FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

SOCIAL MEDIA AND WEB ANALYTICS

Report

Group 1

Cenk Can
Felix De Poorter
Gust Bossuyt
Juntian Si
Iris Cappoen
Helena Peters
Ingmar Van Laethem

Prof. Dr. M. Bogaert

UNIVERSITEIT GENT

2022-2023

Contents

1 Targeting social media users using restricted information

1.1 Introduction

Social media has transformed the way we communicate and share information, making it an integral part of our daily lives. With users following other users, social media platforms create a network of interconnected individuals that can be analyzed using network analysis. By examining this network structure, we can identify influential followers who have the potential to reach a wide audience with information dissemination. In this study, we present a methodology for identifying influential followers on social media through network analysis, with a focus on Mastodon.

1.2 Part 1

Based on Dirk Van den Poel's network of 138 followers, we developed a five-step methodology to identify influential followers on Mastodon, as detailed in this study. To retrieve all of Dirk Van den Poel's followers, we used the rtoot package. We then expanded the network by retrieving the followers of these followers to obtain a complete picture of the follower network. In the third step, we created an adjacency matrix to represent the relationships between users, using the igraph and statnet packages. By calculating the degree of each node in the network, we could determine the number of followers each user had. Next, we ranked followers based on their degree and extracted the degree of only Dirk Van den Poel's followers to establish a ground truth rank. Finally, we created a data frame containing usernames, degree, and ground truth rank, and saved the ground truth as "ground_truth." Figure 2 illustrating Dirk Van den Poel's network can be found in Appendix 1.

1.3 Part 2

In the second part of our study, we sought to ascertain the ranking of a subset of the network by limiting the analysis to only two followers of the followers and gradually expanding it until the complete network was considered. Following this, we leveraged this ranking alongside that of our ground truth to compute the Spearman's rank-based correlation coefficient. Furthermore, we generated a visual representation in the form of a plot to aid in the assessment of the required network size for predicting the ranking of the followers. This plot is available in Figure 3 of the Appendix. The x-axis represents the size of the network used to compute the degree and the y-axis represents the rank-based correlation. The plot reveals that if we can extract roughly 30 percent of the followers of the followers, our predictions will be highly reliable.

1.4 Conclusion

In conclusion, our study presented a comprehensive methodology for identifying influential followers on Mastodon and analyzing social media networks. By ranking followers based on their degree, we established a ground truth and calculated the Spearman's rank-based correlation coefficient to determine the reliability of our predictions. The results from this study can be useful in various applications, such as social media marketing, political campaigns, and influencer outreach. Our methodology can be adapted to suit specific research questions, making it a valuable tool for analyzing social media platforms. Overall, this study highlights the potential of network analysis for understanding social media dynamics and identifying influential users who can shape the flow of information.

2 Predict a performance indicator from a certain sector by using social media and web data

2.1 Part 1: Descriptive

2.1.1 Youtube API

2.1.1.1 Introduction

Can you make predictions about the popularity of an upcoming music album based on reactions on social media? We researched whether it is possible. We chose to analyze Metallica's new album, 72 Seasons. This seems like an ideal choice since the album was only released April 14, and second because Metallica is hugely popular. This is also evident from The Heaviest List, an all-time heavy metal hit list released by Studio Brussel this week, where Metallica is thus ranked #1. So we knew we would get plenty of data, since Metallica's popularity is immense. We chose to analyze the Youtube comments on 3 songs already released in this new album, namely 72 seasons, If Darkness Had A Son and Lux Aeterna. We did a scraping of the comments on April 7, for all three songs. The rest of the album has been released on April 14.

To have a better understanding of this part we made a wordcloud, wordgraph, topic modelling and sentiment analysis for this descriptive part. At the beginning we started scraping the comments on the 3 recent videos, LuxAeterna, IfDarknessHadASon and 72 Seasons, using the Youtube Data API of Google. We scraped all the comments since the time these songs were released. The latest song '72 Seasons' has been released on March 30th while the earliest one has been released last year. This implies that there is a difference in time since the release date for each of our three songs and caused a bias when comparing these songs' absolute number of comments and views. The input text corpus is filtered to include only English-language texts. After that emojis, emoticons, contractions, internet slang, word elongations, and unnecessary spaces and punctuation are removed before we proceed to the next part.

2.1.1.2 Wordcloud

In the wordcloud part we created a document term matrix (DTM) from the preprocessed text using the text clean and tm packages. We also made a table counting the frequency of the words in the text and we used the function to anti the stopwords in English. Then we transformed the DTM into a term document matrix (TDM) and a word frequency table is generated from it. We used the dplyr package to remove the words such as 'Metallica' and 'Album' and also removed the null value during creating the matrix, as these didn't add much value to our analysis. The three wordclouds are attached in the appendix in Figures 4, 5 and 6.

A quick look at all three wordclouds give insight in the sentiment of the listeners about the songs. You can conclude that most words indicate a positive feedback, such as 'love', 'awesome' and 'amazing'. It is also worth mentioning that in the context of heavy metal, words as 'fire', 'wrath' and such indicate positive feedback. All wordclouds also show the word 'James', which is the name of the very popular lead singer and artist. Bad feedback is almost non-existing in the wordclouds, but you can find some if you look closely.

2.1.1.3 Wordgraph

In the wordgraph we also used the DTM matrix created in the wordcloud. We used the create_adjacency_matrix function. The create_adjacency_matrix function takes a document term matrix as input, creates an adjacency matrix, and applies filtering based on the degree of the vertices. Then we used the function plot_network, input the DTM and transform it into a graph with clusters. In the wordgraphs, colors and lines serve to represent different relationships between words. Colors can be used to represent different clusters or topics, while lines indicate co-occurrence or proximity between words.

When analyzing some of the clusters, we can see the following. All wordgraphs include a cluster with general positive feedback. 2 of the wordgraphs also include a cluster that combines words about the sound, guitar and riffs with James, the lead artist and guitarist as mentioned earlier. This indicates again the popularity of him and the talent people see in his instrumental abilities. The three wordgraphs can be found in Figure 7, 8 and 9.

2.1.1.4 Topic Modelling

In the topic modelling part we used the LDA algorithm. Initially we created a loop to generate multiple LDA models for different number of topics ranging from 2 to 10. We set a seed for the LDA algorithm to ensure reproducibility of the results then used AIC values to test the performance of the models and selected the model with the smallest AIC as the final model.

After fitting the final LDA model, we used the "tidy" function to obtain the beta matrix which contains the probabilities of each term being associated with each topic. Then we selected the top 10 terms per topic, arranged them in descending order of their probabilities, and visualized them using a bar chart with one facet per topic. The results are attached in Figure 10, 11 and 12.

2.1.1.5 Sentiment Analysis

Finally, we performed a sentiment analysis. In this part we firstly applied lemmatization with the textstem package, creating a dictionary from the text and storing it into text_final. Then we extract the sentiment from the text_final and store it into sentiment. We used the dictionary during the course and recode all the columns to make the neutral between -4 to 4. Then We focused only on English comments, which were identified using the textcat function. We also removed comments that did not have a timestamp. We then split up the comments into words and found the positions of the words in the dictionary. We selected the valence of the words present in the dictionary and computed the mean valence of the comments. If none of the words were in the dictionary, we assigned a score of 0. We grouped the comments by minute and hour and computed the average sentiment per day. We plotted the sentiment by time to visualize the sentiment trends.

This was the first approach, namely the Lexicon Based Approach. We also used 2 other approaches for sentiment analysis, which are SentimentR and SentimentVader. With these we got similar results as with the Lexicon Based Approach.

The sentiment analysis with time as an independent variable did not give extra insights. We observe a horizontal asymptote that indicates the general sentiment of +1. We do however observe that for Lux Aeterna, the song that was released first, the sentiment rose as the release date of the whole album came closer.

More interesting graphs are the histograms we made about the sentiment of each of the songs. With all three, we can again clearly observe that overall there is a mainly positive to very positive sentiment, since all the histograms are left-skewed.

A limitation of our model however, could be that the lexicon misinterpreted some of the words. As we mentioned earlier, heavy metal fans often use words such as ‘wrath’ or ‘death’ to express their positive sentiment about the songs. This can possibly have had an impact on the sentiment, which would mean that we underestimated the real sentiment.

2.1.1.6 Conclusion

Both the wordclouds and wordgraphs, as the sentiment analysis indicate a positive feedback of the listeners. This corresponds with some of the critical reviews of well-known radio stations, podcasts or blogs we found on the Internet.

Metallica still shows they have it and that their music won’t die in the upcoming years. Simple manual lookups with hashtags on social media also show that they engage with their fans. As an example, on the platform TikTok they allowed their fans to play certain parts of the instrumentals, such as guitar, drums... in a so-called ‘duet-video’, which gives really cool results as it then seems that they play together with the frontman James. This engagement that they incentivised resulted in a lot of interest from new, younger fans as well. Heavy metal is popular in a very wide range of age!

2.1.2 Spotify API

In addition to Youtube, Spotify was also used for the performance part of this assignment. The Spotify API is very useful as it provides its own variables and metrics. A full list is given in Appendix 2 in Figure 19.

Data on Spotify can be scraped by creating an application on the Spotify Developers website and loading the 'spotifyr' package on R Studio. By using the get_artist_audio_features function we obtained a dataframe 'metallica' that consists of all the songs Metallica has released on Spotify. However, we still needed to remove duplicates and adjust misinformation, older albums may have wrong release years on Spotify sometimes. The data frame 'metallica_cleaned' is used for the descriptive analysis.

First we will look for some outliers, then we will compare the new album with the older ones and lastly we look for trends. We chose four variables that seemed the most important for this analysis. Tempo, Danceability, Energy and Valence. Valence is a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Outliers (one sided) were found by looking for the tracks with the highest score for each of the variables. The results can be found in Appendix 2. It is worth mentioning that Lux Aeterna, a song of the new album, has the highest energy over the whole Metallica repertory. Furthermore, no other song of the new album appeared in the results.

