Decoding the rhythms of emotion: a sentimental journey through music genres

Project Report

Text Mining

NOVA Information Management School

Bachelor's Degree in Data Science

2023/2024

Afonso Cadete | 20211519

Bruna Faria | 20211529

Inês Vieira | 20211589

Rita Centeno | 20211579

Table of Contents

Table of Figures

Abstract

This project aims to explore the intersection of music, natural language processing and machine learning to develop a predictive model to predict a song's genre based on its lyrics. Additionally, the project also extends its scope to include a sentiment analysis of songs' lyrics, exploring the emotional nuances associated with each genre.

The main goal is to leverage advanced techniques in the area of text analysis to accurately categorize songs into their predefined genres. To achieve this goal, the song's lyrics and tag were extracted from a dataset with more than thirteen thousand observations which also contained the songs' respective title, artists, featuring artists, release date, and number of views and used to train a machine learning model. This model's performance was evaluated using a hold-out method and achieved a f1 score of 0.71 and 0.66 for the train and validation sets, respectively.

The outcomes show that it is possible to predict the genre of a song based on its lyrics with high performance and provide interesting insights into the intricate relationship between language, emotions, and musical genres.

Introduction

In a day and age where data science and the availability of artistic content have become more prominent in day-to-day life, the convergence of natural language processing (NLP) and machine learning (ML) has come to offer a powerful tool for examining the complex interaction between lyrical content and musical genres. The motivation behind this project lies in the positive impact that, if proven successful, automatically classifying songs by genre will have in different areas, such as improving music recommendation systems or playlist generation.

The main goal of this project was to develop a machine learning model that could accurately predict the genre of a song based on its lyrics and explore the relationship between the sentiment of a song's lyrics and its genre. While there has been previous research on the topic of music genre classification, much of this research focused on audio features rather than lyrics. The few studies found that explored the use of lyrics for genre classification, generally used much smaller datasets than the one used here.

Nevertheless, based on the research found, an accurate prediction of a song's genre by its lyrics was expected. Additionally, insights into which strategies to adopt were also gathered. Among these insights are the irrelevance or even prejudicial impact of removing stopwords [2][1] and the possibility of having to face dimensionality problems [1][4].

In the second phase of this project, the focus changes to examine the sentiment behind each of the tested genres - pop, rap, rock, r&b, country, and miscellaneous (which are texts, such as poems or excerpts from the Bible, and not songs). For this part of the project, there were expectations of finding significant relationships between genres and sentiment [3].

Data Exploration & Preprocessing

Data description

The dataset in which this project is based regards information on a list of a hundred and thirty-four thousand nine hundred and sixty seven songs. The data included the song's title, main and featuring artist(s), year of release, lyrics and genre (the target).

Data cleaning and preprocessing

As an initial phase of the data exploration and preprocessing phase, it was important to understand the data in question. To do so, the data types of each individual variable were checked for. In this process it was found that there were missing values in one of the features – the title columns – which were then imputed with the string "Unknwn". It is important to note that this will not affect the model since information from this feature ended up not being included in the model.

With this understanding that the dataset might have some issues, duplicate rows were also checked for. Even though, looking at the whole set of columns no duplicates were found, that was not the case when testing for the lyrics column by itself, where 178 duplicated rows were found (13 of these songs changed tags). Among these observations, there were 85 where the year was also duplicated, 1 regarding the number of views, 24 for the title, and 111 for the artist. The observations where lyrics and title are duplicated probably represent covers of a song by another artist, while observations where lyrics and artist are duplicated likely represent different versions of the music from the same artist (acoustic and extended versions, remixes, among others). Having said this, these insights were kept in mind and these duplicates were later dropped, after the preprocessing phase, as there could be some slight variations, such as differences in punctuation, or misspellings, that would cause some duplicates to not be found. Other combinations of features were tested for duplicates on the training set, however, no more results were found. As a last step, overlapping between training and test sets was also checked to see if there were songs present in both these sets, risking biasedness in the model. This last part was checked based on the lyrics column alone and 72 observations were found, and these observations were removed from the training set.

Categorical variables

The general distributions for each column were checked and some of them plotted. A first and important insight taken from the plotting of the target variable distribution – tag – (Figure 1), was that the dataset is imbalanced having a high prevalence of pop, rap and rock (from descending order of prevalence) and having somewhat few observations of the remaining genres – r&b, miscellaneous and country (also in descending order).

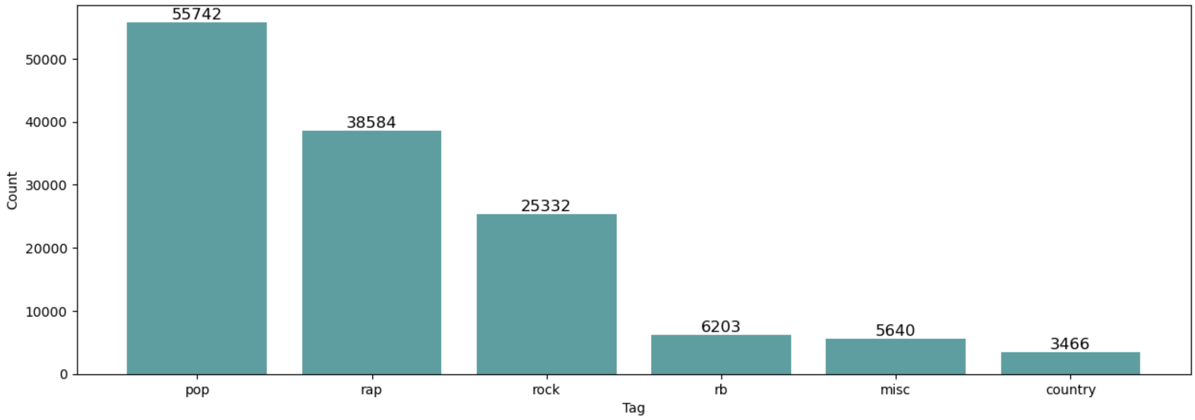

Figure 1 - Target variable distribution

Plots were also made for the song's titles, the artists' names, and the features artists' names. From the first (Figure 5), it was clear that there were certain song names that were plausibly common, such as 'Home', 'You', 'Stay' or 'Love'. However, there was one that stood out, 'Intro', which had almost double the occurrences of the second highest frequency one. This is explained by the common use of this name for the initial introduction of several albums. Having this said and adding the fact that more than half (approximately 54%) of song's lyrics already include the title, this variable was not treated nor included in the model, as it would vastly increase the dimensionality of the data and intensify the computational problems already faced.

