ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis in Data Science and Marketing Analytics

The Poetics of Music: Enhancing Content-Based Music Recommendation with Lyrical Features

Author: **Igor Gukasyan** (491911ig)

Supervisor: **dr. Sean N. Brüggemann**

Second Assessor:

Date: **29 July, 2023**

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Acknowledgements

I want to thank my supervisor, dr. Sean Brüggemann, for giving me the freedom to pursue a topic I was passionate about and guiding me throughout the process. His detailed and extensive feedback significantly contributed to my understanding of the topic and the development of this thesis.

I am thankful for the overwhelming mental support I received from my friends, Timofey Bespalov, Sam Rink, Djien, Matvei Lukianov, Egor Fadeev, Ilia Ischenko, Dmitry, Mikhail and Pavel Kukins, Dmitry Kuznetsov, and Vyacheslav Valeryevich Mashnov. I am grateful for the patience, understanding, and dedication of Ana, who did everything in her power to support me during the last year. She was largely the reason this thesis happened.

Most importantly, I want to express my deepest gratitude to my family, who believed in me with their heart and soul and supported me at every step of my journey. Despite our inability to meet often, I always felt their constant support and knew they had my back.

Contents

1 Introduction

Music streaming has become a primary source of music revenue: according to IFPI, 67% of total industry revenue was generated by streaming in 2022 (IFPI, 2023). The music streaming market is growing rapidly in revenue and the volume of offerings. At the moment, there are more than 100 million songs available on Spotify, and 100,000 songs are uploaded to Spotify every day (Brooks, 2023).

The sheer number of content on a platform can leave the users "paralyzed" as they are subject to the "paradox of choice" (Schwartz et al., 2002): the relationship between satisfaction and the number of choice options has an inverted U-shape, with both too few choices and too many choices leading to user dissatisfaction. Thus, streaming platforms need to find ways to alleviate this problem, e.g., by better navigating users through their vast libraries of content. One solution deployed by streaming platforms consists of user-curated playlists, which allow users to create playlists of songs they like and share them with users who want to discover new music. Streaming platforms also use data on user preferences to provide (personalized) recommendations to users, e.g., algorithmic playlists. These recommendations are an important factor through which streaming platforms can differentiate, as the song libraries of most major streaming platforms are nearly identical. The importance of providing relevant and high-quality recommendations is significant and can impact the company's market standing. For example, Spotify, the largest music streaming service, has the music recommendation algorithm as its unique value proposition relative to other streaming platforms.

Most streaming companies employ a collaborative filtering (CF) approach in their music recommendation: recommending songs based on what similar users are listening to. CF is known for its superior accuracy, and will generally outperform content-based approaches (Slaney, 2011). However, CF suffers from the cold-start problem: Since CF relies solely on user data, it is impossible to recommend a song that has never been listened to before, and it is unlikely that a song with few streams will be recommended. According to *Luminate Year-End Music Report* (2023), out of 184 million songs that they tracked, 79.7 million were streamed less than 10 times. For such cases, hybrid approaches to recommendation were proposed, combining collaborative filtering and content-based recommendation.

Content-based approaches evaluate the content of the song with possible features ranging from audio features to semantics. As mentioned by Wang & Wang (2014), traditional music features such as Mel-frequency cepstral coefficients (MFCCs are a way to summarize the key sounds in a piece of audio and are usually represented as a sequence of vectors) might not be effective in prediction tasks, as they are high-level descriptors of music concepts such as genre, timbre, and melody. Bogdanov et al. (2013) showed that content-based approaches that include semantic features (i.e., features of music that are not conveyed by musical sounds) of music outperform those that do not. However, music recommendation using semantic features might be affected by the semantic gap. The semantic gap is the discrepancy between what can be extracted from music (i.e., semantic or audio features), and high-level human perception of these features. For example, two songs with similar timbral features and topics might be perceived completely differently by a user due to personal tastes and subjective experiences, which are not accounted for by the algorithms.

One way to come closer to bridging the semantic gap is to capture user preferences for music more exhaustively. For example, Starr (2014) found that the vividness of imagery consistently predicts the appeal of a poem. Similarly, Obermeier et al. (2013) show that the meter and rhyme of a poem significantly influence emotional responses. In this thesis, I focus on the effect of vividness of imagery, rhyme, meter, and linguistic creativity on the performance of music recommendation systems.

Research question

- How do poetic features (e.g., vividness of imagery, rhyme, meter, and linguistic creativity) affect the performance of music recommendation systems?

2 Literature Review

2.1 Algorithmic music recommendation

Music streaming is a relatively new field, which has changed the music industry, especially after the launch of the streaming platform Spotify in 2008. Today, with tens of millions of tracks in commercial music libraries (Schedl, 2019), it has become difficult for users to discover engaging content without incurring high search costs. Over the past decade, recommender systems (RSs) have emerged and evolved to lessen users' burden of finding relevant items, such as tracks, driven by commercial interests (Deldjoo et al., 2024).

Currently, two primary approaches to music recommendation exist. The collaborative filtering approach (CF) (Herlocker et al., 2000) recommends items to a user based on data of similar user interactions with items, e.g., ratings, likes. The second approach, content-based filtering (CBF) (Van Meteren & Van Someren, 2000), utilizes the content, i.e., information about the attributes of songs that a user has consumed, to make recommendations. Furthermore, hybrid systems (Ko et al., 2021), which are called content-*driven* approaches, combine both collaborative filtering and content-based filtering, enhancing recommendation quality. However, CF and CBF remain fundamental to these systems.

The amount of research on recommender systems has increased significantly since the launch of services like Spotify and competitions such as the Netflix Prize and the ACM Recommender Systems Challenge 2018 (Deldjoo et al., 2024). Still, it faces many challenges that are yet to be tackled. This literature review aims to provide insights into the currently used approaches and their limitations, particularly the cold start problem (i.e., the inability of CF recommender systems to make predictions for items with little data), semantic gap (i.e., the discrepancy between what can be extracted from music and high-level human perceptions), discuss existing research on features used in music recommendation systems (MRSs), and introduce a rationale for the use of new semantic features in MRSs.

2.2 The cold start problem of collaborative filtering

Though music recommendation traditionally employed content-based filtering (Deldjoo et al., 2024), collaborative filtering has gained popularity in recent years (Schedl, 2019). By design, CF relies on user behavior data as it implies that users who have similar behaviors act similarly on other items. Due to that design, CF faces several challenges. The cold start (CS) problem describes situations where classic collaborative filtering recommendation models cannot provide relevant recommendations due to a lack of observed behavioral user data. CS is reflected in different problems.

The data sparsity problem is related to the sparsity of user-item interaction matrices. Naturally, a user interaction dataset with a large number of items and users is sparse. It is common to have a sparse user-item matrix of 1% (or less) coverage (Celma, 2010). Users can be matched as similar only if they have interacted with the exact same items, and items can be classified as similar if similar users have interacted with them. Therefore, the CF-based approach can fail to classify similar items as similar due to the sparsity of the data, which hinders the recommendation quality.

New user and new item problems represent situations where a new user with no or few item interactions enters the platform or when a new item with no or few user interactions enters the platform. Due to data scarcity, CF cannot provide useful recommendations in such cases. More generally, any item with low popularity (whether new or not) also causes similar problems (Deldjoo et al., 2024), as well as users with atypical taste (known as ‘grey sheep’). That is, such items, despite not being new, are difficult to match as similar, and users with less popular tastes are difficult to be matched to many others.

Recommendations and data sparsity interact in a reinforcing circle: when a new user shows their preference for multiple tracks, the system tends to recommend more popular items since popular items of the dataset are similar to lots of items (Celma, 2010). Conversely, less popular but potentially interesting items are seldom recommended due to the limited data available about user interactions with them. For music recommender systems, one theory is that the issue of data sparsity and hence the cold start problem could be mitigated by using music descriptors with a more uniform distribution.

For example, Deldjoo et al. (2024) propose an "onion model" of the content in the music domain (Figure 1). They explain that it reflects a path from strictly objective and numeric descriptors to more subjective properties that stem from cultural practices of dealing with music. As per the "onion model", the layers of the music domain are:

- audio data, e.g., MFCCs, timbre, key,
- embedded metadata, such as artist name, lyrics, title,
- expert-generated content, e.g., genre, style, mood,
- user-generated content, such as tags and playlists,
- derivative content, e.g., remixes or parodies.

The authors believe that there is a shift from data sources not suffering from the cold start problem in the inner layer to the data being more prone to the cold start problem in the outer layer. In other words, Deldjoo et al. (2024) believe that some forms of content-based filtering inherently do not suffer from cold start.