Then we take a look how the new album scores on these variables and if we can find some trends. The plots in Appendix 2 were derived by calculating the mean for each album and ordering them by album release year. For tempo (BPM), the new album scores high in comparison with the older ones. Additionally, there is a clear upwards trend visible. In terms of danceability, 72 seasons follows the downwards trend. For valence we see an average score and no trend visible. Almost all the albums have a high energy, which is no surprise for a heavy metal band.

Before we start with the predictive analysis, we need to highlight one other very important variable, popularity. The popularity is calculated on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. The value will be between 0 and 100, with 100 being the most popular. In Appendix 2 there is an overview of the popularity of almost each Metallica track and a density plot. At this moment 72 Seasons (Single) has the 9th highest score. However, we do not know if this will still be the case in the future since recent plays have a higher value. The popularity distribution is a right skewed distribution showing us how truly rare it is to have a popular song. Most of the songs have a popularity in the range of 40 to 50.

2.2 Part 2: Predictive

2.2.1 Introduction

This section of the report is dedicated to forecasting the Spotify popularity of Metallica's 72 seasons album, using solely the data obtained from Spotify scraping. The manner in which this scraping was done, is described in the section above. First the data preprocessing is described, then the training and evaluation of four models is discussed and lastly several deficiencies of the predictive performance of the models are listed.

2.2.2 Data Preprocessing

The "metallicafull" dataset makes up the training set and consists of 100 observations and 59 variables. One of which is track.popularity, the dependent variable. The "MetallicaNewAlbum" dataset makes up the test set. Since the album came out 14th of April, the Spotify algorithm already calculated the popularity of each song. However the goal of this report is to predict the popularity of the songs in the long term, so this variable is purposefully ignored.

First a 80/20 split is used to split the "metallicafull" dataset into a training and validation set and next the dependent variable is separated from the independent variables. Most of the independent variables however do not contain useful information. Therefore only the following 15 variables are kept: track.id, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, track.duration_ms, track.name, track.track_number.

Note that the track.album.name and track.album.release_date variables are also left out. This is because this report will predict the popularity of a new album. Keeping the name of the album or their release date would therefore lead to data leakage between the training and validation set and thus biased models.

Lastly the track.duration_ms is converted from milliseconds to minutes and one-hot encoding is used on the variable key. The variable key is a categorical variable in which the numbers 0 to 11 each represent the key in which the song is recorded.

2.2.3 Model Training and Evaluation

In this section of the report, the following 4 models are discussed: the Lasso Regression Model, the Linear Regression Model, Extreme Gradient Boosting Model and Random Forest Model. In Appendix 3 an overview of the predictions is provided. The models are compared by their RMSE which is shown in the table at the end of this section.

2.2.3.1 Lasso Regression

The Lasso Regression Model is used as a baseline model for the more complex models. The model is trained using the “glmnet” function in R with an alpha of 1. After which the parameter lambda is tuned, the optimal value for this parameter results in the lowest MSE and is further used to train the regression model. Because most of the variables in the training set are continuous variables, the simple model does not perform much worse than the complex models discussed later. It has an R-squared of 0.1940116 and an RMSE of 9.438426.

2.2.3.2 Linear Regression Model

Linear regression is a commonly used statistical method for predicting a continuous dependent variable (in this case the popularity of the album) based on one or more independent variables. It is a simple yet powerful method that allows for easy interpretation of the relationship between the dependent variable and the independent variables. Additionally it provides a baseline performance that can be compared to more complex machine learning methods. The linear regression model has an R-squared of 0.1176606 and an RMSE of 9.875361, this may suggest that the linear model has quite low explanatory power.

2.2.3.3 XGBoost

XGBoost (Extreme Gradient Boosting) is a suitable algorithm to predict the popularity of an upcoming album using Spotify data because it is a powerful and efficient algorithm for regression tasks. XGBoost is an ensemble method that combines multiple decision trees, and it can handle high-dimensional data with many features, such as the Spotify data that contains various artist, album, and track features.

XGBoost can also handle missing data and impute them using the most common value or a user-specified value. It has the ability to handle outliers and noisy data through regularization techniques. Additionally, it can capture non-linear relationships between variables and identify the most important features that contribute to album popularity through feature importance scores.

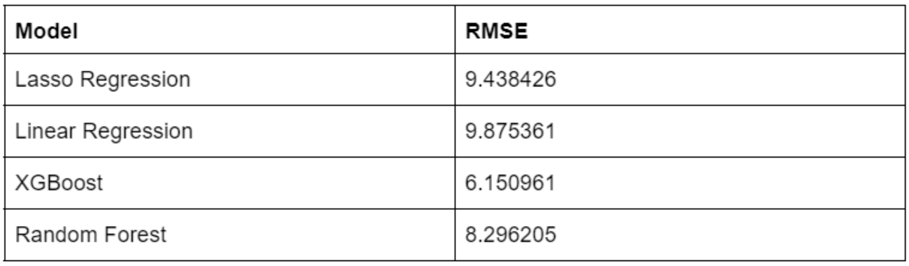
In this specific implementation, the XGBoost algorithm was able to achieve a high R-squared value of 0.6576931 and low RMSE value of 6.150961, indicating that it is a strong model for predicting album popularity. Overall, XGBoost is a powerful and widely used algorithm for regression tasks and is an appropriate choice for predicting the popularity of an upcoming album using Spotify data.

2.2.3.4 Random Forest

The use of Random Forest algorithm in predicting the popularity of an upcoming album using Spotify data is particularly advantageous because it can handle complex and high-dimensional datasets. Spotify data is rich with variables such as artist features, album features, and track features that can be used to predict album popularity. Random Forest's ability to handle missing data by replacing them with the most frequent value of that variable makes it more resilient to real-world datasets, where missing data is common.

Moreover, Random Forest can handle outliers, noisy data, and nonlinear relationships between variables. These features are essential in ensuring that the prediction model is reliable and robust. In addition, Random Forest can provide feature importance scores that can help identify the most important features that contribute to album popularity. Our Random Forest algorithm was able to achieve an R-squared value of 0.3772861 and an RMSE value of 8.296205

2.2.3.5 Overview



Model	RMSE
Lasso Regression	9.438426
Linear Regression	9.875361
XGBoost	6.150961
Random Forest	8.296205

Figure 1: Overview

2.2.4 Model deficiencies

1. The spotify algorithm that calculates 'popularity' is based on the number of times the track has been played on spotify and how recent these tracks have been played. Metallica however has popular albums that came out before spotify was even invented. So their popularity is probably underestimated by the Spotify algorithm.
2. Random forests and XGBoosting are prone to overfitting if the model is too complex or if there is not enough data to train the model. This can lead to poor generalization performance on new, unseen data.
3. While our models may be able to predict album popularity, it is important to note that predicting the popularity of an upcoming album is a complex task that involves many factors beyond the Spotify data. Factors such as marketing campaigns, industry trends, and the current cultural climate can also impact the popularity of an album, and these factors may not be fully captured in the Spotify data.

2.2.5 Conclusion

The results displayed in Appendix 3 provide an overview of the analysis conducted and suggest that the album is unlikely to gain substantial attention from Spotify users, according to the models utilized. However, it is important to note that these models are not perfect and have certain deficiencies that must be taken into account when interpreting these findings. As discussed in the previous section, the models may have limitations in terms of their accuracy and the extent to which they capture all relevant variables and factors. Therefore, while the results presented in Appendix 3 offer valuable insights into the potential popularity of the album among Spotify users, they should be interpreted with caution, and further research may be necessary to fully understand the dynamics at play.

