Lyrics of The 500 Greatest Songs of All Time

Callie Boogaert & Joseph Davis

[cboogaer@syr.edu](mailto:cboogaer@syr.edu) & [jcdavis@syr.edu](mailto:jcdavis@syr.edu)

Lyric Analysis

Decision Trees, Multinomial Naive Bayes, SVM, LDA

09/17/2023

| **Introduction** | Music is a human tradition dating back to at least 40,000 B.C. by evidence of bone flutes in the upper paleolithic. While the beginnings of music aren’t known exactly, it is a deep-rooted part of the human experience. This tradition of music is seen in the modern day practiced across all known cultures around the world. To enjoy music is to partake in being human.  As of September 2023, Spotify’s market cap has surpassed $30 billion cementing it as a top 1000 company world-wide. This is up from just $15 billion in 2021. This massive market share can be attributed to the growing popularity of music streaming services in the past decade. Currently Spotify has 551 million users, 220 million of which are premium subscribers.    With such a massive market share and influence it follows that Spotify and similar music streaming services have an incentive to optimize their platforms for users. One way in which Spotify has improved their product is the implementation of recommended music and playlists. This feature uses songs the user already listens to and curates them a playlist with a similar style of music. Improving this system would benefit not only the company but also the user.    In addition to Spotify, recording artists and studios also seek to improve their product. With over 100 million songs and an estimated 11 million artists/creators currently on Spotify, artists and studios need ways to stand out from the crowd. One way to improve their music and music making ability is to hone in on the lyrical aspect of their songs.    The intention of this investigation is to utilize popular song lyrics to help optimize music classification using lyrics. This knowledge could be utilized by both music streaming services but also recording artists and studios to optimize their lyric selection. Understanding the importance and influence lyrics have in music and the music industry could not only help streaming platforms better understand their product but could also help evolve music beyond what it currently is. |
| --- | --- |
| **Analysis** | **data preparation and cleaning** This investigation consisted of two data sources. The first data set is the Rolling Stone 500 Greatest Songs of all time. The dataset consists of the following variables: Rank, Artist, Title, Year, Writers, Popularity, danceability, energy, key, loudness, mode speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration ms. The below image shows a snippet of the dataset when first read in.    **Figure 1.** Rolling Stones Dataset  A Genres Dictionary was manually created, this took the Title and Artist and found the Genre that it belonged to. Once this was acquired it was read in and merged in with the Rolling stones dataset. The below image shows the genres dataset when first read in.    **Figure 2.** Genre Dataset  The Rolling Stone Dataset and Genre dataset were merged on the Artist and Title variables. The below image shows the merged dataset.    **Figure 3.** Merged Dataset  This merged dataframe was examined and cleaned to ensure ease of coding and repeatability for future analysis. To begin the column “Unnamed: 0’ was renamed to “Rank. The values of this column were recoded, in the above image the 0 next to Kanye West would indicate that his song was ranked #1, however he is actually ranked 500th. This rank column was then re-coded to begin with 500 and decrease by one every row. Next, there were some Titles or Artists that were not formatted in a way which the API could pick them up. Song Titles and Artist Names were replaced with a version that the API could use. Next multiple new variables were created. The first is the 100s Rank, this variable groups the Ranks of each song by 100s, so the first group will be the Top 100 songs. A 50s Rank variable was also created, this was the same concept as the 100s Rank but just used a smaller bin size. Next the Popularity variable is binned, this was done based on an American grading system. That is if Popularity was 90-100 it would go in the ‘A’ bin. The last variable created was Decade, this variable was generated by binning the Year variable. It is important to note that in the Decade variable 00s stands for 2000’s, 10s stands for 2010’s, and 20s stands for 2020’s. The Below image shows a snippet of the cleaned Rolling Stones dataset.    **Figure 4.** Cleaned Rolling Stones Dataset  The second data set used was sourced using an Application Programming Interface, API. The API used was Lyrics Genius. This API works by taking the Artist and Title and retrieving the lyrics for each song. In order to use this API, the Title and Artist from the Rolling stones dataset was input to create a dictionary. The API will then looped through the Dictionary to return the Lyrics for everything in the dictionary. Before calling the API a few songs had to be removed as it was known that these songs would cause the API to crash as it wasn’t able to retrieve the Lyrics. Once this data has been pulled in, a dataset was created with the Lyrics, Artist, and Title. The below image shows a snippet of the Lyrics data set when it is first read in.    **Figure 5.** Lyrics Dataset  The lyrics in the lyric dataframe had to be cleaned in order for analyses to be run on them. First the Introduction and Outro were deleted. Then ‘/n’ was removed from the end of the lines. The beginning Contributors were removed. Next anything up until the word ‘Lyric’ was removed and the ‘Embed’ was removed from the end of each line. This cleaning ensured that only the lyrics were present in the dataset. Next all special characters and numbers were removed from the lyrics. Lastly, Spaces were inserted between words where needed. The below image shows the cleaned lyrics dataset.    **Figure 6.** Cleaned Lyrics Dataset  The next step was to merge the Lyrics into the Rolling Stone dataset. In order to do this, song Titles and Artists needed to be replaced so that they match in each datafame. In most cases a word is capitalized in one dataset but not the other. Or one dataset uses “&” while the other uses “and”. Once these edits were made the Lyrics dataset was merged onto the Rolling Stone dataset by Title and Artist. The below image shows this final dataset.    **Figure 7.** Final Cleaned Lyrics Dataset **data exploration** Exploratory analysis was begun to get a better understanding of the Rolling Stones data and the relationships between variables. The below bar plot shows the Top 10 Artists with the most songs appearing in the top 500 songs. The Beatles have the most songs in the top 500 with around 12 songs appearing on the list. Other top artists were The Rolling Stones, Bob Dylan, David Bowie, Bruce Springsteen, Joni Mitchell, Prince, Elton John, Stevie Wonder, and Aretha Franklin.    **Figure 7.** Most Common Artists  The below bar chart shows the Top 15 Years with the most songs appearing on the top 500 List.1971 had the most songs of the list with around 21 songs. The other years are 1965, 1972, 1980, 1967, 1977, 1973, 1969, 1968, 1975, 1964, 1976, 1966, and 1992.    **Figure 8.** Most Common Year  The below bar chart shows the number of songs included in the Top 500 list by decade. The decade with the most songs was the 1970's with around 140 songs. The decades with the least songs were the 1930’s, 1940’s, and 2020’s all having less than 5 songs on the list.    **Figure 9.** Most Common Decade  The final bar chart shows the number of songs included in the Top 500 list broken up by genre. Rock is by far the most popular genre on the Top 500 with around 180 songs on the list. Other popular genres are Pop, Hip-Hop, and R&B.    **Figure 10.** Most Common Genre  The below scatterplot was used to understand if there was a difference between Popularity and Danceability over the years. The blue dots are graphing the Popularity and the orange dots are graphing the Danceability. The scatter plots overlap pretty accurately and overall have the shape and direction over the years.    **Figure 11.** Danceability and Popularity over the Years  The below graph shows a box plot of the Popularity by each of the 100s Rank. The popularity for the top 100’s rank has the highest median out of all other ranks and only one outlier around 20%. The remaining boxplots all have a handful of outliers and larger ranges in the plots. This indicates that the Rolling Stones Top 100 songs generally line up with public opinion on the songs.    **Figure 12.** Popularity by 100’s Rank  Next a WordCloud was generated to better understand the lyrics. Below is a WordCloud of all the lyrics of every song in the Top 500 List. The image shows that the most common words are ‘love’, ‘baby’, ‘might’, ‘time’, and ‘know.  **Figure 13.** Song Lyrics WordCloud  The next WordCloud generated was of all the song Titles in the Top 500. This image shows that the most common words are ‘Love’, ‘Time’, ‘Song’, ‘Feel’, and ‘Girl’.  **Figure 14.** Song Titles Word Cloud **models and methods** After data exploration the data needed to be vectorized. In doing so, stop words were removed, all tokens were made lowercase, and any words with less than 3 letters were removed. 3 methods for count vectorization were performed. Regular count vectorization, where everytime a word appears the count increases by one. Term frequency inverse document frequency where the number of counts a word contributes is inversely related to its popularity in the entire list of documents. And bernoulli, which is binary, the word either appears in the document or doesn’t. A snippet of each dataframe can be seen below. Note, after much cleaning and joining between dataframes many songs couldn’t be consolidated with the final dataframe. The final Dataframes consisted of 415 songs (rows) and 2984 words (columns). These are the base dataframes and from each of these labels can be added.    **Figure 15.** Count Vectorized Dataset    **Figure 16.** Term Frequency Inverse Document Frequency Dataset    **Figure 17.** Bernoulli/Binary Count Vectorized Dataset  For this investigation 4 models were utilized; Decision trees, Naive Bayes, Support Vector Machines, and Latent Dirichlet Allocation. For Decision trees and Naive Bayes 3 dataframes (count vectorizer, term frequency inverse document frequency, and bernoulli/binary) were trained across 5 different variables (100’s, 50’s, genre, popularity, and decade). This resulted in 30 models. SVM was trained for the same 5 classifications using the count vectorizer, adding an additional 5 models. Finally, LDA was generated using the count vectorizer dataframe.  