Deldjoo et al. (2024) (p. 18) write that the audio signal "constitutes a perfect source to alleviate CS issues in content-based MRSs, because audio features can always be extracted from the raw audio, even for content that does not contain any descriptive tags yet". While audio data does not suffer from cold start, it does not reflect the whole reason why people might prefer one song over the other. The reasons behind musical interests are unclear (Ben Sassi et al., 2021), and features other than audio –which might be more affected by the cold start problem– might play a key role in user preferences (Deldjoo et al., 2024), therefore requiring a multifaceted approach.

2.3 Content-based filtering and the semantic gap

Content-based filtering classifies items as similar based on item characteristics. While it is not as subject to CS as collaborative filtering is, several complications related to feature selection

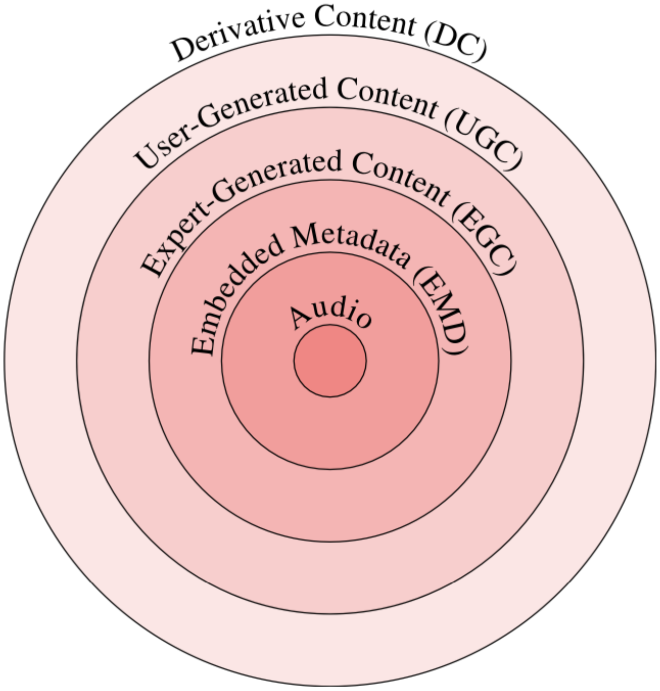


Figure 1: An ‘onion model’ of content in the music domain

exist. For example, expert-generated content is richer in semantics, yet also more biased and difficult to obtain (Deldjoo et al., 2024). User-generated content is not immediately available after the song’s release, and takes time to be generated. Moreover, the creation of user-generated content is dependent on the community of listeners and is, therefore, less reliable than other features in terms of accessibility, quality, and amount. Derivative content “has, so far, not received as much attention for content-based music recommendation as to the underlying layers” (Deldjoo et al., 2024, p. 5), and, therefore, research on it is limited.

Moreover, research consistently indicates that low-level audio features are not suitable for music recommendation. In a study by Barrington et al. (2007), two models aimed at finding similar tracks were compared: one using audio features, and another utilizing semantic features inferred from acoustic data. As an example of a semantic representation, the sound of a gun might score high in “weapon” and “war,” but low in “quiet” and “whistle”. Their analysis showed that the model based on semantic features demonstrated a significant improvement in mean average precision (mAP) compared to the one using purely acoustic features. Mean average precision is an average precision (i.e., fraction of relevant instances among the retrieved instances) across multiple queries. At a recall (i.e., the fraction of relevant instances that were retrieved out of all relevant instances) rate of 0.1, the semantic model showed a relative improvement of 26%. In a study by Bogdanov et al. (2010), three models were developed using inferred semantic characteristics, while two were trained with MFCCs (Mel-Frequency Cepstral Coefficients). The results showed that all three models using inferred semantics outperformed those based on MFCCs in terms of "hit" rate. Bogdanov et al. (2010) defined "hit" as a situation where the recommended track is not very familiar to the listener, yet is rated highly. While low-level audio features are not subject to cold start, they do not perform well compared to high-level features. One explanation for the poor recommendation performance of models using traditional music features might be the semantic gap (Wang & Wang, 2014). The semantic gap is related to the difficulty in linking low-level features to high-level human concepts, i.e., emotions, understanding, expectations, etc. Celma (2006) also argues that only using audio signals does not allow for bridging the semantic gap.

Embedded metadata in the form of lyrics provide a solution. Lyrics are affected by cold start less than other features, apart from audio data. They are more accessible than UGC, and EGC, and they offer a source of high-level semantic information that might help bridge the semantic gap and improve the performance of content-based filtering recommendations.

2.4 A link between music and poetry

According to Schedl (2019), one of the reasons why music recommendation is a fundamentally different task from that of other services or products is the strong emotions that music evokes in people. Emotion is a high-level human concept that is hard to model using low-level audio features. To do so, one needs to create other meaningful descriptors of music. Zeman et al. (2013) find that the areas of the human brain that send "shivers down the spine" (pp. 140, 148-149) as a reaction to music get aroused as a response to emotionally charged poetry, therefore linking enjoyment of poetry and music on a biological level. Musical lyrics and poetry both share the usage of language. Thus, descriptors for poetry such as rhymes, meter (i.e., the basic rhythmic structure of a verse or lines in a verse, and is characterized by stressed syllables coming at regular intervals), and expressive language can also be used as high-level descriptors of music.

These descriptors are largely important in human perception of poetry. According to Belfi et al. (2018), the vividness of imagery is the most important factor in the appeal towards specific poems. While the vividness of imagery is subjective, it can be inferred from descriptiveness, the strength of sentiment, and lexical richness. Obermeier et al. (2013) argue that meter and rhyme significantly influence emotional responses. As the rules for the identification of meter and rhyme are objective, it is possible to derive meter and rhyme patterns from musical lyrics. Similarly, Kshenovskaya (2020) and Hoffmann (2024) suggest that linguistic creativity enhances the aesthetic experience of poetry as well as its appeal. Linguistic creativity is the ability to produce new linguistic units (words, phrases, or sentences) that are novel and appropriate (Kshenovskaya, 2020). While it is difficult to quantify, it might be possible to infer it through lexical diversity, density, and sophistication. Singhi (2015) uses rhyme, meter, and syllable features to predict whether a song would be successful. He reported that a model using only lyrical features (i.e., rhyme, meter, and syllable features) achieved a higher precision and recall than the model based on the audio features, setting a precedent for lyrical features outperforming audio features in popularity prediction tasks.

In summary, content-based music recommendation approach helps alleviate the cold start problem. However, finding the right set of features and bridging the semantic gap is an issue. An important contribution of this thesis is to identify which lyrical features help improve recommendation quality. In this thesis, I introduce new lyrical features of songs to assess their effect on the performance of music recommender systems. These features build on the link between music and poetry, and, more precisely, on human enjoyment of music and poetry.

3 Data

As discussed above, the focus of this thesis is to assess the effect of lyrical features on the performance of music recommendation systems. Audio features were retrieved from The Million Song Dataset (MSD) (Bertin-Mahieux et al., 2011). Lyrical features are engineered from the lyrics extracted from lyrical databases. User stream counts are extracted from the The Echo Nest Taste Profile Subset, which is also a part of The Million Song Dataset. The Echo Nest Taste Profile Subset includes user streaming data for more than a million users in the form of user-song-play count triplets. A random sample of 2000 users (later shrunk to 1993 users due to low stream count of seven users) is drawn from the subset, and 40627 songs, which the sampled users streamed, are selected for analysis.

Songs with no lyrics are not considered as they are irrelevant to the posed hypothesis. Moreover, songs in any language other than English are not included as well in order to make the engineering of lyrical features feasible.

Below, data sources and how the data was accessed is discussed in greater detail. Also, features and how they were engineered is explained.

3.1 Data sources

Audio features and metadata are extracted from the Million Song Dataset (MSD) (Bertin-Mahieux et al., 2011). MSD is a large dataset containing a million songs and various features related to those songs. It was compiled by a music intelligence and data platform, The Echo Nest, which is currently owned by Spotify, and The Laboratory for the Recognition and Organization of Speech and Audio in 2011. It also includes data on user behavior, user-created song-level tags, etc.

At the moment, a full copy of MSD is only accessible through Amazon Web Services (AWS). In order to extract only relevant data from the dataset, an SQLite database, provided on the Million Song Dataset website, is used to extract track IDs belonging to the relevant songs by matching song IDs from the The Echo Nest Taste Profile Subset and song ids from the database. This step is essential as MSD files containing song features are named based on the track ID of the song. Finally, relevant files containing audio data and metadata are extracted from MSD using a virtual machine and obtained track IDs. The files, which are in an h5 format initially, are processed using the *rhdf5* R package, and relevant features are extracted. Most of these features are not altered and are used in the format they are presented in the dataset. A list of those features, as well as their descriptions, are presented in the next section.

