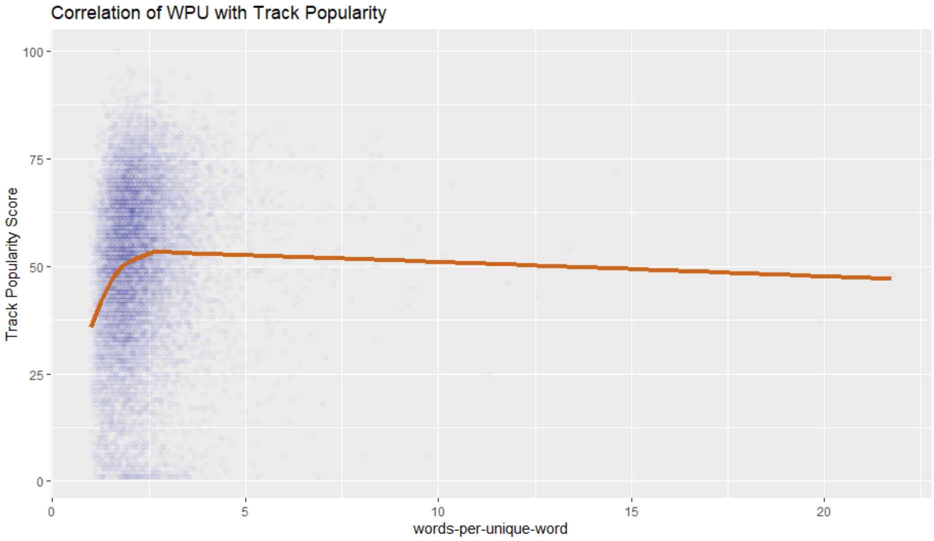


Figure 12: Correlation of WPU with Track Popularity

In Figs. 13 and 14 I plot the same correlation as before but split by decade. In the latter, I keep only the 70s, 80s and 2020s, which are more relevant for my argument. The reader might be thrilled to notice how different the fans behavior has been in these 3 decades! In the 70s, as a song grew in WPU (became more repetitive), the listeners tended to dislike the song and make it unpopular, shown in the red curve. It is as if listeners back then had a "demanding" ear and wanted to hear profound and dense lyrics. Then we have the 80s, which were a sort of transition era for music. From this decade, we can see a fairly flat blue curve, which implies that people were a bit indifferent to the WPU. There were so

