Music Words Slap*

Do Words Make Hits?

Cole Gaito†
Applied Comp Sci Student
University of Colorado Boulder
Littleton, Colorado, USA
cole.gaito@colorado.edu

Abstract

Can music that tops the Billboard Top 100 Charts be predicted through analysis and utilization of the words in-of-themselves current and previously songs on the chart?

The short answer is that the data is inconclusive, but indicates that there is a possible positive predication capability..

Using a locally implemented Recurrent Neural Network (RNN), I trained a model on text data (lyrics of songs from the Billboard chart) and incorporated numerical metadata (rank and weeks-on-board) from the dataset, along with normalized popularity data as the target. After running this model on approximately 18 million data points, the training process took around 8 days to complete. The RNN provided a prediction **range of +/-19%.**

Future work certainly exists and the initial value appears promising. As detailed in the report below there are also strong collations and interesting graphs that are provided as well.

Overall, I would state that the initial exploration yielded positive results and that the question is worth further exploration. The music industry certainly has a financial interest in developing a prediction model if possible as even a slight improvement in the highly competitive space could yield a large monetary gain.

In addition to the positive aspects of this question's posited line of inquiry, there are additional draw-backs consisting of the curse of dimensionality, the overall question of certain word importance, and the lack of the melodic component as expressed by the individual artist. These are also explored within the paper.

Introduction

When it comes to the business of Music, we see a method of communicating with one another. Whether it's the latest cringe Tic-Tok video theme or yet another one of Taylor Swift's break-up songs… music has the ability to pull people in and capture the audience's attention, heart, emotions, and ultimately wallet. Understanding this continuously growing 28.6 billion dollar[1] industry and what captures the attention of this consumer group is the focus of this research project. I am specifically interested in utilizing the Billboard Hot 100 list as the basis for measure. While this list reflects the interests of the American consumer, this group is often the one driving interest and trends worldwide. Most of major music stars arise from the United States, and the typically the US spends the most on music.[2] It also doesn't hurt that a majority of the record labels and technology platforms also reside in or were developed within the borders of the United States.

In this research project in addition to the Billboard Hot 100, would like to do analysis based on Spotify (given the company's pre-eminence within the streaming market and availability to API calls), and lyrical word analysis. Based of the categories available I seek to derive additional data points and see if there is a correlation between the songs as a whole within their genre, album, time and list on the billboard chart.

While ambiguous, I am also interested in seeing if there is a correlation that can be derived to world events and the Billboard's top 100 chart. While I personally highly double the availability of this to measure, I'm interested to see if the songs, by virtue of their words point to influential events in the world. Again, due to the highly subjective nature of lyric interpretation, this is unlikely.

In addition to an analysis of the words, based on feedback received, there is a section of this paper that explores the ability to utilize machine learning to understand if it is possible to predict with accuracy the length of time on the Billboard Hot 100's Charts list based on lyrical analysis.

As a point of acknowledgement, this project will not survey the rhythmic form of songs, nor its melody. Only the words of the songs themselves and their associated metadata will be analyzed.

Related Work

Looking around the research landscape, there are a couple of previous works that I have come across. In *Analysis of Billboard's Top 100 Songs and Lyrics (1964-2015)* by *Elaine Hsu & Hattie Xu* there is a similar analysis of the ranging and the word counts to study the decrease in one-word, two-word, and three-word phrase complexity. There is a generalized perspective that with each passing decade there is an increase in pessimism that grows out of the lyrics. Ultimately some of their hypothesis' held true – more profanity, less complexity, though their idea of a more pessimistic outlook was incorrect. See the reference for additional details.[3]

There is also a Machine Learning Algorithm that was created by GitHub user *yamnihcg*. Here data was used across Spotify and Billboard data pipelines to model if a song would be a future hit.[4]

A Medium author Josh Viner, also used data from Billboard Hot 100 and Spotify's labels identify artists, songs and popularity.[5]

Ultimately though, I could not find anyone who is specifically looking at the text to see if there is a correlation.

Data Set & Main Techniques Applied

After having hinted at the work proposed, I'll look at outlining what is necessary for understanding and working through the questions stated above. The following numerical values are based on what was theorized in my previous submission.