The Million Song Dataset includes lyrics for more than 230 thousand songs in a bags-of-words format. However, due to the nature of some of the lyrical features that are engineered, a bags-of-words format is not sufficient. Full lyrics are extracted from multiple databases (i.e. Musixmatch, Genius, NetEase, LRCLIB, and Deezer) using the *syncedlyrics* Python package (Momeni, 2022). A search for the lyrics is conducted using the song's title and the name of the artist. Before conducting the search, song titles are cleaned by removing parentheses and contents inside of the parentheses in order to omit irrelevant details (e.g. "Live" or "Session Version") which can prevent the song from being found in the databases. As a result, lyrics are obtained in a plain text format.

Finally, 27 lyrical features are calculated using "Rhyme Analyzer" software created by Hirjee and Brown (2009). Rhyme Analyzer calculates "a variety of statistical features about the detected rhymes in the input lyrics, providing a quantitative characterization of rhyming style" (Hirjee, 2009). The application's back-end is public and can be accessed via GitHub. It includes a command line interface (CLI) application which can be used to output several of the lyrical features in the command line. Two modifications of the CLI application were created to retrieve the lyrical features as well as the transcriptions of the lyrics. R was used to pass the lyrics of each song to the created CLI applications and obtain the required data.

3.2 Feature engineering and variable descriptions

Audio and metadata features

The following section provides brief descriptions of audio features retrieved from the Million Song Dataset (MSD). Feature descriptions are taken from the MSD website and adjusted for better explanatory value.

Pitch_1:12 and timbre_1:12 are features related to twelve different pitch and timbre measurements. In MSD, pitch and timbre are given per segment, and each song has a different number of segments. In order to aggregate these measures, an average per feature was taken, which resulted in 12 pitch and timbre features per song. The rest of the features were not altered.

As can be seen, MSD features, apart from “hotttnesss” and “familiarity” features, represent low-level concepts, which, as mentioned above, might be unsuitable for content-based recommendation and do not allow for bridging the semantic gap (i.e., the discrepancy between what can be extracted from music and high-level human perceptions) (Barrington et al., 2007; Bogdanov et al., 2010; Celma, 2006; Wang & Wang, 2014).

Table 1: Audio features and metadata

Source	Variable	Description
MSD	duration	Duration of the track in seconds
MSD	end_of_fade_in	Time of the end of the fade in, at the beginning of the song, according to The Echo Nest
MSD	start_of_fade_out	Start time of the fade out, in seconds, at the end of the song, according to The Echo Nest
MSD	key	Estimation of the key the song is in by The Echo Nest
MSD	mode	Estimation of the mode the song is in by The Echo Nest
MSD	loudness	General loudness of the track
MSD	tempo	Tempo in BPM according to The Echo Nest

Source	Variable	Description
MSD	time_signature	Time signature of the song according to The Echo Nest, i.e. usual number of beats per bar
MSD	song_hotttnesss	Indication of how much attention the song is getting *at the moment* according to The Echo Nest in December 2010, on a scale from 0 to 1
MSD	artist_hottness	Indication of how much attention the artist is getting *at the moment* according to The Echo Nest in December 2010, on a scale from 0 to 1;
MSD	artist_familiarity	Indication of how well-known the artist *generally* is according to The Echo Nest in December 2010, on a scale from 0 to 1
MSD	pitch_1:12	Twelve chroma features (tone, retrieved from the audio signal) averaged across all segments of the song
MSD	timbre_1:12	Twelve MFCC-like (Mel-Frequency Cepstral Coefficients) features averaged across all segments of the song

Lyrical features

A table summarizing lyrical features is presented below. These features were selected based on the conducted review of the literature, both related to recommender systems as well as to literature and poetry. While the chosen set of features is not exhaustive, it extensively covers multiple facets of lyrics: rhythmic structure, metric structure, and lexical richness. Descriptions of features, calculated using Rhyme Analyzer, are taken from the paper written by one of the creators of Rhyme Analyzer, Hirjee (2009). A discussion of the features follows the table.

Table 2: Lyrical features

Variable	Description
Syllables per Line	Average number of syllables per line

Variable	Description
Syllables per Word	Average word length in syllables
Syllable Variation	Standard deviation of line lengths in syllables
Novel Word Proportion	Average percentage of words in the second line not appearing in the first
Rhymes per Line	Average number of detected rhymes per line
Rhymes per Syllable	Average number of detected rhymes per syllable
Rhyme Density	Total number of rhymed syllables divided by total number of syllables
End Pairs per Line	Percentage of lines ending with a line-final rhyme
End Pairs Grown	Percentage of rhyming couplets where the second line is more than 15% longer in syllables than the first
End Pairs Shrunk	Percentage of rhyming couplets where the second line is more than 15% shorter in syllables than the first
End Pairs Even	Percentage of rhyming couplets neither grown nor shrunk
Average End Score	Average similarity score of line-final rhymes
Average End Syl Score	Average similarity score per syllable in line final rhymes
Singles per Rhyme	Percentage of rhymes being one syllable long
Doubles per Rhyme	Percentage of rhymes being two syllables long
Triples per Rhyme	Percentage of rhymes being three syllables long
Quads per Rhyme	Percentage of rhymes being four syllables long
Longs per Rhyme	Percentage of rhymes being longer than four syllables
Perfect Rhymes	Percentage of rhymes with identical vowels and codas
Line Internals per Line	Number of rhymes with both parts falling in the same line divided by the total number of lines
Links per Line	Average number of link rhymes per line
Bridges per Line	Average number of bridge rhymes per line
Compounds per Line	Average number of compound rhymes per line
Chaining per Line	Total number of words or phrases involved in chain rhymes divided by total number of lines
Iambic proportion	Percentage of lines in iambic meter
Trochaic proportion	Percentage of line in trochaic meter
Spondaic proportion	Percentage of line in spondaic meter
Anapestic proportion	Percentage of line in anapestic meter

Variable	Description
Dactylic proportion	Percentage of line in dactylic meter
Amphibrachic proportion	Percentage of line in amphibrachic meter
Pyrrhic proportion	Percentage of line in pyrrhic meter
Number of Lines	Absolute number of lines in the lyrics
Number of Syllables	Absolute number of syllables in the lyrics
Number of Rhymes	Absolute number of rhymes in the lyrics
MTLD score	A measure of textual lexical diversity
Descriptiveness ratio	A proportion of descriptive words out of all words
Polarity	Overall sentiment of the lyrics
Lexical density	Proportion of functional words out of all words
Lexical sophistication	Percentage of words not in 5000 most used words
Iambic meter proportion	Proportion of iambic meter of all meters
Trochaic meter proportion	Proportion of trochaic meter of all meters
Spondaic meter proportion	Proportion of spondaic meter of all meters
Anapestic meter proportion	Proportion of anapestic meter of all meters
Dactylic meter proportion	Proportion of dactylic meter of all meters
Pyrrhic meter proportion	Proportion of pyrrhic meter of all meters
Amphibrachic meter proportion	Proportion of amphibrachic meter of all meters

Lyrics of relevant songs were retrieved from several large databases. Instrumental songs (those with no lyrics) were omitted as their presence could have skewed the results of the experiment.

Songs in languages other than English were also not included in the analysis due to the specificity of the feature engineering methods. For example, as will be discussed in more detail below, features related to rhymes were extracted from a pre-made application, which only supports English. The language of a given lyrics was determined using the *cld3* (Compact Language Detector 3 by Google) package in R.

Rhyme features

Rhymes are a correspondence of sounds between words or the endings of words. As discussed above, 27 features (25 rhyme-related features and two representing a number of lines and a number of syllables) were calculated using Rhyme Analyzer.

Meter

Meter is the basic rhythmic structure of a verse or lines in a verse, and is characterized by stressed syllables coming at regular intervals. In the table below, the stress patterns of all meters used in this thesis are briefly explained (Singhi, 2015).

In poetry, various verse forms prescribe a specific meter or a combination of meters. For instance, the novel Eugene Onegin by Alexandr Pushkin is mostly written in a combination of iambic meters. In musical lyrics, however, there are no strict rules that artists adhere to. Therefore, identifying the "main" meter by finding the most common meter can lead to meaningless results. In this thesis, each lyric is assigned the proportion of each meter out of all meters observed.

Meters are identified using a transcription of the lyrics, which is extracted from the Rhyme Analyzer software. Transcriptions of the lyrics include lexical stress markers, i.e. indication of which syllable is stressed, which can equal zero, one, or two. Zero indicates no stress, while one and two indicate primary and secondary stress respectively. Syllables that are under a secondary stress are considered stressed in this thesis.

Rhyme Analyzer uses the CMU Pronunciation Dictionary (CMU, 2019) in order to produce the transcriptions. The CMU Pronunciation Dictionary contains 134000 words for North American English and was created at the Carnegie Mellon University. While the CMU Pronunciation Dictionary is extensive, it does not account for the context in which the word is pronounced, only providing the correct transcription. However, both in musical lyrics and poetry, words can be pronounced differently from the norm in order to produce a better rhythm. Therefore, this approach imposes a limitation on the quality of the identified metric patterns.

