NOVA IMS Information Management School

TEXT MINING

DATA SCIENCE DEGREE

With the orientation of professors Carina Albuquerque, Artur Varanda, Mohamed Elbawab and Ricardo Santos

Decoding the Rhythms of Emotion:

A Sentimental Journey through Music Genres

Project Report

20211595 | António Oliveira
20211628 | David Martins
20211637 | Mariana Ferreira
20211619 | Mariana Takimura
20211639 | Rui Lourenço

Abstract

In this report we will describe our approach to deal with a dataset containing music lyrics, where we aim to classify them into different genres and perform Sentiment Analysis. To do this, we decided to split the project into three different code notebooks and an extra file with functions used across all notebooks. These consist of EDA, Classification, Sentiment Analysis and the *py* file is called functions.

In order to explore the data we were given, we started by making sure we fully understood each feature. Then we performed some commonly used methods to better explore our data. Following this, we looked for missing values, outliers and incoherencies, treating them accordingly. After this we cleaned and normalized our data, where we used regex to remove unwanted text elements, like HTML tags. We also performed tokenization, stop word removal, lemmatization and stemming.

Finally, we exported our data that will be used for Genre Identification and Sentiment Analysis. In both notebooks we performed methods learnt in class, as well as some other methods we believed were adequate. We decided to keep most methods we tried, even though some produced much better results than others.

The model that gave the best results in the Genre Identification section was a Multilayer Perceptron. This model will later be fully explained.

Key Words

Text Mining; Music; Supervised Learning; Genre Identification; Sentiment Analysis

Contents

1. Introduction and Objectives

"Music is the universal language of mankind." – Henry Wadsworth Longfellow. With this statement one might say that music is the basis of humanity. Not everyone will agree with this statement, but in fact, music is present everywhere, in our day to day lives. It is a form of expressing oneself through the melody and lyrics of a song across various genres. This captures the essence of our project, which seeks to explore the relationship between music and emotion.

Through the lens of Natural Language Processing (NLP) we want to complete this project with the goal of creating a classification model that can accurately predict the genre of a song based on its lyrics, investigating the relationship between a song's lyrics and its genre and to perform Sentiment Analysis, trying to understand the emotional undertones present within each song.

For this project, we were provided two datasets that contained several attributes about each song and its musical genre. The **training set** includes the features and the ground truth being the basis to build our model. The **test set**, as the name indicates will be our dataset to test the created model, where the music genre is not known. When doing the Sentiment Analysis, we will aim to answer questions such as how each genre's sentiments change over the years, and which are the most positive or negative musical genres.

We also did some research for papers with similar objectives. We started by looking for papers that aimed to classify musical genres based on their lyrics. "*The Music Genre Classification using Song Lyrics*" paper has a very similar goal to ours. They used Kaggle data, treated it and used GloVe Embeddings, along with an LSTM model to make predictions, having achieved an accuracy of 68%. As they also had an imbalanced dataset, they used oversampling for the classes with the least number of observations. This paper gave us some ideas we will keep in mind for later sections of the project.

We also discovered that the paper "*Sentiments mining and classification of music lyrics using SentiWordNet*" had similarities with our project, even though their approach was a little different. They started by cleaning the data, removing repeated content (verses) and removing punctuation. Then, they took each lyric as input, passed it to a POS tagger and before applying *SentiWordNet*, they lemmatized and stemmed the words. We do not believe that removing punctuation would be a good step in our case, as the punctuation is very important to understand the sentiment that lyrics give. There were other papers, like "*Improving mood classification in music digital libraries by combining lyrics and audio"* that performed sentiment analysis using both the lyrics and the audio components of songs.

By applying Text mining techniques, we aim with this project not only decode the rhythms of emotion, but also to explore the huge landscape that is the music world and uncovering the emotional undertones of the various genres.

Our project code is divided into three Notebooks: 01_EDA, 02_Genre_Identification, 03_Sentiment_Analysis and a .py document named functions.