3 Appendix 1: Targeting

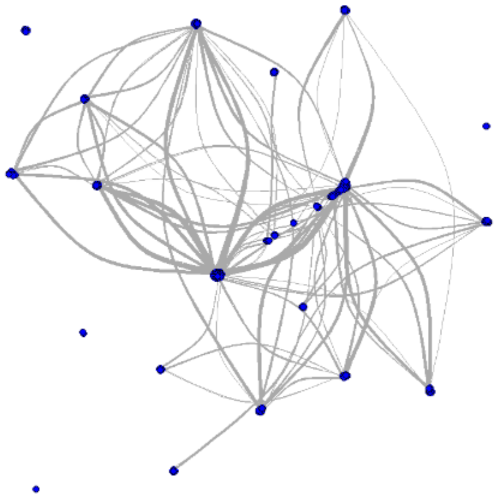


Figure 2: Network Dirk Van den Poel

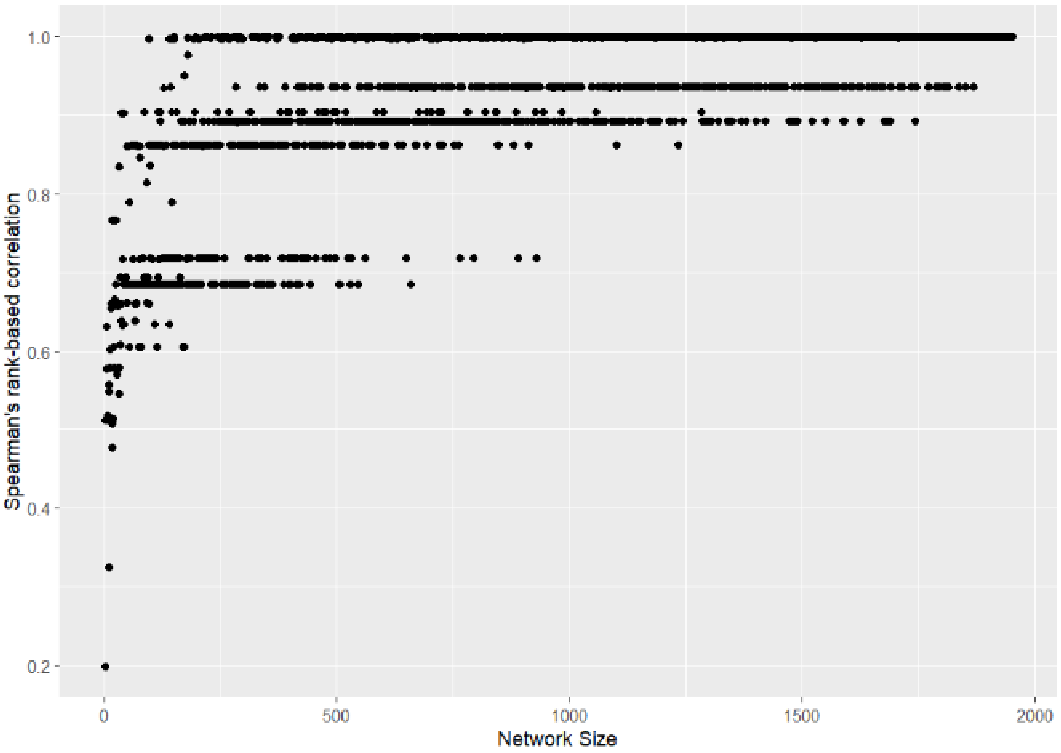


Figure 3: Rank Based Correlation

4 Appendix 2: Descriptive

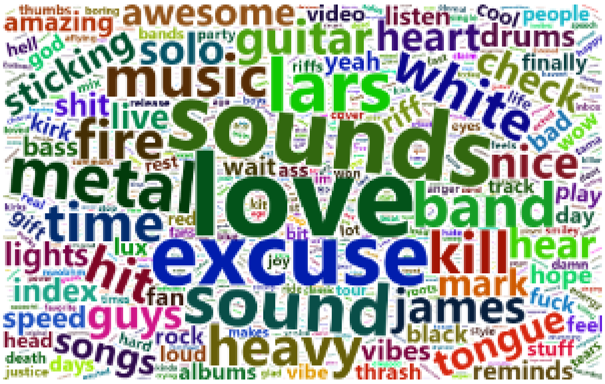


Figure 4: Wordcloud: 72 Seasons

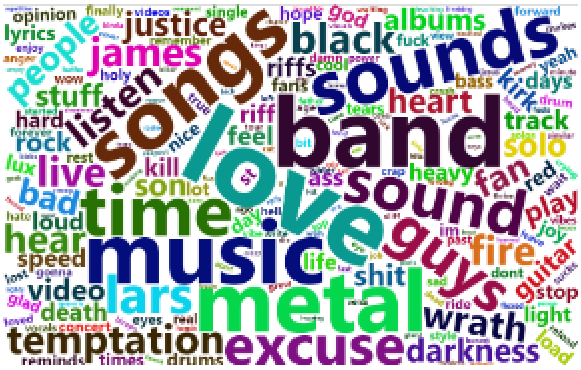


Figure 5: Wordcloud: If Darkness Had A Son

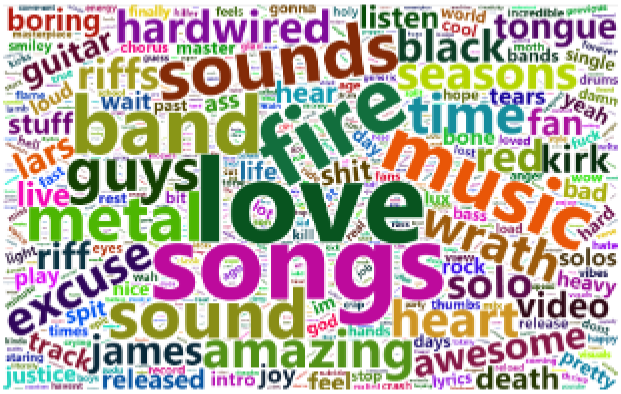


Figure 6: Wordcloud: Lux Aeterna

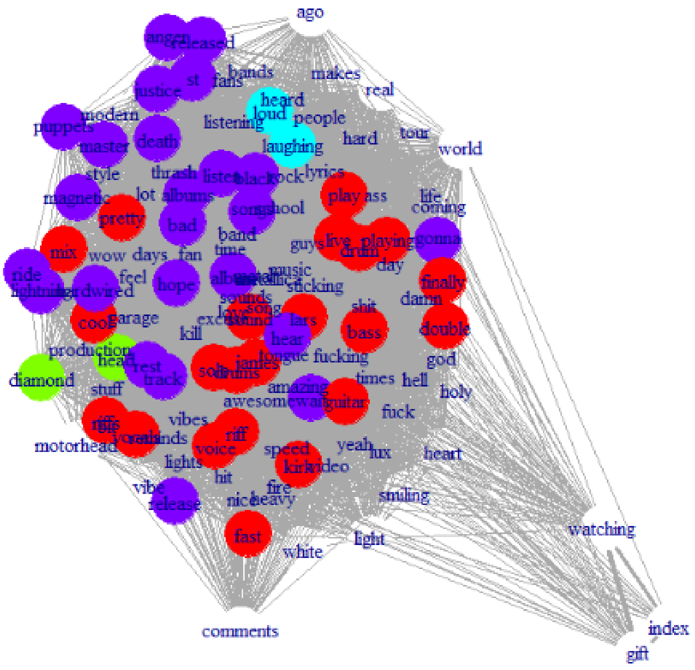


Figure 7: Wordgraph: 72 Seasons