Meter	**Stress pattern**
Iambic	Unstressed + Stressed
Trochaic	Stressed + Unstressed
Spondaic	Stressed + Stressed
Anapestic	Unstressed + Unstressed + Stressed
Dactylic	Stressed + Unstressed + Unstressed
Pyrrhic	Unstressed + Unstressed
Amphibrachic	Unstressed + Stressed + Unstressed

Table 3: Stress patterns of different metrical feet

Measure of textual lexical diversity (MTLD)

"Lexical diversity (LD) refers to the range of different words used in a text, with a greater range indicating a higher diversity" (McCarthy & Jarvis, 2010, p. 381). A lexical diversity measure, namely MTLD, is calculated for each lyric as it, along with other measures, might aid in portraying the linguistic creativity of the artist, which is important for the perception of the lyrics. MTLD is an approach to lexical diversity assessment, proven to correlate highly with established sophisticated approaches and produce consistent results regardless of the length of the text (McCarthy & Jarvis, 2010).

MTLD is calculated by sequentially calculating a text's type-token ratio (i.e. total number of unique words divided by the total number of words) until a specific threshold is reached (TTR of 0.72 by default (McCarthy & Jarvis, 2010)). When the threshold is reached, the TTR evaluation is reset, and the factor count increases by 1. In case at the end of the sentence there are words left that do not form a full factor, they are accounted for by adding a partial factor to the sum of full factors. By definition, the TTR of such a segment is higher than the chosen threshold. A partial factor is then calculated as a percentage of the range between one and the chosen threshold that is filled in by the TTR of a partial segment, going from one to the threshold, divided by 100. For example, a TTR of 0.8 forms 71.4% of the range between 1 and the threshold of 0.72, calculated as $(1 - 0.8)/(1 - 0.72) * 100$ and a value of 0.714 will be added to the sum of full factors. Finally, the total number of words is divided by the factor count to arrive at the MTLD value.

In this thesis, MTLD was calculated using the *koRpus* R package (Michalke, 2021), and the default threshold value of 0.72 was used.

Descriptiveness ratio

Vividness of imagery, as discussed in the literature review, is an important factor in the reason for the appeal of specific poems. While it is inherently subjective, it is reinforced by how descriptive the text is, as descriptions drive the author's ability to convey emotional contexts, and the reader's ability to form mental images of those contexts. Descriptiveness of the text is calculated as a proportion of descriptive words from all words.

Adjectives are used to modify nouns, and adverbs can be used to modify verbs, adjectives, adverbs, or whole sentences. Therefore, to calculate the descriptiveness ratio of a lyrics, the number of adjectives and adverbs is divided by the total number of words in the lyrics. To identify the parts of speech in the lyrics, a pre-trained open-source part-of-speech (POS) tagging model provided by the UDPipe community is used (Wijffels, 2024).

Lexical density

Lexical density measures the complexity of human communication in language (Halliday, 1985). In this thesis, it is used as one of the metrics to infer the author's linguistic creativity from, as linguistic creativity is an abstract and multifaceted concept. To calculate lexical density, the number of lexical units, or functional words, is divided by the total number of words. Functional words are nouns, adjectives, verbs, and adverbs. The calculation of lexical density relies on the same POS tagging model used to calculate the descriptiveness ratio.

Lexical sophistication

Lexical sophistication refers to the use of rare words by the author. In this thesis, a rare word is defined as a word that is not among 5000 most common words in the English language, similarly to the definition of a rare word of a typing assistant Grammarly.

Before rare words can be identified, it is ensured that all words that are present in the lyrics exist in the English language. Lyrics are first lemmatized (i.e., brought to their dictionary form, or lemma) using the *textstem* R package (Rinker, 2018). Afterward, lyrics are matched with a comprehensive list of English words from the SCOWL (Spell Checker Oriented Word Lists) database, and words that are not in the list are omitted. The list includes more than 152000 words.

Most common words are extracted from The Corpus of Contemporary American English or COCA (Davies, 2015). COCA provides word frequencies, or lemma frequencies, for the most common 5050 words in the English language. Finally, lexical sophistication is calculated as a proportion of non-common words in the total word count of a lyrics.

Polarity

Sentiment score, or polarity, is a measure of the sentiment of a document. A negative polarity value indicates a negative sentiment, a positive polarity value indicates a positive sentiment, and, finally, a polarity of zero indicates that the document is neutral in terms of the sentiment. The strength of the sentiment is a relevant variable as it can refer to the vividness of imagery, portrayed by the lyrics.

Polarity is calculated using the *qdap* R package (Rinker, 2023). Each word in the document is assigned a polarity score of one, negative one, or zero based on whether the word is tagged as positive, negative, or neutral (e.g., the word “cheer” has a positive sentiment, and the word “chore” has a negative sentiment, and the word “item” is neutral). A sentiment dictionary by Hu and Liu (2004) is used to tag polarized words with those values. Around each polarized word, a window of six words (four before and two after) is investigated to find valence shifters. Valence shifters affect the score of the polarized words: negators (e.g., not, didn’t, isn’t) reverse the score of a word, amplifiers (e.g., quite, really, very) increase the intensity of a word, and deamplifiers (e.g., barely, only, rarely) decrease the intensity of a word. Amplifiers and deamplifiers cause the polarity of the word to be multiplied by 1.8 or by 0.2 respectively.

The final polarity score (δ) is calculated as:

$$\delta = \frac{x_i^T}{\sqrt{n}}$$

Where:

$$x_i^T = \sum \left((1 + c(x_i^A - x_i^D)) \cdot w(-1) \sum x_i^N \right)$$

Individual components:

$$x_i^A = \sum (w_{neg} \cdot x_i^a)$$

$$x_i^D = \max(x_i^{D'}, -1)$$

$$x_i^{D'} = \sum (-w_{neg} \cdot x_i^a + x_i^d)$$

$$w_{neg} = \left(\sum x_i^N \right) \mod 2$$

Where:

- x_i^T is the total adjusted sentiment score for a given polarized word,
- x_i^A is the sum of amplifier weights,
- x_i^D equals the adjusted de-amplifier value,
- $x_i^{D'}$ refers to the intermediate de-amplifier value calculation,
- w_{neg} is the negator weight,
- $\sum x_i^N$ is the sum of negator values,
- c is the weight for amplifiers and de-amplifiers (i.e., 0.8),
- w is the weight of a polarized word based on the dictionary,
- n indicates the word count.

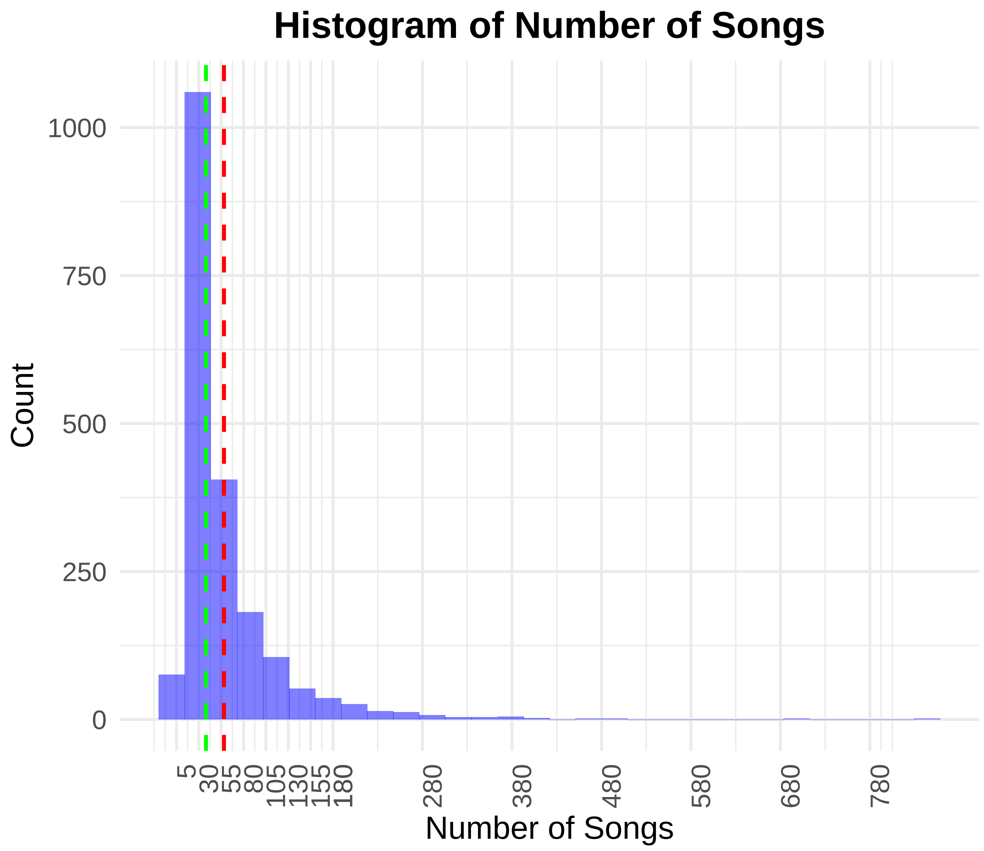
Polarity is calculated per each lyrics, or document. Since the polarity algorithm, implemented in *qdap*, treats documents as large sentences, removing and ignoring periods, it can lead to unexpected results. For instance, the polarity of "very. Good" is equal to that of "very good". In the first example, the word "Good" refers to the second sentence and should not be amplified by the word “very” which comes from the sentence before. Commas, however, do separate the context clusters, and, therefore, all periods are replaced with commas in order to ensure a correct calculation of the polarity score.

