

Reading songs: A Computational Analysis of Popular Songs Lyrics

Silvia Corbara¹, Alessio Molinari²

¹*Scuola Normale Superiore, P.za dei Cavalieri, 7, 56126 Pisa (IT)*

²*Università di Pisa, Pisa (IT)*

Abstract

There is no doubt that certain songs are so easily liked by the public because of their melody. However, certain songs strike us for their lyrics, either because they convey an important (for us) meaning, or for their captivating sound when sung.

Since the year 1958, the *Billboard* magazine held the special section *Hot 100*, with a rank of the 100 most popular songs of the week. Exploiting this invaluable source regarding the musical taste of the past decades until our days, we perform an analysis over various aspects of popular songs lyrics, especially focusing on the question: what is the importance of lyrics, when classifying musical artists and genres? We find out that, as we expected, many artists are not immediately recognizable only by their lyrics; some of them, however, and especially if they belong to some specific genres (such as rap), stand out, opening to the possibility of further analysis over their styles and themes.

Keywords

Lyrics analysis, popular songs, music

1. Introduction

Together with melody, lyrics are what characterizes songs, and by extension the artists creating and performing them. It could be replied that certain songs are not the direct production of a single artist (or groups of artists), but are written under commission, or subject of marketing demands. However, even if this is true in many circumstances, it is undoubted that each artist or group have a peculiar style, be it uniquely 'theirs' or the result of the combination of many factors. A similar, if not identical, point can be made for musical genres, represented by artists and songs sharing common, more or less defined, elements.

Since the 1950s, popular music spread and evolved, constantly changing through the decades. Almost each decade has had its own very peculiar taste in music and in what was considered popular - so much in fact, that we can usually tell if a song is from the '60s or from the '80s, for example. Hence, a question raises: can we actually recognize an artist or a musical genre only by a song lyrics? More generally, how much can lyrics tell us about popular genres and artists?

In this work, we firstly perform some preliminary analysis over our dataset, made of the *Hot 100* songs listed in the *Billboard* magazine (see Sec. 3.1 for details). In particular, we experiment by clustering lyrics embeddings and plot them in a bidimensional plane to see if


IIR 2021 – 11th Italian Information Retrieval Workshop, September 13–15, 2021, Bari, Italy

✉ silvia.corbara@sns.it (S. Corbara); alessio.molinari@phd.unipi.it (A. Molinari)

🆔 <https://orcid.org/0000-0002-5284-1771> (S. Corbara); <https://orcid.org/0000-0002-8791-3245> (A. Molinari)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

such a visualization holds meaningful insights (Sec. 3). Secondly, we focus on investigating the employment of an automatic text classifier for the task of artist and genre classification, by exploiting various features sets extracted from the texts, i.e., the lyrics (Sec. 4).

The methods and experiments in this project are developed in Python, and the code is available on GitHub: https://github.com/silvia-cor/lyrics_analysis.

2. Related work

It is no surprise that songs lyrics have been employed for many different musical analysis - of course, with some restrictions given by some songs being purely instrumental, or not available due to copyrights issues. Such analysis can have different goals: from studying specific artists' careers (see for example [1], where stylistic and emotional differences in some songs by The Beatles are enlightened, both between the artists Paul McCartney and John Lennon and across the band years), to the more general musical trends (see for example [2], where it is manifest that songs lyrics become progressively simpler over the years due to a bigger pool of novelty), to the changes in society preferences over time (see for example [3] on how meaningful contents are more popular in difficult historical times).

In the field of Music Information Retrieval (MIR), the specific value of such features in different classification tasks is unclear: for example, they appear to be less effective than features extracted from the audio of the songs in genre classification [4], while they exceeds audio features (among others) in specific settings of mood classification [5]. However, it has been proved that combining audio and textual features often results in a better performance than the employment of a single features category, both in genre classification [4, 6] and artists similarity detection [7]. Similarly, lyrics have been proved to be linked with the popularity of their respective song, even though their relative contribution compared to audio features depends on how the concept of 'popularity' is computed [8]. Finally, Fell and Sporleder [9] employ lyrics to tackle different classification tasks (by genre, 'goodness' intended as popularity, and time); in particular, they exploit n-grams TfIdf combined with other different features, such as POS-tags, rhymes, word and sentence lengths, and semantics.