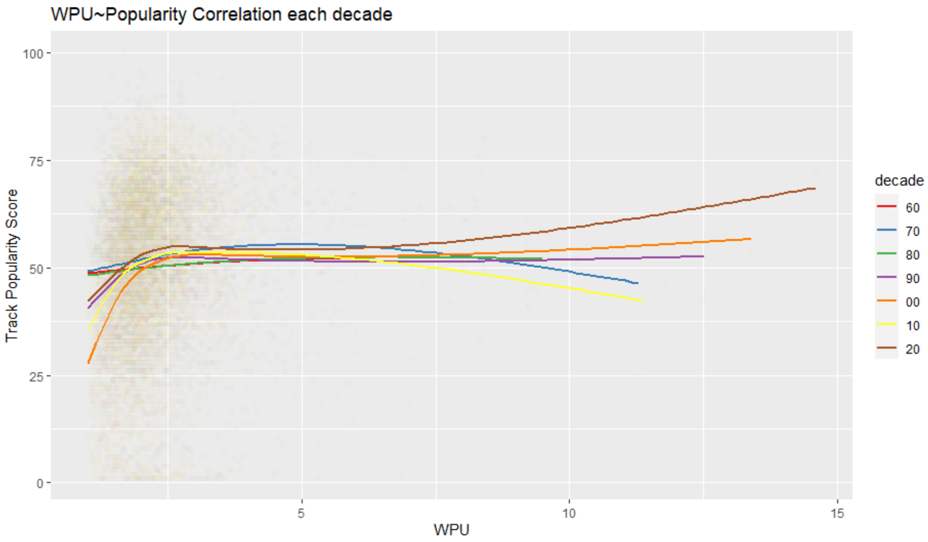


Figure 13: Correlation of WPU with Track Popularity, by decade

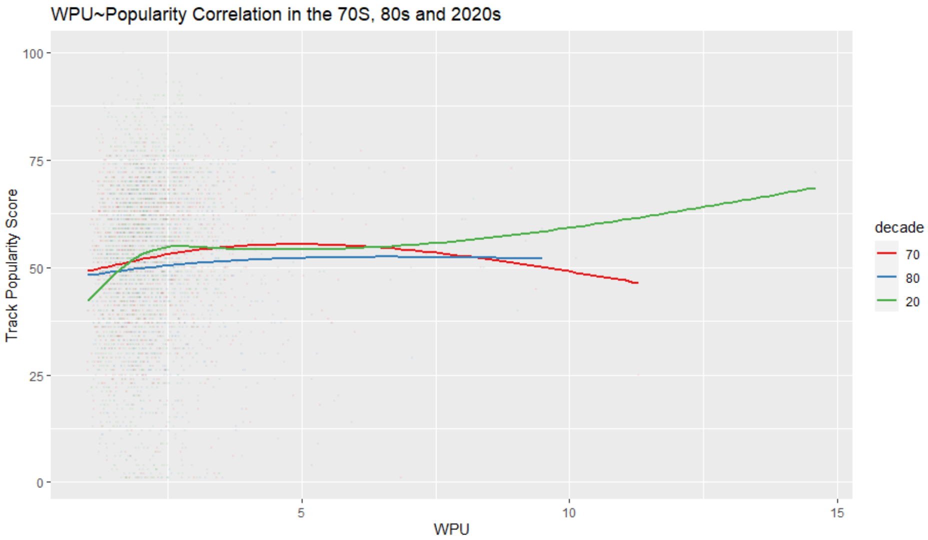


Figure 14: Correlation of WPU with Track Popularity: a tale of three decades

many new elements appearing in the songs that the lyrics stayed in the background. Fast forward to 2020, we find a pronounced vertical green curve which is letting us know that listeners nowadays tend to appreciate songs that have very repetitive lyrics with very little real content. **To me, this plot if one of the most shocking findings of my work and I really like how just 3 curves can tell a clear story of music preference from the last 5 decades.**

The last plot made it clear how, in general terms, some decades responded positively to lyric entropy, others responded negatively and other were too busy to even care. However, if we were too focus on only the each decade's hits, is this still the story? Or does something change? To answer this I decided to split the songs in 3 equally sized groups according to popularity and label them as not popular, average or very popular. Then I proceeded to create a line chart of the average WPU through the decades for each of these 3 groups. The result, shown in Fig. 15 also brings about a pretty sharp discovery. In all decades, the top third songs by popularity (A.K.A. the hits), represented in the blue curve, have had a higher than average WPU than the 66% rest of the songs of its corresponding time span. Even in decades like the 70s, in which we already saw that the relationship between these variables was negative, if we focus on the hit tunes, the gap is still in favor of higher WPU values.