3.3 Descriptive analytics

In this section, descriptive analytics is presented in order to gain insight into the data at hand. Given a large number of features, only some are going to be discussed.

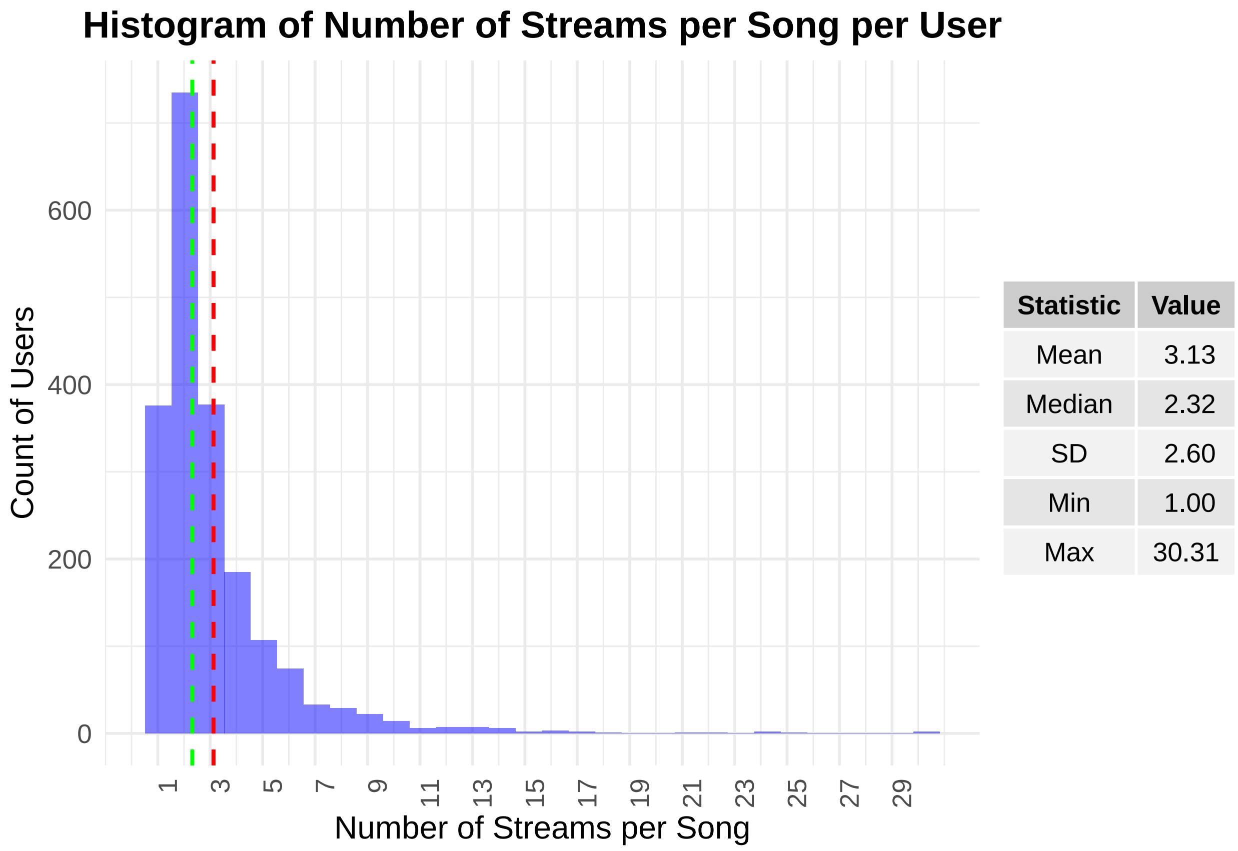
A histogram of the total number of unique songs streamed by each user is shown below. The red line represents the average number of songs streamed, and the green line represents the median number of songs streamed. On the right, a table with summary statistics is presented.

An average user streamed a total of 58 unique songs. The distribution is skewed to the right, therefore median is lower and equals 38. The most active user streamed 849 unique songs. The variation is very large with a standard deviation higher than the mean, yet it represents real user behavior. It is important to mention that the values below correspond to the sample. The population mean is lower and equal to 47 songs, and the population standard deviation is around the same and equal to 57.



Statistic	Value
Mean	58.15
Median	38.00
SD	60.03
Min	5.00
Max	849.00

In the second histogram depicts the average number of streams per song. Red and green lines indicate the mean and median, respectively. Most users in the sample streamed a song, on average, around three times, while, as can be seen from the minimum, some users streamed all the songs they listened to only once. Once again, variation of the metric is high and indicates largely varying user behaviors. The population mean is around the same and is equal to 2.9.

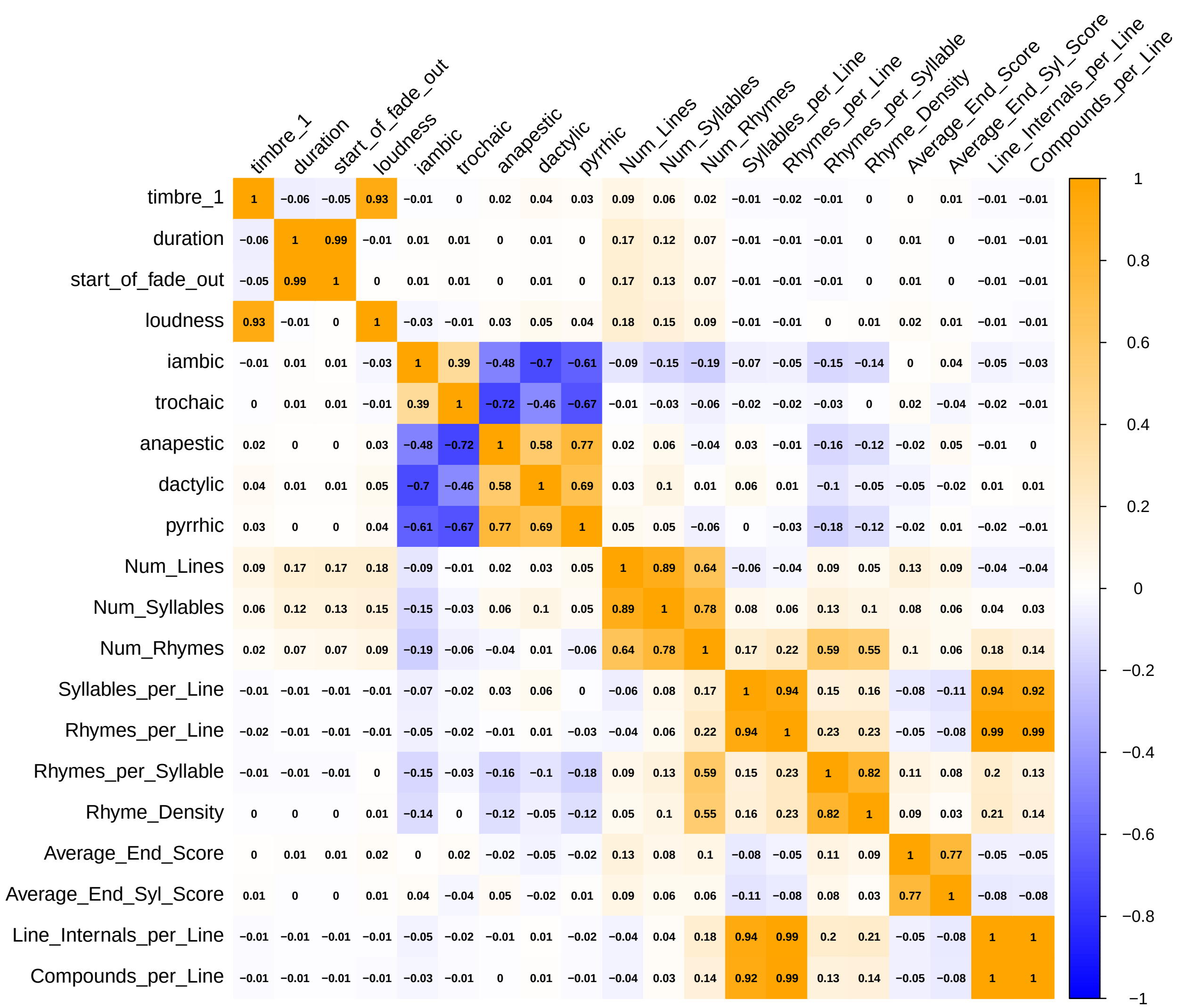


Statistic	Value
Mean	3.13
Median	2.32
SD	2.60
Min	1.00
Max	30.31

Due to the large number of features, a correlation table depicted below only features variables which correlate strongly (absolute correlation coefficient above 0.7) with at least one other variable. The numbers inside the squares refer to the Pearson correlation coefficient. Pearson correlation coefficient measures linear correlation between sets of data and ranges from -1 to 1, where -1 indicates a perfect negative linear relationship between the variables, and 1 indicates a perfect positive linear relationship.