From the second graph (Figure 6), the presence of some famous song artists is noticeable – Frank Sinatra and Elvis Presley – but it is also visible that there are ones that most probably do not relate to the music industry and likely belong to the miscellaneous category – Abraham Lincoln, Holly Bible, or William Shakespear. The artist's name that mostly popped out was Genius English Translations, which is a website to access song lyrics, therefore misrepresenting the artist's real name. This along with the fact that there were almost 75 thousand different artists led to this feature being discarded from being treated and included in the model.

From the third plot (Figure 7), it is clear that the vast majority of songs do not have any featuring artists – the case for almost 112 thousand songs – which is to be expected since most of the released songs have only one artist or band. Moreover, it is also noticeable that many of these artists' names are incorrectly written, having unnecessary slashes and signs in the middle of the strings, leading to this variable also being discarded.

Faced with the fact that the highest occurring artist was not an actual person, but a lyrics database website, this column was checked to see how many of the main and featuring artists' names included the word "Genius". From this analysis, it was possible to conclude that there were many misrepresented artists from lyrics websites from several different countries. This only reinforced the fact that dropping these variables was a good practice.

Numerical features

Focusing on the year's feature, some inconsistencies were found. First of all, there was a song dated in a year that has not yet happened (2024). Then, after applying the logarithm to the graph, it was visible that the range of this variable went until year 1. Having this said, some of the observations of these earlier years were observed, and, even though some of them were plausibly dated – as the ones from the Holly Bible – there were also observations where this feature was misrepresenting the truth – the song "Get ready" by John Campbell Munro, in the dataset listed under year 1, in reality was released in 2019. This can be something useful to be considered in the sentiment analysis part of the project, as insights will be taken that need the consideration of this feature.

As for the views' variable, nothing uncommon was found, however, it was possible to conclude that the vast majority of the dataset's observations belonged to the lower spectrum of the number of views – some even having 0 views – however, observations on the opposite side of the spectrum also exist (popular songs) – some even reaching values above three and a half million views.

Lyrics preprocessing

Regarding the preprocessing of the data phase of the project, several changes were made to the lyrics' column. Firstly, conversion to lowercase and expansion of contractions was performed in order for the following preprocessing steps to be applied consistently. Among these steps are the removal of non-alphanumeric characters, URLs, emails, social media tags and overall noise from the data. Additionally, tabs, newlines, and sets of spaces were replaced with a single whitespace. Afterwards, two tokenization processes can be applied, depending on the desired outcome, one which includes emojis and one which does not. With the words already tokenized lemmatization was applied along with the optional step of removing stopwords. In the end, the lyrics are converted back to strings.

In the context of the modeling stages, in the tokenization mentioned above, emojis were removed, contrarily to what happened with the stopwords (the reasoning behind this decision is explained below). In what regards the sentiment analysis phase, both options were used.

Concerning the lemmatization process, two alternatives were tested. Firstly, an approach where POS tagging was used to attribute a label to every word and therefore make lemmatization more efficient was tested. This experiment was, however, unsuccessful since it was found that certain words were not being changed accordingly – verbs from the third person of the plural were being kept unchanged (loves, remembers, …). Having this said, a more naïve methodology for lemmatization was opted for. In this case, a loop was used to go through every POS tag for each of the words and make the desired changes if the word belonged to that tag. A possible drawback of this method would be if there were to be words that could be classified into more than one tag and end up having their content changed twice, risking an incorrect lemmatization. Nevertheless, based on further testing this last approach was the one implemented.

Log-ratio analysis

Then, having all the initial preprocessing done, a log ratio analysis was performed. The goal of this analysis was to select the words with better identification capabilities for each genre. To do so, the word frequencies were calculated for each genre individually and in total. For each genre, the number of words selected was proportional to the dataset's percentage of observations regarding that genre (if a genre represented 40% of the dataset, it had the right to choose 40% of the words deemed important). Since different genres could end up selecting the same words, it was ensured that if a genre was to select an already selected word, it would instead select the next best word. The log ratio values were calculated using the logarithm of following calculations:

$$\frac{\frac{word\ frequency\ in\ genre}{total\ number\ of\ genre\ words}}{\frac{word\ frequency\ overall}{total\ number\ of\ words}}$$

Using this approach, it was possible to address the computational problems faced when performing vectorization in the original data, as the dataset is reduced to a more manageable size. It is also important to note that during preprocessing stopwords were not removed as the log ratio intends to select useful words for genre detection and there might be useful words in this set. If not, these words will not be selected. Nevertheless, tests where stopwords were excluded were also performed.

To decide the total number of words to be selected by the log-ratio analysis a grid search was performed to consider every possibility between including 1000 and 5000 in steps of 100 words and apply this to a simple logistic regression model. The outcome showed that the inclusion of 3000 words would lead to optimal results. The ordering of which were the first genres to select words was also defined. The decision was made that the selection phase would be in descending order from genre prevalence, as it would allow for the least represented genres to possibly keep more of its more important words. Results that support this decision can be seen in Figure 10.

Later, word clouds were created to visualize the most relevant words (Figure 8). Additionally, these words were also plotted in the context of each genre separately (Figure 9).

Vectorization

To convert the dataset into a format that could be fed into machine learning models, the data – in this case, the lyrics column – needed to be vectorized. To do so, several vectorization techniques were used. One-hot encoding was used as a simpler vectorization method, in which word frequencies are not accounted for, only focusing on the presence or absence of the word in the text. A normal bag of words was used as a slightly more detailed version of the previous method, in which frequencies are accounted for. Lastly, TF-IDF (Term-Frequency – Inverse Document Frequency) was also used as a technique that, not only gives importance to the high presence of a word in a document, but also to its rarity in regards to the entire corpus.

Genre Identification

In the modelling phase, for genre classification, there were several approaches experimented. As a first step several models were tested with their default values. From this initial analysis two models were discarded from further analysis as they required a high dimensionality and presented poor results. These excluded models were Decision Trees and K Nearest Neighbours Classifier. The models that were deemed fit for further testing were Logistic Regression and Multinomial Naïve Bayes, both from *sklearn*, XGBoost Classifier from the *xgboost* library, and Neural Networks, from the *keras* library.

Initially, cross-validation was used to guarantee the generalizability of the obtained results. However, this approach was wasting computational power unnecessarily as the results showed consistent patterns independently of the split made. This is likely due to the vast quantity of data from which our dataset is constituted, which makes it unnecessary to create several separate tests to ensure consistency among the results. Therefore, cross-validation was excluded, and the remaining analysis was made using a simple hold-out method.

During the model testing process described above, the data resultant from the three vectorization techniques from the preprocessing phase were used – one-hot-encoding, bag of words, and tf-idf. The combination of these vectorization techniques with the selected models showed better results in the one-hot encoded data, therefore this was the vectorization type on which the remaining analysis was based.