1. Data Cleaning **(Genuis API)**– Almost None

Having worked with the data so far, this statement remains mostly accurate. The data coming from highly used datasets provides me with little "cleaning". I have found though that there is roughly a ~2%, totaling to 850 items, with an issue between all the unique items. Whether lyrics are not found, names incorrectly identified, or with odd labels for the set there doesn't seem to be a large issue.

2. Data Incorporation **(Genuis API)**– Significant

I knew that this would be the main bulk, capturing the data. As I am pulling the lyrics from a website, I spent ~2 weeks with my machine constantly pinging Genuis' API for Lyrics data. This was also after I managed to get a script written to scrape the data, and then periodically save it into an excel file. The complexity is not necessarily the issue, rather the amount of time to ping, then compile the file was more then I anticipated.

3. Data Preprocessing **(Genuis API)**– Some

Here I recommended having to perform some data manipulate to extract and understand the lyrical text.

Overall, I expect that the interesting part is still to go. I have generally completed the pull of data from the Genuis API scraping the lyrical data from the internet. I want to incorporate additional data point from Spotify and the associated Metadata around the artist, album and song, this is work yet to go.

In addition to the data itself, deriving the most common word in a song or subset and other statical metrics whether by decade, year, month etc. is also outstanding work. I'm not overly concerned about this part though. There are a variety of tools that have and will simplify the process.

The original sources for the data can be located at Kaggle[5] and within the self-maintaining GitHub[7]. The dataset can be found in my public facing GitHub account in an excel file called gaitocole/

Final_Project/ Music_Trends/ *charts.xlsx*[8]. This format is accessible to Pandas, MatplotLib and other coding tools and provides an easy-to-read format that is universally available to both coders and non-coders alike.

Currently there is 3.4 million pieces of data, will additional data to be imported. There are roughly 31.5 thousand unique songs across the dataset starting from August 4th, 1958. As previously mentioned, importing the additional metadata and extracting the metrics is the work to go.

Concerning the data extraction **from Spotify** I have sought the following **metadata** categories for interest: *Track Popularity*; *Track Explicit*; *Album*; *Album Release*; *Artist Release*; *Artist Popularity*; *Artist Genre*; *Track ID*; and *Tracks in Album*. These categories are tethered to the Title and Artist that have been extracted from the Billboard Hot One Hundred data that has been collected. Due to the nature of pulling and process the ~31,000 unique songs this process has consumed much of the previous two weeks. In addition, I ran into the first issue of a web-hosted API having a limit. Luckily the developer site is rather flexible and allows the creation of multiple endpoints from which you can request the data from the Spotify Database Servers. In this way I've reversed the data and split it between two lists to process a bit more quickly.

At this point in time there are roughly 5.5 million pieces of data when aggregating in the 6 categories created through the Spotify API request. It still wildly blows my mind the sheer volume of data that is generated simply by adding in a few more details surrounding metadata.

Data cleaning for the process was a bit more extensive than I anticipated. The Spotify Library surprisingly lacked the total history of all songs dating back to 1958 when the Billboard Hot 100s were initially started. I find this rather surprising due to the extensive library that both Spotify and Apple maintain. There is definitely an exploratory rabbit hole that could be delved further into on that side of the house.

Overall, I did also notice that the Spotify Data Category of Artist Popularity is wildly subjective. We can see the overall limitations of the dataset being correlated with a time dependent activity. I did not necessarily have an expectation surrounding this topic but I did find it rather interesting that the popularity is dependent on the time of API call request and as such the ability to have general popularity is something that can wain over the weeks and years.

1. Data Cleaning (Spotify API)– Significant

While having described the data, previously, when manipulating and importing the Spotify Data, I noticed that there was significant holes in the Spotify Metadata dataset. While Spotify came into existence 2008, and a large portion of the popular back history data has been captured, the origin of this licensing and metadata has been captured from previous compilations not the original Artist's release record.[9] As such a significant amount of the metadata was skewed.

2. Data Incorporation **(Spotify API)**– Significant

The reason that manipulating the data here was difficult was not due to any complexity constraints, but the sheer volume of memory that was required. Due to the large number of additional metadata columns, there was a need for additional time for the software to manage the aggregation of the data.

3. Data Preprocessing **(Spotify API)**– Some

Again, some data preprocess was required to add the data together and adjust the datatypes but overall the bulk of the work needed for the data rested in aggregate the data and manipulating it to compile it all together.