2. Data Exploration and Preprocessing

In this section of the report, we began our analysis by performing an Exploratory Data Analysis (EDA) to explore and better understand or data. This step consists of different sections where our main goal is to clean and normalize the data, as well as treat possible outliers, missing values and incoherencies to increase our Classification model performance and to perform a better Sentiment Analysis. We started by importing the necessary libraries and loading our data.

2.1. Preliminary Analysis

To start the analysis, we performed simple yet crucial operations, where basic methods such as *.info* and *.describe* were employed to uncover patterns and insights from the data.

Our first insights were that we had seven columns (variables) and 134967 observations (music entries) to work with. From the seven variables, only two are numeric and only the variable *title* appeared to contain null observations, a problem that will later be addressed. We noticed that the column *year* had strange values as its minimum and maximum values were respectively 1 and 2024. As the median of the column *views* is significantly lower than its mean, we also concluded that this feature had a right-skewed distribution.

We decided to further explore the target variable, *tag*, to understand which were its distinct values and how many observations of each *tag* we had in our dataset. There are six unique music genres in our dataset. The music genre with most observations was ***pop*** and the one with the least observations was ***country***.

During our exploration of random lyrics, it was noticed that there was some text between square brackets and regular parentheses that could be irrelevant to the project, the content within them was inspected concluding that the contents within square brackets did not add relevant

information such as tags for specific sections of the lyrics, and for the regular parentheses, it was concluded that for most part, they contained relevant information such as actual lyrics.

Then, we delved into these occurrences in more detail by displaying a random lyric for each existing *tag*. This allowed us to identify that the **misc** column contains objects that are not songs and confirmed the suspicion that the information within square brackets was in fact, not relevant in differentiating each *tag*.

All these issues will be addressed later in the report.

2.2. Missing Values Identification

Addressing the two missing values in the *title* variable, these two rows were visualized and subsequently removed from the dataset, given that they were not a significant amount, verifying the success of the process after.

2.3. Incoherencies Check

The prior exploration revealed certain inconsistencies in the numerical columns' *year* and *tag*. When addressing the column *year*, there were values such as 1 and 2024 in the dataset. Even though humans started producing music many, many years ago, we do not believe it is realistic for our dataset to contain these. As we are unaware of the existence of the ability to travel to the future, we also agreed that it is impossible that a song that is yet to be launched to be present in our dataset. To further explore this issue, we verified the number of songs that were produced before 1900 and in 2024 within each genre, showing that most of the songs associated with these unusual years belong to the **misc** *tag*, leading us to consider it a problem. The identified anomalies were treated in the outliers section.

Addressing the incoherencies in the column *tag*, we displayed some examples which were identified through random sampling in the Preliminary Analysis section, like an interview of Eminem to ESPN in 2013, and a sacred text from the Bible, leading us to conclude there are several cases such as these. Observations like this will be later addressed in Outliers section.

2.4. Outliers

Here we will address the previously identified outliers and search for potential outliers. Starting with the column *year*, we removed the improbable song associated with the year 2024. Then we visualised the remaining years by plotting histograms, where we noticed that songs produced before 1950 are significantly infrequent. As songs from before 1950 constitute less 0.1% of our data, they were removed, keeping the songs produced in 1950 or after.

Considering now the column *views*, we checked for outliers since it had a wide range, more precisely a minimum of 0 and maximum of approximately 3.6 million. This variable is significantly skewed, given how low the values are in its quartiles when comparing to the maximum. We decided to plot a histogram to better visualise this variable, verifying that the *tag* **rap** was associated with having a larger number of views, something it is worth keeping in mind for the following steps. Regarding this variable, it was considered that it is reasonable for songs to have such a wide range of views, not removing any observation.

Then we addressed the *lyrics length*, which was a variable created by us that measures the length of each observation in the dataset. It was plotted a histogram allowing us to conclude that the majority of the songs belong to a somewhat small range of lyrics length, even though there are occasional lyrics with large, and perhaps extreme lengths. The next step taken was to display the *lyrics length* by *tag*, to see if there was a specific tag with considerably higher values than the rest. We noticed that the *tag* **misc** had a considerably higher maximum value followed by the *tag* **rap**. To address this issue and considering the previously stated incoherencies associated with the **misc** *tag*, we looked at the top 10 **rap** songs with the highest length, although these are unusual high values, they are songs that exist, hence we kept them in our dataset. We then proceeded to look at the **misc** *tag* being this a little bit trickier, as we noticed that the lengthier observations were not songs. To resolve this, any song that exceeded the maximum length song in the **rap** *tag*, was removed. All these songs represented less than 0.1% of the dataset and were part of the **misc** *tag*.