As a next step, a grid search (for logistic regression, naïve bayes and xgboost) and hyperband tuning (for the *keras* neural network) were performed to find the best parameters for the selected models. The models were trained with their respective selected parameters on the one-hot encoded data and results were observed. After this analysis, some slight alterations were made in the models' parameters to further test possibilities and find combinations that led to higher model performances.

As a model elimination method, logistic regression was considered as a base model for inter-model comparisons. Logistic regression was the model chosen for this task as it showed consistently good results during the testing process. Therefore, models that did not perform as good as or above the logistic regression's performance threshold were excluded from further analysis. Having said this, in the next phase of testing, naive bayes, and xgboost were excluded from the analysis. The first one, mainly because of its poor results compared with the remaining models, and the second one because it had the tendency to overfit (Figure 2).

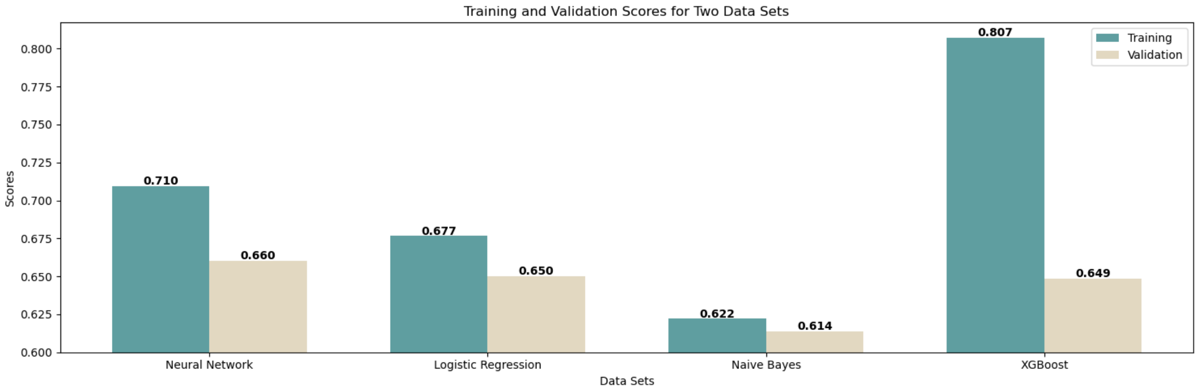

Figure 2 - model performances comparison

In the end, the best model found was a neural network from the *keras* library, with an f1 score of 0,71 and 0.66 in the training and validation sets, respectively. This neural network was constituted of three layers, one input layer (with the number of nodes equal to the number of observations), one hidden layer (with 100 neurons, a relu activation function, and a ridge regression regularization technique), and one output layer (with 6 neurons, corresponding to the number of genres). The model was evaluated using the sparse categorical crossentropy loss function, the adam optimizer., and based on accuracy (keras does not have an f1 score evaluation option).

To further analyze the chosen model, confusion matrixes were plotted (Figure 3). From their examination, it was clear that where the model failed the most was in distinguishing between the pop and rock genres (labeled as 2 and 5, respectively). It was also noticeable some misclassifications in the pop category which were predicted as belonging to the rap genre. Overall, errors happened in what regarded the model predicting songs as pop. However, this is most likely explained based on the significantly higher prevalence of this genre in the dataset.

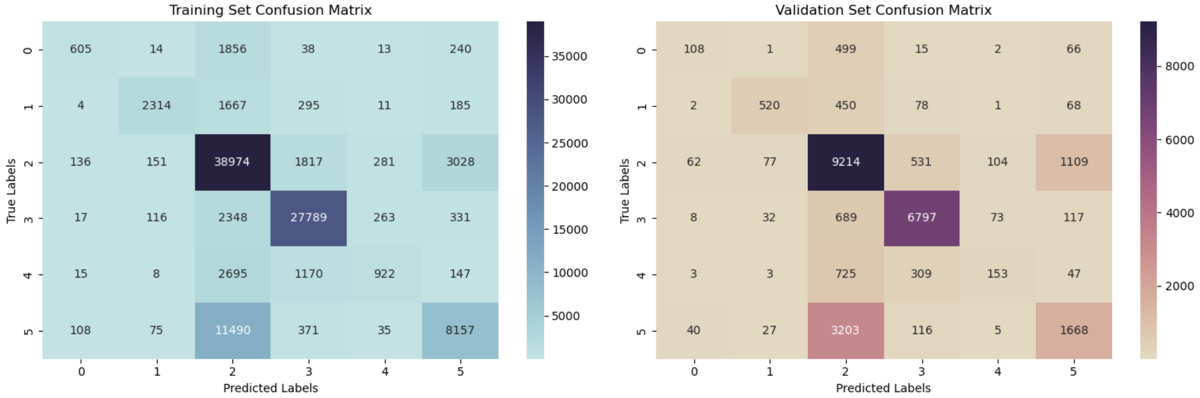

Figure 3 - best model's confusion matrix

Sentiment Analysis

With the completion of the modeling phase, the next and final part of the project involved conducting sentiment analysis. In this part of the project there were three different python libraries that can be used for sentiment analysis were tested, Vader, TextBlob and NRCLex[5].

Before the preprocessing, the polarity of the stopwords was checked to understand if it would make sense to keep them when performing the sentiment analysis or not. To do so, Vader library was used. After checking, the polarity of the stopwords was 0.0015, in light of this, it was decided to remove them. Considering this neutrality and that, in the previous steps of this project, it was discovered that the stopwords can already be a part of the most common and relevant words, it was decided that this added dimensionality would not be fruitful for the subsequent analysis. The preprocessing stage for this phase was done as described previously, having two types of preprocessing applied, one that kept the emojis in the lyrics and the other that deleted them. This was done to later check if the inclusion of the emojis in the analysis would impact the results.

Three different sets of words were tested to check the polarity of the different genres, the set of common words – words that most frequently appear in the collection of lyrics among all genres – the set of relevant words – set of words that best classifies genres – and finally the set of all the words – the entirety of the words in the dataset.

The first algorithm used to analyze the polarity of music was *Vader*. This algorithm has an output that contains four different values – negative, neutral, positive and compound score. The first three values measure the strength of the respective sentiment in the text. Nevertheless, it was decided that the compound score would be the only value used since this value is calculated using a specific formula that takes into account the relative intensity and balance of positive and negative sentiments. This value ranges from -1 to 1, in which closer to -1 is negative, closer to 0 is neutral and closer to 1 is positive. A threshold was defined to divide the result of the compound score into positive, negative, and neutral. If the compound value was between [-1, -0.05[the overall sentiment was considered negative, between [-0.05, 0.05] was considered neutral and between]0.05, 1] was considered positive. [6]

For a first analysis, the dataset in which the emojis were not removed was used. When comparing results from the three different sets of words the majority of the genres' polarity did not change – pop, country, miscellaneous and r&b were always classified as positive music genres. In regard to the rock genre, it was only considered a positive music genre when the common set of words was used; in the other two sets, it was considered a neutral genre. In general terms, the relevant words tended to have more negative results while the more common words tended to be more positive. When using the set of all words it was more balanced, always being in between the polarities of the two previously mentioned sets (Figure 4).