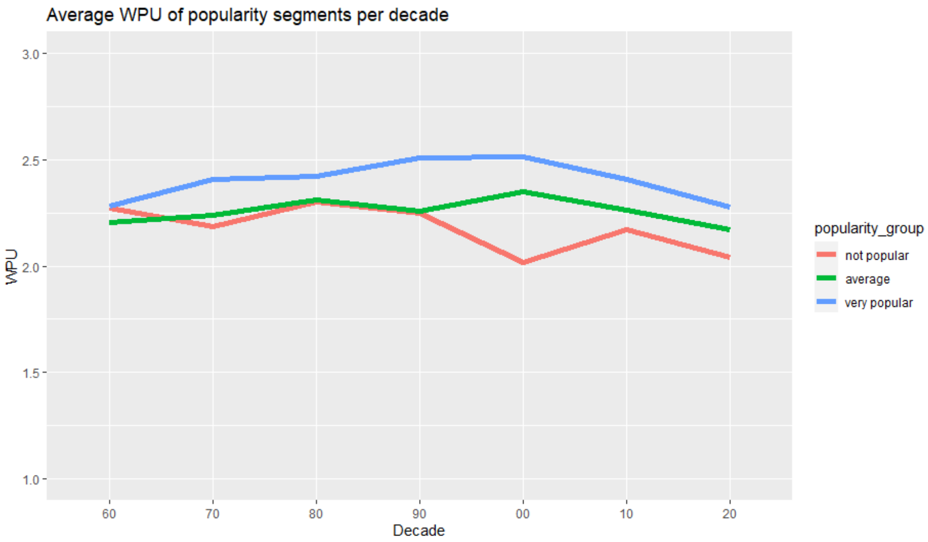


Figure 15: Average WPU of popularity groups per decade

So, does this mean that it is always preferable to produce a higher-than-average WPU if you are aiming for world fame? Well, not really. It will depend on what genre you play. For example, based on the data shown in Fig. 16, in the 2000s, even though indie, pop, and rap had a positive correlation, rock still possessed a negative one with WPU. In conclusion, it is not only about knowing your audience, but also knowing that they will change their preference every decade. **More interesting perhaps is the case of indie genre, so trendy nowadays, which had a negative relationship with WPU back in the 80s, but has a positive relationship in the 2000s and onward.**

In the next Section 4, we will see how this pattern is learned by a tree-based classifier and embedded in its predictions, allowing us to produce some entertaining discoveries.

4 Modeling with Lyrics Entropy

4.1 Predictive Model: modifying release date to evaluate impact in expected popularity

The first experiment I deployed was to train a model that predicts the popularity of a song. This task is rather trivial and has been discussed many times, even in this course's sessions, so I'm not going to focus very much in the modeling aspect of it. However, I realized that by having such model, I could be able to "modify" certain data points and see if the model changed the predicted popularity of a song given, for example, an alteration in the song's decade of release.

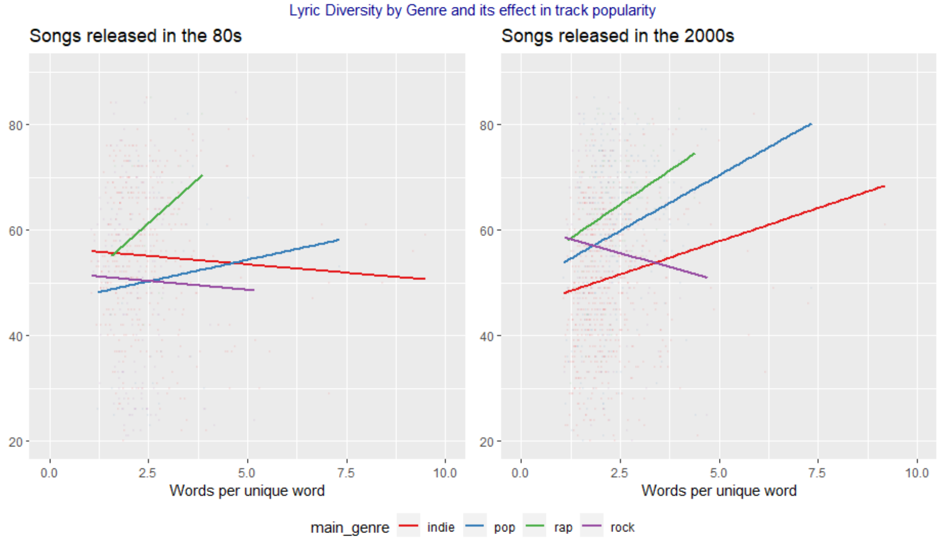


Figure 16: Average WPU of popularity groups per decade

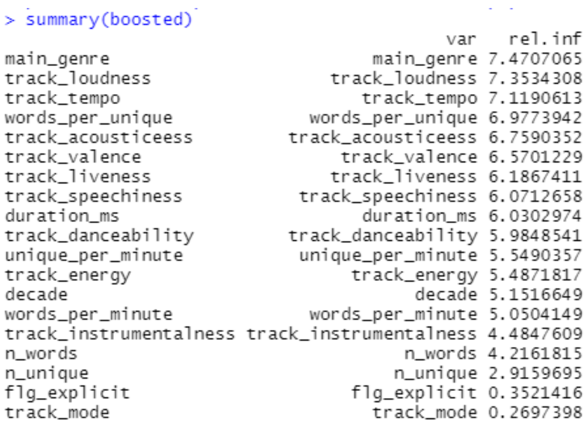
Table 1: Description of GBM Model

Technique	Gradient Boosted Machine # of trees:10K t Max depth: 4
Variables	10 audio features Duration (ms) Explicit Flag 8 Hand engineered features: Decade Main Genre No. of words No. of unique words Unique-per-minute Words-per-minute Words-per-unique
Target	Track Popularity (0-100)
Error Rate	5%

