Heavy Metal Lyric Metallurgy

ISyE 6740 Final Project

Matthew Holland
Fall 2021
GT ID# 903564707

1 Problem Statement

Throughout human history, humans have felt the need to classify objects into groups. This dates back to needing to be able to distinguish between prey and predator on the open savannah in Africa. Similarly, but much less importantly, humans have been organizing music into genres. According to the website musicmap, even in 1870, music could be classified into the genres, utility, folk, gospel, classical, and world. Currently, it is almost impossible to settle on a number of genres of music due to the subjectivity of classifying music. Complicating things further is the fact that genres of music have their own subgenres that can often be difficult to distinguish, especially among those who are not familiar with the genre.

One such type of music is heavy metal, or metal. It evolved from rock and blues, and it is characterized as "heavy" due to the distortion and amplification in the instruments. Further, it uses difficult, diverse voice

techniques, and it often features technically challenging instrumentals such as fast guitar solos. The performers tend to dress in dark clothing and have an overall "black" look. Finally, the lyrics tend to focus on dark themes. The many genres of metal are distinguished both by varying musical style and diverse lyrics. They include mathcore (with complex time signatures), funeral doom, and space metal to name just a few.

My project will focus on four genres of metal.

1. **Death Metal -** Death metal is heavy, versatile, and fast. The lyrics can involve many topics including death, political conflict, religion, or true crime.
2. **Pirate Metal -** Pirate metal can be performed like traditional metal or can be performed with folk instruments such as mandolins. The lyrics focus heavily on pirate mythology.
3. **Power Metal -** Power metal is high and fast with higher-pitched vocals. The lyrics tend to have elements of fantasy, and they have an emotionally "powerful" sound..
4. **Viking Metal -** Viking metal is performed in diverse ways such as with folk instruments and growled vocals. The lyrical content focuses on Norse mythology and the Viking age.

Death Metal

Death metal band, Cannibal Corpse [1]

Pirate Metal

Pirate Metal Band, The Dread Crew of Oddwood [2]

Power Metal

Power metal band, Dragonforce [3]

Viking Metal

Viking metal band, Thyfring [4]

Machine learning algorithms can be used to classify music into genres, but can a machine learning algorithm be trained to classify metal music into genres based solely on the lyrics? Or are the lyrics too similar for an ML algorithm to differentiate between them?

2 Dataset

2.1 Building the Dataset

There are several datasets that exist with the lyrics of popular music, but I was not able to find one for genres of metal. So, I built my own dataset. I built the dataset through a Google search of "top _____ songs" as the criteria for each of the genres. Since a Google search is imperfect and doesn't always yield bands from the proper genres, I cross referenced the bands with Wikipedia to make sure that they were classified in that musical genre.

2.2 Word Clouds

I created word clouds from the lyrics for the songs that I chose in the dataset. In the word clouds, I removed stop words and chose a minimum word length of three characters. They are quite representative of the lyrical themes of each type of music. However, it should be noted that the small nature of the dataset means that words in repetitive songs end up with "bigger" representations in the word clouds.