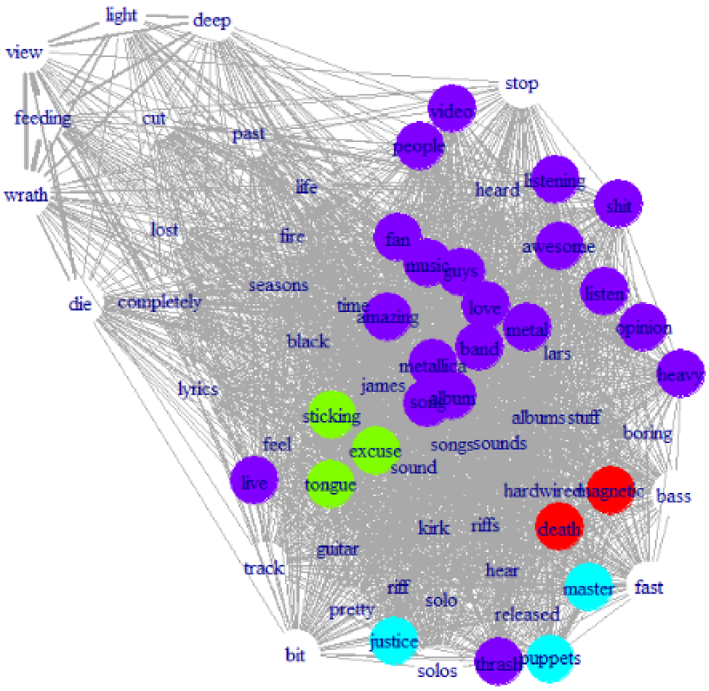


Figure 8: Wordgraph: If Darkness Had A Son

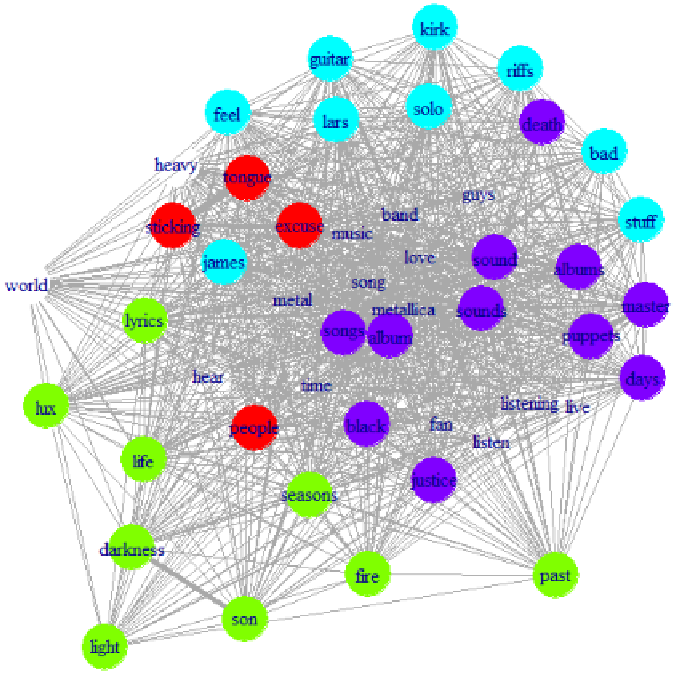


Figure 9: Wordgraph: Lux Aeterna

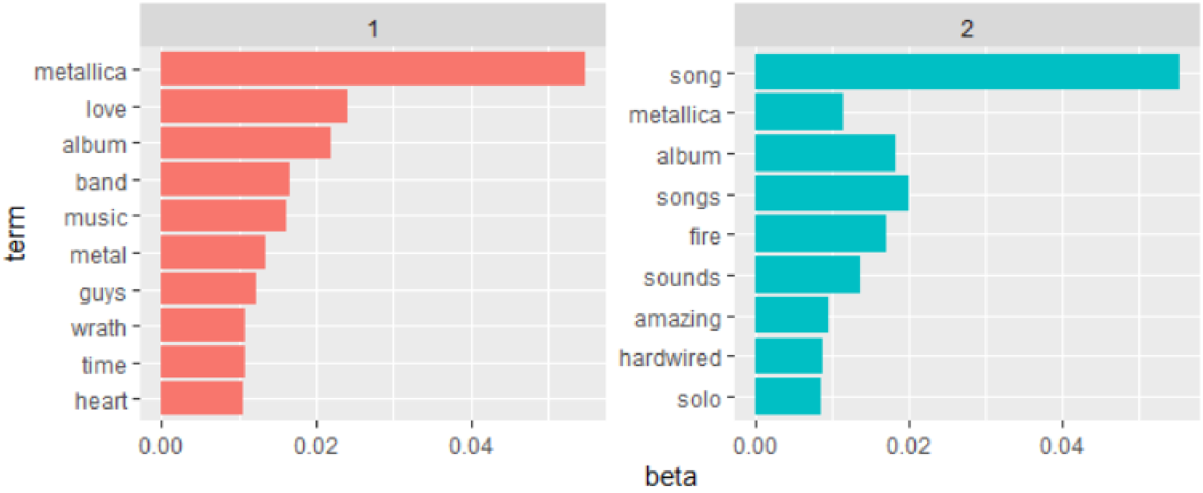


Figure 10: Topic Model: 72 Season

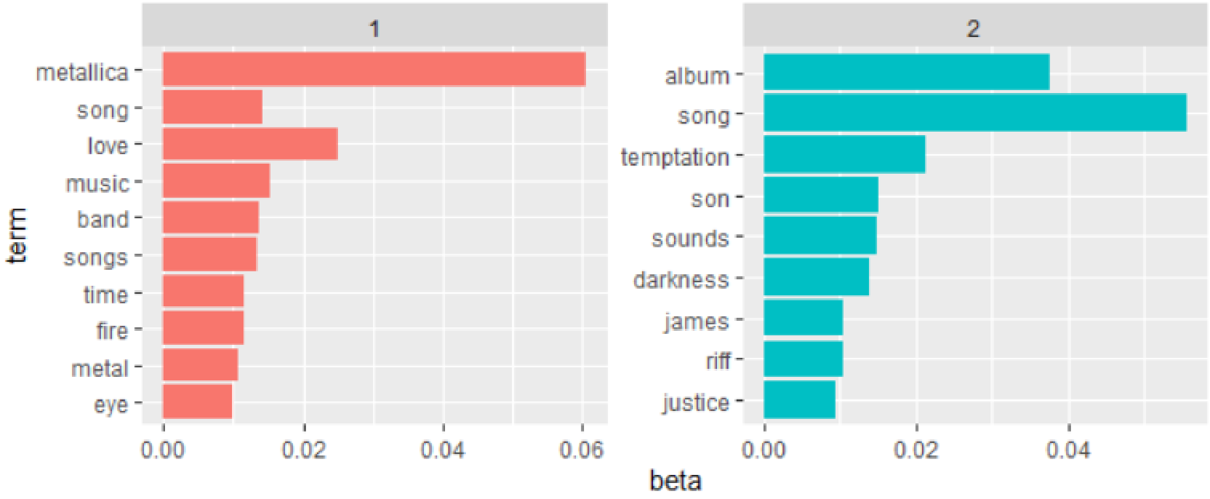


Figure 11: Topic Model: If Darkness Had A Son

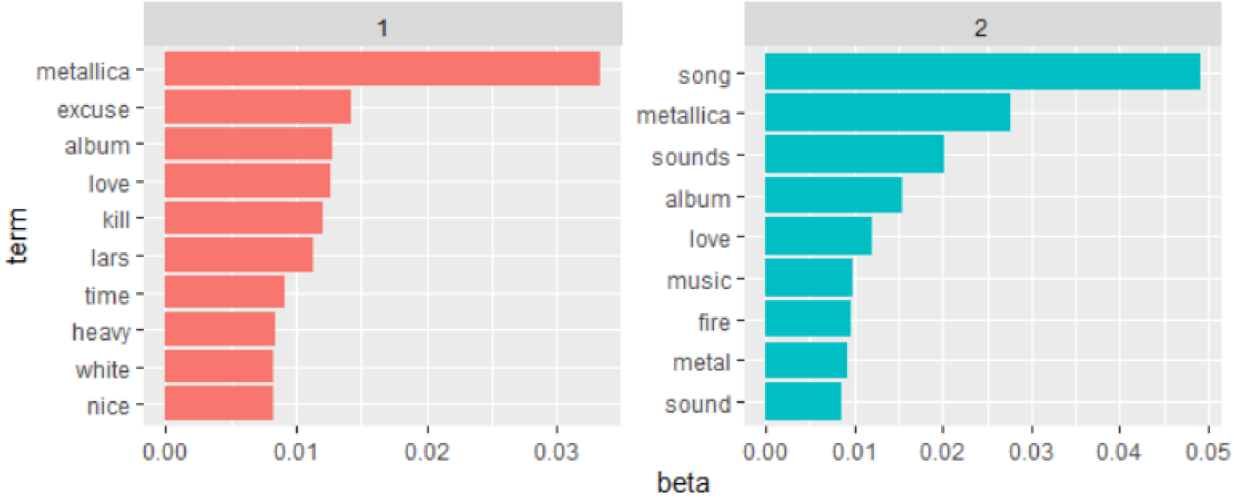


Figure 12: Topic Model: Lux Aeterna

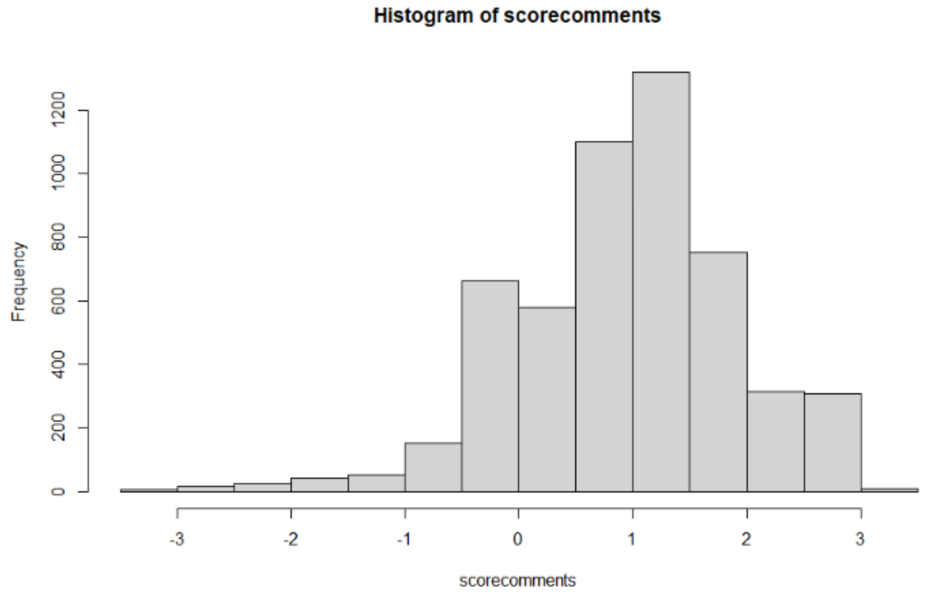


Figure 13: Sentiment Analysis: 72 Seasons - a