My objective with this model was to share cases in which a popular song from the 70s but with very low lyric entropy (not very repetitive) would probably achieve less popularity if it were released nowadays; and, in contrast, a song from the 80s with very repetitive and catchy lyrics would most likely be a hit again if it were released in our present-day. To prove this, I had the model trained on the complete dataset and manually altered the variable "decade" in hand picked rows to analyze the outcome.

The model consists of a GBM with $10,000$ weak learning trees and a max depth of 4. Besides the features available from Spotify, I included all the hand engineered features of my previous analysis steps. On average, the model incurs in an error of 5 points in its predictions (see Tab. 1). When analyzing the feature importance, it is noteworthy that our "lyrics entropy" score, WPU, appears as high as the 4th more important feature (see Fig. 17), implying a good validation of the reasoning behind the creation of this metric.

My bow and arrow for this analysis are Eric Clapton's "Wonderful Tonight" (read appendix C for a curious story about this tune) and the always catchy Culture Club's "Karma Chameleon". A gradient boosted trees regressor originally predicts a popularity score of 66.3 for Clapton's word-filled ballad, but when we change the decade to 2020s, *ceteris paribus*, the model drops the score in almost 4 points, to 62.5. If we repeat the same process with 80s disco hymn Karma Chameleon, the story is another



```
> summary(boosted)
                                              var   rel.inf
main_genre                             main_genre 7.4707065
track_loudness                     track_loudness 7.3534308
track_tempo                           track_tempo 7.1190613
words_per_unique                 words_per_unique 6.9773942
track_acousticeess             track_acousticeess 6.7590352
track_valence                       track_valence 6.5701229
track_liveness                     track_liveness 6.1867411
track_speechiness               track_speechiness 6.0712658
duration_ms                           duration_ms 6.0302974
track_danceability             track_danceability 5.9848541
unique_per_minute               unique_per_minute 5.5490357
track_energy                         track_energy 5.4871817
decade                                     decade 5.1516649
words_per_minute                 words_per_minute 5.0504149
track_instrumentalness track_instrumentalness 4.4847609
n_words                                   n_words 4.2161815
n_unique                                 n_unique 2.9159695
flg_explicit                         flg_explicit 0.3521416
track_mode                             track_mode 0.2697398
```

Figure 17: Feature Importance of the GBM (R output)

Table 2: Results of altering release decade in 2 songs from the dataset

Song	**Words-per-unique-words ratio**	**Decade released**	**Original Predicted Score**	**Predicted score if released in 2020s**
Wonderful Tonight - Eric Clapton	1.37 (barely repeats any word)	1970s	66.3	62.5
Karma Chameleon - Culture Club	4.1 (very repetitive lyrics)	1980s	74.8	74.8

A non-repetitive song like Wonderful Tonight has its score penalized when released in 2020s, whereas a loopy song of the likes of Karma Chameleon stays as popular as before.

one. This time, the original popularity score and the hypothetical, present day score, are the same: 74.8 (after all, they did say it comes and goes)

Obviously, just because this holds true for one hand-picked pair of songs, it doesn't mean this is systematically true. Also, no one can guarantee that this phenomenon is occurring specifically because of the lyric entropy feature we included in the model: it could very well be owed to other variables. All things considered, I still think it is interesting enough to make one wonder, and it lays grounds for a future, more-robust way of investigating the matter. Please refer to Table 2 for a summary of the inputs and outcomes of this experiment.