2.5. Special Comments

It was observed that newline characters " \n " commonly appeared in each sample, and as it is not something useful for us, we decided to later remove them. Some other inadequate and irrelevant characters were detected, such as " \ " and tags for specific sections between either square brackets or regular parenthesis such as "*(Chorus)*" or "*(Chorus: (…))*", "*[Verse(…)] *", "*[Chorus]*", "*[?]*" and "*[Outro]*" which again, does not add any information needed to perform Sentiment Analysis nor Genre

Identification and can mislead the model given these portions of text. The information within square brackets was addressed in the upcoming section.

2.6. Data Cleaning and Normalisation

To only keep the essential information in each observation, we knew we needed to clean and normalise the data. To do so, we took advantage of a function, *preprocessor*, which allowed us to input what we wanted to remove or keep. This function had parameters that allowed us to lowercase the hole text, as well as to leave or remove punctuation and stopwords, to correct the spelling, to perform lemmatization or stemming, and to tokenize or transform the output into sentences. This function allowed us to try many different combinations, for example between lemmatization or stemming, as the function does not allow to do both at the same time. We also experimented with *TextBlob,* to correct the spelling of words like *u* which mean *you*. Independently of the selected parameters, this function would always remove the words between square brackets, URLs, isolated consonants, newline characters, number and letters from other alphabets (like Cyrillic). It also converts words to the same font.

We applied this function with different goals: for Sentiment Analysis purposes, where we kept capitalisation and punctuation, and for Classification.

2.7. Data Visualisation

With our data cleaned, we performed some visualisations to complement the insights taken so far. Word Clouds were particularly insightful (Annexes-6.1.WordClouds), revealing if there were any specific words more commonly present in certain tags than others. We realised that the tags **country** and **pop** commonly contain the word *love*, whereas *rap* contained a lot of swear words, being the most unique one. Then, we went ahead and plotted some bar charts and treemaps.

For a first visualisation, we plotted a bar chart that showed the Total Number of Songs per Genre, indicating that the *tag* **pop** had more songs in the dataset and **country** the least number songs, as seen before. Next it was plotted the Sum of Total Views per Genre, which allowed us to conclude that the genre with most visualisations is **rap** even though **pop** has a higher number of songs and country has the least number of views in total, taking into consideration that it has less songs. Following these insights, we decided to take a closer look at the Average Lyrics Length per Genre it showed us that the **misc** genre has the highest average of "lyrics" length by far, this was to be expected

from previous analysis. Total Views per Year and Average views per Year allowed us to say that there are many more views in recent years, which can be due to the easy access we have to music streaming platforms and social media nowadays. However, the average of views per year is evenly distributed. We followed plotting Visualisations per Year per Genre, that summarizes previous insights, as expected. Songs made per Year and Unique Artists that Produced at least a Song per Year from the plots, we can see that there are more songs being made and that there are more artists releasing songs in recent years, this leads us to relate to previous insights taken about the hight view count recently. We then proceed to plot a treemap for Artists with the Most Songs to get a general view of the artists with most songs in our dataset, "The Grateful Dead" is the second one. Since "Genius English Translations" had so many songs produced, we decided to remove it from this plot. Then, to finalize our visualisations, we plotted Artists with the Most Songs per Genre, for **rock** the artist with most songs is "The Grateful Dead" with 75 songs, for **rap** and **pop** are the "Genius English Translations" with a total of 163 and 309 songs, respectively, for **country** is Willie Nelson with a total of 42 songs, for **rb** is also "Genius English Translations" with a total of 30 songs and for **misc** the top artist is Abraham Lincoln with a count of 57 "songs".

