The Rolling Stone's 500 Greatest Songs of All Time

Mikayla Scott, Michael Harper, Evan Helig

IST 718 - Big Data Analytics
Syracuse University
June 19, 2023

Specification

Problem

The Rolling Stone is a magazine that is best known around the world for their reporting on music and pop culture. With the first issue being published in 1967, Rolling Stone has stood the test of time, and has reported on music legends from John Lennon to rising stars like Lizzo and Drake. In 2004 Rolling Stone published the first ever "500 Greatest Songs of All Time" list, they followed that list with 26 updated songs in 2010, and most recently released the 2021 version of the list. The new version of the list was created by polling more than 250 musicians, artists, producers, writers in the music industry on their greatest songs of all time. Each of the participants ranked their top 50 songs and Rolling Stone calculated the results. Of the resulting songs there were more than 4,000 songs suggested to be on the list.[1]

As the music industry faces new challenges with the growth of artificial intelligence, music producers and musicians are turning to analytics to make decisions. The purpose of this study was to find patterns and relationships that would help to advise artists and music producers on musical choices that will maximize streams, revenue and popularity.

There are several business questions that this study aims to answer, those questions are outlined in the list below:

1. Are there any key indicators that would demonstrate to a producer or record company that a song has a greater chance at becoming a hit?
2. What are the relationships between the songs, specifically their lyrics, on the Greatest List? What songs are most closely related?
3. What are the most popular topics of songs in the list?
4. What song characteristics influence the popularity of a song?

The rest of this document aims to provide answers to the proposed questions through exploratory data analysis, and different machine learning models.

Hypothesis

As a team the general hypothesis for this analysis was that there would be a correlation between certain variables and the other songs on the list. For the topic modeling it was predicted that there would be a general consensus of topics that would be derived from the lyrics. With the topics modeled, it would provide insights into the topics of songs that are considered amongst the greatest of all time. With the graph ML implementation of the semantic search between song lyrics, it was predicted that there would be a high number of high scoring relationships between the lyrics of each of the songs. This also includes high scoring relationships between the years that songs were released. For the logistic regression model, the team believed that there could be a potential relationship that could be defined between the characteristics that make up a song (variables in the set) and the overall ranking of the song on the the Rolling Stone Top 500 Greatest Songs of All Time list.

Data

[1] The Editors of Encyclopaedia Britannica. (2023, June 17). *Rolling Stone | American magazine*. Encyclopedia Britannica. https://www.britannica.com/topic/Rolling-Stone

The data for this study was a dataset found on kaggle. The dataset was composed of 29 total variables. 9 categorical values, such as key of the song, the song title, and artist, and 20 numeric variables. The numeric variables in the set range from tempo, to duration of the song, to different spotify metrics such as danceability, energy and speechiness. There were several columns that were added by the team so that various versions of sentiment analysis and natural language processing techniques could take place. This included the scraping of lyrics. Lyrics were scraped from various websites using a user defined python function utilizing the beautiful soup library. Once the lyrics were scraped, they went through standard text processing phases like stop word removal. Once cleaned the lyrics were merged into the original dataset using a left join on the song name. The sentiment of each of the songs was determined using the NLTK VaderLexicon package. The scores for negativity, positivity, neutrality and the overall compound score of each of the song lyrics were also stored in the original data set. Data was stored within a Pandas data frame and was used for analysis throughout all stages of the study.

Observation

Exploratory Data Analysis

The exploratory data analysis (EDA) performed on this dataset was done with the goal of gaining a baseline understanding of the distribution of data across several different variables. As stated in the data section, after the data was prepped there were 500 songs, and more than 30 variables to play with for EDA. The distribution of songs over the years was among the first aggregates of data to be visualized. The treemap below demonstrates the years on the list, what songs were produced in those years, and the total count of songs within those years. The bigger the box, the more songs. When the graphic is in its HTML format, it is completely interactive and the song names and artists that sing the songs that fall into that year become visible to the user. The year that has the most songs on the list is 1971, with a grand total of 21 songs. In 1971. There are several artists that had two songs on the list including David Bowie, The Who and Joni Mitchel. Some notable songs include “Imagine” by John Lennon, and “Stairway to Heaven” by Led Zeppelin. See the treemap below for the distribution of songs over the years.

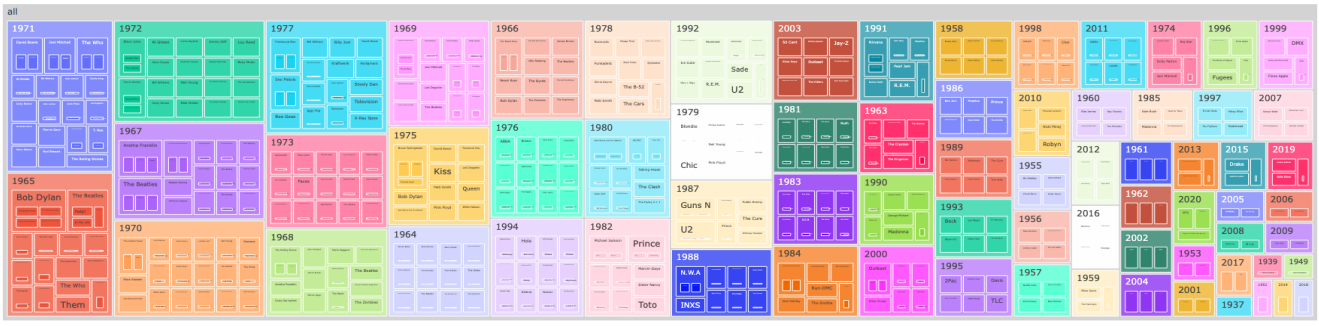


Figure 1: Treemap Distribution: Artists and Songs by Year

To look further into the year variable, and for better visualization where the interactive components of charts are limited, the songs were broken down into 5 year timeframes to find the date range with the most songs on the list. The range with the most songs on the list is from 1965 to 1974 with a grand total of 159 songs. There were two other notable spikes, one of them occurring around the early 1980’s, from 1980 to 1984 and then again in the early 1990’s, specifically from 1990 to 1994. The spike in the early 80’s accounts for 46 of the songs on the greatest list, and the spike in the early 90’s accounts for 44 songs.

With these four date ranges alone, the 227 combined songs account for more than 45% of the entire dataset. The distribution described above can be seen in *Figure 2*, displayed below:

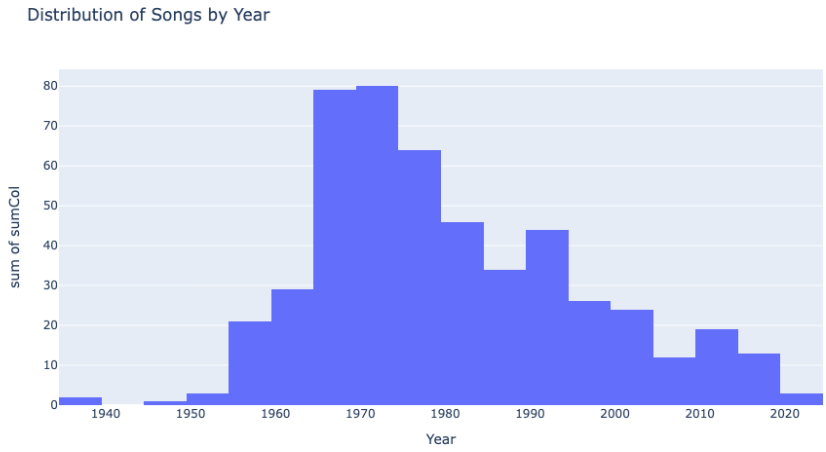


Figure 2: Distribution of Songs by Year

Lyrics for each of the songs were scraped and added to the original dataset. Sentiment was calculated using NLTK Vader on the lyrics. After running the sentiment analysis each song was given four scores, negative, neutral, positive and a compound score. The next phase of the EDA looks at the compound sentiment scores of the songs grouped by year. Most of the years across the dataset have a positive compound score, meaning that most of the lyrics inside that year had a positive connotation; however, there are a handful of years that have overall negative compound scores. Some of these years include 2020,, 2006, 1999, and 1993. In the chart below, negative scores are represented by darker colored bars, where positive scores are shown in the lighter colors.