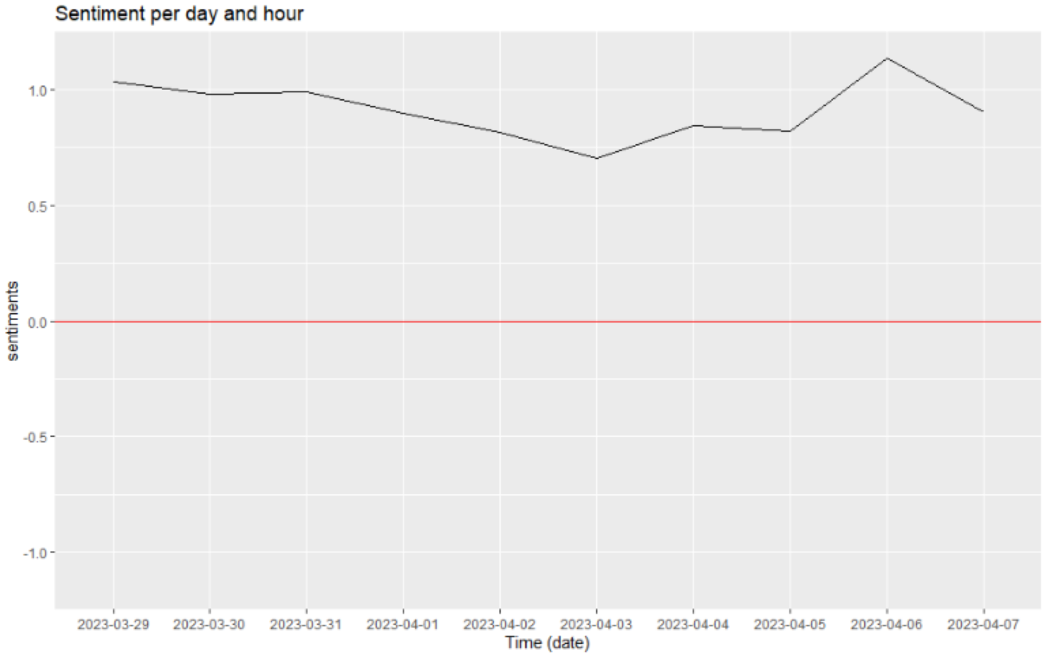


Figure 14: Sentiment Analysis: 72 Seasons - b

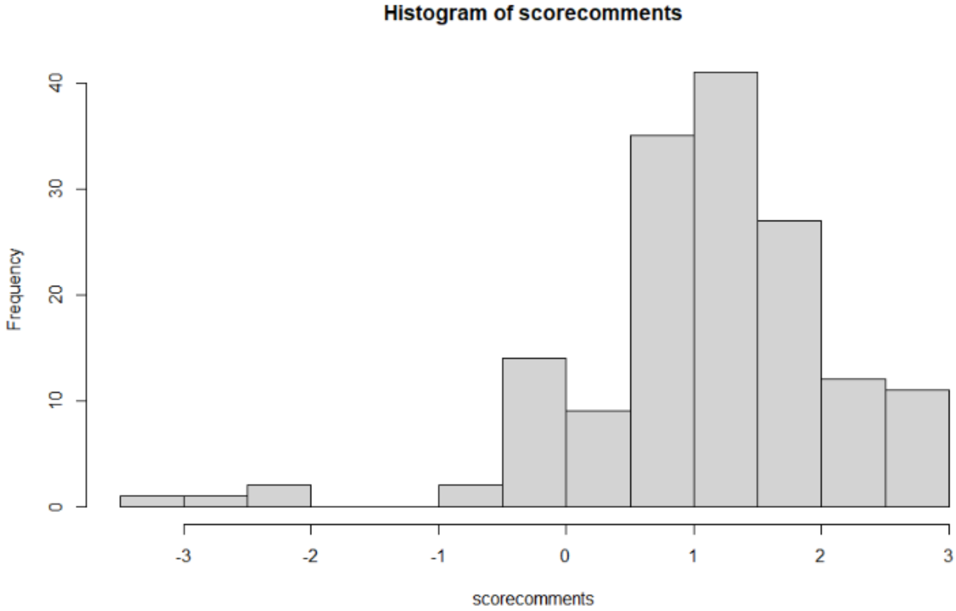


Figure 15: Sentiment Analysis: If Darkness Had A Son - a

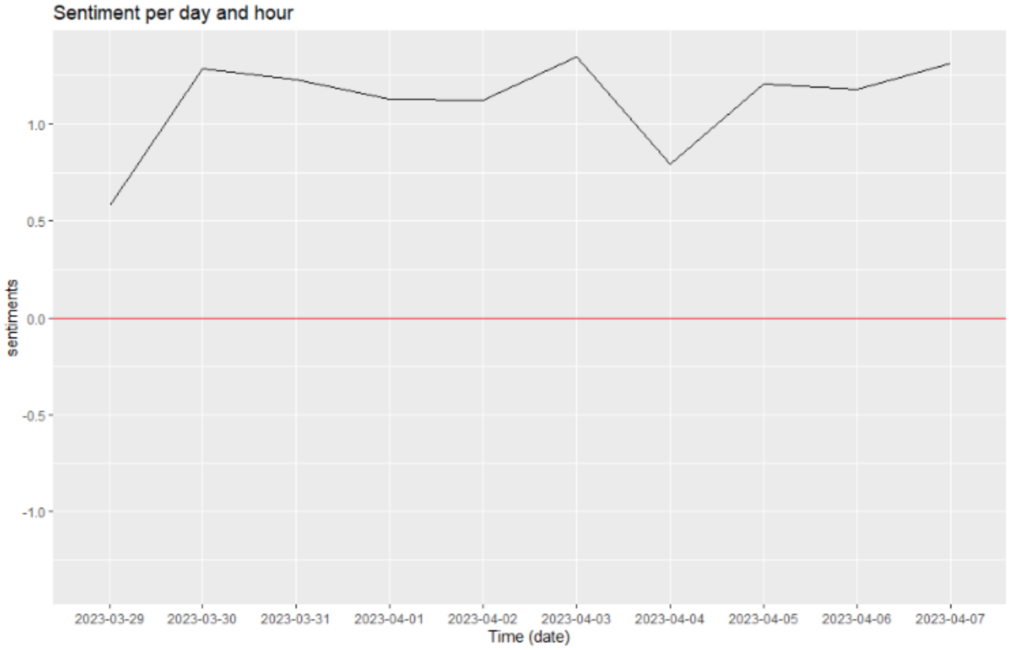


Figure 16: Sentiment Analysis: If Darkness Had A Son - b

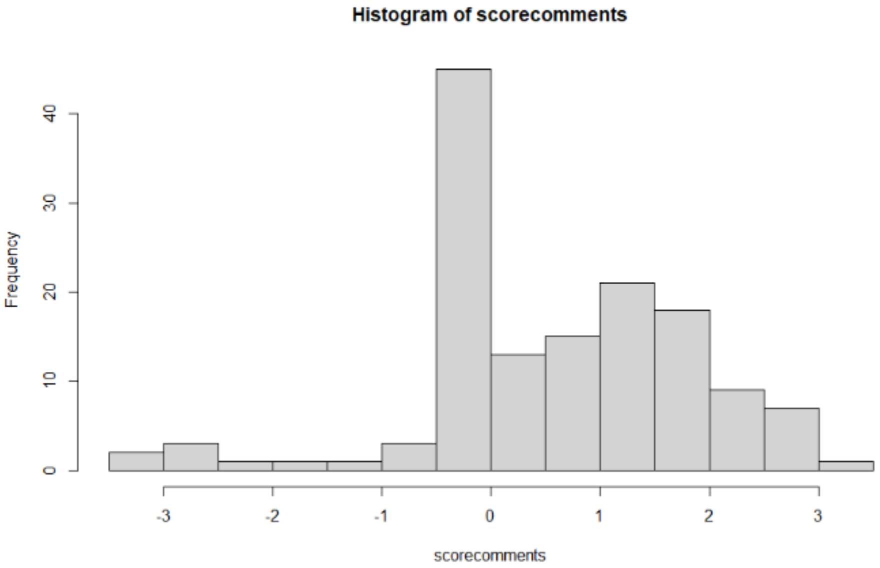


Figure 17: Sentiment Analysis: Lux Aeterna - a

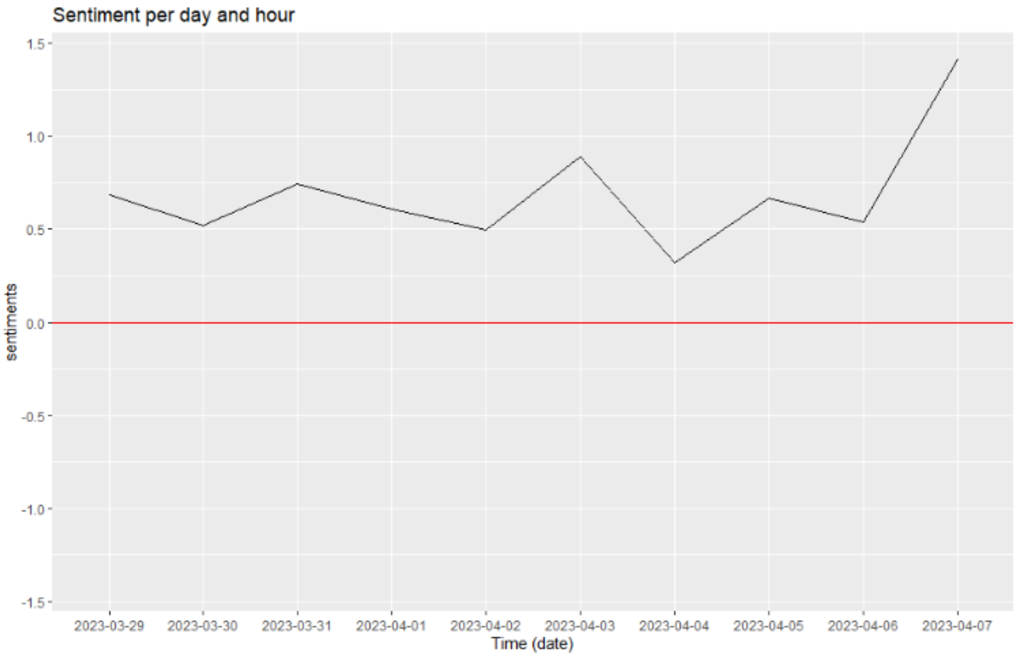
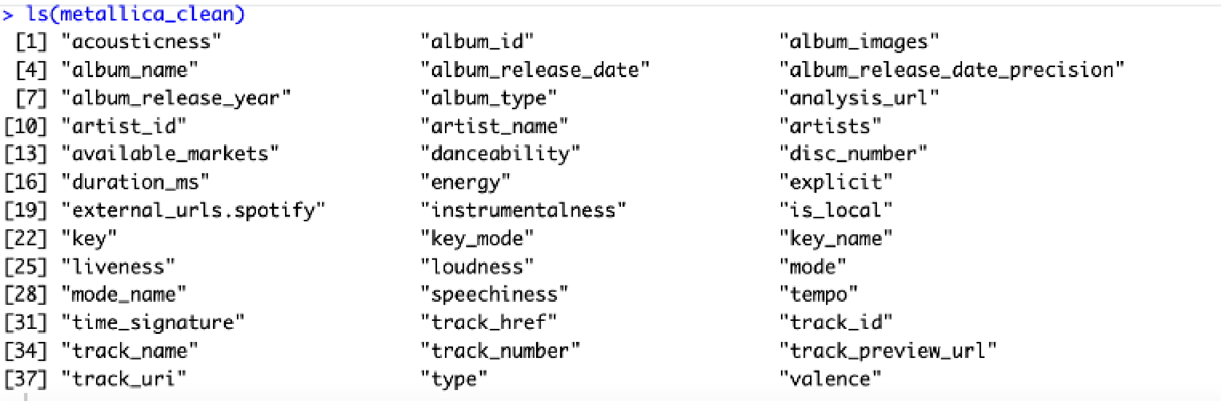
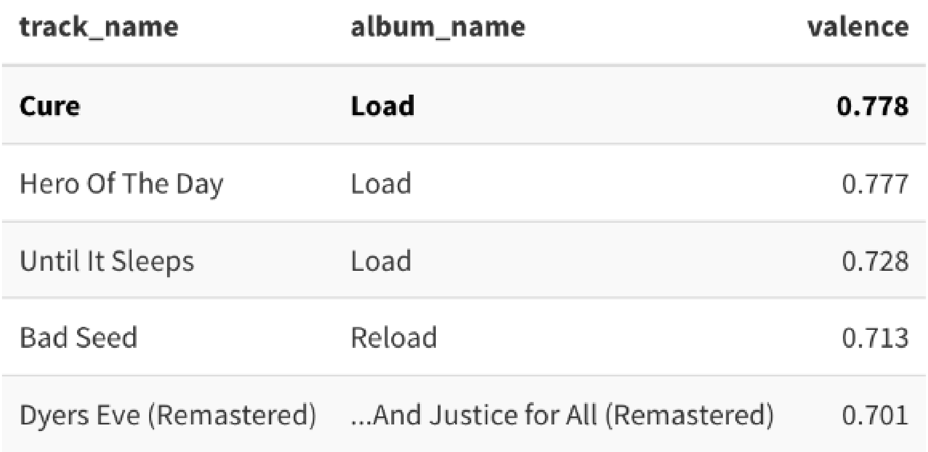


