**SI 630 Project Final Report**

Musical Genre Classification

|  |  |  |
| --- | --- | --- |
|  | Chris Peña  chpena@umich.edu |  |

Abstract

This project explores the classification of music lyrics as documents to their respective genres. If executed properly, the classifier could be used, for example, to categorize new songs, more accurately classify artists and albums, and explore trends or progressions of artists. A small series of classifiers were used to predict genres with poor results. A pre-trained BERT model was also used, providing mixed results.

# Introduction

Music, like almost all art, is subjective, which leaves classification up to consensus. Wouldn’t it be helpful to have some sort of data-based assessment to help with categorization? Should the message or story of a song ultimately be what determines the genre that song belongs in? Should it go beyond just the lyrics? Is “Old Town Road” by Lil Nas X country or hip-hop? When did Taylor Swift really transition from country to pop? Is Cannibal Corpse death metal or horror metal? These types of questions are what I hope to be able to answer by somehow developing a genre classification model.

Other projects related to musical genre classification have primarily focused on one genre. The aim of this project is to broaden the scope of genres and lyrics.

The initial approach to solving this problem was to assemble a series of classifiers and compare the results. The classifiers chosen were a Naïve Bayes, a linear SVC, and random forest. To my disappointment the classifiers performed poorly when attempting to predict the genres based on lyrics provided.

Along with the goal of classifying lyrics to genres, an added benefit of this project is the opportunity for users to explore various genres of music. If the prediction of a given set of lyrics produces an unexpected genre / subgenre, it may lead to the exploration of that genre by the curious, lover of music.

As previously stated, the predictions by the series of classifiers were poor, with the BERT-based also performing not as well as expected.

# Data

The original intent was to use the Genius app and Spotify app API to extract lyrics and genres. The Genius API nor the Spotify API provided the genre label for artists.

A dataset for this project was created with API calls for a semi-random list of artists, and another dataset was derived from RateYourMusic.com’s top 500 albums as of October 2020 dataset from data.world.

The first source of lyrical data came from the Genius app API utilizing the LyricGenius package. The list of artists, and subsequently genres, is generated semi-randomly by soliciting ‘any artist’ suggestions from social network outreach. Once the semi-random artist list was set, API calls were used to extract five song lyrics from that artist. The resulting lyrics dataset was the first dataset used with a total of 135 song lyrics, but a very lopsided distribution.

The Rate Your Music (RYM) top 500 dataset contained artist names and genres, which was sufficient to build the larger lyrics dataset. Preprocessing the dataset included dropping unnecessary columns, dropping repeated artists, and exacting one genre per artist from the list of multiple genres / subgenres the artist / album was identified as. Genres with less than five appearances in the dataset were also dropped. The resulting column of artists was converted to a list, and the process of API calls to the Genius app was repeated to complete the larger dataset. This was the second dataset used, with a total of 311 song lyrics.

Chart, bar chart, histogram

Description automatically generated

Lyric distribution from social network outreach.

Chart, bar chart

Description automatically generated

Lyric distribution from RYM.com after preprocessing.

# Related work

Others have worked with the LyricGenius package to produce similar types of projects, but they tend to mainly focus on one particular genre. In contrast I will be attempting to classify any given lyrical set to a genre. The main advantage this provides over the other related projects is the ability to repeatedly apply the final work to future lyrical contributions.

The first focuses on country music and the correlation between certain words. The author wanted to find if a country artist’s gender could be predicted based on certain words, the relationship between ‘truck’ and ‘beer’, and ‘love’ and ‘girl’ / ‘boy’.

A project was developed to test how neural networks would attempt to derive meaning behind lyrics. It appears the project eventually shifted towards predicting text of lyrics instead of finding meaning.

A slighter older (2017) project was a sentiment analysis of rap lyrics. There was not much information available on the project, and I don’t believe it was completed.

Another word correlation project wanted to identify the usage of alcohol related words to musical genres. While this might be the closest project to my project, as it is genre classification of sorts, it focuses purely on alcohol related lyrics.