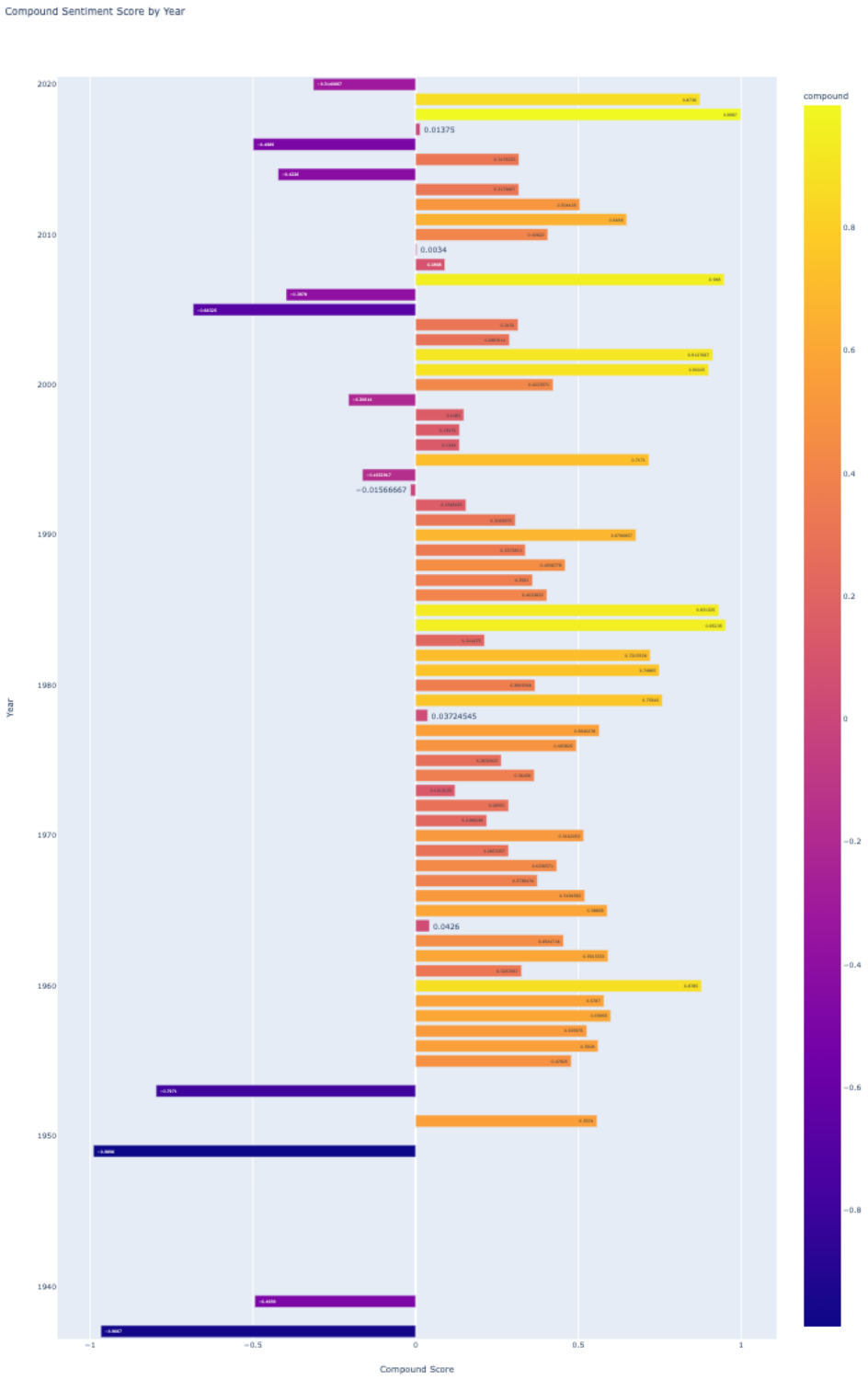


Figure 3: Compound Sentiment Score of Songs by Year

The songs on the list were also aggregated by artist to show the artist on the list with the most songs. There were 370 different artists represented on the top 500 list. Of those 370 artists, there are names from Bruce Springsteen to Lizzo, covering a vast amount of genres and music styles. There are 297 artists on the list that only have one song, accounting for 80% of all the artists represented in the set. The figure below shows the top 50 artists by the count of songs that they have on the Top 500 list:

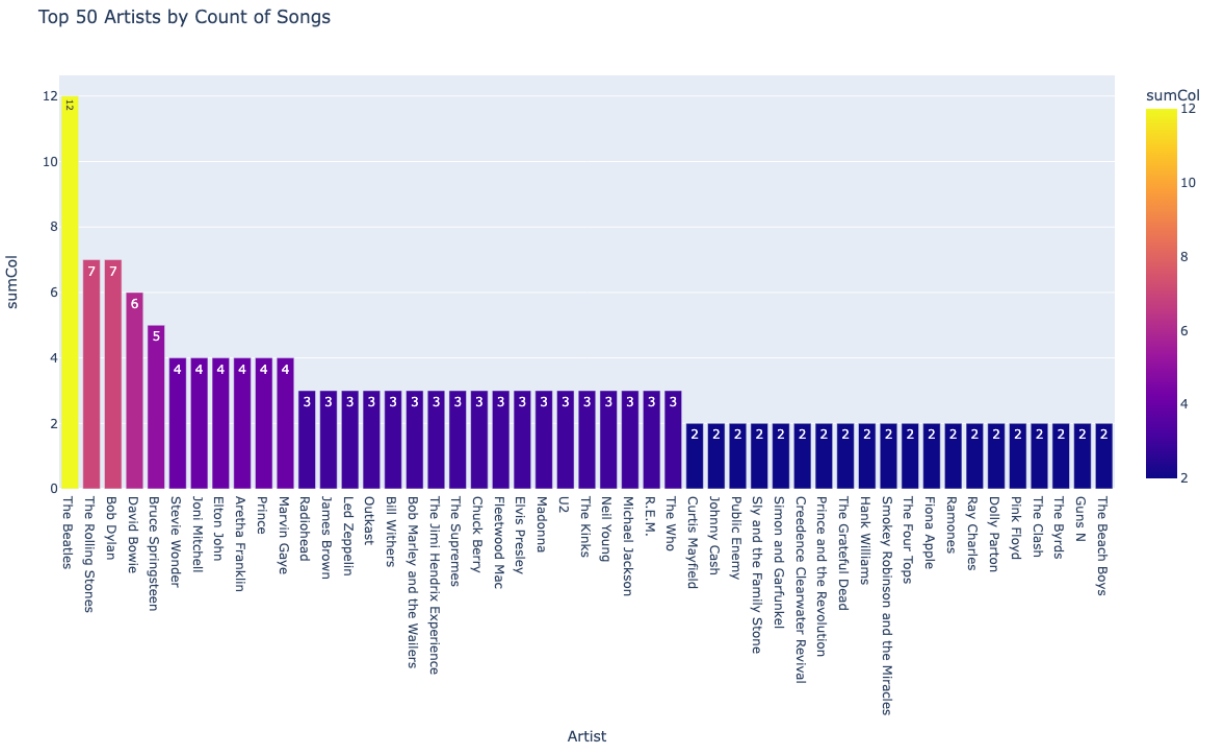


Figure 4: Top 50 Artists by Count of Songs

There were nine numeric variables that were brought in for spotify to help describe each of the songs. The variables included popularity, danceability, energy, loudness, speechiness, liveness and valence. The distribution of each of these values was plotted against the number of songs. After analyzing the distributions the most songs that occur most on the lists are songs that have a high popularity score, have low speechiness and acousticness scores, low instrumentalness scores, and low liveliness scores. Variables that were somewhat evenly distributed throughout included danceability, energy, loudness and valence. See the distributions plotted below:

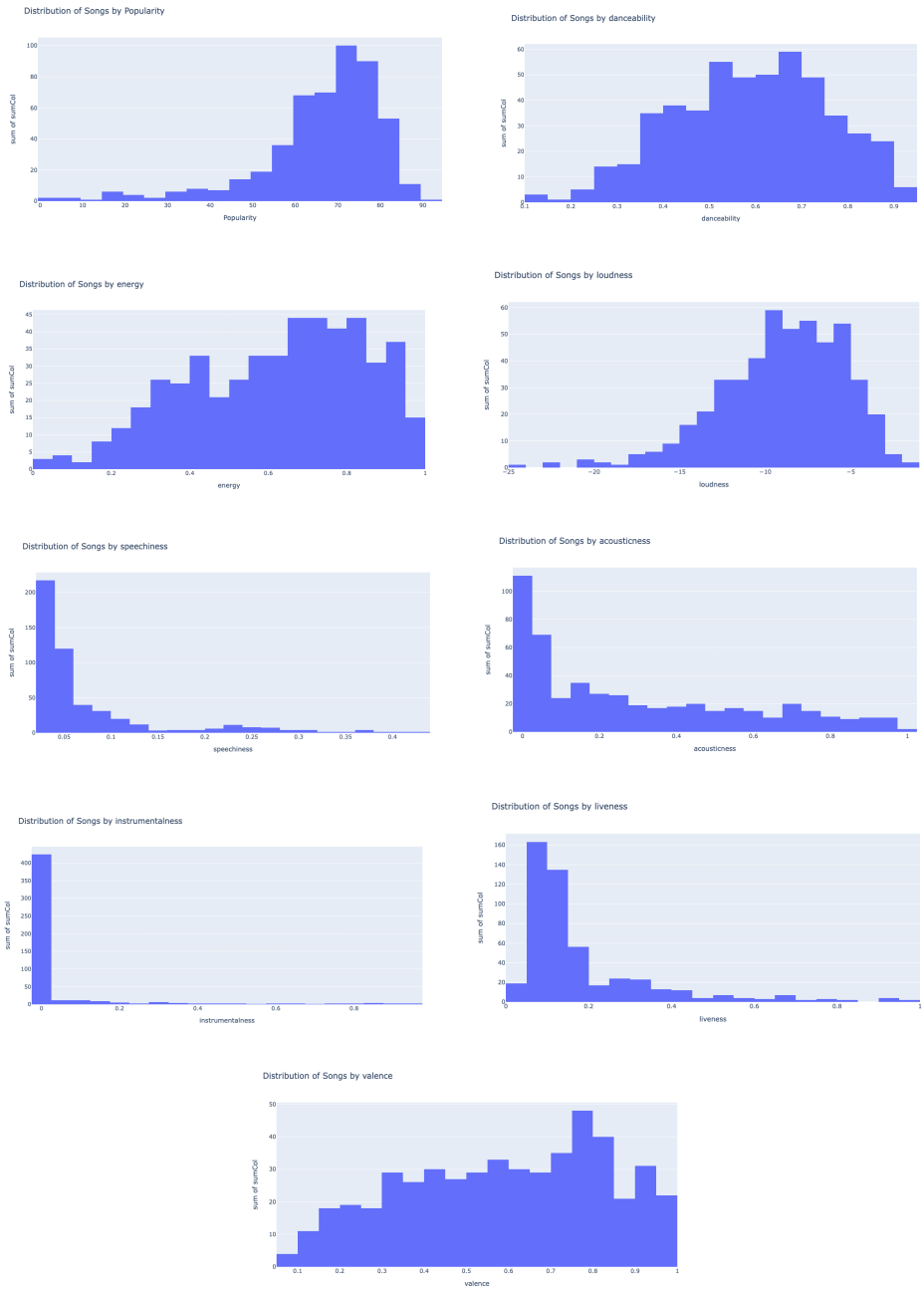


Figure 5: Spotify Numeric Variable Distributions

The average duration of a song overtime was plotted to see if there was any correlation or patterns that could be identified with the release of music streaming technologies such as the LP record and iPod. LPs came out in 1948, the Sony Walkman was released in 1979, CDs were released in 1982, the first MP3 player hit the markets in 1991, the iPod was released in 2001, and the first iPhone was released in 2007. As technology was developed, more storage became available for artists to store and share their music. Notice in the figure below that there is a steady upward trend from the early 1960's to around the 1990's, then after the 90's the length of songs starts to plateau again.

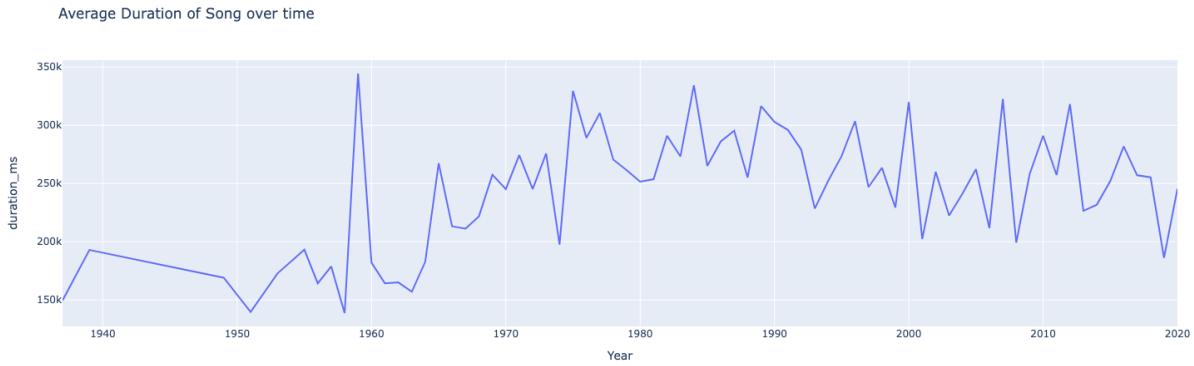


Figure 6: Average Duration of Top 500 Songs by Year

Analysis

There were three main modeling initiatives to help answer the business questions that were outlined in the *problem* section. The first model attempted to answer ____ through the _____ process. Something about the second, something about the third.

Logistic Regression Classifier

To answer the question, "What song characteristics influence the popularity of a song?", the power of machine learning was leveraged to predict the popularity of songs based on various song characteristics. Logistic regression was used, to help understand how the song's features contributed to a song's popularity. For each song this dataset included various quantitative ratings, such as those measuring its level of Energy, Danceability, Loudness, Liveness, Valence, Acousticness, Speechiness, and Popularity. For this regression, Popularity was our *Y* and all other quantitative ratings were *X*.