3. *Hot 100*: an analysis of popular songs

In this section, we explain the characteristic of our dataset (Sec. 3.1) and show some interesting insights, mostly related to the popularity and duration of the various songs over the six decades considered, from the 60s to our current days (Sec. 3.2). Moreover, we experiment by projecting the embeddings of the songs from an all-time popular band (The Beatles) into a bidimensional space, and evaluate the result of a subsequent clustering process (Sec. 3.3).

3.1. The dataset

All of our analysis and experiments are conducted on a dataset available on Kaggle¹. The dataset is a collection of all the weekly *Hot 100* charts released by the *Billboard* magazine, since its start

¹<https://www.kaggle.com/dhruvildave/billboard-the-hot-100-songs>

in 1958 up to May 2021. This follows a common practice: many studies in literature make use of *Billboard's Hot 100* [2, 3], or some other list of songs ranked by popularity (such as weekly UK top five singles charts [8]). The dataset consists of 327, 587 entries (with 24, 333 unique songs), for a total of 10, 044 unique artists. Beside the artist and song title, the dataset also contains information on the song position in the chart and how many weeks it stayed ‘on-board’. We integrated this dataset with lyrics downloaded via the *genius.com* HTTP API. For each song, we also gathered information on the genre and duration via the *last.fm* HTTP API. As it was expected, we could not retrieve all these information for all the entries in the dataset. Pruning out all of those songs for which we could not retrieve the lyrics or the genre, we remain with 7, 045 unique songs for a total of 596 artists².

Furthermore, we grouped the genres of the songs in such a way that we could have a pre-defined and controlled list of genres to work with. Specifically, the complete list of genres is composed of 17 elements: POP, PROGRESSIVE ROCK, ROCK, METAL (which also include HARD ROCK), COUNTRY, RNB (which also include DOO WOP), FUNK, HIP-HOP, ALTERNATIVE, RAP, DISCO (which also include ELECTRONIC, DANCE and DANCEHALL), FOLK, JAZZ, BLUES, INDIE, CHRISTMAS and SOUL. Moreover, since in most of our analysis we work with songs and artists grouped by decade, we merge all the charts from 1958 and 1959 to the charts in the 60s, and the charts from 2020 and 2021 to those in 2010s.

Finally, we add to our dataset two metrics that could be seen as average rank/popularity scores across one or all the decades taken into consideration. Recall that we have the position for each song in the charts and for each week it stayed on the chart. We could therefore think of an average rank measure for a given time span as the sum of the ranks given to a certain song, penalizing it by summing a constant for each week it did not appear in the charts. This allows us to give a higher weight to songs that stayed in the charts for many weeks, rather than to songs which were short-lived meteors. More formally, the rank r^s for a given time span T and song s is given by:

$$r_T^s = \frac{\left(\sum_{t=0}^T r_t^s\right) + k \cdot (T - W_T^s)}{T} \quad (1)$$

where r_T^s is the average rank for a given time span T (which is composed of n weeks), r_t^s is the rank of song s at time t , W_T^s is the total number of weeks song s stayed in the charts, and k is a constant set to 101 (i.e., the first available position out of the chart).

Another possibility is a simpler popularity score p_T^s , given by the sum of the inverse of the ranks of a song in the time span:

$$p_T^s = \sum_{t=0}^T (101 - r_t^s) \quad (2)$$

3.2. Duration and genre trends, from 1960s to 2010s

Considering the duration by itself, an interesting question regards whether songs have shortened or lengthened their duration over the years, and/or how this is related to the decade, eg.: were

²We exclude the songs coming from the artist GLEE CAST, since they come from the popular tv series *Glee* and are mostly cover songs.