3. Genre Identification

With our data properly cleaned, we began the next step, and consequently the next notebook, where our goal was to identify the genre of each music. To do so we started by importing the data outputted by the previous notebook and followed this by performing the necessary steps to take advantage of methods learnt in class, like Bag of Words, TF-IDF and Word Embeddings, namely *Doc2Vec*. With these methods we aimed to correctly classify musical genres.

Having imported the data from the previous notebook, we implemented the code for the previously mentioned methods. Before doing so we had to use pre-process the stopwords, since this method is useful when implementing BoW and TF-IDF and experimenting with an extended list of stopwords, more specific to our problem, instead of only using the traditional ones. We also experimented with using oversampling through the *imblearn* library's method SMOTE. However, the initial results were not promising, and due to time constraints, we moved on without oversampling.

As we wanted to try different combinations of pre-processing techniques along with different Vectorisation approaches, different models and parameters, we took the following approach: 1. Import the pre-processed data (for example, lemmatized data); 2. Run a Vectorization method (like TF-IDF); 3. Run a Stratified K-Fold using a designated model (for example, Random Forest); 4. Make predictions on the test set and export the results to a CSV file that will be updated on Kaggle.

In terms of pre-processed data, we experimented with lemmatization, stemming and tokenization. For vectorization, we experimented with Bag of Words, TF-IDF and *doc2vec*. For modelling purposes, we experimented with Logistic Regressions, Random Forests, Multinomial Naïve Bayes, Multilayer Perceptron and Neural Networks. We also used GridSearch to optimise each model's parameters.

The best result we got was while using lemmatization, with the doc2vec vectorization , using as parameters {vector_size=300,negative=5, hs=0, min_count=2,dm=0, sample = 0,epochs=30,workers = 8}, and the Multilayer Perceptron model with the best parameters {'activation': 'logistic', 'batch_size': 256, 'hidden_layer_sizes': (128, 64), 'warm_start': True}, obtained with a GridSearch. We also used an Early Stopping callback to prevent the model to continue running when the model was not improving. When using the same model, but with 'activation': 'tanh' we also achieved similar results. This model granted a weighted average of 64%, performing at its best when predicting **rap** and **pop.**

4. Sentiment Analysis

The Sentiment Analysis section will dive into the emotions expressed in lyrics, providing insights into the overall sentiment of the song using computational methods to categorize emotions in text as positive, negative, or neutral. We applied and compared different algorithms: VADER, TextBlob, SentiWordNet and AFINN Lexicon.

To perform this task we, imported the data outputted from the EDA notebook. This data is different from the one used in the Genre Identification section, as this time we are interested in keeping text components like capitalization and punctuation, which are crucial to Sentiment Analysis.

4.1. VADER

We started our Sentiment Analysis using the VADER algorithm. With it we computed the polarity scores, which consisted of individual negative, neutral and positive scores, as well as the polarity compound, where the compounded scores are stored. After computing these scores for our entire dataset, we used the method *describe* to further explore and gain insights, having also plotted histograms to better understand the sentiment analysis' distribution.

4.2. TextBlob

Following this, we employed the TextBlob algorithm, which was very straightforward to apply, being a lexicon-based sentiment analyser with predefined weights for a set of words. We then analysed the results.

4.3. SentiWordNet

We proceeded to employ *SentiWordNet*, an algorithm that takes advantage of the part of speech tags of each word to perform Sentiment Analysis. We started by Tokenizing our data, and then assigning a POS tag to each token. As the *SentiWordNet* algorithm can only deal with a limited number of tags, we had to create a function that transformed the NLKT POS tags to the *SentiWordNet* ones. Finally, we computed the sentiment score, and looked at the results.

4.4. AFINN Lexicon

We also decided to implement the *AFINN* Lexicon. This algorithm just needed as input the pre-processed sentences outputted from the EDA notebook. We computed the AFINN scores and again looked at the results.

As the *SentiWordNet* and the *AFINN* algorithms outputted values with a very wide range, we decided to scale them using the MinMax Scaler.

4.5. Comparing the Results

Having experimented with different algorithms, we compared their results. We started by using the method *describe*, making sure all the explored polarity scores had ranges between -1 and 1. Additionally we used the same method, but to obtain the values per musical genre.