Figure 18: Sentiment Analysis: Lux Aeterna - b



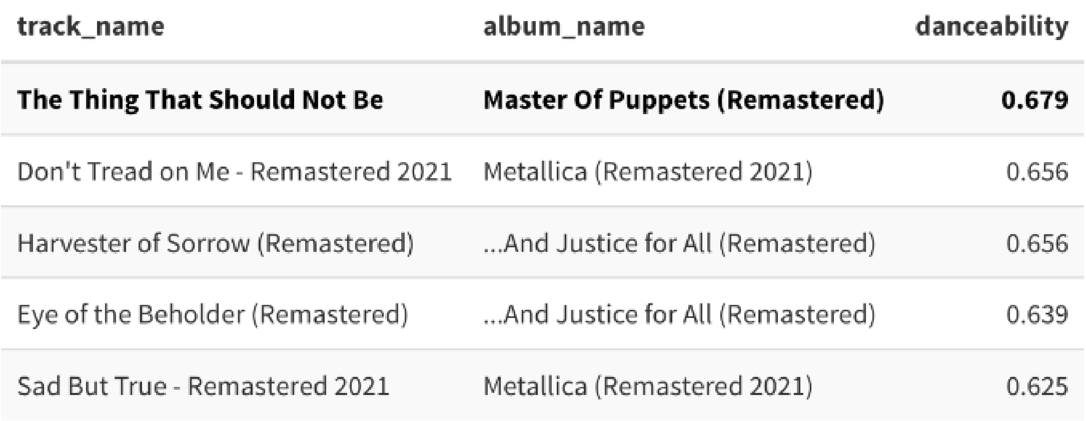
```
> ls(metallica_clean)
 [1] "acousticness"            "album_id"                "album_images"
 [4] "album_name"              "album_release_date"      "album_release_date_precision"
 [7] "album_release_year"      "album_type"              "analysis_url"
[10] "artist_id"               "artist_name"             "artists"
[13] "available_markets"       "danceability"            "disc_number"
[16] "duration_ms"             "energy"                  "explicit"
[19] "external_urls.spotify"   "instrumentalness"        "is_local"
[22] "key"                     "key_mode"                "key_name"
[25] "liveness"                "loudness"                "mode"
[28] "mode_name"               "speechiness"             "tempo"
[31] "time_signature"          "track_href"              "track_id"
[34] "track_name"              "track_number"            "track_preview_url"
[37] "track_uri"               "type"                    "valence"
```

Figure 19: Spotify API - List Of All The Variables



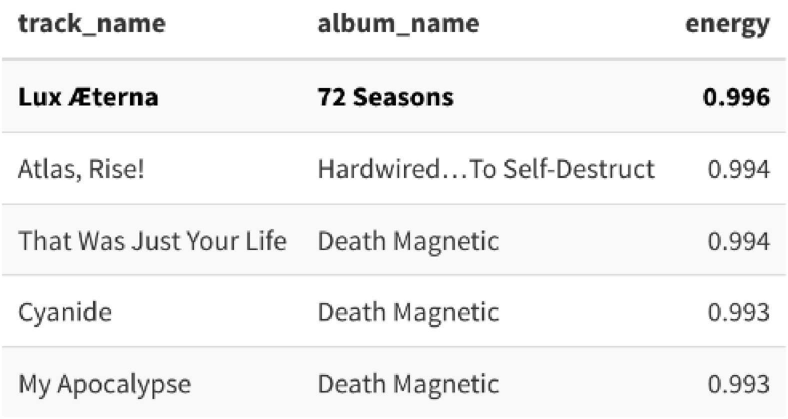
track_name	album_name	valence
Cure	**Load**	**0.778**
Hero Of The Day	Load	0.777
Until It Sleeps	Load	0.728
Bad Seed	Reload	0.713
Dyers Eve (Remastered)	...And Justice for All (Remastered)	0.701

Figure 20: 5 Most Joyable Tracks



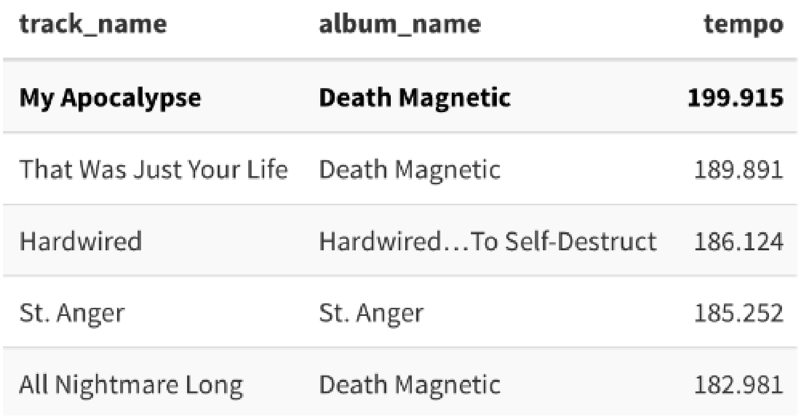
track_name	album_name	danceability
The Thing That Should Not Be	**Master Of Puppets (Remastered)**	**0.679**
Don't Tread on Me - Remastered 2021	Metallica (Remastered 2021)	0.656
Harvester of Sorrow (Remastered)	...And Justice for All (Remastered)	0.656
Eye of the Beholder (Remastered)	...And Justice for All (Remastered)	0.639
Sad But True - Remastered 2021	Metallica (Remastered 2021)	0.625

Figure 21: 5 Most Danceable Tracks



track_name	album_name	energy
Lux Æterna	**72 Seasons**	**0.996**
Atlas, Rise!	Hardwired…To Self-Destruct	0.994
That Was Just Your Life	Death Magnetic	0.994
Cyanide	Death Magnetic	0.993
My Apocalypse	Death Magnetic	0.993

Figure 22: 5 Most Energetic Tracks



track_name	album_name	tempo
My Apocalypse	**Death Magnetic**	**199.915**
That Was Just Your Life	Death Magnetic	189.891
Hardwired	Hardwired…To Self-Destruct	186.124
St. Anger	St. Anger	185.252
All Nightmare Long	Death Magnetic	182.981