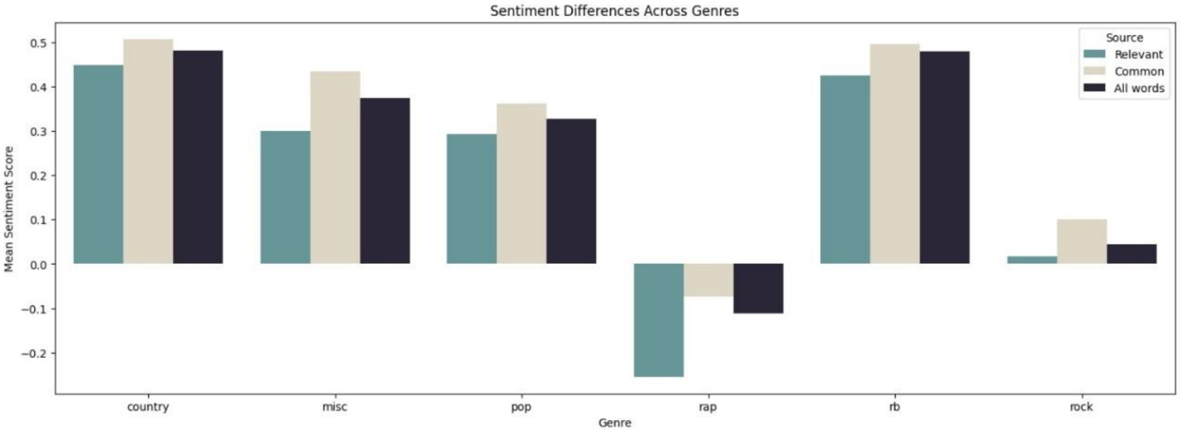

Figure 4 - Sentiment Difference Across Genres

After checking the results of the three different sets of words with and without emojis, the results seemed to be exactly the same. As to confirm this theory, a graph was made using the set of all words to make sure that were no emojis discarded from this analysis. It was important that this set was the one used, as emojis could be excluded as uncommon or non-relevant words in the other sets. As expected, the graphic showed us that the emojis did not have any impact in the analysis of the polarity of the lyrics (Figure 11). With that knowledge, and the fact that *Vader* performs better than the *TextBlob* with emojis, it was decided that they were not important for the sentiment analysis, so they were discarded and not used for the rest of the analysis and exploration.

The next algorithm that was tested was *TextBlob*. A similar approach to the one with the *Vader* algorithm was used – the three different sets of words were compared to compare results. In this case the results showed that all the genres were positive except for rap – represented as neutral – this result was the same across all the sets of words. As it was expected the results with *TextBlob* are less discriminative than the results with *Vader* and the compound scores are closer to each other when using *TextBlob* than when using *Vader*. This most likely happens because *TextBlob* performs best in formal texts like papers and books, while *Vader* is better for non-formal text.

After analyzing these results, it was decided that the rest of the sentiment analysis would only be done with the relevant set of words. Seeing as the *TextBlob* algorithm did not perform well, its results were not used to make the decision about what set of words to use in the remaining analysis. The decision to focus on the relevant set of words in the remaining analysis was made because, with *Vader*, it was observed that, for the common set of words, the rock genre was considered positive. However, knowing the genre, neutral would be a more accurate description of its polarity. Therefore, this set of words was discarded as a potential set. The relevant set of words was chosen over the set of all words due to the computational power required by this last one, so, since the results were the same for both sets, except for the fact that the relevant set had significantly less words than the complete one, and since this set would allow for this exploration to be more coherent with the modeling part, the relevant words set was used from this point on.

The third and final algorithm used was the *NRCLex*. The inclusion of this algorithm into our analysis was made as to expand the exploration from pure polarity into emotion identification. This algorithm works by using the *NRC affect lexicon* [7] and the *NLTK* library's *WordNet* synonym sets. The *NRC affect lexicon* is a list of English words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). This lexicon is widely used in research for emotion analysis and sentiment analysis. The *NRCLex* library uses this lexicon to measure the emotional affect from a text. This works by inputting a text to NRCLex, which is then tokenized into words and sentences. After this, it checks each word of the desired text against the *NRC affect lexicon*. If a word from the input text is found in the lexicon, it is tagged with the corresponding emotions, which can be more than one. For example, love would be tagged as positive and as joy.

The final output after using this algorithm is the set of all genres in which all emotions with the respective score were displayed. Each genre has several songs associated with it, and each song has a specific score for each emotion that exists in this lexicon. The overall score of the emotions in the genre is the sum of the emotion scores in the corresponding genre songs. Having the scores of each emotion for every genre, the emotion with the biggest one became the emotion that represented/was more recurrent in the genre. It was decided not to count the emotions 'positive' and 'negative' seeing as one of them was always the most representative emotion, resulting in a redundant and similar analysis to what was done previously, therefore removing the potential interest of performing this emotion analysis.

After using this algorithm, the sentiment more predominant in each genre was joy in the case of pop, country and r&b, sadness in the case of rock, anger in the case of rap and anticipation in the case of miscellaneous.

In light of this sentiment analysis, some graphs were made to explore this new gathered information, in order to get more interesting insights about the music world that is being observed.

The first graph that was analyzed was the scatter plot. (Figure 12) When observing this graph it is possible to reach the conclusion that the songs with more views correspond to the extremes of the polarity, either very close to -1 or very close to 1. That is, the songs with more extreme negative polarity and the songs with more extreme positive polarity.

The second graph that was analyzed were boxplots of key emotions. (Figure 13) The boxplots show the relation between the sentiments and the genre popularity. A logarithmic function was applied to the views feature, in order to better visualize the results. This was done since they were very close to 0, which would make the interpretation and visualization a lot harder. An insight that can be taken from the graphic is that sadness is the emotion that in average has more views while anticipation is the one that has less.

The third and final graph that was observed was a line chart. (Figure 14) The objective of it was to check if there was a change of the predominant sentiments over the years for each genre. By observing the chart, it is possible to see that the one that changed the most through the years was the rap genre, that in the early 60s was very positive and has been decreasing a lot until today's days. The country genre was the most stable, however, it is good to note that there are few records of this genre in the dataset, seeing as they only start in the 90s.