Possibly the most interesting of the related works was a project analyzing the evolution of song lyrics over the past few decades. Unique words, word complexity, and use of profanity are examples of what was analyzed.

# Methods

Collection of the first dataset was an attempt to collect an array of genres in the hopes of training classifiers exposed to a broad range of lyrics and genres. A request for ‘any five artists’ was sent to various individuals in my social circle with the goal of achieving the broad range. As noticed after collection and distribution plotting, the genres, though varied, are highly skewed.

The use of the Spotify API was abandoned as it was not providing much use to the project, and I instead focused primarily on the Genius app API[[1]](#footnote-1). The list of artists provided by the social circle was placed in a Python ‘for’ loop to make API calls for lyrics. A HTTPAdapter from the Requests[[2]](#footnote-2) library was implemented after having issues with the loop crashing due to timeout errors. The API calls did not provide genre classification, and manual annotation was used to classify each artist.

Due to the skewed dataset, I searched for an additional dataset that could be used for classifier training. The top 500 albums from the Rate Your Music website as of October 2020 from data.world[[3]](#footnote-3) provided artists, albums, genres, and several other details, and I decided this dataset could be used after some preprocessing of the dataset. The ’genre’ column contained multiple labels for each artist, and I extracted only the first (main) genre of that artist to use for training. Many artists were repeated in the original dataset, and duplicates were dropped while keeping their latest album as the genre to classify as. There was also a high number of genres listed only once, and I decided to also drop those genres with less than 5 occurrences. The resulting column of artists was set to a list, and it was implemented into the ‘for’ loop mentioned for lyrics.

A TF-IDF vectorizer[[4]](#footnote-4) was implemented for each dataset. A multiclass Naïve Bayes[[5]](#footnote-5), linear SVC[[6]](#footnote-6), and random forest[[7]](#footnote-7) classifier was initialized to train on each dataset. Grid Search Cross Validation[[8]](#footnote-8) was used in the random forest classifier to find the best parameters. The parameter grid used included estimators of 150, 200, 250, 300, and 500, with max depths of 3, 4, and 5, and a max features mode of ‘auto’ and ‘log2’.

The models were combined into a function in order to run them simultaneously with any given set of test lyrics.

A BERT-based model was also used as a classifier, and a lyrics specific model was found in the Hugging Face repository. The model is titled ‘AutoNLP Song Lyrics’ by user Julien Simon[[9]](#footnote-9).

# Evaluation and Results

Each dataset – the social circle generated and the RYM top 500 – were ran through each classifier. After poor results were generated from the social circle dataset, the next goal was to achieve higher accuracy from the RYM dataset.

The Naïve Bayes classifier was identified to be used as the baseline classifier for this project. The linear SVC and random forest classifiers were set to be the classifiers that would ideally outperform the Naïve Bayes. The first pass through the Naïve Bayes classifier with the social circle dataset predicted genres with 40% accuracy. Though not great, this turned out to be the highest accuracy of all the tests; possibly due to the high number of two genres. The same dataset produced an accuracy of 37% with the linear SVC pipeline - removing stop words. I re-ran the data through the linear SVC, but without removing stop words to test whether accuracy would improve. The result was once again 37%. Having selected the best parameters using grid search cross validation, I had high hopes the random forest classifier would produce higher accuracy results than the previous models. It unfortunately had the lowest accuracy score of about 30%.

Thinking the dataset could have been the culprit to such results, I found, processed, and ran the RYM dataset through the models after re-fitting the TF-IDF vectorizer with the new dataset. The Naïve Bayes classifier produced the worst results of the project, coming in with 19% accuracy. The linear SVC did improve accuracy and predicted almost 43% correctly. The random forest also improved accuracy to almost 40%.

The AutoNLP BERT-based model was the last model used, and has produced some unexpected results. The first test set of lyrics was a progressive metal category, and it did successfully predict different metal genres, such as groove and heavy metal, as the output. When the ‘hybrid’ lyrics of “Old Town Road” by Lil Nas X were introduced, groove metal was once again predicted by the model. Trying a known country song by George Strait again produced groove and thrash metal predictions.