Evaluation Methods

Proposed Method 1:

The evaluation methods will be the derivation of the words based in the lyrics, the correlation to the Billboard Hot 100 chart. I assume that this will be along the range of having similarity to the idea of a correlation co-efficient. I would also like to see the demonstration based off a word cloud. I often find myself to be a visual person, so having the visualization is a huge point for me personally.

Proposed Method 2:

Extracting the relevant data from the now captured lyrics, I plan on pulling the most common words bucketed within each of the songs while excluding the articles of speech (i.e. the, a, an, etc.). This will occur on a per-song basis. In addition, I would like to analyze to see if there is a general theme within either the longevity on the charts or particular brackets. Think of hit songs numbers one through ten, again eleven through twenty and so on. I find that this topic can also be interesting to understand considering the category and genre that the songs fall into. Do particular themes within particular genres do better than others?

In such a way the main words can be seen as a method for understanding how themes chart based on genre and thus how long that theme may last. A difficulty may be dealing with synonyms rather than exactly the same word. I don't think that other than the artist would typically use the same word over and over again. Hence bucketing the various hits into 10 brackets may turn out to be the most reasonable way of accomplishing this task.

Proposed Method 3:

In this approach being able to align the weeks and see the linear progression rather than a single flat file could allow the user or analyst to understand the progression over time. In this way genres and other metadata categories can be utilized to track the predicted popularity. The incorporation and work done for Method 1 could also be useful as a means of folding in additional datapoints to make the algorithm more accurate.

From a personal perspective this will certainly be more challenging as creating the data cubes discussed in class will become necessary. However, challenging this could certainly make the end result more fascinating and provide a greater experience trying to digest and discern what is occurring within the data itself.

Potential Pitfalls

With the methods listed above there are several significant assumptions that may prove to be costly in the pursuit of trying to capture and predict the longevity of a song on a chart and ability to predict future hits. I have taken the time to break out the issues into the categories below:

Insufficient Word Diversity

This issue may arise from a lack of repeated words in both the songs in-of-themselves and between the songs as they are placed into their respective casts. It would be interesting to see if there exists an application which can identify words and synonyms but I find this unlikely. Even if such an application or API exists into a database trying to find all the synonyms that exist for each word in the song would be feasibly impractical.

Arrangement of Data

I see there arising the inability for pandas, the current data manager to handle the complexity of the dataset that I'm trying to present to it. While this issue certainly is not as severe, due to the general time constraints I can foresee the becoming an unrulily question that I simply do not have the skillset to manage. As a simple solution, I want to explore working with TensorFlow or some other data management tools. I am sure that this problem is not novel and as such finding an appropriate tool may be the most difficult part of this task.

Lack of Correlation

While this is not necessarily a problem, it would be unfortunate to find that the data lacks a correlation between any of the categories. What could be worse is the problem of having strange items identified as the driving cause of a song's popularity, like the spelling of an Artist's name or something akin to that issue. Again, while it is not necessarily an outright issues, it could become a headache if there is nothing interesting found.

Complexity between Repeated Items

Here another issue that may end up simply requiring the analysis to be performed twice is the situation of having two similar but different datasets. The dataset which lists the weeks on the charts and thus has the single long list of every time a song appeared on the Billboard Hot 100 and the Unique Song List Data Set that I have created. The later was an ends to speeding up the API requests, but may end up being useful both in the need to have a smaller data pool, and the need to expedite learning. From my previous courses learning how to effectively manage the resources to train AI is necessary. While not an explicit issue, I wonder if analysis on both sets would yield the same results. Theoretically it should if I have structured my analysis correctly, but if not I could merely be opening the door to future questions about the work done here.

Tools

The current list of tools are as follows:

GitHub – Data Configuration Management

Git-LFS – Open-Source Large Dataset Git Extension

MS Excel – Basic Data Storage File Handler

Python – programming language v3.13.0

VScode – Integrated Development Environment

Genuis API – Website with API call interface for Lyrics gathering

Spotipy API – Spotify website call interface

Various Python Libraries

Jupyter Notebook – Data Science Code Workbook

Tensor Flow – Initial Machine Learning Library

Pandas – Data Framework for Data Manipulation

MatplotLib – Graphing Tool

NLTK – Natural Language Toolkit

String – Pandas Dataframe Library Extension

Key Results

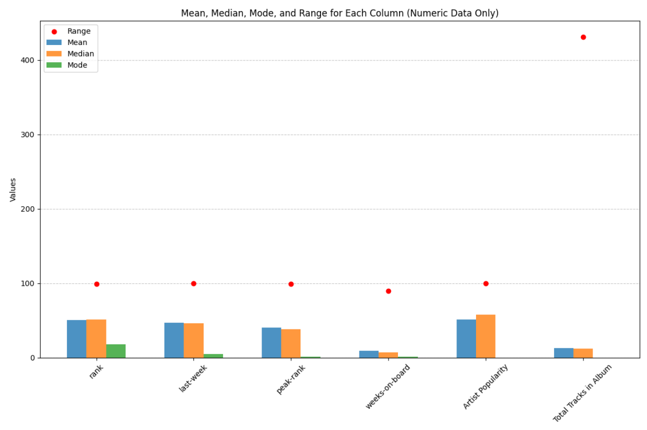
In the following section I will outline the various methods and results that were utilized, not just proposed, along with chart descriptions that were created to evaluate the data.

Methods Implemented:

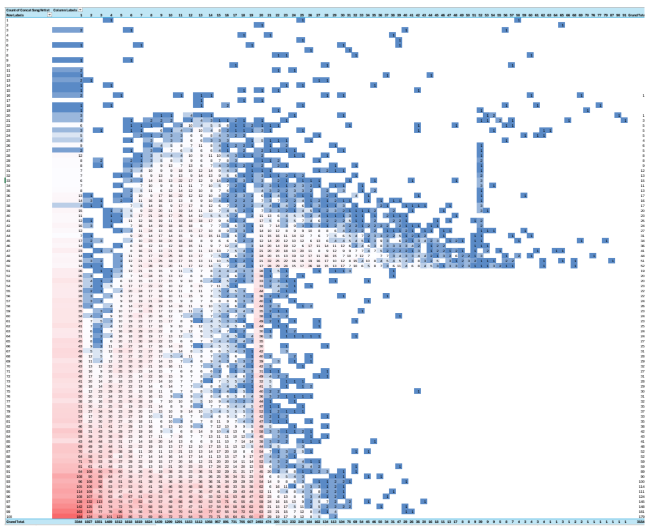
Method 1

As described above, I was able to take the data and create and implement comparison charts casting a given variable against another for the various data sets. Working within excel and the pandas data frames, I was able to export and create a variety of descriptive charts that help to gauge the value of the data, and what various attributes therein. I'm going to take a moment to walk through and highlight the values that I feel add significant insight.

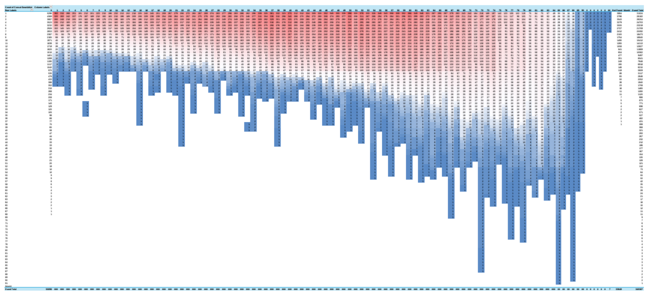
In this first chart we have the mean, median, mode, and range of the various billboard hot 100 chart values. You can see the rank, weeks on board, popularity and total tracks on the album. While the data in of itself may not feel to be very descriptive, I note that this chart is quite interesting due to the range feature for the number of albums on a track. The interest on this value being significantly above the rest is due in part to the pointing of the use of conglomerate hits from earlier decades. This piece of data is pulled from Spotify and was surprising to me due to the fact that Spotify aggregates and licenses only the hits from earlier decades. I'm sure that is due to licensing restrictions and the desire to save money on the part of Spotify. The unfortunate issue with this though is the fact that you end up skewing the data since the original artist's metadata from the album which a song was released on is captured. As such we have already entered a risk for our AI/ML program.



These next two charts take the unique songs and the entire Billboard Top 100 Charts weeks list aggregate and plot the data.

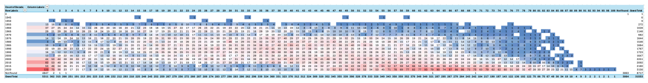


This chart graphs the rank of the song based on popularity and weeks on the chart. So it captures where the song ended the week prior to falling off the top 100 chart. I created a heat map to make it easier to view on the eyes. Blue is low values (typically 1, while red represents higher values. The while portion between is the middle ark transition). The x-axis shows the number of weeks on the billboard top 100 list, while the y-axis show the overall popularity rank during that last week.