Conclusion

In conclusion, this project successfully demonstrated the ability to predict song genres from lyrics using natural language processing and machine learning techniques. The neural network (Dense[100] + Dense[6]) model built in keras showed strong performance in recognizing the linguistic patterns behind the seven studied musical genres, achieving 0.71 and 0.66 values of f1 score for the train and validation sets respectively. The integration of sentiment analysis provided interesting insights into the emotional tones connected to each of the genres.

While believing this technology will certainly impact the music industry, especially song-related enterprises that rely on personalized song recommendations, such as Spotify or AppleMusic, it is crucial to acknowledge the limitations of our work. Since it is expected that the number of musical genres exceeds 41 categories, which are then subdivided to more than 300 subcategories, applying a similar approach using a dataset where more genre tags are included, could lead to a more in-depth analysis and bring more value to this study.

Additionally, a more detailed analysis could have been done to deal with songs in other non-English languages. In this case, translating the lyrics could be an option, however, being important to keep in mind that automatic translation still has its limitations.

The inclusion of the remaining dataset's textual features could have also been tested as to see if performance results would increase. Namely, the inclusion of the song artist's name might lead to some interesting results, since an artist is commonly associated with only one or a small set of genres. However, it is important to note that this would lead to an increase in dimensionality. In this specific dataset, there were almost 75 thousand different artists which also had some noise in its representation that would require a further and detailed analysis. In this project, the tradeoff between spending time treating these variables inconsistencies or working in other aspects relevant to the project was considered and deemed unnecessary, due to the lack of computational power and possibility of these words later being discarded by the log-ratio analysis.

In the context of model building, it could also be interesting to further test the xgboost classifier, since it showed promising results, despite the overfitting that led to its elimination. This challenge could have been mitigated with further testing of parameter combinations, however, due to the vast number of parameters it has, computational capacity led to it being discarded.

Finally, more nuanced approaches to sentiment analysis, where additional information about the songs was included or where other lexicons were explored, might lead to interesting insights and provide a deeper understanding of the emotional nuances in lyrics.

References

[1] Mayer, R., Neumayer, R. & Rauber, A. (2008, June). Rhyme and Style Features for Musical Genre Classification by Song Lyrics. In Ismir (pp. 337-342).

[2] Howard, Silla Jr., N., & Johnson, G. (2021, November 16). Automatic Lyrics-based Music Genre Classification in a Multilingual Setting. Kent Academic Repository.

[3] Girase, Advirkar, Patil, Khadpe, & Pokhare. (2014). Lyrics Based Song Genre Classification. Journal of Computing Technologies, 3(2).

[4] Neumayer, & Rauber. (2007, April). Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. ECIR 2007.

[5] NRCLex. (2022, August 31). PyPI. https://pypi.org/project/NRCLex/

[6] Elbagir, & Yang. (2019, March). Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment. IMECS 2019.

[7] Sentiment and emotion lexicons. (2019, June 3). National Research Council Canada. https://nrc.canada.ca/en/research-development/products-services/technical-advisory-services/sentiment-emotion-lexicons

[8] Introduction to Boosted Trees — xgboost 2.0.3 documentation. (n.d.). https://xgboost.readthedocs.io/en/stable/tutorials/model.html

[9] How XGBoost Works - Amazon SageMaker. (n.d.). https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html

[10] Bhattacharyya, J. (2021, October 7). Understanding XGBoost Algorithm In Detail. Analytics India Magazine. https://analyticsindiamag.com/xgboost-internal-working-to-make-decision-trees-and-deduce-predictions/

[11] Islam, Chen, & Jin. (2019). An Overview of Neural Network. American Journal of Neural Networks and Applications.

Annexes

XGBoost Classifier

XGBoost, or eXtreme Gradient Boosting, is a machine learning algorithm that operates within the gradient boosting framework. This algorithm is recognized for its computational capability, it applies ensemble learning by combining multiple models to increase the accuracy of the predictions. By incorporating the principles of gradient boosting, XGBoost builds a robust predictive model through iteratively adding weak models and correcting errors along the way. As a tree-based model, it builds trees and optimizes split points based on purity scores. This model also incorporates regularization techniques to handle overfitting, allowing for a better control over model complexity through its hyperparameters. In sum, XGBoost is a versatile and efficient algorithm, known for its speed and performance in tasks such as classification and regression.

Keras Neural Networks

Keras is a user-friendly tool for building and training neural networks. It simplifies the process by using layers to construct a model. This library offers tools like activation functions and optimizers to enhance the network's learning. In sum, Keras helps in the process of making neural networks without the complexities of coding from scratch.

The feed forward neural network is one of the most basic forms of Artificial Neural Networks, and is the approach used in this project. It involves a unidirectional flow of data or input, going from input nodes to output nodes. Data flows through the network without forming cycles or loops. In simpler terms, it's like a pipeline for information. Data goes in, passes through the layers, which can include hidden layers or not and comes out the other side, with no backward or feedback loops. It is commonly used for tasks like classification and regression.

In the specific case of this project, the neural network was built using fully connected layers, which means that each neuron from a previous layer will be connected to every neuron from the current one. The activation function used to introduce non-linearity to the model in the hidden layer was the relu (rectified linear unit), which sets negative values as zero, while positive values remain unchanged. For the output layer, as this project regards a classification problem, the activation function used was softmax, which will give a probability for each of the target classes. In this specific context the L2 regularizer, also known as Ridge Regression, was used. This technique is commonly used in machine learning to prevent overfitting. It works by adding a penalty to the loss function during training, ensuring that the model does not focus all its knowledge in a specific feature. The regularization factor, will determine the weight of the penalization. The higher the regularization factor the higher the penalties and the less likely is the model to overfit.

The sparse categorical crossentropy is used as a loss function, as it’s a classification problem. The loss function measures the difference between the true values and the model’s predicted values. The adam optimizer and the early stopping callback was also used. The optimizer is designed to help the model learn and improve its performance over time, while callbacks are used to end the model’s training cycle if or when the validation loss stops improving or worsens.