4.2 Unsupervised Learning: Using PCA to detect artist-level patterns

Lastly, to finish the discussion, I also ran a PCA fit over the data selecting 5 audio features and most of the hand-engineered ratios mentioned before. Before the model, I grouped the data into artists. I plotted the first 2 PC which explain 42% of the variance in the data and I kept visible only the artists with a popularity score above 75, which is very high (less than 10% of the artists belong to this elite group). In the plot, shown in Fig. 18, it is noticeable that our already familiar WPU is very alligned with track popularity, confirming once again the correlation between these two variables as we saw in the GBM and in the scatter plots. I also notice how this elite group of artists and bands are leaning towards positive values in the WPU axis, meaning that, at an artist-level, the same insight as discussed in the previous paragraph holds true.

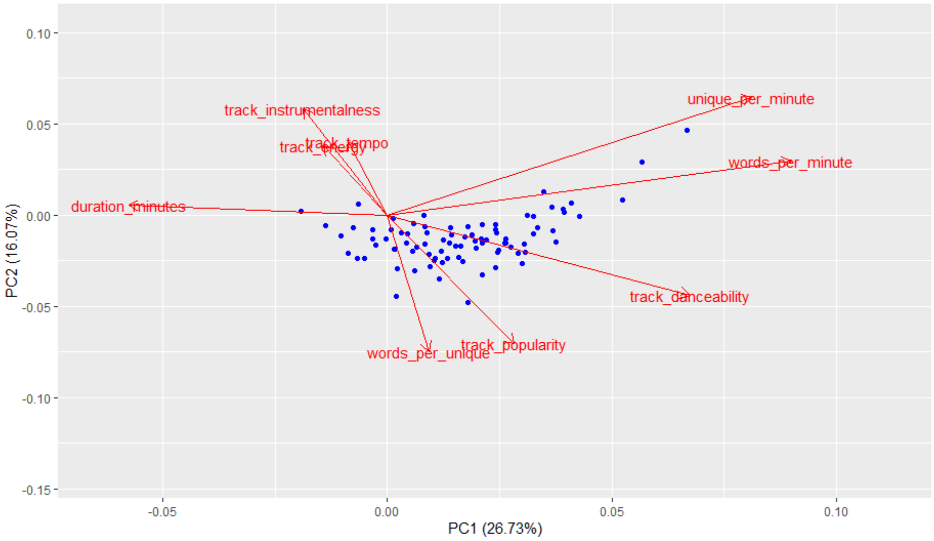


Figure 18: PCA plot of PC1 and PC2 with very popular artists shown as blue points

5 Conclusions

5.1 Executive Recommendations for the music industry

Assuming that Spotify's song popularity is a good approximation to a track's commercial success, and that all artist seek to maximize said metric:

- In this project, I've shown that rather than number of words, number of unique words, words per minute or unique per minute, it is the words-to-unique ratio what ultimately has a larger influence in a song's popularity.
- Across genres, all listeners seek some degree of repetition in the lyrics, but the value of that degree changes genre-to-genre, and in some cases decade-to-decade. We can also expect that the same listener will have a different tolerance towards lyrics repetition depending on the genre. For example, I argue that most people can tolerate more repetition of lyrics in a pop song than in a rock song.
- Some formulas that worked in the past are still effective nowadays. In a genre like pop, for example, the relationship between WPU and song popularity has been constantly negative through the decades. This means that it is a good idea for some artists to revisit old songs and attempt new covers. We can think how this is happening more frequently recently (Dua Lipa's cover of Elton John is a perfect example). This insight was strongly proved with the predictive model example of Karma Chameleon tune in Section 4.1
- In a genre like indie, however, the musician and record label owners have to pay more attention to the trend, since I have shown that it varies from decade to decade. Reusing formulas from past decades might not be a safe bet in these kinds of genres.