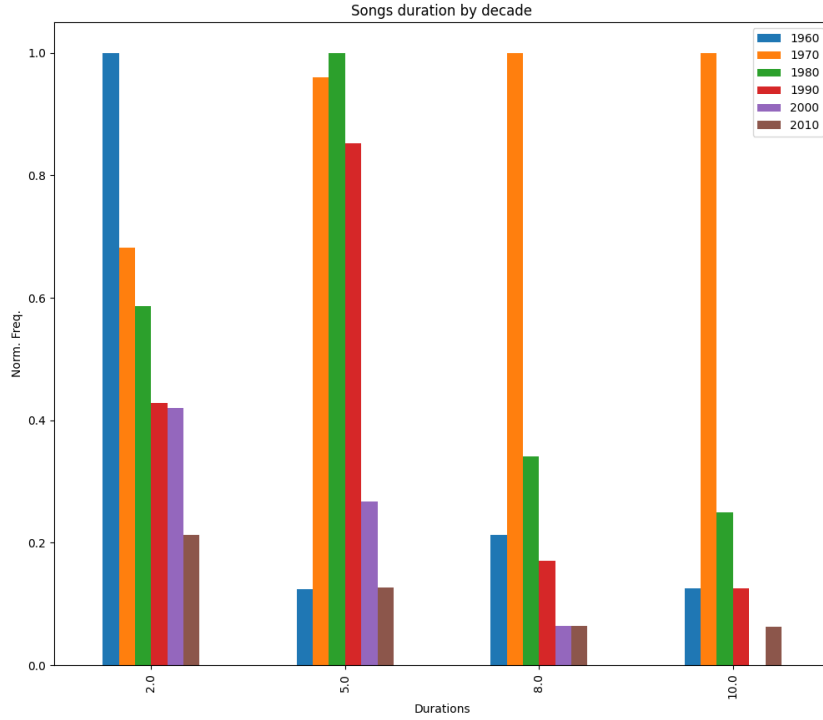


Figure 1: Duration of songs over decades.

there years where people enjoyed longer songs? Or again, which is the decade where most tracks in our chart had a rather small duration? We show the histogram for the duration of the songs over the decades in Fig. 1. Frequency has been normalized by decade in order to improve visualization, i.e. for each decade d , its normalized frequency f^d is given by:

$$f_i^d = \frac{n_i^d}{\max_j n_j^d} \quad (3)$$

where f_i^d is the frequency of duration i in decade d and n_i^d is the count of songs with duration i over that decade. From the bar chart, we notice how 2-minutes songs were very popular in the 60s and monotonically faded over the next decades. On the other hand, as we may expect, 5-minutes songs were fairly popular in most decades. Most notably, notice how the 70s favoured very long songs with respect to the other decades: this is mostly due to the rise and fall of progressive rock in those years. Finally, the decades 2000-2010 seem to favor shorter songs, a trend noticed also elsewhere [2, p.13].

Regarding genres, we show in Fig. 2 the frequency of the top 3 genres for each decade. Notice that in this case, we applied no normalization, and simply plot the count of songs with the given genre. We notice how SOUL was the prevailing genre by a fair margin in the 60s, but

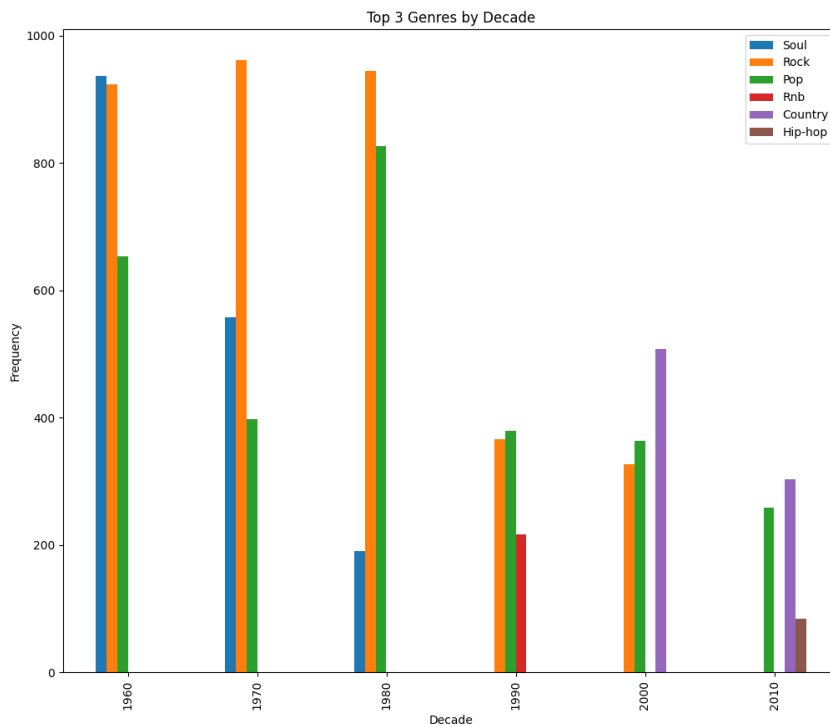


Figure 2: Top 3 genres of songs by decade.

disappeared from charts since the 90s. On the other hand, as we would expect, ROCK and POP have been rather stable over the decades. We should point out, however, that these are very inclusive labels, and pop music in the 80s is a very different genre from what was considered ‘pop’ in the early 2000s, for instance. Also, rock is not to be confused with the hard rock of the 70s (indeed, the latter is but one of the subgenres of the former).