Noticeably, correlations between the meter proportions are rather strong and logically consistent. For instance, the iambic meter, which follows a pattern of a stressed syllable following an unstressed syllable, correlates negatively with anapestic, dactylic, and pyrrhic meters, none of which correspond to that pattern. Also, it looks like one of the twelve timbre variables represents loudness as they are almost perfectly correlated. Generally, there are many strong correlations between rhyming features, such as, for instance, rhymes per line, syllables per line, compounds per line, and line internals per line. This makes sense as compounds per line represent compound rhymes, which are characterized by containing two or more syllables, therefore increasing the number of syllables. Line internals per line, i.e., the number of rhymes that fit on the same line divided by the number of lines, are, by definition, connected to rhymes per line.

High Correlations in Song Features



Finally, a short table with summary statistics of some features is presented below. Only features that can be intuitively understood and assessed are included, so, for example, timbre and pitch features are excluded as they represent low-level audio signals, and their values are not meaningful without context.

It can be noticed that some variables have a high standard deviation, such as, for example, the number of syllables. In fact, the maximum number of syllables of 11434 refers to the song by DJ Kay Slay, which lasts for 40 minutes and includes 110 verses. The song with 1102 syllables per line is actually a track by a comedian Lewis Black, and, while it is just a clipped recording of his show, it is considered a song. This table shows the large variability in the content as well as potential data issues (e.g., a live show recording labelled as a song in music databases).

Table 4: Summary statistics

	N	Mean	SD	Median	Min	Max
duration	40627	242.20	91.72	230.16	2.17	3024.67
end_of_fade_in	40627	0.81	3.26	0.18	0.00	399.99
start_of_fade_out	40627	232.86	89.78	220.65	2.17	3020.30
MTLD_score	40627	42.14	27.12	36.51	2.57	468.23
descriptiveness_ratio	40627	0.12	0.05	0.11	0.00	0.65
polarity	40627	0.00	0.81	0.00	-14.27	29.00
lexical_density	40627	0.43	0.06	0.43	0.00	0.89
lexical_sophistication	40627	0.36	0.30	0.28	0.00	3.03
iambic	40627	0.27	0.03	0.27	0.00	1.00
trochaic	40627	0.25	0.03	0.25	0.00	0.69
spondaic	40627	0.10	0.05	0.09	0.00	0.93
anapestic	40627	0.08	0.02	0.08	0.00	0.32
dactylic	40627	0.07	0.02	0.07	0.00	0.24
pyrrhic	40627	0.10	0.03	0.10	0.00	0.43
amphibrachic	40627	0.11	0.02	0.12	0.00	0.28
Num_Lines	40627	40.08	24.00	36.00	1.00	967.00
Num_Syllables	40627	319.71	241.96	274.00	6.00	11434.00
Num_Rhymes	40627	47.95	58.08	34.00	1.00	2475.00
Syllables_per_Line	40627	8.41	12.02	7.70	1.59	1102.00
Syllables_per_Word	40627	1.24	0.37	1.22	0.87	29.62

4 Method

In order to assess the impact of the engineered lyrical features on the performance of music recommender systems, a content-based recommendation system was built. Below, the design of the model is justified and explained, the metric used to assess the results is presented, and the experiment setup is described.

A content-based music recommendation system provides recommendations utilizing exclusively the information about the attributes of the songs consumed by a user. In the data section, audio and lyrical features to be used to make recommendations were described. Moreover, it was mentioned, that stream count is treated as an indicator of a user's preferences. A content-based recommendation system was built in accordance with the design proposed by Leskovec et al. (2020): attributes of the items that the user had interacted with were aggregated to form a vector, indicating the user's preferences and cosine similarity was calculated between the user's preference vector, i.e., user profile, and song attribute vectors, i.e. song profiles.

Cosine similarity was chosen to generate recommendations for the following reasons:

1. When creating regression models in a high-dimensional space, one needs an appropriate number of observations in order to avoid overfitting. As shown in the data section, the number of songs a user streamed can be as low as 5, while more than 30 features are used when generating recommendations in each experimental setup. While utilizing tuning and dimensionality reduction to avoid overfitting is possible, it presents significant limitations in terms of efficiency. An approach featuring vector similarity measures, however, is less prone to overfitting and does not suffer as much as the former from a large number of features combined with a small number of observations, also sometimes referred to as the Curse of Dimensionality.

2. As many experiments need to be run on large amounts of data, cosine similarity provides a more feasible solution compared to regression or classification models. Moreover, cosine similarity is less computationally expensive in high dimensions than, for instance, Euclidean distance.

3. Cosine similarity is a popular choice in content-based recommender systems literature and research (Gorakala & Usuelli, 2015; Leskovec et al., 2020; Niyazov et al., 2021).

Cosine similarity is the cosine of the angle between two vectors in a multi-dimensional space. Cosine similarity between two n-dimensional vectors A and B is represented by:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where n is the number of dimensions, A_i and B_i are the ith components of vectors $\mathbf{A}$ and $\mathbf{B}$ respectively. Cosine similarity ranges from -1 to 1, where 1 indicates that the vectors point in the same direction, or the angle between them is equal to 0 degrees, -1 indicates that the vectors are opposite, or that the angle between them is 180 degrees, and 0 indicates that the vectors are perpendicular. In the context of this thesis, the higher the cosine similarity between a user's preferences and song attributes, the more similar they are.

4.1 Recommender system design

Below is a high-level conceptual model representing the structure of the recommendation system to be built. The explanation of the elements, in order, is the following:

1. The recommendation system takes historical user activity data as input. User preferences are inferred from observed user behavior. User activity data is formatted as a vector populated with the count of streams per song, and is called a user utility vector.

2. Two groups of data are extracted: audio data and metadata, extracted from the Million Song Dataset, and lyrics as plain text, from various lyrical databases (i.e. Musixmatch, Genius, NetEase, LRCLIB, Deezer).

3. Audio and metadata from the Million Song Dataset do not need to be processed, while lyrics are used to create lyrical features (e.g. lyrical sophistication, rhyme features, etc.). Combined, these constitute a table with song features. Each row represents a vector of features for a specific song and is called a song profile, or an item profile. The song feature table is then standardized to ensure that no feature dominates the calculation of the cosine similarity due to differences in the scales of the features.

4. Then, the user profile is created by averaging song features, i.e., song profiles, weighted by the stream counts, i.e., user utility vector. This step ensures that features belonging to songs that the user streamed more have a higher impact on the final user preferences vector. The user profile represents the aggregate musical preference of the user.

5. Finally, cosine similarity is calculated between feature vectors of all songs the user has not listened to (and that are not used for training, in case of testing), and the user profile. A sorted list of songs is returned, starting with the highest cosine similarity. Those songs on top of the list are assumed to be most likely preferred by the user.