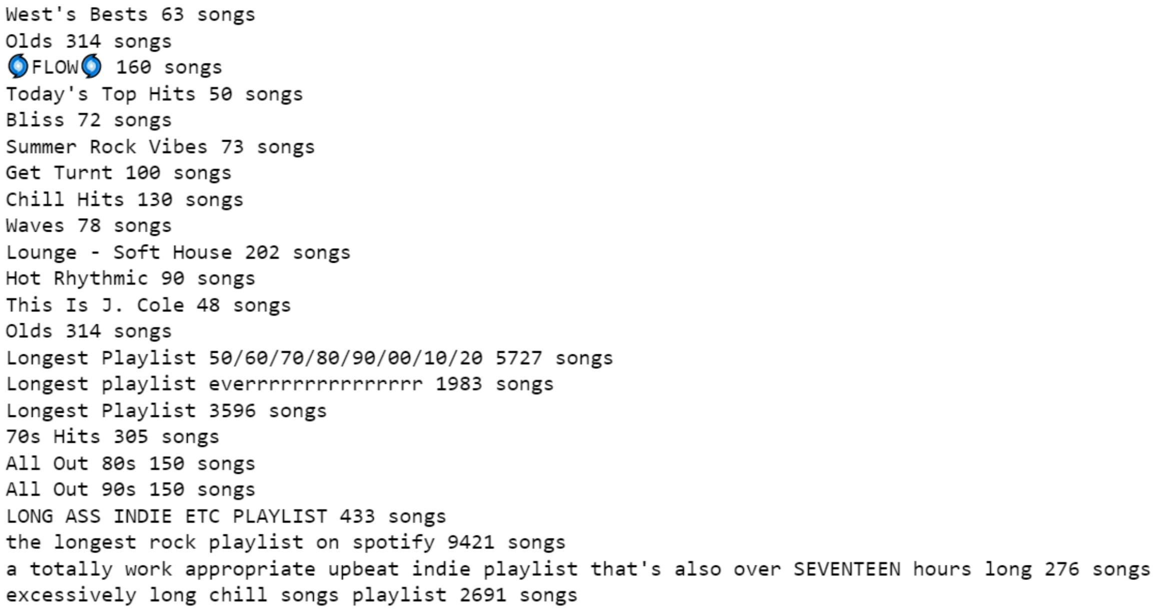
5.2 Final thoughts

What I really enjoyed about my work is that I didn't have to use any particularly complex measurements to be able to arrive to these insights. A formula as simple as counting unique words and designing some straightforward ratios was enough to tell a complete and compelling story about a phenomenon that on a first sight can sound esoteric. The simplicity of the metrics used means that one doesn't have to be a mathematician or scientist to understand the insights founds. The metrics are intelligible and memorable since I decided to a stick to a "unitA-per-unitB" type of nomenclature along the whole discussion.

I decided to build the dataset from scratch because I thought it was important to ensure by myself the quality and representation of the data and I'm glad I did because that was essential in my arrival to valid insights.

My objectives set in 1.2 were met and, to the best of my knowledge, I have provided novel, valuable, and actionable insights for the music industry and its main actors.