**6 Discussion**

Overall, the project did not produce the desired outcome it set out to accomplish. The primary contributor to that is likely the datasets used. The initial dataset was both skewed and small. The second, though larger in size and more balanced, likely was still too small of a sample to train on.

It was disappointing to see none of the models successfully predicted over 50%. The BERT-based model seemed to have been the ‘winner’ after testing metal-based lyrics and predicted metal-based genres, but it seems to predict metal-based genres given any set of lyrics.

A much larger dataset of lyrics and genres may be required to get better results from the models. Seeing as how this dataset does not yet exist (that I’m aware of), similar preprocessing steps taken for this project would need to be completed to generate the desired dataset. With the possibility of tens or hundreds (maybe more?) of thousands of song lyrics required, the time needed would exceed that of a course semester. The presumption of the BERT-based model was that it had been trained on a much larger dataset, which leaves the metal-based predictions to be a bit of a mystery.

**7 Other things tried**

As mentioned, manual annotation was used for the first, social circle derived dataset. While it was very straightforward, it was also a bit time consuming.

The method of setting the dataframe index, stacking, and resetting the index used to preprocess the artist and lyrics for the second dataset is much more efficient, but finding that efficiency took time. That said, once it was successfully implemented it would be the preferred method if a larger dataset collection were to be done.

**8 Conclusion**

The idea of a lyrics-based genre classifier seemed, on the surface, to be a relatively straight forward task. As discussed in lectures for this course, what may seem trivial to a human is extremely complex to a machine. This project has demonstrated that natural language processing is a complicated field especially when being applied to machines.

**9 What would have been done differently or next?**

If time allowed, a larger collection of lyrics would be generated as a next step. A limit of subgenres for every main genre would likely be a good idea to implement, with a more even representation from each genre and subgenre.

Merging the two datasets used in this project and re-running the models could be another step taken to try improving accuracy of the models.

**10 Resources**

Data.World, Rate Your Music Top 500 Albums dataset. <https://data.world/notgibs/rateyourmusic-top500-albums>

Genius API. <https://docs.genius.com/#/getting-started-h1>

Goldman, Jacob. Song Lyric Corpus. 2017.

<https://github.com/JacobGo/nltk-lyric-corpus>

Gruss, Richard. Text Classification with Python. YouTube. 2019. <https://www.youtube.com/watch?v=EfEW3_RLnGA>

Miller, John. LyricsGenius package. <https://github.com/johnwmillr/LyricsGenius>

Simon, Julien. Hugging Face website. 2021. <https://huggingface.co/juliensimon/autonlp-song-lyrics-18753417>

Spotify Web API. <https://developer.spotify.com/documentation/web-api/>

1. Genius App API - https://docs.genius.com/#/getting-started-h1 [↑](#footnote-ref-1)
2. Requests used with API calls - https://docs.python-requests.org/en/latest/ [↑](#footnote-ref-2)
3. Data.world dataset repository website - https://data.world/notgibs/rateyourmusic-top500-albums [↑](#footnote-ref-3)
4. SciKit Learn TF-IDF Vectorizer - https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfVectorizer.html?highlight=tfidfvectorizer#sklearn.feature\_extraction.text.TfidfVectorizer [↑](#footnote-ref-4)
5. SciKit Learn Multinomial Naïve Bayes - https://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html [↑](#footnote-ref-5)
6. SciKit Learn Linear SVC - https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html?highlight=linear%20svc#sklearn.svm.LinearSVC [↑](#footnote-ref-6)
7. SciKit Learn Random Forest Classifier - https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random%20forest#sklearn.ensemble.RandomForestClassifier [↑](#footnote-ref-7)
8. SciKit Learn Grid Search Cross Validation - https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html?highlight=grid#sklearn.model\_selection.GridSearchCV [↑](#footnote-ref-8)
9. Hugging Face, Julien Simon. AutoNLP-data-song-lyrics Multi-Class Classification - https://huggingface.co/juliensimon/autonlp-song-lyrics-18753417 [↑](#footnote-ref-9)