Figures

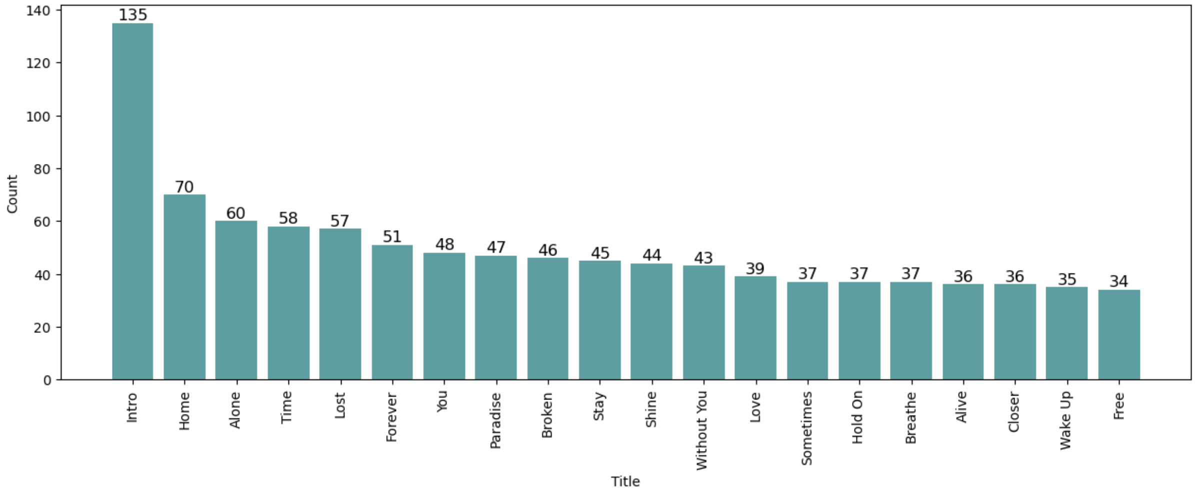

Figure 5 - 20 most common song titles

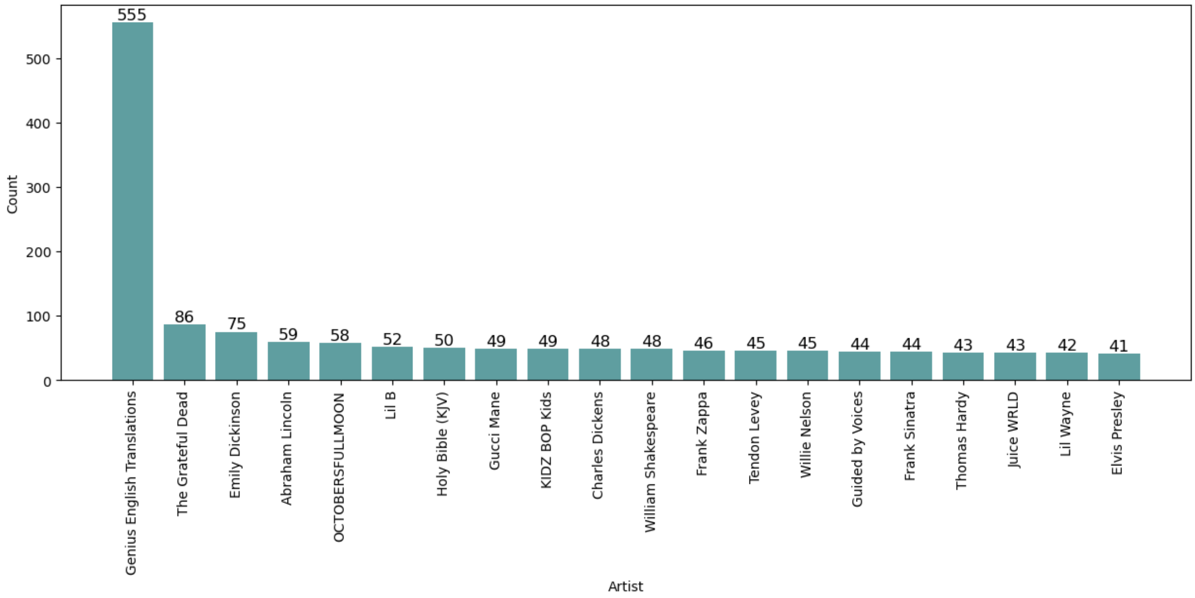

Figure 6 - 20 most common song artist

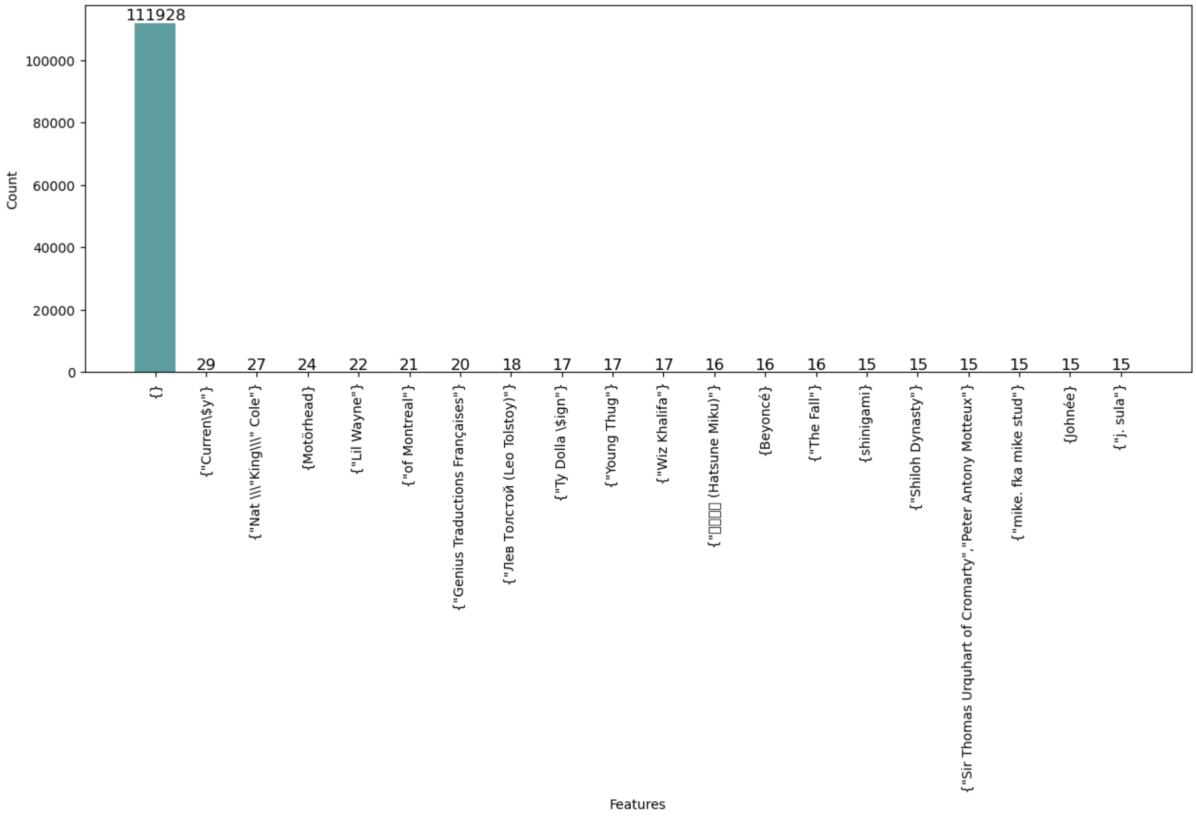

Figure 7 - 20 most common song featuring artists

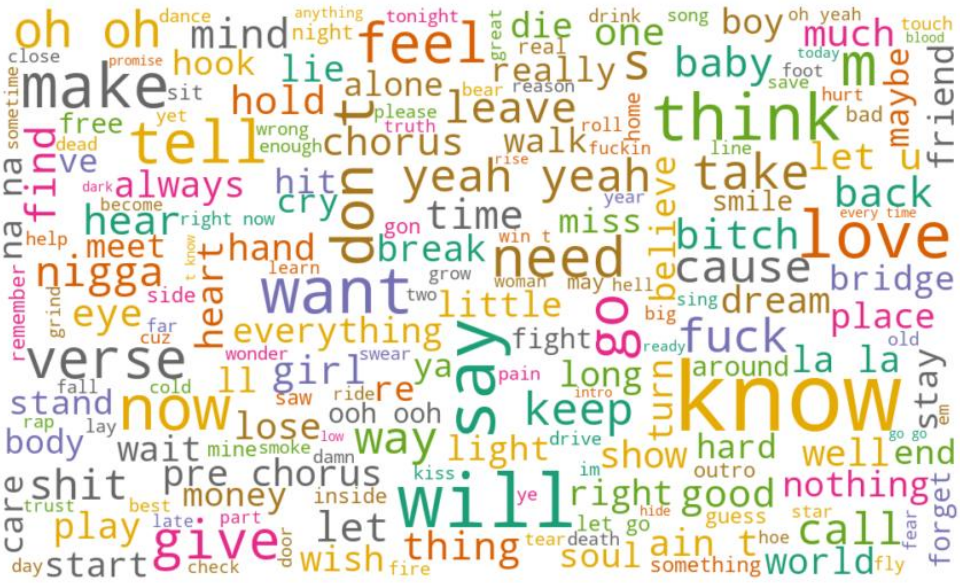