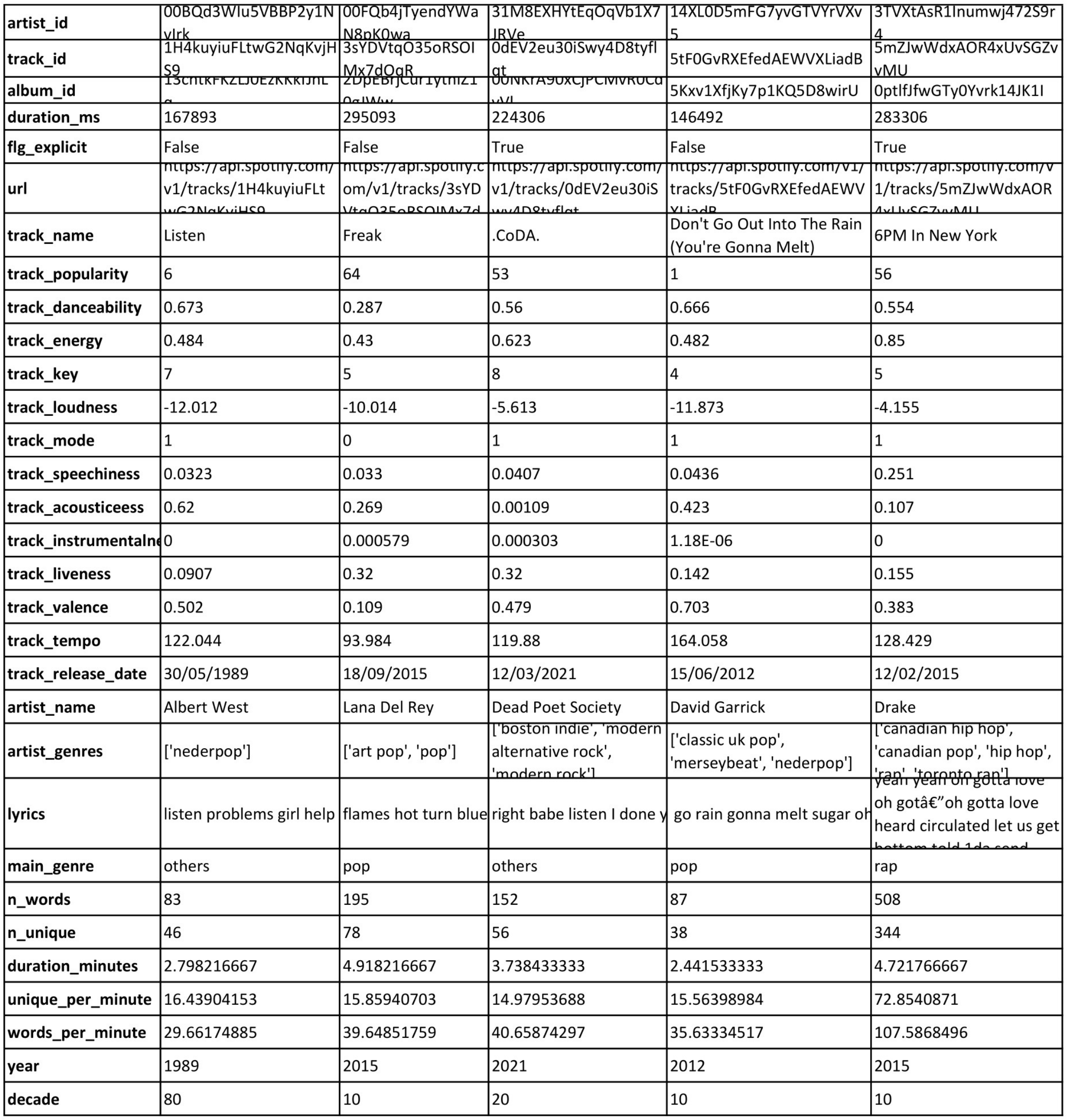
Appendix A Playlists and size



```
West's Bests 63 songs
Olds 314 songs
🌀FLOW🌀 160 songs
Today's Top Hits 50 songs
Bliss 72 songs
Summer Rock Vibes 73 songs
Get Turnt 100 songs
Chill Hits 130 songs
Waves 78 songs
Lounge - Soft House 202 songs
Hot Rhythmic 90 songs
This Is J. Cole 48 songs
Olds 314 songs
Longest Playlist 50/60/70/80/90/00/10/20 5727 songs
Longest playlist everrrrrrrrrrrrrr 1983 songs
Longest Playlist 3596 songs
70s Hits 305 songs
All Out 80s 150 songs
All Out 90s 150 songs
LONG ASS INDIE ETC PLAYLIST 433 songs
the longest rock playlist on spotify 9421 songs
a totally work appropriate upbeat indie playlist that's also over SEVENTEEN hours long 276 songs
excessively long chill songs playlist 2691 songs
```