Death Metal

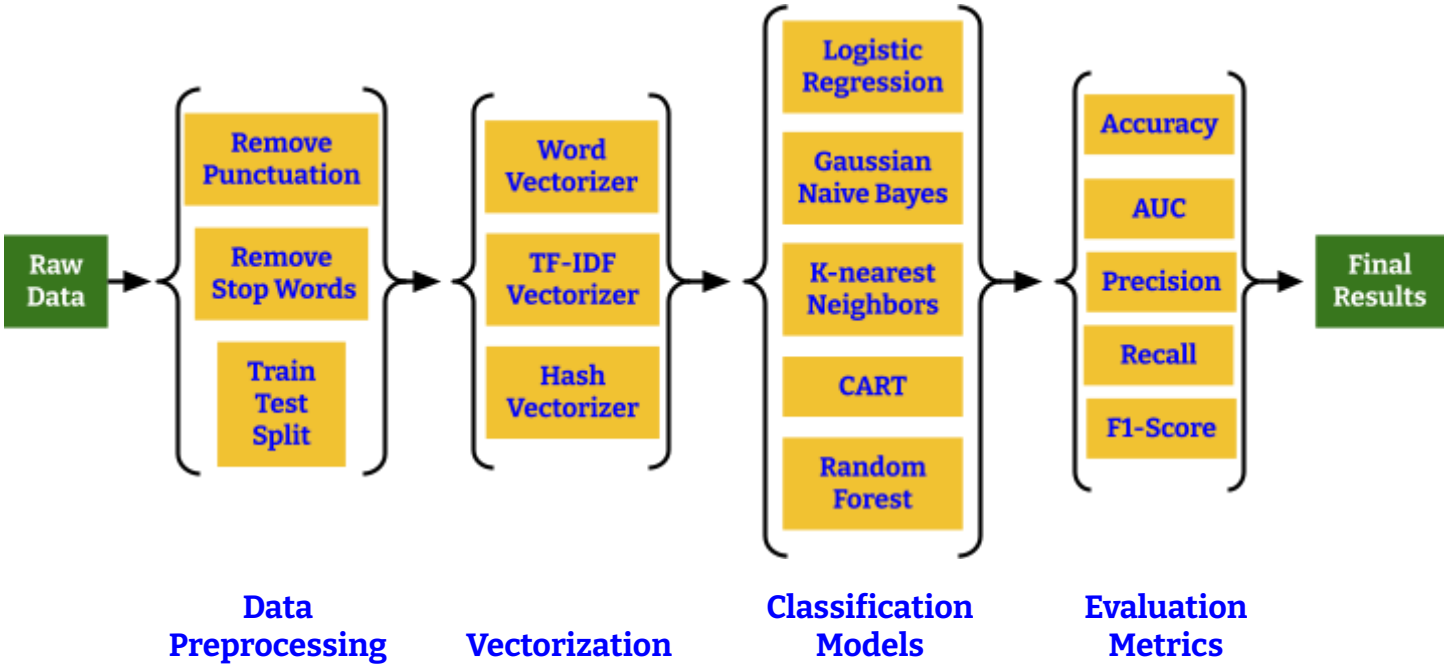
Pirate Metal

Power Metal

Viking Metal

3 Methodology

The goal of my project is to see if a machine learning algorithm can be trained to classify metal songs based solely on their lyrics. First, I will process the song lyrics to prepare them for classification. Next, I will split the lyrics into training and testing sets. After that, I will vectorize the lyrics so that they can be run through classification models. Finally, I will run the vectorized lyrics through several classification models to see which one is superior. The metrics that I will use for evaluation will be classification accuracy,precision, recall, f1-score, and AUC. The process is summarized in the flowchart below.



4 Data Collection and Preprocessing

4.1 Data Collection

I obtained the lyrics for each song from a simple Google search, and I pasted them into a .csv file along with the song title, band name, and album title. Within the .csv file, I removed the punctuation and line spaces. I obtained a sample size of 30 songs for each of the four genres, for a total of 120 songs.

4.2 Data Processing

In Python, I replaced capital letters with lowercase letters. Then, I removed the stop words in order to simplify the lyrics. Stop words are common English words that are not predictive of the category of a song. I used 179 stop words in total.

4.3 Train-Test-Split

The song lyrics were then split into training and testing sets with 30% of the lyrics going into the testing set. This was done so that I could create and model with the training sets, and test the quality of the model on the testing sets. Each time that the train-test split occurred, the data were shuffled in order to give a truer representation of the accuracy of the model. Without shuffling, results could be skewed high or low

based on the particular songs in the training or testing sets. Along with shuffling, I stratified the training and testing sets to have the same number of songs from each genre. The training sets contained 21 songs from each genre, and the testing set contained 9 songs from each genre.

4.4 Vectorization

Since computers cannot extract meaning from words and sentences, the lyrics need to be converted into a numerical structure. For this project, I used vector space models or Bag-of-Words models to convert the song lyrics. The main advantage of this approach is that it is fairly straightforward to implement. The method to create a bag-of-words model has three steps.

1. **Tokenization -** Tokenization is the process of breaking text into smaller parts such as characters or words. I used word tokenization rather than character tokenization for my project since my goal is to build a model based on the lyrics of songs.
2. **Weight Assignment -** In weight assignment, a weight is given to each token based on how often it occurs in the document.
3. **Matrix Creation -** Finally, a matrix is created with the rows depicting the document and a column for each token.

I used three vectorization techniques to process my song lyrics.

1. **Count Vectorizer -** A count vectorizer is quite simple. It counts how many times a token shows up in a document. The count of each token is its weight.
2. **TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer -** This vectorizer is also dependent on the count, but it also takes into account how often each term appears across the entire dataset. The weights are determined by the formula below. TF-IDF is computed by multiplying the term frequency (TF) term by the inverse document frequency (IDF) term.

$$W_{x,y} = tf_{x,y} * \log\left(\frac{N}{df_x}\right)$$

$W_{x,y}$ = TFIDF

$tf_{x,y}$: TF (the frequency of each term, t)

$\log(N/df_x)$: IDF (measure of how much information a word provides)

df_x: number of documents where term t appears

N: total number of documents

3. **Hash Vectorizer -** This vectorizer stores tokens as numerical indexes using feature hashing. Feature hashing stores the tokens as numerical indices rather than strings. The advantage of this technique is that it is memory efficient. The downside is that the names of the features cannot be extracted after vectorization has occurred.

The vectorizers convert the song lyrics into sparse matrices. These bags-of-words can now be used as feature vectors in machine learning models.

4.5 Classification Methods

I decided to use several machine learning models to find the one that works the best. I chose to use logistic regression, Gaussian Naive Bayes, K-means Neighbors, Decision Trees (CART), and Random Forest. Since these were explained in great detail during class, I won't go into detail about these different methods.

5 Evaluation and Final Results

5.1 Evaluation Methods

I will evaluate the quality of the models based on five metrics.