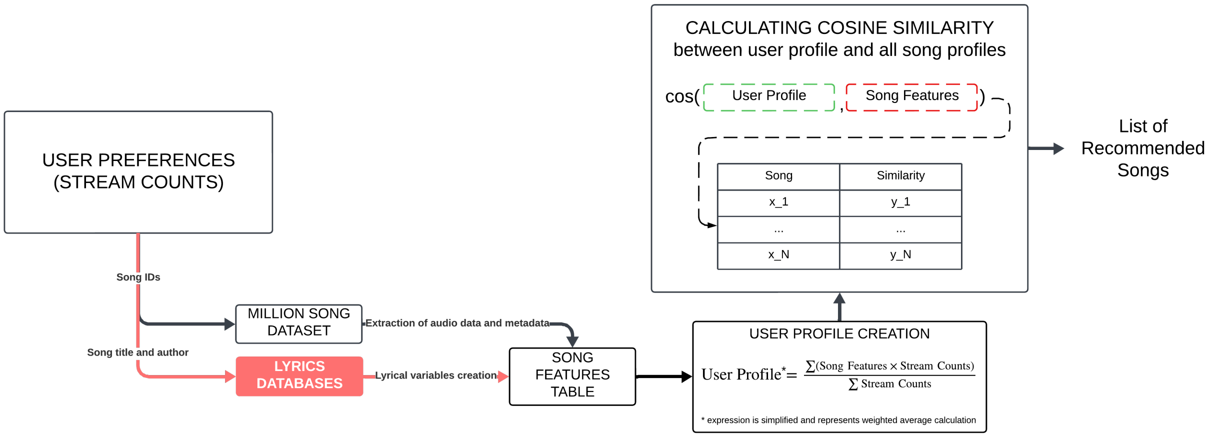

CALCULATING COSINE SIMILARITY
between user profile and all song profiles
cos(User Profile , Song Features)
Song
Similarity
x_1
y_1
...
...
x_N
y_N
List of Recommended Songs
USER PREFERENCES (STREAM COUNTS)
Song IDs
Song title and author
MILLION SONG DATASET
Extraction of audio data and metadata
LYRICS DATABASES
Lyrical variables creation
SONG FEATURES TABLE
USER PROFILE CREATION
User Profile*= ∑(Song Features × Stream Counts) / ∑ Stream Counts
* expression is simplified and represents weighted average calculation

Figure 2: Conceptual Model

4.2 Model performance assessment

Mean average precision at 500 (MAP@500) was chosen as a metric to assess and compare the performances of various iterations of the recommender system. MAP@500 was also used as the evaluation measure in the Million Song Dataset Challenge, which posed a similar challenge of predicting users' missing listening history based on available streaming data. K was set to 500 in order to capture most users' entire libraries.

Mean average precision at K is calculated as following:

$$\text{MAP@K} = \frac{1}{U}\sum_{u=1}^{U} \text{AP@K}_u$$

Where:

- K represents the chosen cutoff point,
- U is the total number of users,
- AP is the average precision.

Average precision at K is defined as:

$$\text{AP@K} = \frac{1}{N} \sum_{k=1}^{K} \text{Precision}(k) \times \text{rel}(k)$$

Where:

- N is the total number of relevant items for a particular user,
- Precision(k) is the precision calculated at each item position,
- rel(k) is an indicator of the relevance of the item, equals 1 if the item at position k is relevant, and 0 if not.

Finally, precision at K is calculated as:

$$\text{Precision@}K = \frac{\text{Total number of relevant items in } K}{\text{Total number of items in } K}$$

Where K is the cutoff point.

To summarize, MAP@K evaluates the quality of the recommendations by both, measuring the relevance of the recommended items as well as their placement. MAP@K ranges from 0 to 1, where 1 can be achieved by placing all the relevant items as close to the top of the list as possible. It is important to mention that relevant items are all assumed to have equal relevance, and, therefore, interchanging the items themselves within the recommendation does not have an effect on the MAP@K.

4.3 Experimental setup

In order to conclude that lyrical features can have a positive impact on the performance of the recommender system that was built, at least a single feature that significantly improves the performance of the baseline model needs to be found. To do so, several iterations of the recommender system were created. In order to achieve a more robust result, MAP@K was cross-validated. Each user's streaming library was divided into five folds, and users with less than five unique streamed songs were omitted. Four folds were used for training, and one fold was used for testing. Each user's final AP@K was calculated as an average of AP@Ks across all five folds.

The following models were created and assessed: a baseline model employing exclusively audio features retrieved from The Million Song Dataset. A model using all features available, both audio and lyrical. A model per lyrical feature, which used all baseline features as well as one of the lyrical features. Finally, to ensure the model built is relevant, its performance is also compared to the recommender system that gives random recommendations.

5 Results

A content-based music recommendation system that uses user stream counts and cosine similarity to produce recommendations was built and tested with multiple variations of features. Table 5 shows train and test MAP@K for four models. It includes a baseline model, built on audio features only and referred to as "audio", a model built solely on lyrical features and referred to as "lyrical", a model built on both sets of features and referred to as "all variables", and a model that gives random recommendations.

As can be seen, models using only lyrical data tend to perform worse than models using audio features and models using a combination of features. On train data, the highest MAP@K of 0.0389 is achieved by using both lyrical and audio features. However, on test data audio features alone perform better than a combination of lyrical and audio features. It might be due to the model failing to generalize to unseen data. In fact, lyrical features might have introduced noise that decreased the performance of the combined feature set to be lower than that of audio features only. Random recommendations, on the other hand, yielded a MAP@K of zero.

Table 5: Models featuring full sets of variables

Data	Feature set	MAP@K
train	audio	0.0249
	lyrics	0.0136
	audio_and_lyrics	0.0389
test	audio	0.0039
	lyrics	0.0007
	audio_and_lyrics	0.0038
random		0.0000

Table 6 shows the MAP@K of an audio feature set combined with individual lyrical features, on test data. Features marked in bold are those which, in combination with audio features, yield a result higher than that of audio features only on test data, and features with a cyan background are those which, according to a paired t-test, increase MAP@K on a statistically significant level. Therefore, combining audio features with the MTLD score or lexical sophistication score increases MAP@K by approximately 7.7%.

Table 6: Models featuring single lyrical features

	MAP@K
MTLD_score	**0.0042**
descriptiveness_ratio	**0.0040**
polarity	0.0039
lexical_density	**0.0040**
lexical_sophistication	**0.0042**
Num_Lines	**0.0040**
Num_Syllables	**0.0040**
Syllables_per_Line	**0.0039**
Syllables_per_Word	**0.0040**
Novel_Word_Proportion	**0.0039**
Singles_per_Rhyme	**0.0040**
Doubles_per_Rhyme	0.0038
Quads_per_Rhyme	0.0039
Longs_per_Rhyme	**0.0040**
Line_Internals_per_Line	0.0039
Links_per_Line	**0.0040**
Bridges_per_Line	**0.0039**
Compounds_per_Line	0.0039
Chaining_per_Line	0.0039
iambic	**0.0039**
trochaic	**0.0039**
spondaic	**0.0040**
anapestic	0.0039
dactylic	0.0039
pyrrhic	0.0039
amphibrachic	0.0039
Perfect_Rhymes	0.0039
Rhyme_Density	0.0039
Triples_per_Rhyme	**0.0039**
Num_Rhymes	**0.0039**
Syllable_Variation	**0.0039**
Rhymes_per_Line	**0.0039**
Rhymes_per_Syllable	0.0038
End_Pairs_per_Line	**0.0039**
End_Pairs_Grown	0.0038
End_Pairs_Shrunk	0.0038
End_Pairs_Even	0.0038
Average_End_Score	0.0038
Average_End_Syl_Score	**0.0039**

Table 7 shows the MAP@K of an audio feature set compared to that of a combined set of audio features and those lyrical features that statistically significantly improved MAP@K as shown in Table 5. An achieved MAP@K of 0.0045 indicates an increase of 15.4% to the MAP@K of exclusively audio features.

Table 7: Models featuring audio and best lyrical features

Feature set	MAP@K
audio	0.0039
audio_and_best_lyrical	0.0045

6 Conclusion

The research objective of this thesis was to investigate the impact of lyrical features on content-based music recommendation systems. To achieve this objective, 39 unique lyrical features were introduced into a content-based music recommendation system and results were evaluated across 1993 users and 40627 songs.

6.1 Discussion

Including all 39 generated features into a content-based music recommendation system did not yield a positive result, and, in fact, decreased the mean average precision. However, four features were found to statistically significantly increase MAP@K: MTLD score, which is a measure of textual lexical diversity; lexical sophistication, which is the percentage of the lyrics that consists of words that are not in 5000 most used words in the English language; number of syllables, which is the total number of syllables in the lyrics, and syllables per word, which is the average number of syllables per word in the lyrics. Combining these features with audio variables produces an increase in MAP@K of 15.4%. Therefore, it can be concluded that poetic lyrical features can improve the performance of content-based music recommendation systems. Including such features can increase the quality of the representation of a user's preferences, and, therefore, increase the precision with which relevant songs are recommended.

Notably, MTLD score and lexical sophistication, if taken separately, increased MAP@K to approximately 0.0042, which is the highest increase among all variables. Both MTLD score and lexical sophistication can be categorized as measures of lexical richness: how unique and non-repetitive the lyrics are, and how uncommon the words in the lyrics are. It can be then stated that users seem to have a stronger preference for lexical richness than other features when it comes to their music taste.