Figure 19: Complete list of playlist's retrieved for songs set

Appendix B Dataset Sample



artist_id	00BQd3WIu5VBBP2y1NvIrk	00FQb4jTyendYWaN8pK0wa	31M8EXHYtEqOqVb1X7JRVe	14XL0D5mFG7yvGTVYrVXv5	3TVXtAsR1Inumwj472S9r4
track_id	1H4kuyiuFLtwG2NqKvjHS9	3sYDVtqO35oRSOIMx7dOqR	0dEV2eu30iSwy4D8tyflgt	5tF0GvRXEfedAEWVXLiadB	5mZJwWdxAOR4xUvSGZvvMU
album_id	13cntkFKZLJ0EzKkkIJnLg	2DpEBrjCur1ytnIZ10gJWw	00NkrA90xCjPCMvR0CdvVl	5Kxv1XfjKy7p1KQ5D8wirU	0ptlfJfwGTy0Yvrk14JK1I
duration_ms	167893	295093	224306	146492	283306
flg_explicit	False	False	True	False	True
url	https://api.spotify.com/v1/tracks/1H4kuyiuFLtwG2NqKvjHS9	https://api.spotify.com/v1/tracks/3sYDVtqO35oRSOIMx7d	https://api.spotify.com/v1/tracks/0dEV2eu30iSwy4D8tyflgt	https://api.spotify.com/v1/tracks/5tF0GvRXEfedAEWVXLiadB	https://api.spotify.com/v1/tracks/5mZJwWdxAOR4xUvSGZvvMU
track_name	Listen	Freak	.CoDA.	Don't Go Out Into The Rain (You're Gonna Melt)	6PM In New York
track_popularity	6	64	53	1	56
track_danceability	0.673	0.287	0.56	0.666	0.554
track_energy	0.484	0.43	0.623	0.482	0.85
track_key	7	5	8	4	5
track_loudness	-12.012	-10.014	-5.613	-11.873	-4.155
track_mode	1	0	1	1	1
track_speechiness	0.0323	0.033	0.0407	0.0436	0.251
track_acousticeess	0.62	0.269	0.00109	0.423	0.107
track_instrumentalne	0	0.000579	0.000303	1.18E-06	0
track_liveness	0.0907	0.32	0.32	0.142	0.155
track_valence	0.502	0.109	0.479	0.703	0.383
track_tempo	122.044	93.984	119.88	164.058	128.429
track_release_date	30/05/1989	18/09/2015	12/03/2021	15/06/2012	12/02/2015
artist_name	Albert West	Lana Del Rey	Dead Poet Society	David Garrick	Drake
artist_genres	['nederpop']	['art pop', 'pop']	['boston indie', 'modern alternative rock', 'modern rock']	['classic uk pop', 'merseybeat', 'nederpop']	['canadian hip hop', 'canadian pop', 'hip hop', 'rap', 'toronto rap']
lyrics	listen problems girl help	flames hot turn blue	right babe listen I done y	go rain gonna melt sugar oh	yeah yeah oh gotta love oh gotâ€”oh gotta love heard circulated let us get bottom told 1da send
main_genre	others	pop	others	pop	rap
n_words	83	195	152	87	508
n_unique	46	78	56	38	344
duration_minutes	2.798216667	4.918216667	3.738433333	2.441533333	4.721766667
unique_per_minute	16.43904153	15.85940703	14.97953688	15.56398984	72.8540871
words_per_minute	29.66174885	39.64851759	40.65874297	35.63334517	107.5868496
year	1989	2015	2021	2012	2015
decade	80	10	20	10	10

Figure 20: Sample of 5 rows (transposed) from the dataset after preprocessing

Appendix C The muse of 1970s rock ballads

Clapton wrote Wonderful Tonight in 1976 for his future wife Pattie Boyd while they were getting ready for a night out. He also wrote other famous songs dedicated to her like Layla and Bell Bottom Blues. Clapton actually met Boyd through George Harrison, who was married to her from 1966 and 1977. It was Harrison who introduced Boyd to Clapton while they were still together. While married to Boyd, Harrison wrote several of his beautiful songs such as Something and For You Blue. Between her relationships with Harrison and Clapton, it is believed that she had a brief love story with Ronnie Wood, future member of the Rolling Stones, and God knows what songs she inspired into Ronnie back when they were together.

To me, it is unbelievable how many of the beautiful 1970s ballads we owe to the same woman. She was really a modern-day muse and even though I don't like her for breaking the hearts of my favorite artists, I can't help but appreciate the by-products of said romances.

Further reading: https://en.wikipedia.org/wiki/Pattie_Boyd

References

[1] R. A. Peterson and D. G. Berger, "Cycles in symbol production: The case of popular music," *American sociological review*, pp. 158–173, 1975.

[2] P. J. Alexander, "Entropy and popular culture: product diversity in the popular music recording industry," *American Sociological Review*, vol. 61, no. 1, pp. 171–174, 1996.

[3] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," *arXiv preprint arXiv:1707.04678*, 2017.

[4] P. Bello and D. Garcia, "Cultural divergence in popular music: the increasing diversity of music consumption on spotify across countries," *Humanities and Social Sciences Communications*, vol. 8, no. 1, pp. 1–8, 2021.

[5] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.