Lastly, we check if there is any correlation between the duration of songs and their popularity/rank, as measured by Pearson correlation coefficient. We expected to see a fairly strong correlation between short songs (2-5 minutes) and popularity/rank, since short songs are usually easier to remember and to follow. Quite the contrary, it turns out there is actually no significant correlation between the two, i.e. the correlation coefficient is a negative -0.13 between the all-time ranking and the duration, and a positive 0.08 between the all-time popularity and the duration.

3.3. Visualizing the lyrics: THE BEATLES through embeddings

Taking into account the lyrics of a song, one interesting analysis to consider is the following: visualizing the projection of the lyrics in a bidimensional plane, in order to see if we can obtain

any useful insight.

We test this idea by considering the production of a single artist. In order to have a reasonable number of songs, we need an artist whose songs have constantly been great hits in the time-frame considered. We hence turn to THE BEATLES, probably the most famous band of the 60s and beyond (also, as an addendum to the study in [1]). As a matter of fact, in our dataset we have 52 songs by the band, spanning the whole 60s decade (plus *Free as a Bird*, 1995).

We then create a bidimensional embedding of each of THE BEATLES' songs in our dataset, and cluster them with a clustering algorithm, in our case K-means. In order to do this, we use the renowned GloVe word embeddings [10]. In particular, we choose the embeddings trained on the Twitter dataset; this is not entirely an arbitrary choice, as we would hope to have a wider vocabulary for slang and similar constructions (present in abundance in lyrics), than with any of the other GloVe embedding. We use the 25 dimensional vectors of the Twitter GloVe embeddings, followed by the t-SNE algorithm [11] (with the Scikit-learn implementation [12]) in order to reduce the embeddings to a bidimensional vector. The embedding for each song is then computed as the mean of its tokens embedding (tokenization is made via the Python NLTK library [13]). We show the results of the K-means clustering (with 4 clusters³) on the so-obtained bidimensional vectors in Fig. 3. Notice how the clustering and the songs position in the space actually convey some meaning:

- the blue cluster (number 0) mostly gathers generally uplifting/hopeful songs;
- the orange cluster (number 1) mostly gathers late-Beatles and dreamy/psychedelic lyrics;
- the red and green cluster (number 2 and 3) seem to gather sad and/or love-related songs.

4. Classification

In this section, we investigate the employment of textual features extracted from the lyrics for different classification tasks. In particular, we aim to investigate two different settings and research questions:

- Multi-class classification:** Is it possible to classify songs by their artist (genre), given a set of possible artists (genres)? How do different features sets perform in this task? (Sec. 4.1)
- Binary classification:** Are some artists (genres) more identifiable than others, taking into account only their lyrics? (Sec. 4.2)

Given the exploratory nature of these experiments, and the low number of songs available for certain artists, in this section we retain only the artists with at least 10 songs. This subset sums up to 292 artists and 4,874 songs in total. This causes the disappearance of the INDIE genre.

In every experiment in this section, we employ a Support Vector Machines (SVM) as learning algorithm, since it is a standard for many classification tasks, due to its resilience to high dimensionality. In particular, we employ the LinearSVC implemented in the Scikit-learn library [12]. Moreover, due to the exploratory nature of these experiments, we employ the default hyper-parameters without fine-tuning, in order to reduce the computational time required.

³The choice for 4 clusters has a dual reason: on the one hand, after various experiments with different number of clusters, 4 empirically yielded the best visualization results; on the other hand, this aligns with the division by Whissell's [1] of the band's compositions in four stages.