This second graph again plots the popularity of the songs verse the weeks on the board, but here instead of only capturing the final week, every single value is plotted even if it is duplicative. The popularity is on the x-axis while weeks on the board is along the y-axis. These were sorted by rank as well. The same heat map scaling has been applied too.

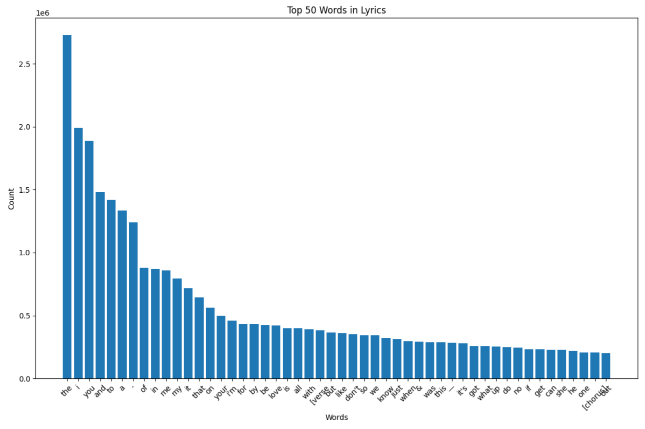
The difference in these resulting charts has been fascinating to mee looking through what to do and how to do it. Typically you can see that a higher percentage of songs only stay on the charts for less then 8 or so weeks, and tend to score lower. If a hit is a hit then it typically does stay on the chart longer, but this is a more few and far between occurrence.



While the chart above is more difficult to see, I've ranked the number of songs and their placement by decade. I thought this chart was interesting since it conveys the overall movement throughout the decades. You can begin to see that the either the songs debut at the top of the charts or closer to the bottom half. This movement and shift indicates that to break into the top portion of the billboard hot 100's chart initial placement and rank does matter.

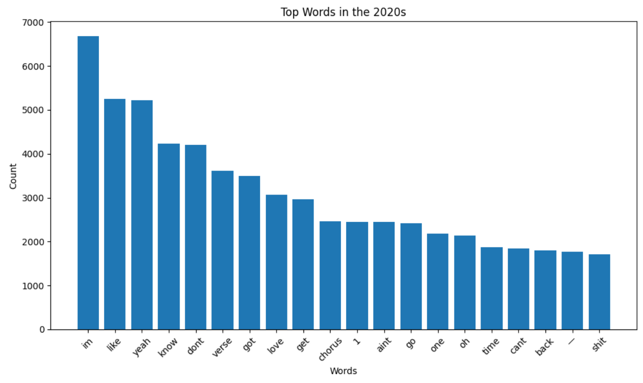
A question you may ask yourself, is why does the analysis of these factors matter when the original question posited concerns the lyrics and words therein? I think a clear and level headed understanding of the data and how it is aggregated is important since it gives context to the overall question being asking and possible ways and means by which the factors of a predication algorithm's confidence calculation is determined. Phrased another way, if you lack the understanding to be able to explain the data, then you certainly will not understand the resulting answer you get from the recursive neural network. On top of this you will also lack the ability to tune the model.

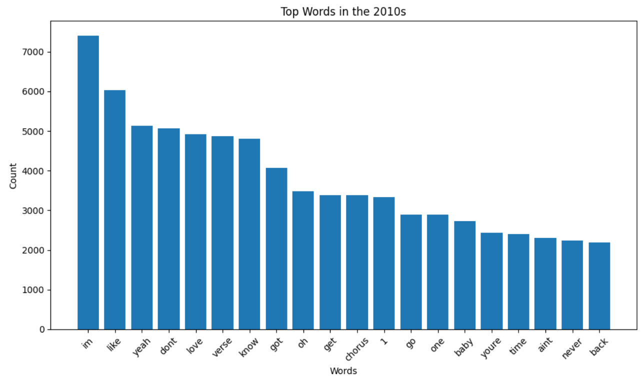
With that in mind, I would like to turn to the lyrics themselves. Here is a basic bar chart of the words and their counts of the 50 most common throughout all the lyrics. Scaled in the millions, this chart is important since it shows both articles as well as important words. You can see and gauge the importance of the words since none of the words have been omitted.