We also decided to present the results through a series of visualizations, which aimed to answer some of the questions we posed earlier, as well as others that arose during this process. We started by displaying the mean polarity of each method per musical genre, which showed us that the Vader algorithm has significantly more extreme values when compared to the remaining algorithms. This algorithm was also the only one that considered a genre to have a negative connotation, this genre being Rap. This was followed by a visualization of the maximum polarity values per label, where we concluded that, in each genre, there is at least one algorithm that achieves a maximum positive polarity. These visualisations allowed us to conclude that the predominant sentiment in the songs' lyrics is positive. The rock and rap genres have a more neutral polarity. This may happen because the songs in these genres can transmit a wider range of emotions.

Following this, we also decided to see how each average sentiment score changed by year per musical genre. The Vader algorithm shows the most changes over the years, whereas the other algorithms appear quite consistent. It is shown that Misc is the most inconsistent genre for all algorithms, due to the variety of songs present in this class.

Then we decided to plot the average sentiment scores for the top three artists that most appeared in this genre. These visualisations allowed us to conclude that the rock band *The Grateful Dead* transmits a positive connotation, whereas the rap artist OCTOBERSFULLMOON transmits negative or neutral feelings in their songs.

We also added a visualisation that compared the mean number of views with the positive and negative sentiments computed by each algorithm. This visualisation made us realize that, on average, the songs predicted to have a negative connotation by the Afinn algorithm had considerably more visualisations than the rest.

5. Conclusion

The encounter between NLP and music led us on the exploration not only to "decode the rhythms of emotion", but also to explore the huge landscape that is the music world and uncovering the relationship between music and emotion.

This project had three main parts, the data exploration and treatment, the creation of a predictive model and the analysis the emotions in each genre. Through our knowledge in Data Science and Text Mining capabilities, we created a genre classification model capable of predicting the songs genre based on its lyrics and performed a Sentiment Analysis to uncover the emotional undertones within each songs genre.

In the Genre Identification section our model was able to achieve a weighted average of 64%. Despite being good at predicting pop and rap, our model is not great at predicting rb and country. This was to be expected, since these classes have the least amount of observations and they had similar wordclouds.

By performing the Sentiment Analysis, we employed algorithms such as VADER, TextBlob, SentiWordNet and AFINN Lexicon, through these algorithms the emotional expression of each genre came to the top. VADER algorithm exhibited the most extreme and variable values while the other algorithms were stable.

However, even though we are satisfied with our results, we acknowledge they could be even better. First of all, in the Genre Identification Section, maybe our models could have been more tuned, but due to the time each one takes to run, this was not possible. Then we also believe that using pre-trained embeddings, adequate for musical lyrics could slightly improve our Sentiment Analysis results. Time constraints also limited the experimentation with some more Sentiment Analysis algorithms to get a more in depth understanding.

This project gave us a new perspective on the universal language, the power that words hold, discovering the intricate relationship between music and emotion. In the words of Henry Wadsworth Longfellow, "Music is the universal language of mankind.".