Decision Trees are non-parametric machine learning algorithm used for classification purposes. The algorithm iteratively splits the data at high polarization points or nodes. Doing so uses the variables to split the data into their various categories. This results in many splits/branches forming a tree like structure of classification.  Multinomial naïve bayes model is one of the models that will be used for this analysis. Multinomial naïve bayes is a classification model that predicts the tag of the text. To do so, it calculates the probability of tags for a given sample and outputs the tag with the highest probability. The model also calculates these probabilities independently, that is the probability of one classification is not related to any other classification. In this assignment the model will predict multiple labels such as 100s Rank, 50s Rank, Popularity, Decade, and Genre.  Support Vector Machine, SVM, was the third model used for this analysis. Support Vector Machines are a machine learning algorithm used for classification problems, regressions, and outlier detections. SVM plots all observations and creates a line or hyperplane that separates the data into classes.  Latent Dirichlet Allocation, or LDA, is the last model that was used for this analysis. LDA is a probability model that uses Bayesian networks for topic modeling. LDA models randomness in topics for words and topics. The LDA model has three parameters which are the number of topics, the number of words per topic, and the number of topics per document.  For each of the Vectorization method 5 sets of testing and training data sets were created, one for each of the labels needed for analysis. These labels were 100s Rank, 50s Rank, Popularity, Genre, and Decade. For each of the datasets, 33% of the data was used for testing data and 67% of the data was used for training data. The labels were removed from these test and training sets and the models were created.  The fundamental analysis tool used for determining the efficacy of the generated models will be a confusion matrix. A sample confusion matrix is shown below as well as the calculations for precision, recall accuracy, and F-score. It should be noted that the values for all of these range from 0 to 1, with a value closer to 1 being desired.     |  | | Actual Values | | | --- | --- | --- | --- | | Positive | Negative | | Predicted Values | Positive | A | B | | Negative | C | D |     **Figure 18.** Sample Confusion Matrix.    Box A will contain the number of songs that the model correctly predicted positive (true positive). B will contain the number of songs that the model predicted as positive, that were actually negative (false positive). C contains the number of songs the model predicted as negative that were actually positive (false negative). Finally, box D contains the number of reviews the model correctly predicted as negative (true negative).  Precision measures the accuracy of predictions. Recall measures the ability of a model to correctly identify all points in a relevant class. F1 uses both values to give a more overarching result that describes the entire matrix. Finally, accuracy is simply the number of correct predictions over the total number of predictions. The formulas for these calculations are shown below.    **Figure 19.** Precision, Recall, F1-score, and Accuracy calculations |
| **results** | **technical results** **DECISION TREES**  For each category classified over (100’s, 50’s, genre, popularity, and decade) all 3 dataframes were used (Count vectorizer, term frequency inverse document frequency, and Bernoulli/binary). This resulted in 15 total models. To improve concision only the most accurate confusion matrices and accompanying decision tree for each classification will be discussed. It should be noted that a depth of 100 was used to generate the models but for visualization purposes a depth 5 tree was used to see the important words.    **Figure 20.** TFIDF Decision Tree Results (100’s)  For 100s classification the term frequency inverse document frequency was found to be the most accurate for the decision tree model (24.09%). This is only a slight improvement over the no information rate of 20%. It’s noteworthy that words such as wind, walking, and groove were seen as polarizing words for rank.      **Figure 21.** CV Decision Tree Results (50’s)    For the 50’s classification all models performed poorly. The count vectorizer and inverse document frequency had the same accuracy (10.2%). This is worse than the no-information rate.        **Figure 22.** Bernoulli Decision Tree Results (Genre)    The Bernoulli dataframe had the highest accuracy when classifying the genre. It was able to achieve an accuracy of 35.8%, significantly higher than the no information rate of 8.33%.      **Figure 23.** BN Decision Tree Results (Pop’s)    For classifying popularity the Bernoulli count vectorizer had the most success. This model yielded an accuracy of 35.8%. This accuracy is due in part to the fact there are only 4 possible popularity categories.      **Figure 24.** CV Decision Tree Results (Decade)    The count vectorizer model was the best at predicting the song decade based off lyrics. It had an accuracy of 27.0%, which is above the no information rate of 12.5%.    