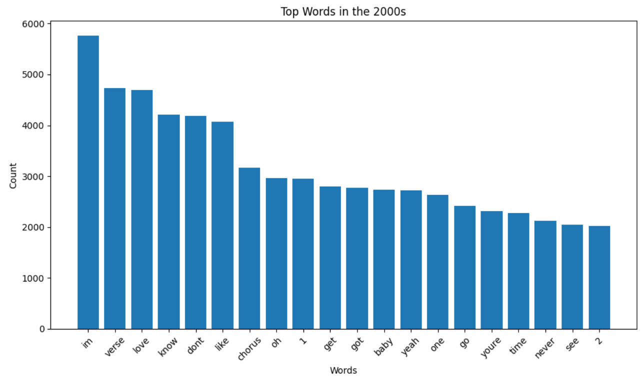


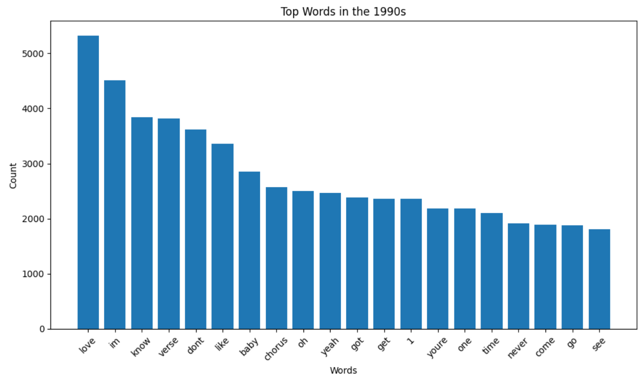
The next series of graphs show the word breakout based on the decades. The rationale for providing this information, although limited to only the top 20 words does provide the reader additional insight into any words that may not be represented or captured in the aggregate collections. We can again validate that there isn't a missing piece to the data and gauge the overall accuracy of the data we are trying to evaluate.

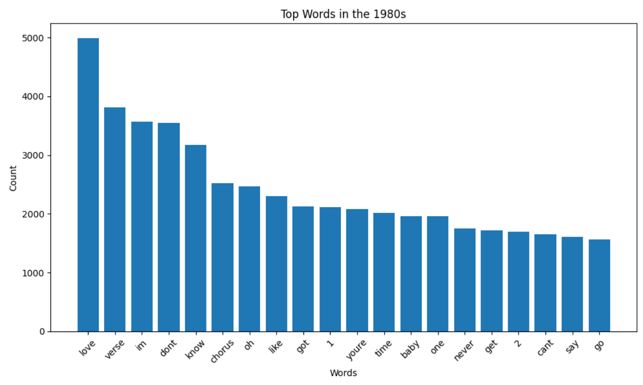
In addition to these charts, I would also like to point out that I used the Natural Language Toolkit and utilized the "Stop Words" library to remove the various non-descript words and articles as well as removing the punctuation. This permits you to better understand and see past some of the more meaningless or supporting words that are used primarily as a sentence structure function as opposed to indicate meaning such as love.