To implement the logistic regression model, *Pandas* was used for data manipulation, *sklearn* for machine learning, and statsmodels for more detailed statistical analysis. Matplotlib was used for data visualization.

One of the features was 'Popularity', which was the target variable. To convert this continuous variable into a binary outcome (popular or not), a popularity threshold was defined. For instance, if a song's popularity was greater than 50, it was classified as popular (1), otherwise not popular (0).

After defining our features and target variable, the data was split into a training set and a test set. The model was trained on the training set, and then its performance was evaluated on the unseen test set. To ensure reproducibility of results, a specific random state was set.

The logistic regression model was created using *sklearn's* LogisticRegression, fitted the model to the data, and then used it to make predictions on the test set. The model's performance was then evaluated by computing the accuracy of the classifier on the test set, generating a confusion matrix, and printing a classification report that included precision, recall, f1-score, and support for each class.

```
Accuracy of logistic regression classifier on test set: 0.88
[[ 0 12]
 [ 0 88]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        12
           1       0.88      1.00      0.94        88

    accuracy                           0.88       100
   macro avg       0.44      0.50      0.47       100
weighted avg       0.77      0.88      0.82       100
```

Figure 7: Logistic Regression Classifier

The logistic regression classifier performed well on the test set, achieving an accuracy of 88%. This means that the model was able to correctly predict the popularity of 88% of the songs in the test set.

When looking at the confusion matrix, the model has classified 88 songs correctly as popular (true positives), and it has not classified any of the songs as not popular correctly (true negatives). However, it has misclassified 12 songs that are not popular as popular (false positives) and it did not misclassify any popular song as not popular (false negatives).

In terms of precision, recall, and f1-score, the model performed excellently in predicting popular songs (class 1), with a precision and recall of 88%, and an f1-score of 94%. This suggests that among the songs that the model predicted to be popular, 88% were truly popular, and the model identified 100% of the popular songs. The high f1-score indicates a harmonic mean of precision and recall, thus confirming the model's robust performance in predicting popular songs.

The logistic regression analysis provided a set of coefficients for each of the song features. The Logit Regression Results table lists the coefficients of the model, along with their standard errors, z-scores, p-values, and confidence intervals.

Logit Regression Results

Dep. Variable:	Success	No. Observations:	500
Model:	Logit	Df Residuals:	487
Method:	MLE	Df Model:	12
Date:	Mon, 19 Jun 2023	Pseudo R-squ.:	0.07579
Time:	18:08:31	Log-Likelihood:	-158.18
converged:	True	LL-Null:	-171.16
Covariance Type:	nonrobust	LLR p-value:	0.01093

	coef	std err	z	P>|z|	[0.025	0.975]
danceability	0.3116	1.153	0.270	0.787	-1.948	2.571
energy	0.5330	1.327	0.402	0.688	-2.068	3.134
key	-0.0430	0.043	-0.992	0.321	-0.128	0.042
loudness	-0.0119	0.066	-0.181	0.856	-0.141	0.117
mode	0.0654	0.352	0.186	0.852	-0.624	0.755
speechiness	-1.0166	2.070	-0.491	0.623	-5.073	3.040
acousticness	0.0055	0.685	0.008	0.994	-1.336	1.347
instrumentalness	-2.1768	0.701	-3.107	0.002	-3.550	-0.804
liveness	-2.4116	0.726	-3.322	0.001	-3.835	-0.989
valence	-0.4367	0.814	-0.536	0.592	-2.032	1.159
tempo	0.0033	0.006	0.534	0.593	-0.009	0.015
duration_ms	-2.723e-06	1.28e-06	-2.129	0.033	-5.23e-06	-2.17e-07
const	3.0151	1.724	1.749	0.080	-0.363	6.393

Figure 8: Logistic Regression Results

The dependent variable for this model is 'Success', indicating whether or not a song is considered popular. With 500 observations in total, the model generated 12 degrees of freedom.

The pseudo R-squared value is 0.07579, which indicates the proportion of variance in the dependent variable that can be explained by the independent variables. A value close to 1 would indicate a perfect fit. In this case, the value suggests a fair degree of unexplained variance. It's worth noting, however, that pseudo R-squared values aren't directly comparable to R-squared values in linear regression.

Looking at the individual coefficients, we see that 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'valence', 'tempo', and 'duration_ms' have p-values greater than 0.05, indicating that these predictors are not statistically significant at the 5% level.

On the other hand, the 'instrumentalness', 'liveness', and 'duration_ms' features have p-values less than 0.05, making them statistically significant predictors of song popularity at the 5% level. Specifically, 'instrumentalness' and 'liveness' have negative coefficients, suggesting that increases in these characteristics are associated with decreases in the likelihood of a song being popular. 'Duration_ms', on the contrary, has a negative coefficient but its magnitude is quite small, indicating only a slight impact on song popularity.

Finally, the constant term (or intercept) in our model is 3.0151, but with a p-value of 0.080, it is not significant at the 5% level. This means the null hypothesis would not be rejected and that this coefficient equals zero.

These results give us important insights into the factors that contribute to a song's popularity, helping to illuminate the characteristics that can make a song a hit.

In order to better assess the performance of our logistic regression model, a Receiver Operating Characteristic (ROC) curve and compute the Area Under the Curve (AUC) score was generated. The ROC curve is a useful tool for understanding the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) for different threshold settings. An AUC score closer to 1 indicates a better performing model.

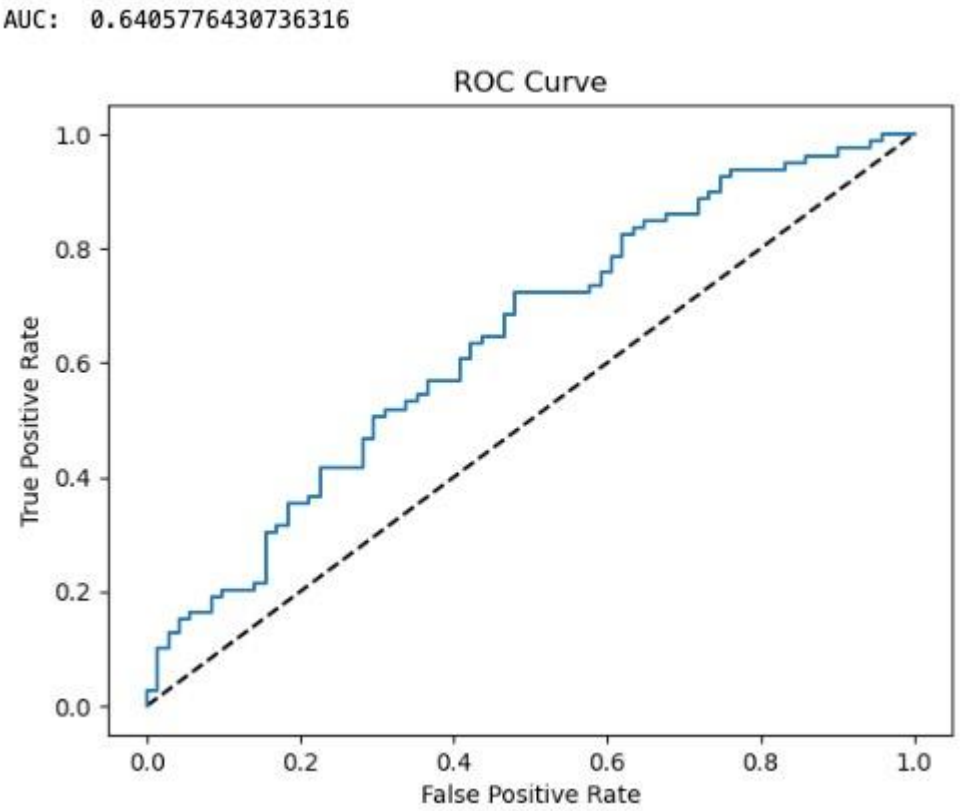


Figure 9: Logistic Regression ROC Curve

In our case, the calculated AUC score is 0.64. This suggests that our model has a reasonable discriminatory ability, i.e., it has a 64% chance of correctly distinguishing between positive (popular) and negative (not popular) classes.