Figure 23: 5 Fastest Tracks

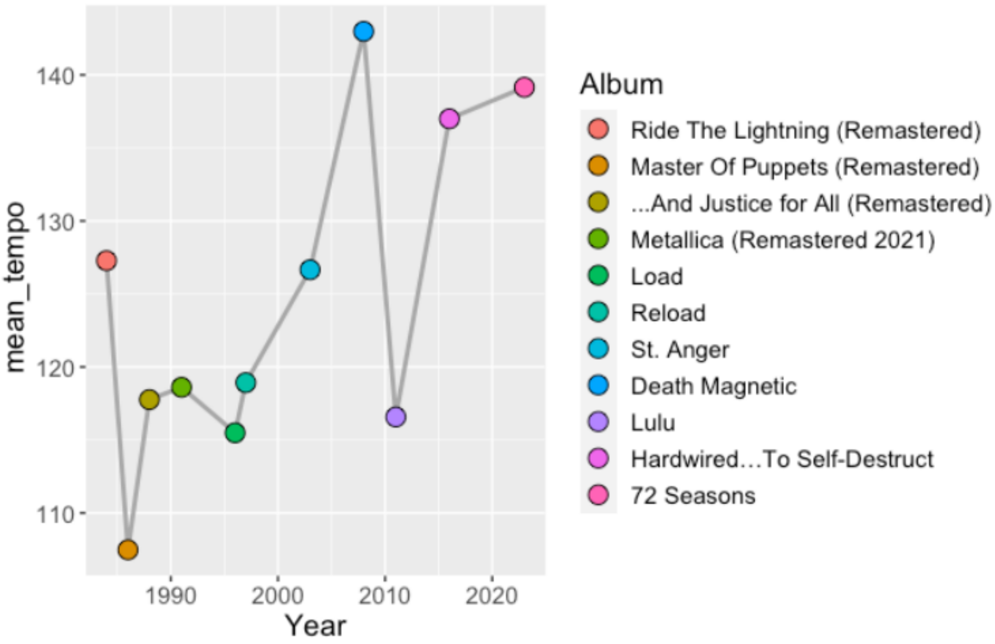


Figure 24: Album Mean And Trendline: Tempo - 1

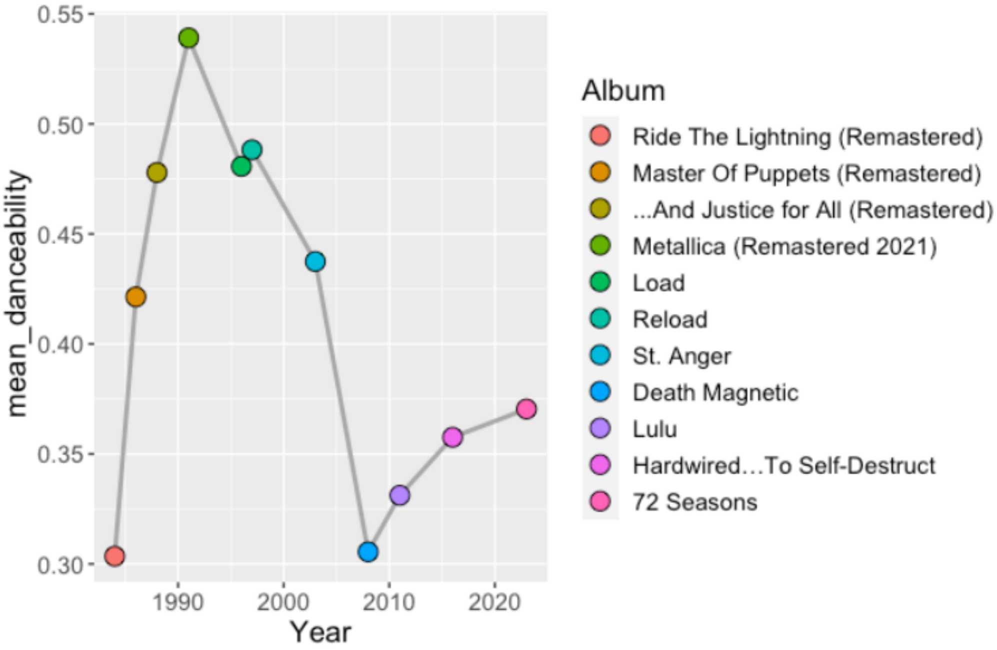


Figure 25: Album Mean And Trendline: Danceability - 1

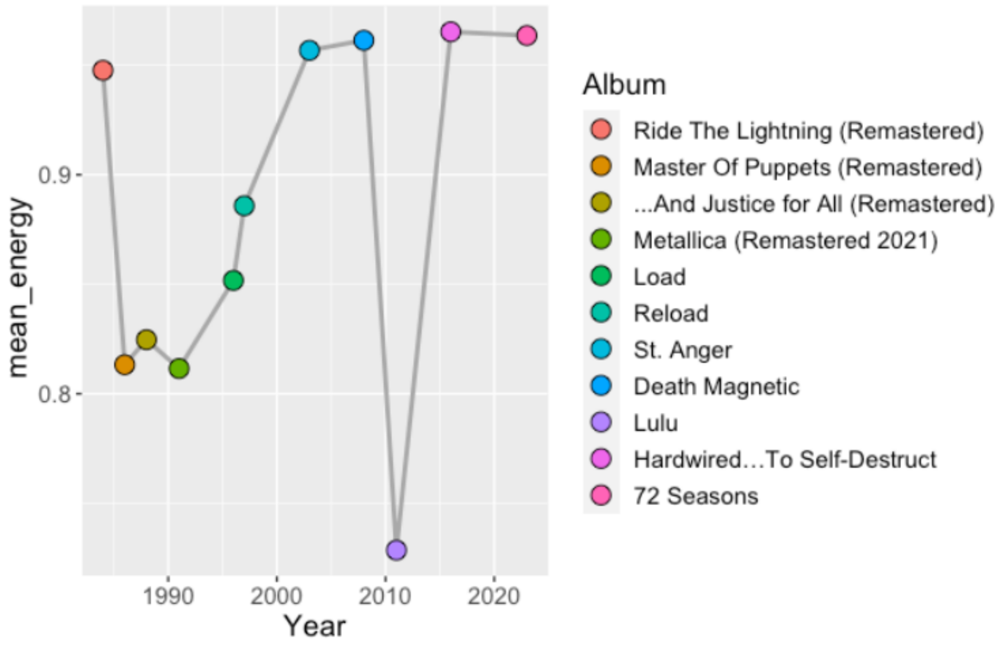


Figure 26: Album Mean And Trendline: Energy

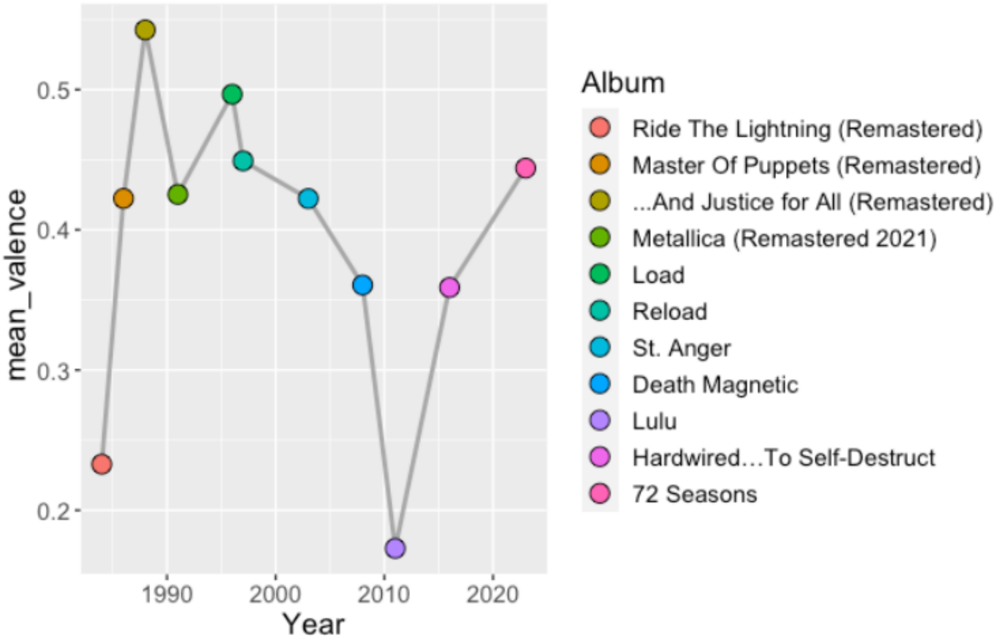


Figure 27: Album Mean And Trendline: Valence

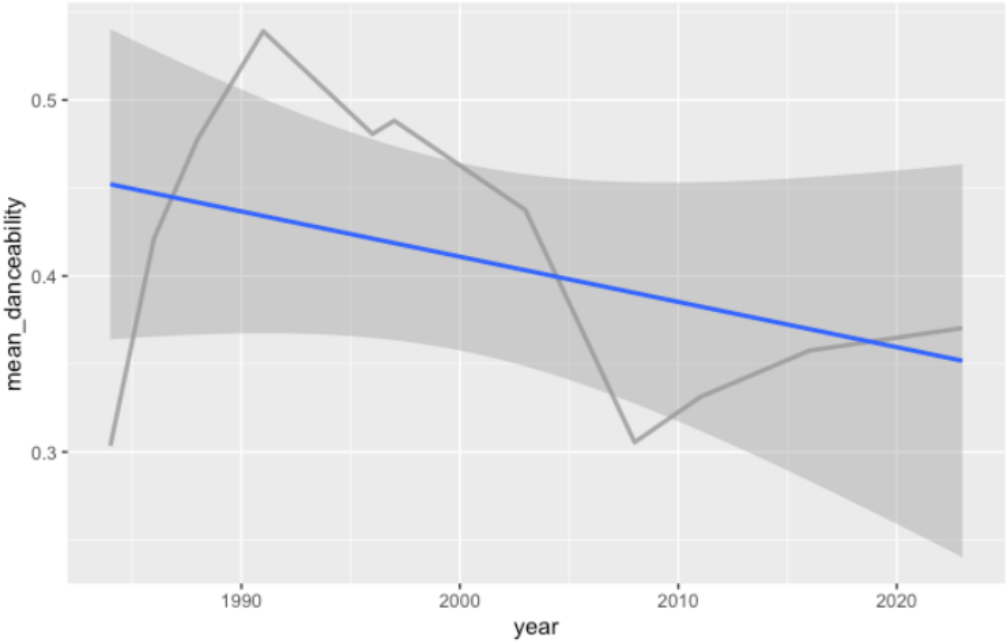


Figure 28: Album Mean And Trendline: Danceability - 2 (Gives the same information as "Album Mean And Trendline: Danceability - 1")

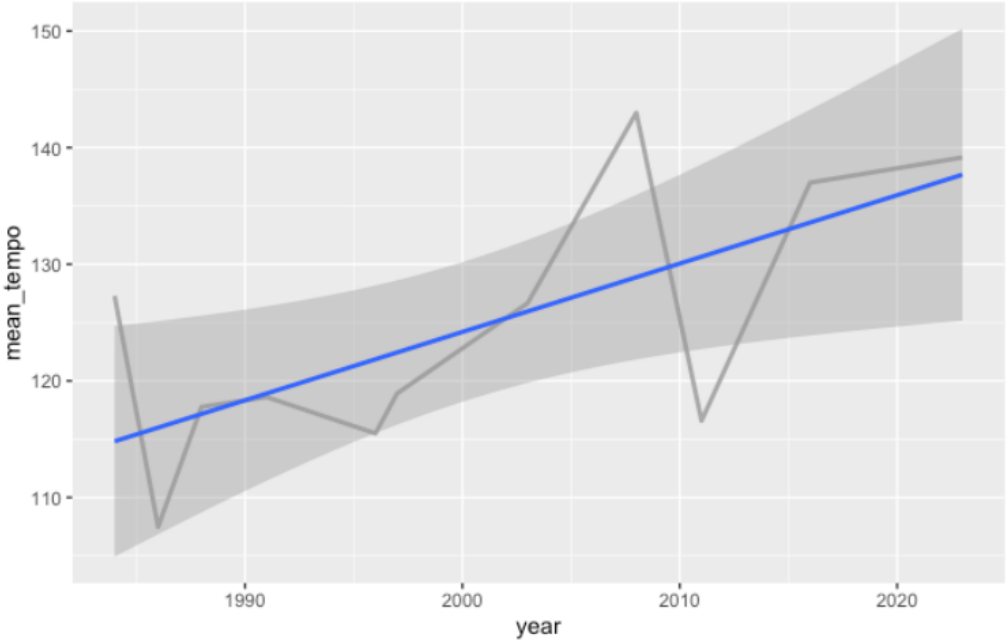


Figure 29: Album Mean And Trendline: Tempo - 2 (Gives the same information as "Album Mean And Trendline: Tempo - 1")