At the same time, no features that directly represent rhyme, meter, or vividness of imagery were found to significantly improve MAP@K. While the number of syllables and syllables per word were in fact statistically significant and introduced by the Rhyme Analyzer software, they are not directly related to rhyme, i.e., correspondence of sounds that is meant to enhance the listener's experience. Intuitively, number of syllables and syllables per word should not directly be in a user's preferences but can represent a higher-level preference. For example, there is a positive correlation between the number of syllables and lexical sophistication.

6.2 Contributions to theory and practice

This thesis contributes to the existing literature on content-based music recommendation systems and specifically on the issue of cold start (i.e., the inability of CF recommender systems to make predictions for items with little data) in music recommendation. It presents a new approach in tackling the cold start problem and bridging the semantic gap (i.e., the discrepancy between what can be extracted from music and high-level human perceptions) by introducing new lyrical features that can be used to increase the performance of content-based music recommendation systems.

The results of this thesis also carry practical implications for companies that have to do with music recommendation. As it was demonstrated that the inclusion of lyrical features into content-based recommendation systems improves their performance, companies can improve the user experience they provide as well as their position in the market through a relatively undemanding adjustment.

6.3 Limitations and future research

Arguably, the most challenging limitation of this thesis lies in the features it aims to introduce. As mentioned in the literature review, modelling the vividness of imagery or lexical creativity is rather difficult, as such concepts are subjective and multifaceted. Features that were meant to represent the vividness of imagery did not prove to significantly improve MAP@K, and it can be argued that the reason for that is poor choice of features. Future research is necessary to identify whether and how such concepts can be modelled to make use of them more effectively.

Next, the approach employed in this thesis aggregates user profiles into a single vector which is meant to represent a user's set of preferences. However, it is somewhat obvious how such an approach can yield unreasonable results: let's assume a user listened to songs that are more or less polar opposites of each other the same number of times. Then, a vector representing his or her preferences is going to lie in the middle, essentially failing to account for the user's varying and unique taste.

Finally, the approach used in this thesis applies the same feature set to all users, disregarding what they might consider more or less important. People's preferences in music are driven by different factors (sometimes, audio and lyrics are even proposed to be the opposites of each other in that sense), and, therefore, it makes sense to account for such individual differences, algorithmically or through user feedback. Further research could focus on the identification of the strength of the user preferences and their implementation in content-based music recommendation systems.

References

Barrington, L., Chan, A., Turnbull, D., & Lanckriet, G. (2007, April). Audio information retrieval using semantic similarity. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07.* https://doi.org/10.1109/icassp.2007.366338

Belfi, A. M., Vessel, E. A., & Starr, G. G. (2018). Individual ratings of vividness predict aesthetic appeal in poetry. *Psychology of Aesthetics, Creativity, and the Arts*, *12*(3), 341–350. https://doi.org/10.1037/aca0000153

Ben Sassi, I., Ben Yahia, S., & Liiv, I. (2021). MORec: At the crossroads of context-aware and multi-criteria decision making for online music recommendation. *Expert Systems with Applications*, *183*, 115375. https://doi.org/10.1016/j.eswa.2021.115375

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The million song dataset. *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011).*

Bogdanov, D., Haro, M., Fuhrmann, F., Gómez, E., & Herrera, P. (2010). *Content-based music recommendation based on user preference examples. 633.*

Bogdanov, D., Haro, M., Fuhrmann, F., Xambó, A., Gómez, E., & Herrera, P. (2013). Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing &Amp; Management*, *49*(1), 13–33. https://doi.org/10.1016/j.ipm.2012.06.004

Brooks, B. C. (2023, November 29). *Spotify wrapped 2023: "Music genres are now irrelevant to fans"*. BBC News. https://www.bbc.com/news/entertainment-arts-67111517

Celma, Ò. (2006). Foafing the music: Bridging the semantic gap in music recommendation. In *The semantic web - ISWC 2006* (pp. 927–934). Springer Berlin Heidelberg. https://doi.org/10.1007/11926078_67

Celma, Ò. (2010). *Music recommendation and discovery: The long tail, long fail, and long play in the digital music space.* Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13287-2

CMU. (2019). *The CMU pronouncing dictionary.* www.speech.cs.cmu.edu/cgi-bin/cmudict

Davies, M. Y. (2015). *Corpus of Contemporary American English (COCA).* https://doi.org/10.7910/dvn/amuduw

Deldjoo, Y., Schedl, M., & Knees, P. (2024). Content-driven music recommendation: Evolution, state of the art, and challenges. *Computer Science Review*, *51*, 100618. https://doi.org/10.1016/j.cosrev.2024.100618

Gorakala, S. K., & Usuelli, M. (2015). *Building a Recommendation System with R.* https://dl.acm.org/citation.cfm?id=2987890

Halliday, M. A. K. (1985). *Spoken and written language.* UNSW Press.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, 241–250.

Hirjee, D., Hussein & Brown. (2009). Automatic detection of internal and imperfect rhymes in rap lyrics. *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009*, 711–716.

Hoffmann, T. (2024). *Cognitive approaches to linguistic creativity.* https://doi.org/10.33774/coe-2024-8tnps

IFPI. (2023). *Global music report 2023.* IFPI. https://www.ifpi.org/wp-content/uploads/2020/03/Global_Music_Report_2023_State_of_the_Industry.pdf

Ko, S., Kim J., K. T., & Choi J., L. J. (2021). *Discovery of user preference in recommendation system through combining collaborative filtering and content based filtering.* 684–695. http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBGHIF_2001_v7n6_684

Kshenovskaya, U. (2020, August). Linguistic creativity: Cognitive and communicative aspects. *European Proceedings of Social and Behavioural Sciences.* https://doi.org/10.15405/epsbs.2020.08.98

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive datasets* (3rd ed.). Cambridge University Press.

Luminate year-end music report. (2023). Luminate. https://luminatedata.com/wp-content/uploads/2024/01/Luminate_Year-End_Report_2023.pdf

McCarthy, P. M., & Jarvis, S. (2010). MTLD, vocd-d, and HD-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, *42*(2), 381–392. https://doi.org/10.3758/brm.42.2.381

Michalke, M. (2021). *koRpus: Text analysis with emphasis on POS tagging, readability, and lexical diversity.* https://reaktanz.de/?c=hacking&s=koRpus

Momeni, M. (2022). Syncedlyrics. In *GitHub repository.* https://github.com/moehmeni/syncedlyrics; GitHub.

Niyazov, A., Mikhailova, E., & Egorova, O. (2021). Content-based music recommendation system. *2021 29th Conference of Open Innovations Association (FRUCT)*, 274–279. https://doi.org/10.23919/FRUCT52173.2021.9435533

Obermeier, C., Menninghaus, W., Koppenfels, M. von, Raettig, T., Schmidt-Kassow, M., Otterbein, S., & Kotz, S. A. (2013). Aesthetic and emotional effects of meter and rhyme in poetry. *Frontiers in Psychology*, *4*. https://doi.org/10.3389/fpsyg.2013.00010

Rinker, T. W. (2018). *textstem: Tools for stemming and lemmatizing text.* http://github.com/trinker/textstem

Rinker, T. W. (2023). *qdap: Quantitative discourse analysis package.* https://github.com/trinker/qdap

Schedl, M. (2019). Deep learning in music recommendation systems. *Frontiers in Applied Mathematics and Statistics*, *5*. https://doi.org/10.3389/fams.2019.00044

Schwartz, B., Ward, A., Monterosso, J., Lyubomirsky, S., White, K., & Lehman, D. (2002). Maximizing versus satisficing: Happiness is a matter of choice. *Journal of Personality and Social Psychology*, *83*, 1178–1197. https://doi.org/10.1037/0022-3514.83.5.1178

Singhi, A. (2015). *Lyrics matter: Using lyrics to solve music information retrieval tasks.* https://api.semanticscholar.org/CorpusID:73655885

Slaney, M. (2011). Web-scale multimedia analysis: Does content matter? *IEEE Multimedia*, *18*(2), 12–15. https://doi.org/10.1109/mmul.2011.34

Starr, G. G. (2014). *Choice Reviews Online*, *51*(09), 51-4959-51-4959. https://doi.org/10.5860/choice.51-4959

Van Meteren, R., & Van Someren, M. (2000). Using content-based filtering for recommendation. *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, *30*, 47–56.

Wang, X., & Wang, Y. (2014, November). Improving content-based and hybrid music recommendation using deep learning. *Proceedings of the 22nd ACM International Conference on Multimedia.* https://doi.org/10.1145/2647868.2654940

Wijffels, J. (2024). *Udpipe: Tokenization, parts of speech tagging, lemmatization and dependency parsing with the 'UDPipe' 'NLP' toolkit.* https://github.com/bnosac/udpipe

Zeman, A., Milton, F., Smith, A., & Rylance, R. (2013). By heart an fMRI study of brain activation by poetry and prose. *Journal of Consciousness Studies*, *20.*