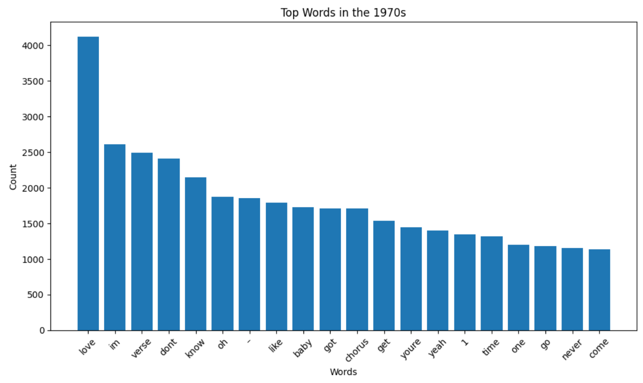


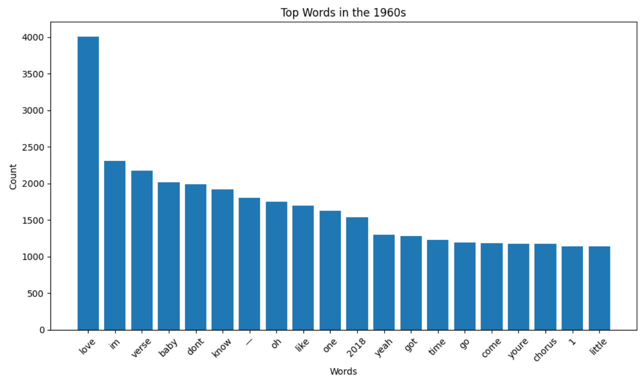


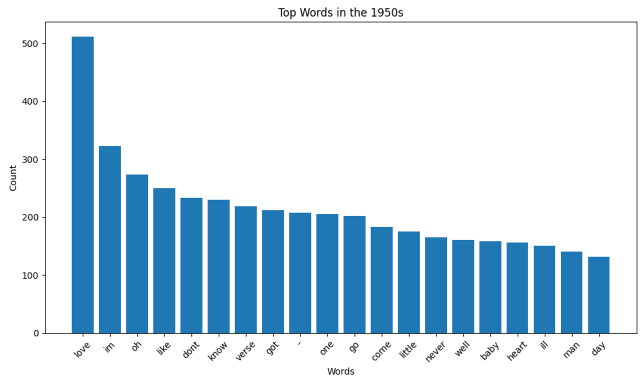


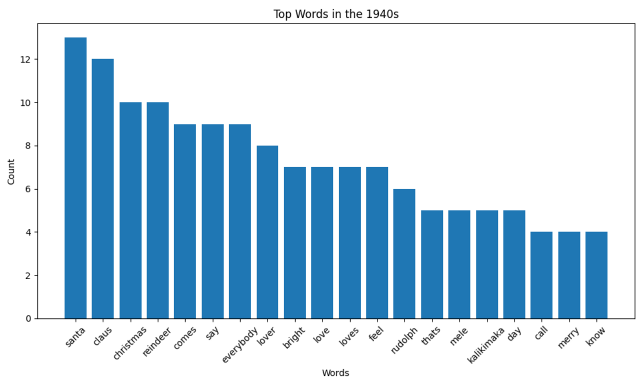


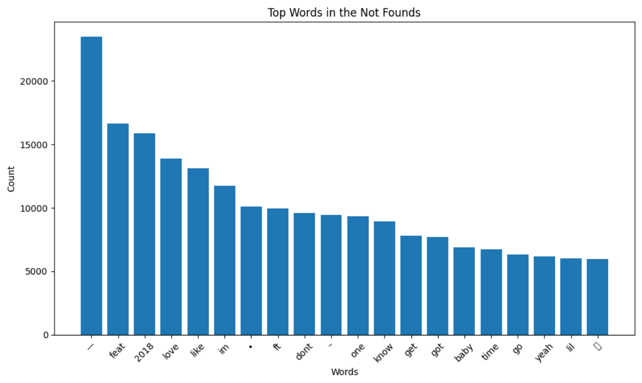












Method 2

Having worked through the various charts that I created throughout the course of the project, I'd like to turn our attention the recursive neural network that I created to evaluate the songs based on the lyrics and gauge the probability of future songs based on their lyrics.

I will say from the outset that I was only able to successfully complete a single RNN cycle of 5 epics. I understand that the dataset was quite large, but I did not realize that it would take me 8 days to run the program for its initial training.

Basic Steps of the Prediction RNN

Key Steps and Functionality

1. Data Preprocessing

Text Data (Lyrics): The lyrics are tokenized using Tokenizer, converting words into numerical sequences. The sequences are padded to a fixed length (max_sequence_len) to ensure uniform input size for the model. The vocabulary size is computed based on the tokenizer's word index.

Numerical Data (Rank & Weeks-on-board): The numerical features (rank and weeks-on-board) are standardized using StandardScaler for consistent scaling. **Target Variable (Track Popularity):** Track popularity is normalized to a range of 0 to 1 for regression.

2. Model Definition

The model combines two types of inputs: **Text Input (Lyrics):** An embedding layer converts each word in the lyrics to a dense vector of 128 dimensions. Two LSTM layers are used to capture sequential dependencies in the lyrics. A dropout layer prevents overfitting by randomly deactivating some neurons during training. **Numerical Input (Rank & Weeks-on-board):** A dense layer with 32 neurons processes the numerical features. A dropout layer is applied to regularize this branch. **Combining Features:** The outputs of the text and numerical branches are concatenated. A dense layer processes the combined features, and the final dense layer outputs a single value (normalized popularity).