Figure 8 - word cloud for all relevant words

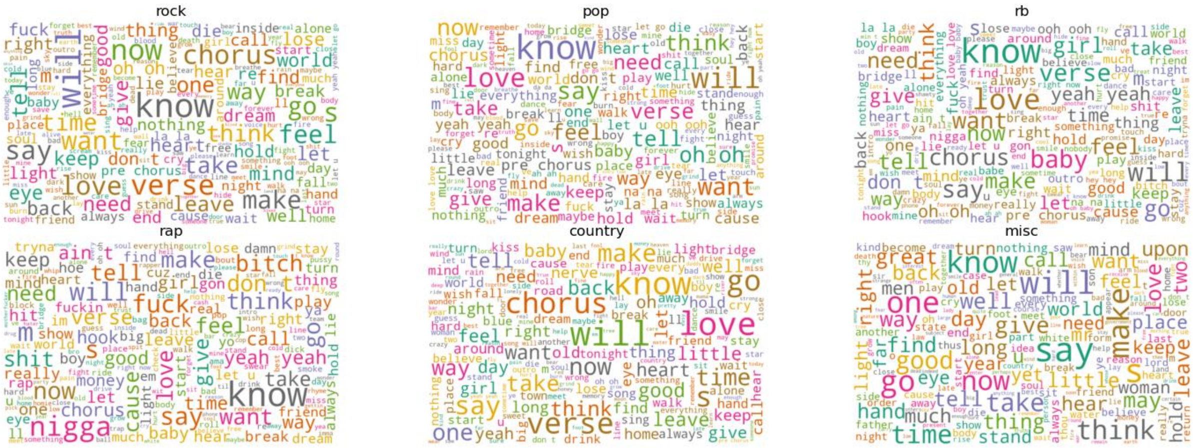

Figure 9 - word cloud of relevant words by genre

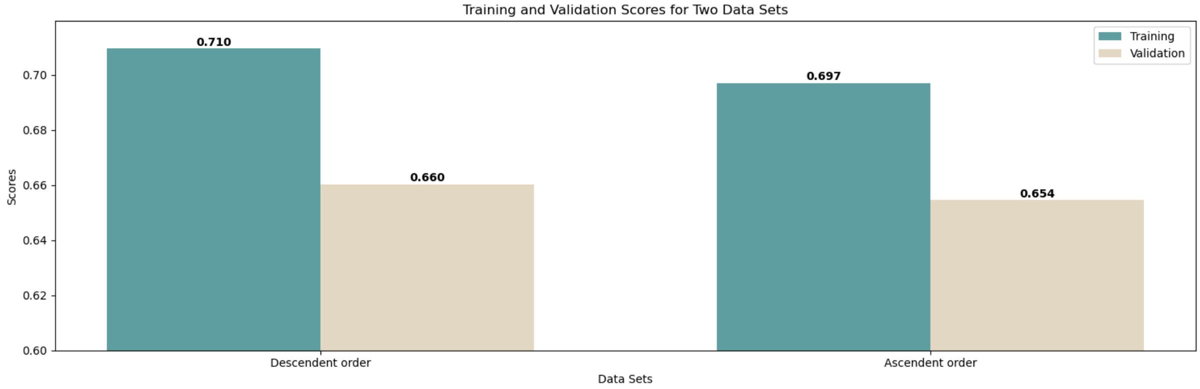

Figure 10 - log ratio order performance comparison

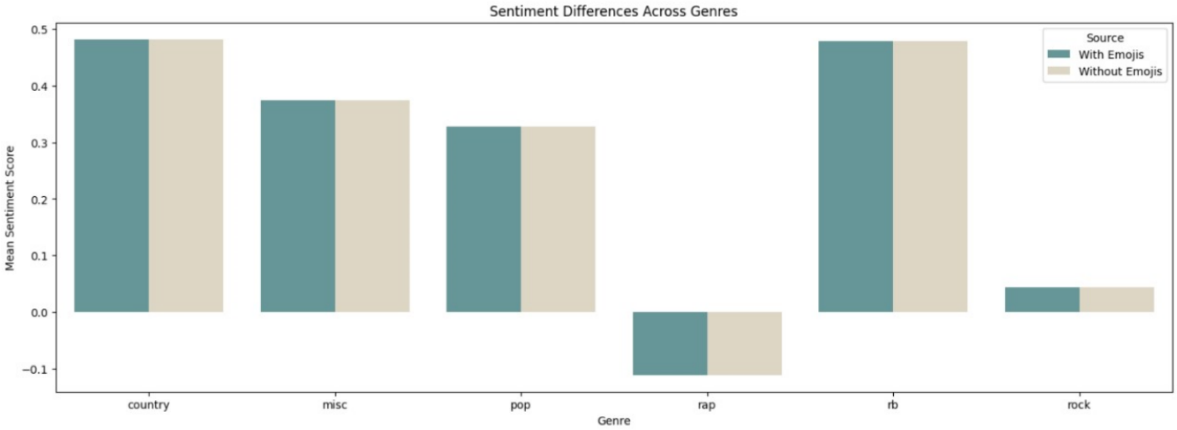