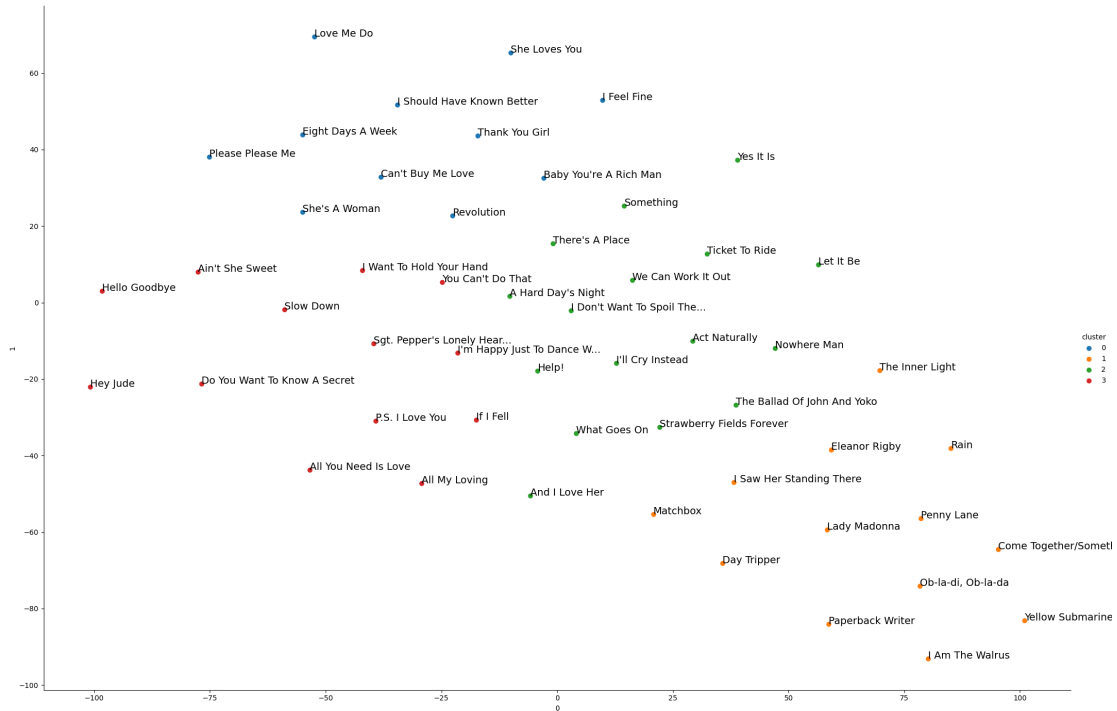


Figure 3: K-means clustering (with 4 clusters) of bidimensional (GloVe) embeddings of The Beatles' songs. Songs embeddings are the average of their tokens embeddings, reduced to a bidimensional vector via t-SNE. We shortened long titles with three dots.

4.1. Comparing features sets: multi-class classification

We experiment with the following features sets:

- **CHAR N-GRAMS:** the TfIdf of character n-grams with n in range $[2, 5]$.
- **WORD N-GRAMS:** the TfIdf of word n-grams with n in range $[1, 3]$.
- **AUTHORSHIP:** this set is made of three different features sub-sets: a) frequency of function words ⁴, normalized by the total number of words in a song; b) frequency of word lengths in the range $[1, 26]$, normalized by the total number of words in a song; c) the TfIdf of word n-grams with n in range $[1, 3]$ over the POS-tags ⁵ labels.
- **PHONETICS N-GRAMS:** the TfIdf of character n-grams with n in range $[2, 5]$ over the text converted into the IPA phonetic transcription ⁶.
- **ALL:** combination of all the previous sets.

⁴The function words, or stopwords, are words without a strong semantic value that mostly help building the grammatical structure of the phrase, such as articles. We employ the list of stopwords available in the homonym NLTK [13] module.

⁵We compute the POS-tags through the tagger available in the homonym NLTK [13] module.

⁶In order to convert the text into the transcription, we employ the Python library *English-to-IPA* available at: <https://github.com/mphilli/English-to-IPA>.

Character and word n-grams are nowadays a standard for many text classification tasks; in particular, character n-grams have the advantage to capture bits of multiple aspects of the text, such as the syntax and the sound. However, it has been observed that, despite their good results, the risk of them actually performing domain classification instead than authorship classification is quite high [14]. Hence, we also experiment with the `AUTHORSHIP` set, that should retain less information regarding the domain. In particular, it is comprised of features widely used in authorship tasks; they have been similarly employed in [6, 7, 9], but we exploit the Tfidf over the word n-grams in the POS-tags encoding, instead than the normalized frequency of the single POS-tag. Moreover, phonetics have been variably employed in music- and poetry-related analysis, but mostly limited to the identification and counting of repetitive sound and rhymes (see for example [6, 9]). Instead, we experiment by computing the Tfidf of character n-grams of the distorted text, i.e., the text transformed into the corresponding phonemes. Note that, in order to limit the dimensionality of the computation within the sets or the subsets that exploit character or word n-grams, we select and retain only the 10% of the best features of the entire set or subset (computed through χ^2).