3. Model Compilation

Optimizer: Adam is used for efficient training. **Loss Function:** Mean Squared Error (MSE) is chosen for this regression task. **Metrics:** Mean Absolute Error (MAE) is tracked to measure prediction accuracy.

4. Training the Model

The model is trained for 5 epochs with a batch size of 32. The training data is split into 80% training and 20% validation subsets to monitor overfitting or underfitting.

5. Evaluation

After training, the model is evaluated on the test set, and the test Mean Absolute Error (MAE) is printed.

6. Saving and Predicting

The trained model is saved in HDF5 format. A function, predict_lyrics_with_features, allows making predictions for new inputs: It processes new lyrics and numerical features through the same pipeline. It outputs the predicted normalized popularity.

Resulting Insights

The model achieved a **MAE of ~0.19**, indicating the average prediction error is about 19% of the normalized popularity value. The example prediction for the input lyrics and numerical features resulted in a normalized popularity of approximately 0.37 (or 37% in unnormalized scale).

Applications of Knowledge Learned

Ultimately, what I found is that the program was able to predict with some accuracy, but it was not enough to be of high value or convincing metrics. In this way I can say that for a first pass the model was able to produce and prove that there is the possibility for expansion on the idea of projecting the Billboard Top 100 Charts list based on lyrics. A better understanding and greater accuracy may result from a finer tuning of the variables used in this model, or even the changing the variables to incorporate the ability to evaluate the rhythms and notes of the music as well.

Further Research:

My suggestion for further research would rest in the following items:

1. Rerunning of the current model to see if the current prediction range holds.
2. Further Data cleaning around the songs utilized in the Spotify Metadata that was pulled.
3. Adding in additional metadata variables and recreating the model.
4. Determining if rhythm and notes can be incorporated into song evaluation.

I have listed the suggestions above in order of what I thing is the most reasonable. If given more time, I would recommend re-running the model to determine the validity of the results, along with further cleaning the metadata. This would require greater data intervention and line by line analysis rather than the API calls that were utilized.

Beyond these simpler research propositions, you can see if additional metadata variable can be added in, and if the translation of the rhythms and notes can be added in as an additional variable to be added into the model. Overall though, I would recommend and would personally perform these recommendations again.

REFERENCES

[1] **Statista.** 2023. Global revenue of the music industry from 2002 to 2023. *Statista.* Available: https://www.statista.com/statistics/272305/global-revenue-of-the-music-industry/#:~:text=In%202023%2C%20the%20total%20revenue,compared%20to%20the%20previous%20year.

[2] **Wikipedia.** 2023. List of largest recorded music markets. *Wikipedia, The Free Encyclopedia*. Available: https://en.wikipedia.org/wiki/List_of_largest_recorded_music_markets.

[3] **Brown University Department of Computer Science.** 2011. CS100 Student Projects - Project 11. *Brown University*. Available: https://cs.brown.edu/courses/cs100/students/project11/.

[4] **Yamnihcg.** 2023.Billboard100. *GitHub Repository*. Available: https://github.com/yamnihcg/Billboard100.

[5] **J. D. Viner.** 2023. What makes a hit song? Analyzing data from the Billboard Hot 100 chart. *Medium*. Available: https://joshdviner.medium.com/what-makes-a-hit-song-analyzing-data-from-the-billboard-hot-100-chart-74c1d5ad3fa3.

[6] **D. Dave.** 2023. Billboard - The Hot 100 Songs. *Kaggle Datasets*. Available: https://www.kaggle.com/datasets/dhruvildave/billboard-the-hot-100-songs.

[7] **UT Data.** 2023. rwd-billboard-data. *GitHub Repository*. Available: https://github.com/utdata/rwd-billboard-data.

[8] **C. Gaito.** 2023. Music Trends - Final Project. *GitHub Repository*. Available: https://github.com/gaitocole/Final_Project/tree/main/Music_Trends.

[9] Wikipedia contributors. 2024. Spotify. In *Wikipedia*. Wikimedia Foundation. Retrieved December 7, 2024, Available: https://en.wikipedia.org/wiki/Spotify.

[10] **Spotify.** 2024. *Spotify for Developers*. Retrieved October 27, 2024 Available: https://developer.spotify.com/

[11] **C. Gaito.** 2023. Music Trends - Final Project. *GitHub Repository*. Available: https://github.com/gaitocole/Final_Project/tree/main/Music_Trends