In summary, the decision tree models had mixed success. The best models generated had accuracies of 35.8% and were made using the binary count vectorizer to categorize the popularity and genre of the songs based on the lyrics. It found words such as “loves”, “whats”, and “yesterday” were the most helpful in polarizing the popularities. For all classifications (100’s, 50’s, genre, popularity, and decade) there was an even spread of dataframe usage. The count vectorizer was used twice, the TFIDF was used once, and the Bernoulli/binary count vectorizer was used twice. The most impressive classifier was arguably the Genre classifier as there were 12 possible categories (11 are shown as not all categories appeared in the testing data). The model correctly predicted 35.8% of the song’s genres based off the lyrics. A quick summary of the models can be seen below, note that only accuracy is shown as this method was concluded to be suboptimal and more focus was put on other model types.   |  | Results | | | --- | --- | --- | | Classification Category | Chosen Model | Accuracy | | 100’s | TFIDF | 24.09% | | 50’s | Countvectorizer | 10.2% | | Genre | Bernoulli/Binary | 35.8% | | Popularity | Bernoulli/Binary | 35.8% | | Decade | Countvectorizer | 27.0% |   **Figure 25.** Decision Tree Summarization Results  **Multinomial Naïve Bayes**  For each category classified over (100’s, 50’s, genre, popularity, and decade) all 3 dataframes were used (Count vectorizer, term frequency inverse document frequency, and Bernoulli/binary). This again resulted in 15 total models. Again, to help with concision, only the most accurate model for each classification will be shown.    **Figure 26.** Bernoulli/Binary Naïve Bayes Results (100’s)    The model with the best accuracy for predicting the 100’s place of the song was the model generated using the Bernoulli/binary dataframe. It yielded an accuracy of 25.5%, a slight increase over the no information rate of 20%.        **Figure 27.** Count Vectorizer Naïve Bayes Results (50’s)    Similar to what was seen in the decision tree model of the 50’s classification, 2 models performed identically. However, this time it was the count vectorizer and Bernoulli/binary dataframe. Both models generated from the dataframes created a model with 10.9% accuracy.      **Figure 28.** Term Frequency Inverse Document Frequency Naïve Bayes Results (Genre)  The “best” model for classifying the genre was found to be the term frequency inverse document frequency model with an accuracy of 35.7%. It should be noted that the model simply predicted that all sets of lyrics would fall into the most popular category. This model is a good example of a relatively accurate model with a very low precision and high recall. This model also had 0 precision, recall, and F-score for all other categories aside from rock. So aside from accuracy this model is very flawed.      **Figure 29.** Term Frequency Inverse Document Frequency Naïve Bayes Results (Popularity)  The term frequency inverse document frequency naïve bayes model performed the best at classifying popularity. This is the best model created, with an accuracy of 39.4% and an F1 score of .213. The model overpredicted the C popularity resulting in a lower precision, however it had a high recall of 0.83.      **Figure 30.** Bernoulli/Binary Naïve Bayes Results (Decade)    The best model for predicting the decade of a song based off the lyrics was the Bernoulli/binary naïve bayes model. This model achieved 24.8% accuracy. Given the 8 possible categories this puts it well above the no information rate of 12.5%. There are only 8 categories as the testing data didn’t include every decade. Some decades, for example the 70's, simply had more songs in the dataset.  In summary, 15 models were created using Naïve Bayes. The results from the best dataframe for each classification can be seen below. In addition, the F1 score has been included. As was the case with the decision tree models, the naïve bayes models had an even spread of dataframe usage. All 3 dataframes (Count vectorizer, term frequency inverse document frequency, and Bernoulli) generated at least one most successful model. The TFIDF popularity model was the best created, with an accuracy of 39.4%, well above the 25% no information rate.     |  | Results | | | | --- | --- | --- | --- | | Classification Category | Chosen Model | Accuracy | F1-Score | | 100’s | Bernoulli | 25.5% | 0.206 | | 50’s | Count Vectorizer | 10.5% | 0.109 | | Genre | Term Frequency Inverse Document Frequency | 35.7% | 0.043 | | Popularity | Term Frequency Inverse Document Frequency | 39.4% | 0.213 | | Decade | Bernoulli | 25.0% | 0.120 |     **SVM**  The first analysis ran using the SVM model was using 100s Rank to predict which rank each song belongs to. The below confusion matrix depicts the results of the model. From the below image the Top 100 rank was the most accurate to predict with 12 correct predictions. The 400-500 rank was not predicted at all in this model.    **Figure 31.** Confusion Matrix for SVM with 100s Rank  The below image shows the results of the model for the 100s rank label. This model had an accuracy of 22.63%. This model is slightly more accurate at predicting 400-500 labels rather than other labels.    **Figure 32.