The ROC curve generated from our model shows the TPR and FPR at various thresholds. The dashed line represents a random classifier (AUC = 0.5), and the solid line is our model's performance. Given that the model's curve is above this line, we can conclude that it performs better than a random classifier. However, there's still substantial room for improvement, as a perfect classifier would have an AUC of 1 and its ROC curve would touch the top left corner of the plot.

This evaluation supports the idea that while our model can be useful for identifying key song characteristics affecting popularity, there are likely other unexamined factors contributing to a song's success. Further iterations of the model could seek to incorporate additional predictors to improve classification performance.

Semantic Search: Relationships between Lyrics

Semantic search was used to gain a deeper understanding of how the lyrics of the songs on the Rolling Stone 500 Greatest Songs List were related. Semantic search is defined by Wikipedia as "searching with meaning, as distinguished from lexical search where the search engine looks for literal matches of the query words or variants of them, without understanding the meaning of the query".[2] To get to the point where text could be compared, the text first had to be embedded. Text embedding is the numeric representation of words and phrases. For this specific experiment the lyrics were cleaned as they would be for traditional sentiment analysis, lowercase, punctuation and stopwords (from NLTK stopword predefined list) were also removed. Once the lyrics were considered to be cleaned, the cleaned lyrics were then fed into the embeddings model from the "sbert" library.

After running through the model each one of the songs were transformed into a vector of 768 numeric values. From there the vectors of each of the songs were compared using cosine similarity to define the distance between each of the song vectors. See the formula for cosine similarity below:

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

The five most similar songs were saved in a dataframe. One to one relationships between each of the songs and their most related top five based on the cosine similarity of the embedded lyrics were used to create a graphic visualizing the results. See an example of the dataframe in the table below:

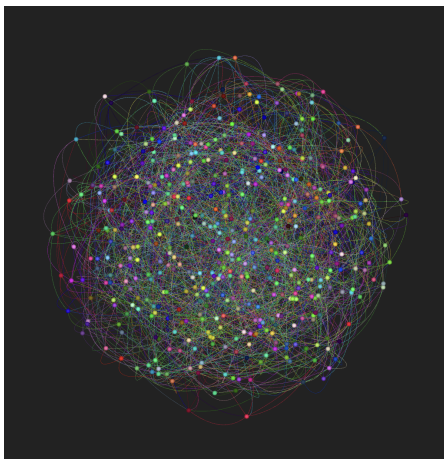
Song 1	*Artist Song 1*	*Song 2*	*Artist Song 2*	*Scores*	*Color*
Respect	Aretha Franklin	Paper Bag	Fiona Apple	0.781108	#FCDEE6
Respect	Aretha Franklin	Idioteque	Radiohead	0.779334	#FCDEE6
Respect	Aretha Franklin	Like a Rolling Stone	Bob Dylan	0.768255	#FCDEE6
Respect	Aretha Franklin	Bad Romance	Lady Gaga	0.765328	#FCDEE6

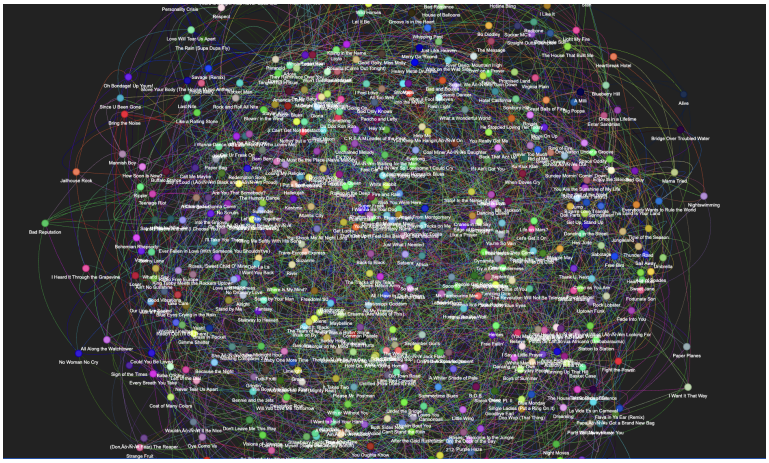
[2] Wikipedia contributors. (2023). Semantic search. *Wikipedia*. https://en.wikipedia.org/wiki/Semantic_search

Respect	Aretha Franklin	Bam Bam	Sister Nancy	0.760423	#FCDEE6

Table 1: One-to-One Relationship Mapping of Cosine Similarity Between Song Lyrics

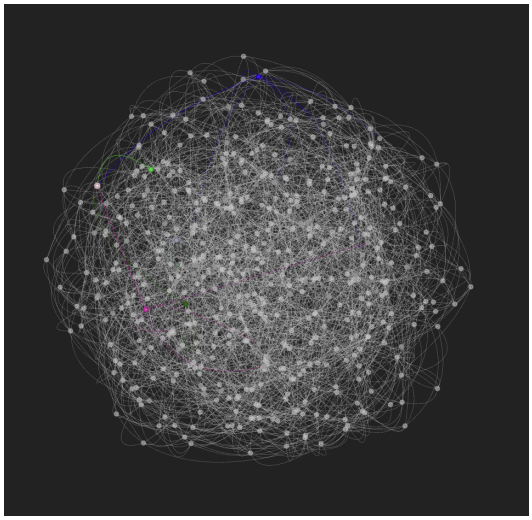
To visualize results the songs were then mapped using a network graph. Each song is a different node, and each artist has a different color, the paths between each of the nodes (edges) are weighted by the cosine similarity between songs. The thicker the edge, the more similar the songs are to each other. See the graphics below for the network of songs:

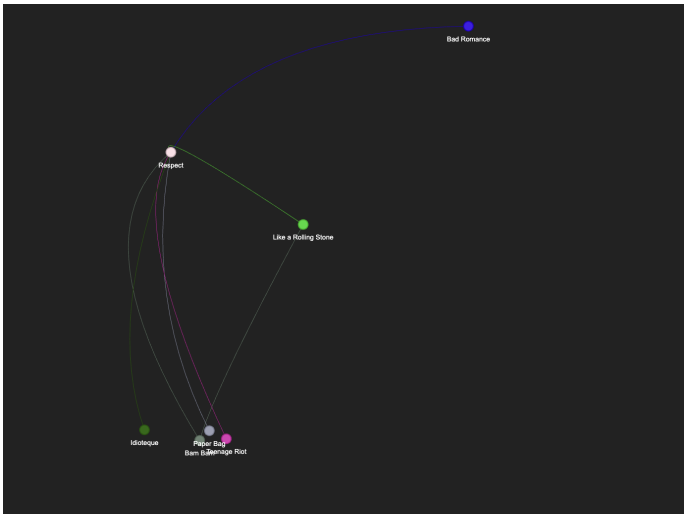




Figure(s) 10 & 11: Top 5 Related Songs for Entire 500 List

Looking at the network can be a lot to take in, patterns, connections and communities can be hard to identify by just looking at the network graph as a whole. To help parse the information the graph can be filtered by name of the song, the artist of the song, the cosine similarity of the song, and any other identifying property that exists in the one-to-one relationship table. To look at the specific example from the table above, Aretha Franklin's *Respect*, see the figures below:





Figure(s) 12 & 13: Top 5 Related Songs for Aretha Franklin's Respect

The songs that were the most similar lyrics are shown in the table below:

Song 1	*Artist Song 1*	*Song 2*	*Artist Song 2*	*Scores*	*Color*
Cissy Strut	The Meters	Life on Mars?	David Bowie	1	#432740
Tangled Up In Blue	Bob Dylan	The Boys are Back in Town	Thin Lizzy	1	#325E09
Subterranean Homesick Blues	Bob Dylan	Life on Mars?	David Bowie	1	#325E09
Untitled (How Does it Feel)	D. Angelo	Super Bass	Nicki Minaj	1	#76FEE9
Green Onions	Booker T. and the MGs	The Message	Grandmaster Flash and the Furious Five	1	#3BEF09

Table 2: Top 5 Most Similar Songs

The songs that were the least similar across the entire top 500 list are shown in the table below:

Song 1	*Artist Song 1*	*Song 2*	*Artist Song 2*	*Scores*
Penny Lane	The Beatles	Never Tear Us Apart	INXS	-0.234849
Penny Lane	The Beatles	Born To Run	Bruce SpringSteen	-0.234849

Green Onions	Booker T. and the MGs	Da Doo Ron Ron	The Crystals	-0.234849
Untitled (How Does It Feel)	D. Angelo	Ripple	The Grateful Dead	-0.234849
Cissy Strut	The Meters	Truth Hurts	Lizzo	-0.234849

Table 3: Least Similar Songs

The purpose behind defining the relationships between each of the songs and their lyrics was to identify potential patterns and/or communities of writers that are similar across the decades. Where this provided the team with some insights into similarities in songs specifically, there was nothing to indicate communities or groups of artists surrounded around a specific node. The artist that had the most similarities across the top 5 similar songs was the Beatles, at 60 connections, then David Bowie at 39, followed by Bob Dylan at 37 connections. The song that has the most top 5 connections was Walk on By by Dionne Warwick. The paths from Walk on By are shown in the figure below:

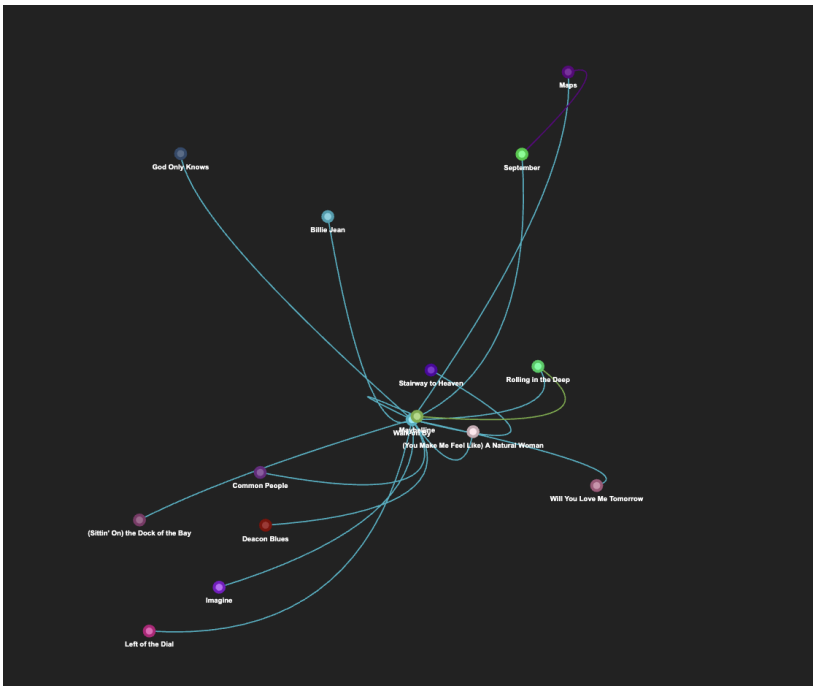


Figure 14: Walk on By Related Songs

The same study was also done on years to see if there was a relationship between the songs that were being released and the year that they were released in. The same methodology was used, the dataset was grouped into song release years, and lyrics for each of the years were combined into a large string. From there the large “year” lyric strings were fed into the same embedding model. Similarity scores were calculated from year to year and results were then mapped on a network graph. See that network graph below:

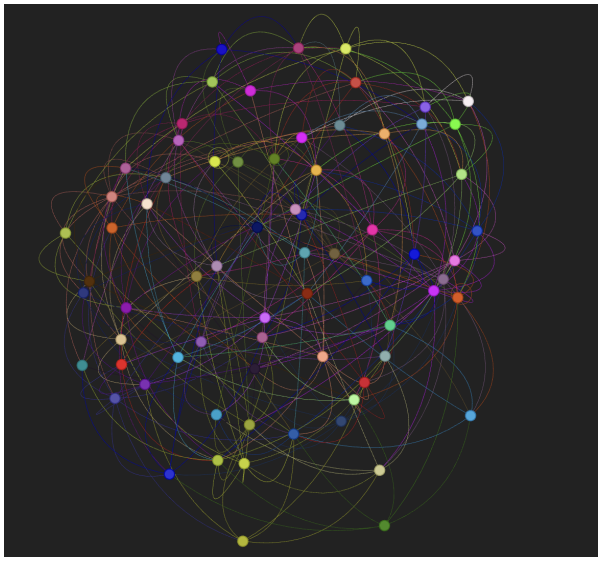


Figure 15: Top 5 Related Years Across Entire Set

There were 71 years represented in this set, ranging from the early 1940's, up into the 2020's. The year's with the most similarities are 1966, 2004, 2013, 1967 and 1939 each with 9 associated years. Those years are shown in the subset of the network graph shown below, where each node is a different year in the dataset:

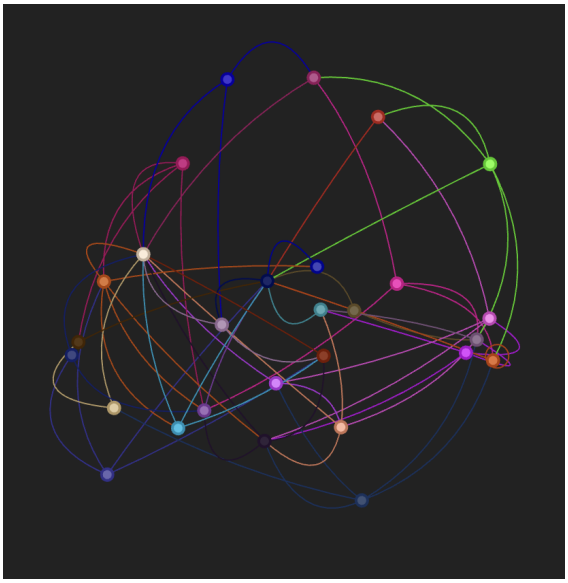


Figure 16: Most Common Related Years

The most similar years are shown in the table below:

Year	*Year 2*	*Scores*
2014	1965	0.895143
1974	2008	0.895143
1968	2016	0.878887
1964	2004	0.878887
2001	1991	0.878867

Table 4: Most Similar Songs by Year

The least similar years are shown in the table below:

Year	*Year 2*	*Scores*
1961	1988	0.282659
2013	1956	0.282659
1982	1974	0.291319
1961	1991	0.291319
1939	1956	0.390807

Table 5: Least Similar Songs by Year

Overall the similarity between the lyrics did show relationships between the lyrics of songs on the list, and the years that songs were released. It did not show general conclusive patterns that could be taken away from as definitive answers to the questions, What are the relationships between the songs on the Greatest List, and what songs are most closely related?