We perform a separate experiment for each domain and for each features set. Each experiment is composed as follows. We select 5 random artists and keep all their songs as dataset (remember that each artist has at least 10 songs). We extract the features for the corresponding features set, and perform multi-class classification via 3-fold cross-validation over the dataset, where the labels are the artists (the genres). We store the predictions for each fold and aggregate them, as if coming from a single classifier. We then compute Macro- F_1 and micro- F_1 over the results. We repeat this process 10 times⁷, and finally compute the mean and standard deviation among the Macro- F_1 and micro- F_1 values.

The results of this series of experiments are in Tab. 1. The general low performance of every set might denote the artist (genre) classification task as a particularly hard one to tackle, at least when using only features extracted from lyrics, as noted in other works [6]. Some interesting insights can be found nevertheless. It is interesting to note, for example, that character and word n-grams are the lowest-performing sets of features (by looking at the Macro- F_1), despite their generally good results in many applications. It could be thus hypothesized that popular songs share many common words and themes, regardless of their author or genre. On the other hand, the `AUTHORSHIP` set and the phonetics n-grams hold very good results, especially the latter; interestingly, the `AUTHORSHIP` set is consistently penalized on the micro- F_1 metric. Finally, the `ALL` set generally provides the best results: this is clear for the classification by artist, while for the classification by genre it is slightly surpassed by the `AUTHORSHIP` set on the Macro- F_1 metric (but does far better on the micro- F_1 metric), and it is similarly surpassed by the phonetics n-grams on the micro- F_1 metric (but does better on the Macro- F_1 metric).

4.2. Binary classification

In this section, we aim to find whether certain artists (genres) are more identifiable than others. To do so, we perform a series of binary classification by artist (genre), and rank the results. In this case, we employ the `ALL` features set.

⁷The random process selecting the artists is seeded, hence the data are the same for each experiment at run k .

Table 1

Results of multi-class classification experiments for different domains and features sets. We report the mean and the standard deviation of the Macro- and micro- F_1 values.

	Char n-gram		Word n-grams		Authorship		Phonetics n-grams		All	
	Macro- F_1	micro- F_1	Macro- F_1	micro- F_1	Macro- F_1	micro- F_1	Macro- F_1	micro- F_1	Macro- F_1	micro- F_1
Artist	0.323 \pm 0.096	0.434 \pm 0.057	0.302 \pm 0.085	0.411 \pm 0.051	0.344 \pm 0.065	0.384 \pm 0.078	0.342 \pm 0.091	0.441 \pm 0.054	0.363 \pm 0.111	0.444 \pm 0.085
Genre	0.214 \pm 0.050	0.508 \pm 0.071	0.216 \pm 0.080	0.492 \pm 0.110	0.248 \pm 0.068	0.456 \pm 0.100	0.223 \pm 0.057	0.519 \pm 0.066	0.235 \pm 0.045	0.512 \pm 0.071

In particular, for each artist (genre), we perform the following experiment. We take all the songs for the specific artist (genre) D , summing up to a total of n songs (the samples for the positive class). We randomly select n songs from other artists (genres) which are not D (the samples for the negative class, creating a balanced dataset). We perform binary classification via 3-fold cross-validation over the dataset, where the labels are 1|0 for the positive|negative class, i.e., whether the sample belongs to D or not. We store the predictions for each fold and aggregate them, as if coming from a single classifier, and we compute the F_1 over the results. We repeat this process 5 times⁸, and finally compute the mean among the binary F_1 values.