** Statistics for SVM with 100s Rank  The next analysis ran using the SVM model was using 50s Rank to predict which rank each song belongs to. The below confusion matrix depicts the results of the model. From the below image it is hard to determine which Rank was the most accurate to predict. The 450-500 rank was not predicted at all in this model.    **Figure 33.** Confusion Matrix for SVM with 50s Rank  The below image shows the results of the model for the 50s Rank label. This model had an accuracy of 8.76%. This model is slightly more accurate at predicting 450-500 labels rather than other labels.  **Figure 34.** Statistics for SVM with 50s Rank  The next analysis ran using the SVM model was using Genre to predict which Genre each song belongs to. The below confusion matrix depicts the results of the model. From the below image both Rock and Pop are the most accurate genres to predict. It is also shown that Pop and Rock are often confused with one another in the model. Many genres were not even able to predict using this model.    **Figure 35.** Confusion Matrix for SVM with Genre  The below image shows the results of the model for the Genre label. This model had an accuracy of 20.44%. This model is slightly more accurate at predicting Pop genres as opposed to other genres.    **Figure 36.** Statistics for SVM with Genre  The next analysis ran using the SVM model was using Popularity to predict which Popularity Grade each song belongs to. The below confusion matrix depicts the results of the model. From the below image Grade C is the most accurate Popularity Grade to predict.    **Figure 37.** Confusion Matrix for SVM with Popularity  The below image shows the results of the model for the Popularity Grade label. This model had an accuracy of 27.01%. This model is more accurate at predicting Popularity Grade C with a precision of 0.42.    **Figure 38.** Statistics for SVM with Popularity  The final analysis ran using the SVM model was using Decades to predict which Decade each song belongs to. The below confusion matrix depicts the results of the model. From the below image the 70s is the most accurate Decade to predict  **Figure 39.** Confusion Matrix for SVM with Decade  The below image shows the results of the model for the Decade label. This model had an accuracy of 20.44%. This model is more accurate at predicting the 2010s with a precision of 0.50.    **Figure 40.** Statistics for SVM with Decade  **LDA**  The first LDA analysis run was the LDA Model using 5 topics. Below shows the results of the LDA model with 5 topics. The “First doc in data:” shows the topic distribution for this specific document. The probability the document is the second topic is 99.8%. For the “Seventh doc in data” the probability the document is the fifth topic is 99%.    **Figure 41.** LDA 1 Probabilities  The below image attempts to show the topics and words associated with each one. Assumptions can be made as to what the topic is based off the words associated with each topics    **Figure 42.** LDA 1 with 5 topics  The second analysis run was the LDA Model using 15 topics. Below shows the results of the LDA model with 15 topics. The “First doc in data:” shows the topic distribution for this specific document. The probability that the document is the eleventh topic is 99.8%. For the “Seventh doc in data” the probability the document is the thirteenth topic is 99%.    **Figure 42.** LDA 2 Probabilities  The below image attempts to show the topics and words associated with each one. Assumptions can be made as to what the topic is based on the words associated with each topic. As the number of topics increases this chart becomes more difficult to interpret.    **Figure 43.** LDA 2 with 15 topics |
| **conclusions** | Many insights were made through the investigation of the Rolling Stones top 500 songs of all time dataset. The intention of this investigation was to learn about the role lyrics had to play in music. Particularly there was an interest in how lyrics affect the genre classification of music and how lyrics could affect the popularity of the song.  It was discovered that a genre classifier could be created with relative success (35.8%). It was discovered that the most polarizing words for determining genre tended to be abstract concept words rather than nouns. For example words like “love”, “nothin”, and “doubt” helped differentiate the different genres more than nouns about people, places, and things.  It was also discovered that popularity, an incredibly important variable for both recording studios/artists and music streaming services, could be classified with very high success (almost 40%). This leads to the conclusion that you could use lyrics to help determine how popular a song will become.  Topic modeling was also achieved. Different categories of lyrics could be grouped together. For example, it was observed that words such as “love”, “baby”, and “night” appeared together in a topic. Perhaps this grouping could involve love songs. Other topics included words such as “away”, “dream”, and “lonely” which could be a heartbreak topic. It was discovered that topic clustering could be performed.  In conclusion many discoveries about lyrics and their importance in music were found and quantified. The insights gained from this investigation should benefit not only music streaming services but also aspirational artists and recording studios. Multiple successful models were created, refined, and tested throughout this process. Overall this project was a great success. |