Topic Modeling and T-SNE Visualization of Lyrics

From the Gensim package, topic modeling was applied to the pre-processed lyrics. The Gensim package contains all the functions necessary to perform all processing and requires a “Bag-of-words” format for applying the Latent Dirichlet Allocation algorithm to the text. Despite the text being processed, the Gensim simple preprocessing functions were applied to obtain the correct data structures. Once a corpus was obtained, the algorithm was applied in a for loop to iterate through the corpus 15 times, adding an additional topic at each iteration. A user defined function was then applied to each topic to determine a similarity measure.

Jaccard Similarity = (number of observations in both sets) / (number in either set)
Notation = J(A, B) = |A∩B| / |A ∪ B|

Jaccard similarity produces a score on a scale of 0 to 1, with 0 representing no similarity and 1 representing complete similarity. This score is useful to evaluate LDA topic modeling when used in combination with the Gensim Coherence score. The idea is to achieve a higher coherence among the topics and a lower similarity score. This ensures that each topic has related words within it, while minimizing the overlap of keywords among the topics. Figure 7 shows a graphical analysis of the average topic overlap in (blue line) and the topic coherence score (orange line). The goal in determining the optimal number of topics is to find the point at which the orange line exceeds the blue line by the maximum distance.

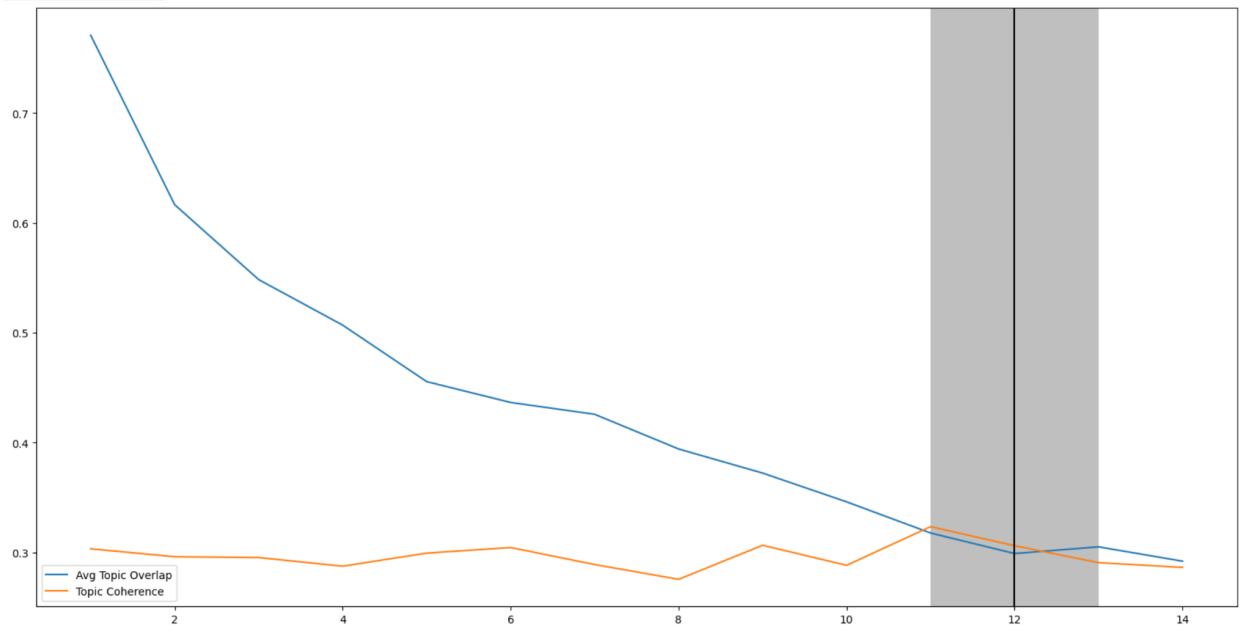


Figure 17: Topic Modeling Analysis

As can be seen in the Fig 17 graphic, the coherence score remained relatively low across all 15 topic quantities, while the average score of Jaccard similarity reduced on a steady exponential rate through 12 topics. This means that the topics of the 500 songs have little coherence overall with respect to subject matter and that they have high semantic similarity until grouped into at least 5 topics.

From this analysis, a single Gensim LDA model was created for 12 topics, holding all other settings identical to the models produced for the above graphical analysis. The resulting model had the following topic keywords output displayed in Figure 18:

```
[(0,
  '0.044*"wish" + 0.022*"never_met" + 0.017*"leg" + 0.014*"deep" + '
  '0.013*"dream_dream" + 0.013*"police" + 0.011*"believe" + 0.011*"watch" + '
  '0.011*"ball" + 0.011*"gonna_fall"'),
 (1,
  '0.085*"shake" + 0.036*"son" + 0.017*"sh_shake" + 0.017*"save" + '
  '0.017*"desire" + 0.013*"player" + 0.013*"folk" + 0.012*"claim" + '
  '0.010*"lean" + 0.009*"lonely_people"'),
 (2,
  '0.023*"think" + 0.022*"never" + 0.018*"long" + 0.018*"try" + 0.017*"know" + '
  '0.016*"away" + 0.015*"time" + 0.014*"hand" + 0.014*"cry" + 0.014*"keep"'),
 (3,
  '0.067*"roll" + 0.037*"fire" + 0.030*"gold" + 0.026*"heart" + 0.025*"crazy" '
  '+ 0.020*"night" + 0.019*"keep" + 0.017*"nasty" + 0.012*"drive" + '
  '0.009*"burn"'),
 (4,
  '0.029*"tell" + 0.018*"go" + 0.018*"boy" + 0.014*"town" + 0.012*"fast" + '
  '0.011*"nice" + 0.011*"road" + 0.011*"round" + 0.010*"kid" + 0.009*"toast"'),
 (5,
  '0.063*"get" + 0.026*"bitch" + 0.024*"shit" + 0.021*"fuck" + 0.018*"nigga" + '
  '0.016*"hit" + 0.013*"gon" + 0.013*"ass" + 0.010*"thank" + 0.009*"bad"'),
 (6,
  '0.107*"want" + 0.045*"let" + 0.034*"get" + 0.026*"night" + 0.026*"funk" + '
  '0.021*"ready" + 0.017*"stop" + 0.015*"set" + 0.014*"light" + '
  '0.012*"common_people"'),
 (7,
  '0.201*"love" + 0.089*"baby" + 0.034*"feel" + 0.033*"good" + 0.025*"need" + '
  '0.022*"give" + 0.020*"heart" + 0.015*"get" + 0.015*"way" + 0.011*"know"'),
 (8,
  '0.047*"make" + 0.028*"right" + 0.026*"feel" + 0.025*"get" + 0.017*"fight" + '
  '0.017*"real" + 0.015*"run" + 0.010*"cause" + 0.010*"problem" + '
  '0.009*"power"'),
 (9,
  '0.082*"stand" + 0.036*"walk" + 0.022*"rain" + 0.019*"purple_rain" + '
  '0.017*"answer" + 0.016*"yesterday" + 0.014*"window" + 0.013*"singing" + '
  '0.013*"like" + 0.012*"search"'),
 (10,
  '0.042*"go" + 0.032*"get" + 0.027*"know" + 0.027*"say" + 0.021*"come" + '
  '0.017*"take" + 0.015*"see" + 0.013*"let" + 0.012*"time" + 0.011*"tell"'),
 (11,
  '0.055*"really" + 0.035*"chain_chain" + 0.028*"fool" + '
  '0.025*"hallelujah_hallelujah" + 0.018*"sugar" + 0.017*"tie" + 0.017*"pussy" '
  '+ 0.014*"instrumental" + 0.014*"ache_someday" + 0.012*"stuff"')]
```

Figure 18: LDA 12-Topic Keyword Output

The author utilized pyLDAvis to produce an HTML output tool (Figure 19) for the purpose of manually assessing these categories and applying a label.