Figure 11 - Polarity by genres datasets comparison

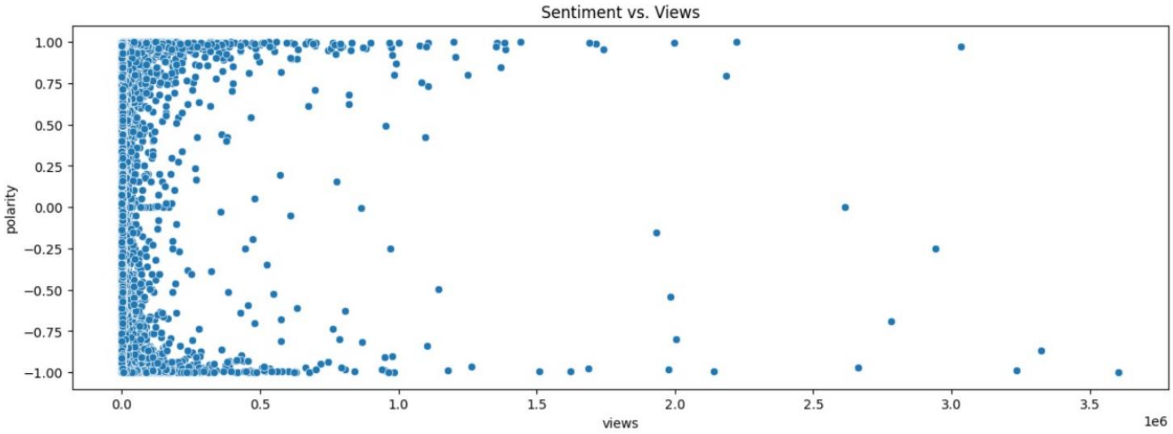

Figure 122 - sentiment vs views

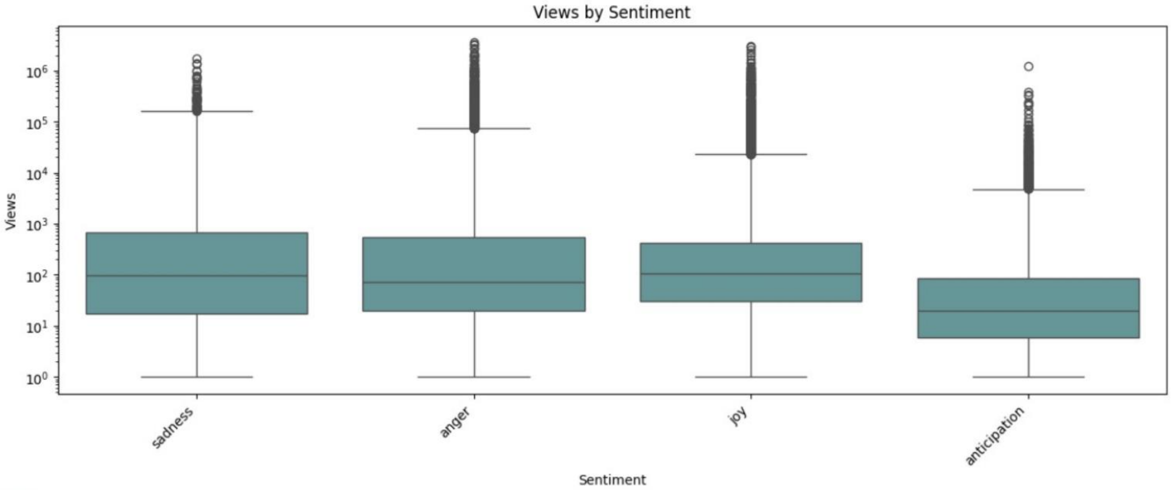

Figure 13 - views by sentiment

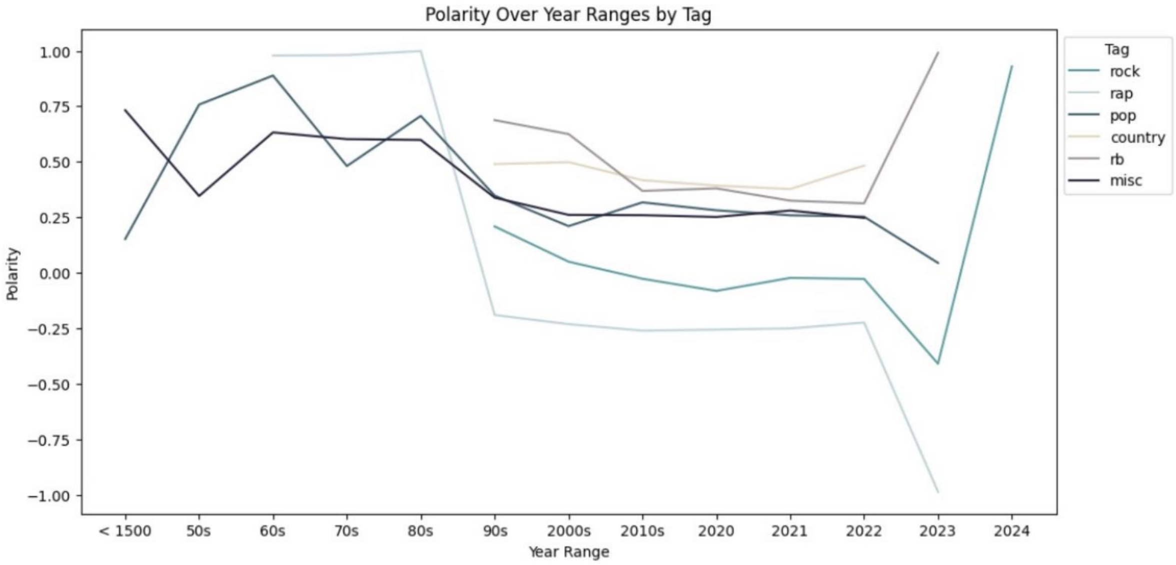

Figure 14 - polarity over year ranges by tag