6. Annexes

6.1. Word clouds

Country BoW

Misc BoW

Pop BoW

Rap BoW

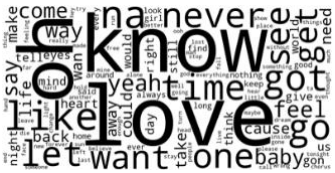
RB BoW

Rock BoW

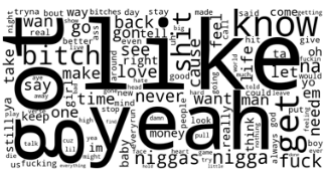
Country TF-IDF

Misc TF-IDF

Pop TF-IDF

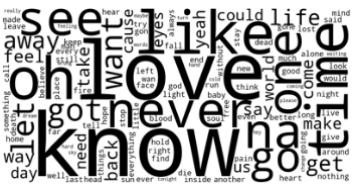


Rap TF-IDF

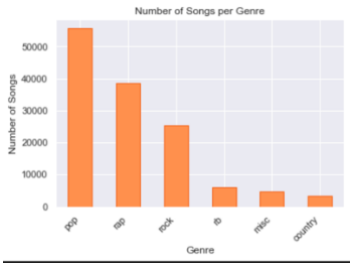


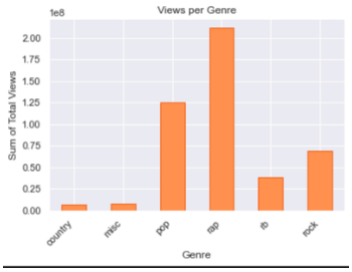
RB TF-IDF

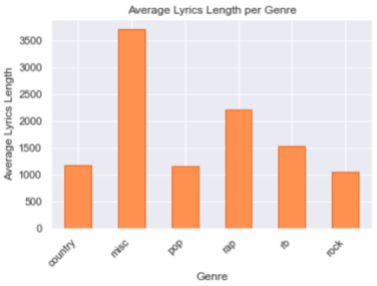
Rock TF-IDF

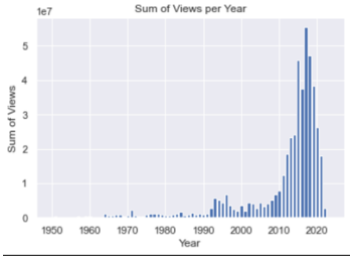


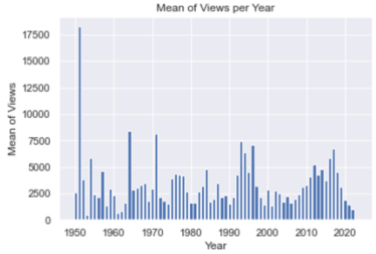
6.2. Other Visualisations

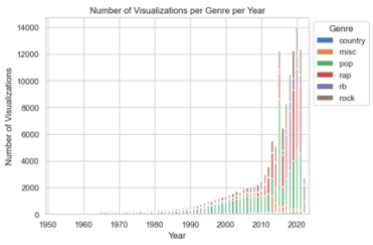


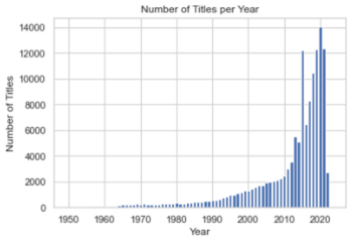


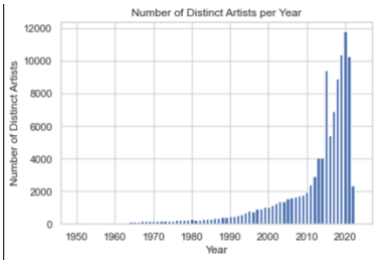


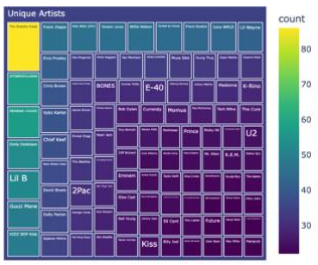


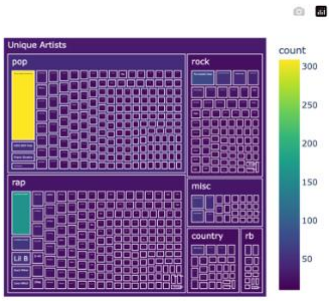












7. References

V. Sharma, A. Agarwal, R. Dhir and G. Sikka, "Sentiments mining and classification of music lyrics using SentiWordNet," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 2016, pp. 1-6, doi: 10.1109/CDAN.2016.7570965.

Xiao Hu and J. Stephen Downie. 2010. Improving mood classification in music digital libraries by combining lyrics and audio. In Proceedings of the 10th annual joint conference on Digital libraries (JCDL '10). Association for Computing Machinery, New York, NY, USA, 159–168. https://doi.org/10.1145/1816123.1816146

Leszczynski, Megan and Anna Boonyanit. “Music Genre Classification using Song Lyrics.” (2021).

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report003.pdf