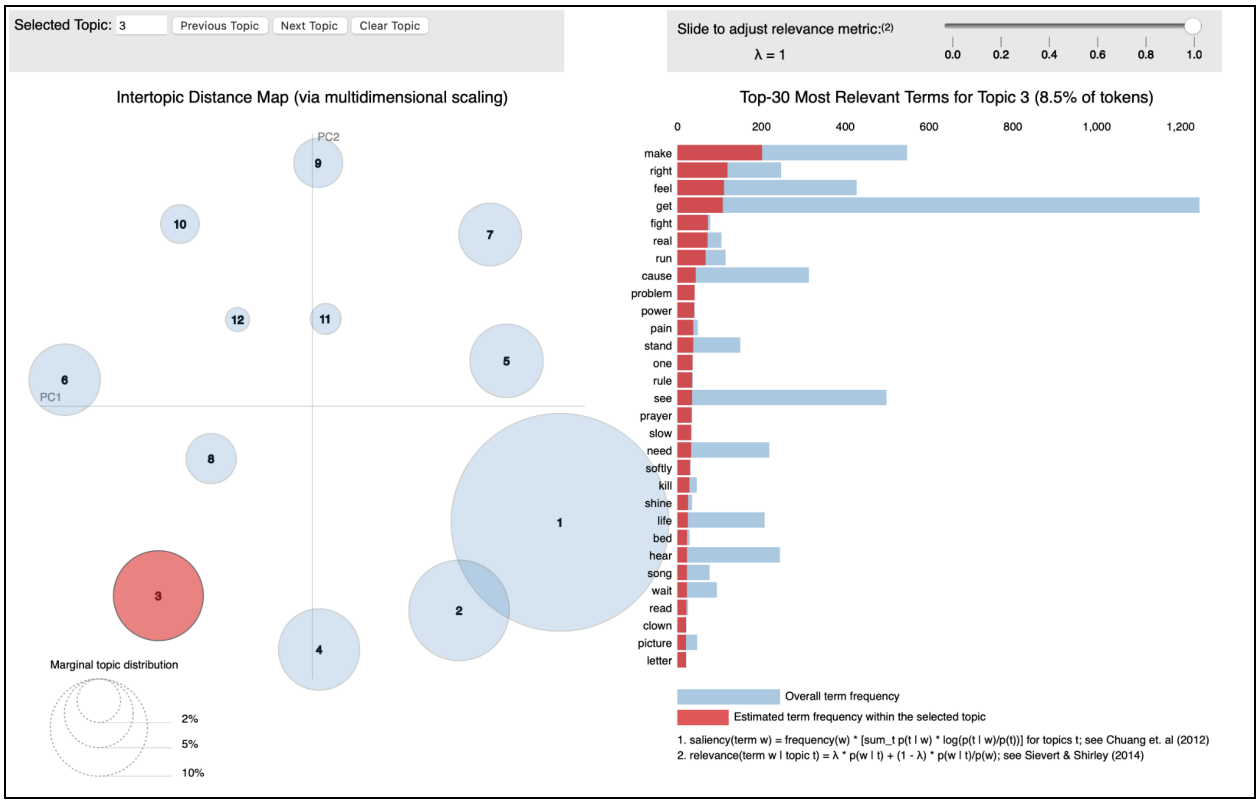


Figure 19: pyLDAvis notebook analysis tool

After assessing these categories, the author produced the following 12 topic labels:

0: "Wishful Thinking",
1: "Emotional Resilience",
2: "Reflective Thoughts",
3: "Passionate Energy",
4: "Journey and Exploration",
5: "Raw Emotions",
6: "Desire and Nightlife",
7: "Love and Affection",
8: "Inner Struggles",
9: "Standing Strong",
10: "Expressive Communication",
11: "Playful and Eclectic"

In order to verify that these topics were relatively independent, the LDA model was transformed using T-distributed Stochastic Neighbor Embedding and then plotted using a Bokeh Scatter Plot (Figure 20).

Figure 20: T-SNE Plot of 10 Topics

T-SNE is a method for disentangling nonlinear relationships and reducing their dimensionality. It is superior to PCA in that it can reduce 3+ dimensional expressions into 2-dimensional plots with relational clusters. As can be seen in Figure 10, only 10 topics survived the transformation from LDA to T-SNE and they are all relatively separate in their clusters.

Conclusion and Recommendation

This study was set out with the goal to find answers to the following four questions:

1. Are there any key indicators that would demonstrate to a producer or record company that a song has a greater chance at becoming a hit?
2. What are the relationships between the songs, specifically their lyrics, on the Greatest List? What songs are most closely related?
3. What are the most popular topics of songs in the list?
4. What song characteristics influence the popularity of a song?

Where there was originally hopes to provide answers to questions through exploratory data analysis, and different modeling, the results provided insights, but were lacking in evidence that could support full

conclusions for each of the questions. The first and fourth questions were addressed through the logistic regression model, the results showed that 'instrumentalness', 'liveness', and 'duration_ms' features are statistically significant predictors of song popularity. Specifically, 'instrumentalness' and 'liveness' have negative coefficients, suggesting that increases in these characteristics are associated with decreases in the likelihood of a song being popular.

Question 2 was answered by the semantic search portion of the study. There were relationships that were defined between each one of the songs using the embedded versions of the lyrics. The songs that were most closely related often came from similar genres of music, or similar time periods. With the network graphing there were hopes that there would be communities and groups of songs identified, but this wasn't the case with the lyrics of the songs. The most relatable artist throughout the list was The Beatles, and the most relatable years were 1966, 2004, 2013, 1967 and 1939. To gain a deeper understanding of the connections and even compare similarity results, different similarity metrics could be used.

Question three, What are the most popular topics of songs in the list, was addressed through topic modeling. This portion of the study confirmed that Topic Modeling may be a good tool for further development. It showed various topics such as Wishful Thinking, Raw Emotions and Emotional Resilience among the top topics across the Top 500 list. The purpose for future research would be to collect a genre of music, produce a model based on economically viable past hits and use the model to cluster the topics of potential new music. This could serve as an indicator of what marketing efforts would yield the best return on invested dollars and time.

References

The Editors of Encyclopaedia Britannica. (2023, June 17). *Rolling Stone | American magazine*. Encyclopedia Britannica. https://www.britannica.com/topic/Rolling-Stone

Stone, R. (2022, March 25). Rolling Stone. *Rolling Stone*. https://www.rollingstone.com/music/music-lists/best-albums-of-all-time-1062063/

Wikipedia contributors. (2023). Semantic search. *Wikipedia*. https://en.wikipedia.org/wiki/Semantic_search

For more on Jaccard Similarity see https://en.wikipedia.org/wiki/Jaccard_index