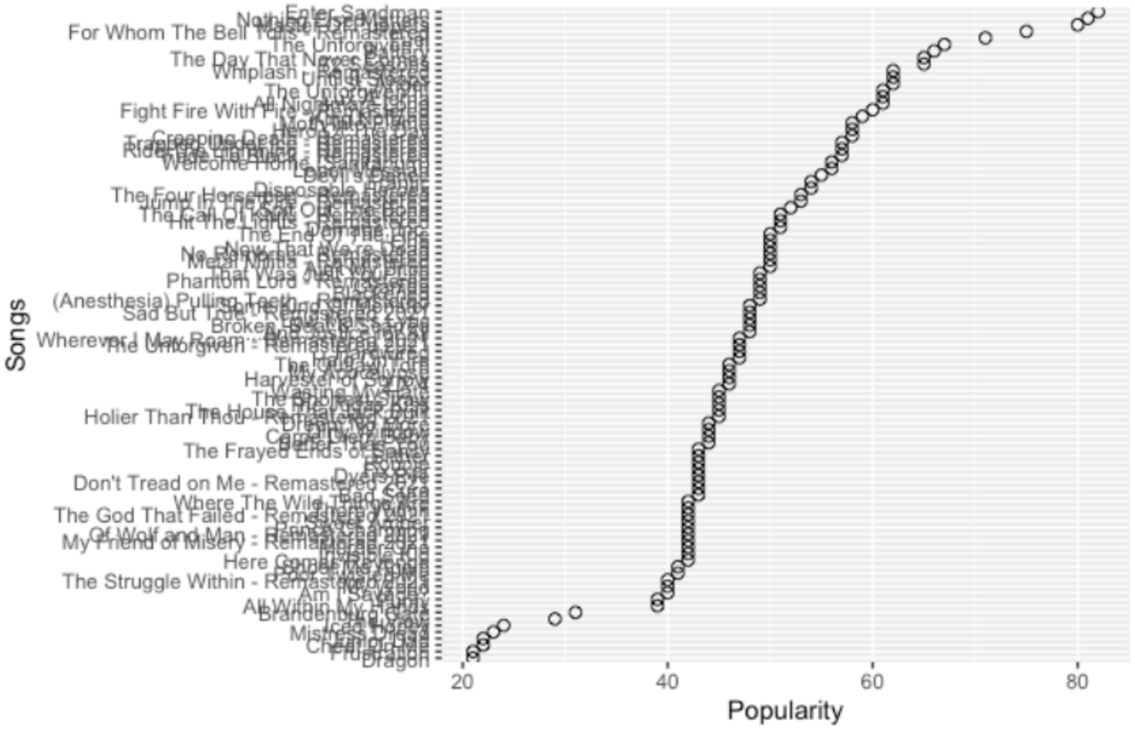


Figure 30: Popularity(density)

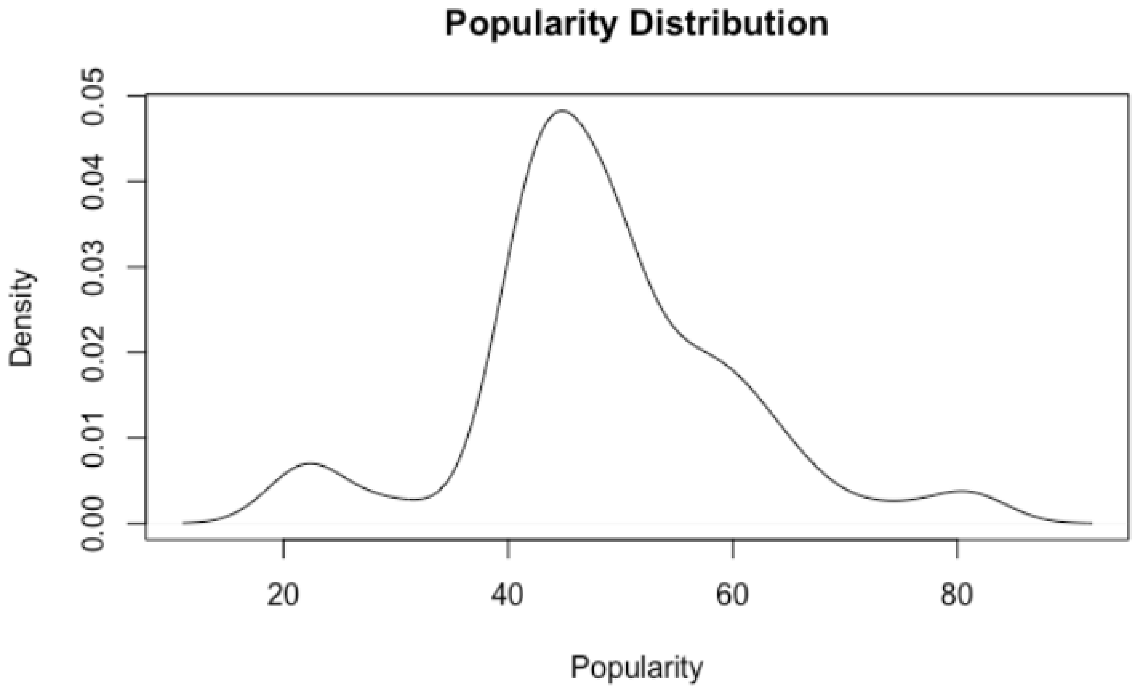
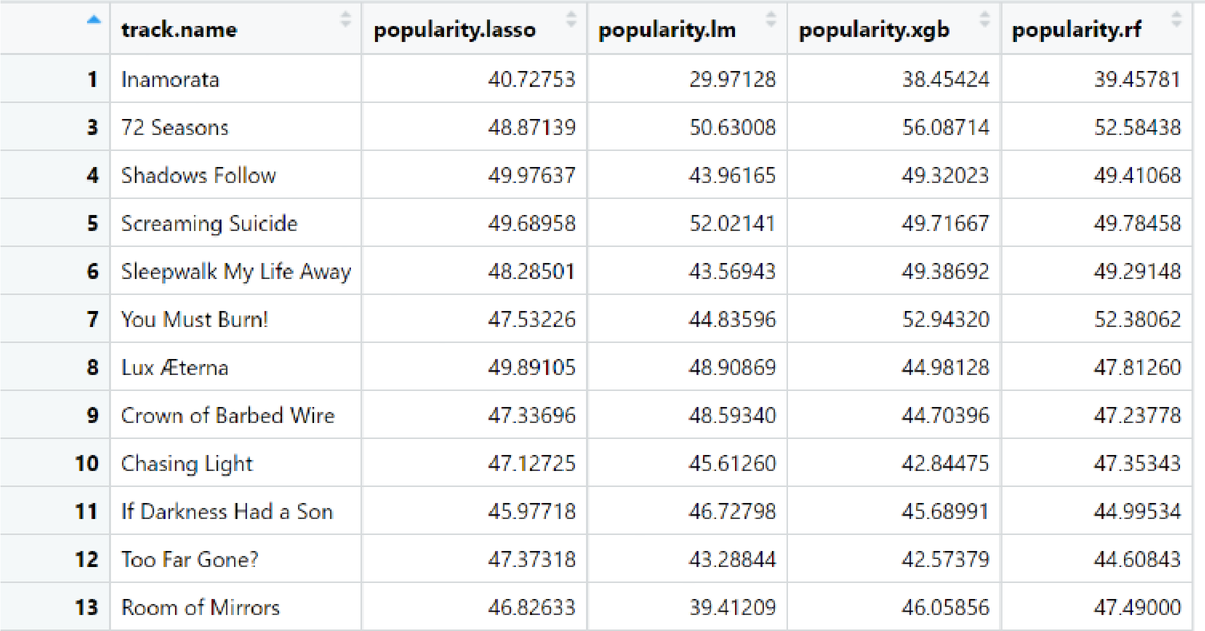


Figure 31: Popularity Distribution

5 Appendix 3: Predictive



	track.name	popularity.lasso	popularity.lm	popularity.xgb	popularity.rf
1	Inamorata	40.72753	29.97128	38.45424	39.45781
3	72 Seasons	48.87139	50.63008	56.08714	52.58438
4	Shadows Follow	49.97637	43.96165	49.32023	49.41068
5	Screaming Suicide	49.68958	52.02141	49.71667	49.78458
6	Sleepwalk My Life Away	48.28501	43.56943	49.38692	49.29148
7	You Must Burn!	47.53226	44.83596	52.94320	52.38062
8	Lux Æterna	49.89105	48.90869	44.98128	47.81260
9	Crown of Barbed Wire	47.33696	48.59340	44.70396	47.23778
10	Chasing Light	47.12725	45.61260	42.84475	47.35343
11	If Darkness Had a Son	45.97718	46.72798	45.68991	44.99534
12	Too Far Gone?	47.37318	43.28844	42.57379	44.60843
13	Room of Mirrors	46.82633	39.41209	46.05856	47.49000

Figure 32: Predictions Overview

6 References

1. Peker, P. (2022, January 6). Predicting Popularity on Spotify — When Data Needs Culture More than Culture Needs Data. Medium. https://towardsdatascience.com/predicting-popularity-on-spotify-when-data-needs-culture-more-than-culture-needs-data-2ed3661f75f1

2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning: with Applications in R. Springer Nature.

3. Mindus. (2021). Spotify Descriptive and Exploratory Data Analysis. Kaggle. - https://www.kaggle.com/code/mindus/spotify-descriptive-and-exploratory-data-analysis