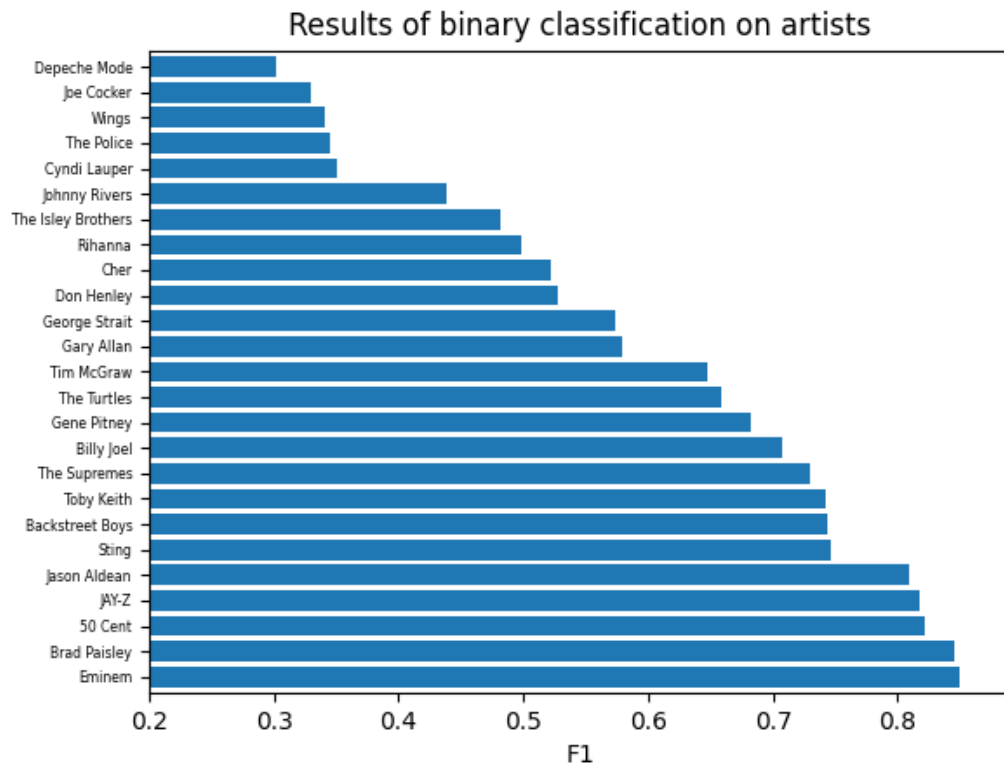
The results of this series of experiments are in Fig. 4. It is worth noticing the general coherence of the two graphics. On the one hand, the less recognizable artists mostly compose for the less recognizable genres (mostly POP and ROCK). On the other hand, the 5 most recognizable artists compose songs mostly for 2 of the most recognizable genres: JASON ALDEAN and BRAD PAISLEY for COUNTRY, and JAY-Z, 50 CENT and EMINEM for RAP. Such discrepancies among artists and, above all, genres, may have multiple answers. In fact, rap songs generally have longer and more constructed texts, which clearly set them apart from other productions, so much as to gain an almost-perfect score for both artists and genres classification. Besides, genres like pop and rock, but even jazz or metal, are musically very heterogeneous and have greatly changed over the decades. It is then no surprise that it would be proportionally harder to identify them.

5. Conclusion and future works

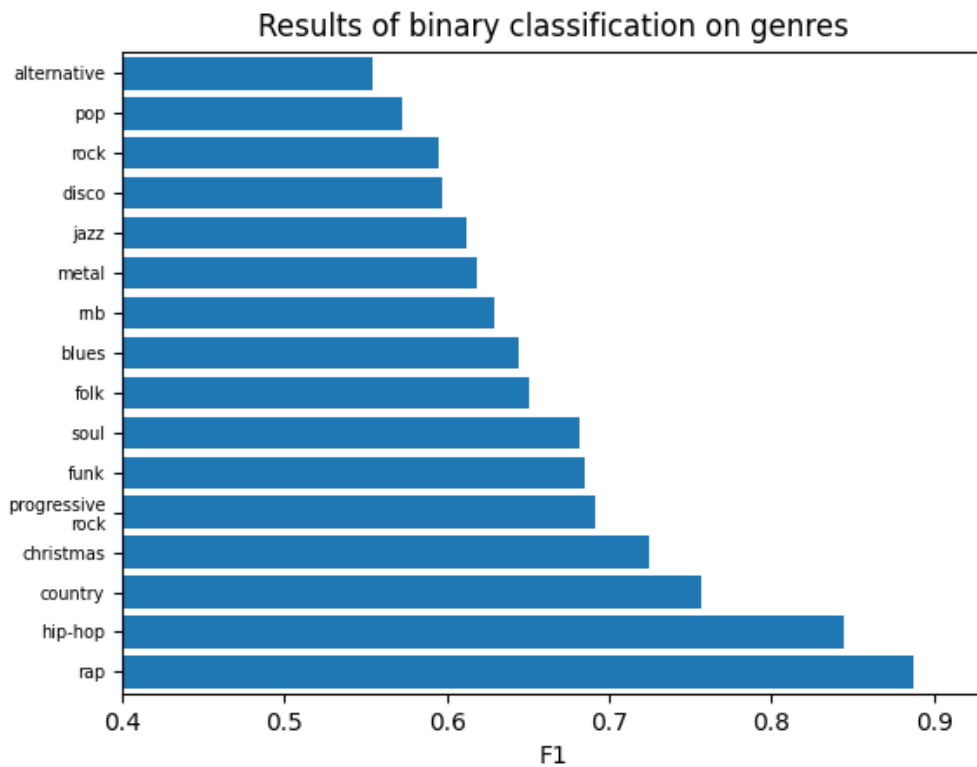
In this project, we analyze the lyrics of songs appeared in the *Hot 100* rank in the *Billboard* magazine in the period 1958-2021. We draw out various observations regarding the role of lyrics in the musical landscape. For example, we show that performing clustering on the embeddings of The Beatles' songs lyrics hold interesting results, in line with the characteristics of the band's works. Moreover, we show that certain artists and genres are generally easier to classify, due to peculiarities in their songs that reflect on the lyrics.

It would be very interesting to further analyze the most discriminative features of the classification (i.e., the features holding the biggest weight in the computation), in order to see if some common patterns can be identified, and if certain artists or genres share similar themes. Moreover, we would like to employ lyrics embedding not only to visualize the production of an artist, but also in order to compare the different artists, for example by employing some form of mean among the embeddings of the various artists' songs. We leave these ideas for future work.

⁸The random process that selects the negative samples is seeded separately for each of the 5 runs; of course, each D will have a different pool from which to pick, but this assures different negative samples and reproducibility.



(a) Results of the binary classification on artists. Since it would have been infeasible to show all the 292 artists of the dataset (and it would have added little to the discussion), we only display the 5 less-scoring artists, the 5 top-scoring ones, and 15 random others in-between.



(b) Results of the binary classification on genres

Figure 4: Results of the binary classification experiments on (a) artists and (b) genres.

References

- [1] C. Whissell, Traditional and emotional stylometric analysis of the songs of Beatles Paul McCartney and John Lennon, *Computers and the Humanities* 30 (1996) 257–265.
- [2] M. E. Varnum, J. A. Krems, C. Morris, A. Wormley, I. Grossmann, Why are song lyrics becoming simpler? A time series analysis of lyrical complexity in six decades of American popular music, *PloS one* 16 (2021) 1–18.
- [3] T. F. Pettijohn, D. F. Sacco Jr., The language of lyrics: An analysis of popular Billboard songs across conditions of social and economic threat, *Journal of language and social psychology* 28 (2009) 297–311.
- [4] C. McKay, J. A. Burgoyne, J. Hockman, J. B. Smith, G. Vigliensoni, I. Fujinaga, Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features., in: *ISMIR*, 2010, pp. 213–218.
- [5] X. Hu, J. S. Downie, When lyrics outperform audio for music mood classification: A feature analysis, in: *ISMIR*, 2010, pp. 619–624.
- [6] R. Mayer, A. Rauber, Musical genre classification by ensembles of audio and lyrics features, in: *Proceedings of International Conference on Music Information Retrieval*, 2011, pp. 675–680.
- [7] T. Li, M. Ogihara, Music artist style identification by semi-supervised learning from both lyrics and content, in: *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 364–367.
- [8] A. C. North, A. E. Krause, D. Ritchie, The relationship between pop music and lyrics: A computerized content analysis of the United Kingdom’s weekly top five singles, 1999–2013, *Psychology of Music* (2020) 1–24.
- [9] M. Fell, C. Sporleder, Lyrics-based analysis and classification of music, in: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, 2014, pp. 620–631.
- [10] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. URL: <http://www.aclweb.org/anthology/D14-1162>.
- [11] M. LJPvd, G. Hinton, Visualizing high-dimensional data using t-sne, *J Mach Learn Res* 9 (2008) 2579–2605.
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [13] S. Bird, E. Klein, E. Loper, *Natural language processing with Python: Analyzing text with the Natural Language ToolKit*, O’Reilly Media, Inc., 2009.
- [14] E. Stamatatos, Masking topic-related information to enhance authorship attribution, *Journal of the Association for Information Science and Technology* 69 (2018) 461–473.