1. **Accuracy -** Accuracy is the ability to correctly differentiate correct classification from incorrect classification. (TP + TN) / (TP + TN + FP + FN)
2. **AUC (Area Under the ROC Curve) -** This is a measure of the area under the ROC curve (FP rate vs. TP rate). It is expressed as a decimal between 0 and 1.
3. **Precision -** Precision is the accuracy of predictions that are positive. TP/(TP + FP)
4. **Recall -** Recall is the fraction of the positive predictions that were accurately classified. TP/(TP + FN)
5. **F1-Score -** F1-Score is the percentage of positive predictions that were classified correctly. F1-Score = 2 * (Recall * Precision) / (Recall + Precision)

5.2 Preliminary Results

	Logistic Regression	Gaussian Naive Bayes	K-nearest Neighbors	CART	Random Forest
Count Vectorizer Accuracy	0.722	0.622	0.317	0.594	0.689
Count Vectorizer AUC	0.928	0.873	0.658	0.729	0.911
Count Vectorizer Precision	0.758	0.693	0.574	0.620	0.734
Count Vectorizer Recall	0.7222	0.622	0.360	0.596	0.688
Count Vectorizer F1-Score	0.710	0.600	0.252	0.594	0.678
TF-IDF Accuracy	0.789	0.661	0.744	0.567	0.733
TF-IDF AUC	0.935	0.831	0.905	0.711	0.912
TF-IDF Precision	0.790	0.678	0.752	0.582	0.752
TF-IDF Recall	0.788	0.660	0.744	0.556	0.734
TF-IDF F1-Score	0.784	0.656	0.742	0.562	0.726
Hash Vectorizer Accuracy	0.783	0.650	0.744	0.600	0.655
Hash Vectorizer AUC	0.925	0.847	0.909	0.733	0.892
Hash Vectorizer Precision	0.786	0.676	0.752	0.640	0.704
Hash Vectorizer Recall	0.784	0.650	0.744	0.600	0.656
Hash Vectorizer F1-Score	0.778	0.638	0.740	0.610	0.644

Above is a chart showing the results from each of the three vectorization techniques and the five machine learning algorithms for a total of fifteen models. Due to the variability of the test results, I ran each model five times and averaged the results above. Each run of five used the exact same training and testing sets.

First, it should be noted that the accuracy, recall, and F1-Score are very similar. This is due to the stratification of the data. Among the machine learning algorithms, CART consistently performed the poorest, while logistic regression performed the best. Among the vectorization techniques, TF-IDF consistently performed the best while count vectorizer consistently performed the worst. K-nearest neighbors performed well with TF-IDF and Hash Vectorizer, but performed very poorly with count vectorizer. I troubleshooted and tried to find out the reason why, but I was unsuccessful in improving the performance of the model.

According to my tests, the best model for classification is logistic regression with a TF-IDF vectorizer. It performed the best in every single metric.

5.3 Final Results

Now, back to the main question. Can heavy metal songs be classified into genres based solely on the lyrics? I ran the logistic regression model with a TF-IDF vectorizer ten times and averaged the results from the classification report, AUC, and confusion matrix.

	Precision	Recall	F1-score	Support
Death Metal	0.874	0.912	0.882	9
Pirate Metal	0.895	0.847	0.866	9
Power Metal	0.663	0.714	0.685	9
Viking Metal	0.733	0.647	0.647	9
Accuracy			0.778	36
Average	0.788	0.778	0.777	36

AUC	0.911

The overall averaged testing accuracy was 0.778. These are fairly good results, but I don't think that they would be good enough for a music service like Spotify to use. In order to more accurately classify metal into genres, other techniques would need to be employed. Across genres, pirate metal had the highest precision. Death metal had the highest recall and F1-score.

		True Classification			
Prediction		Death Metal	Pirate Metal	Power Metal	Viking Metal
	Death Metal	8.20	0	0.40	0.40
	Pirate Metal	0.10	7.60	0.90	0.40
	Power Metal	0.70	0.40	6.40	1.50
	Viking Metal	0.60	0.50	2.10	5.80

The averaged confusion matrix showed that death metal had the most correct classifications on average while Viking metal had the fewest correct classifications on average. This shows that in the lyrics that I chose, death metal lyrics are the most distinct while Viking metal lyrics are the least distinct. Since power metal and Viking metal both have similar lyrics rooted in hero stories and times long ago, it makes sense that a computer would have more trouble correctly classifying them.

6 Conclusion

6.1 Further Analysis

I am pleased with the results from my analysis, but it could be made better with some improvements. First, a larger dataset could be used to diversify the lyrical content of the feature vectors. Second, different techniques could be used such as deep neural networks, convolutional neural networks, and recurrent neural networks. Finally, bigrams and trigrams could be used instead of just unigrams.

6.2 Final Thoughts

It is clear from my analysis that lyrics alone are not enough to classify music into genres. Adding more features would improve predictive power. An analysis that included band name, album title, song title, lyrics, and music (processed with a library like Python's librosa) would almost assuredly have better predictive power.

While looking through Soundcloud, Spotify, and YouTube Music, I noticed that none of these streaming music services classify metal into genres. Over the years, I've learned that metal fans can be quite particular about the genre of metal that they prefer. I believe that it is possible, through machine learning, to build a model that accurately classifies metal songs into genres. Maybe one of these gigantic streaming music companies will use one some day.

7 Images

[1] https://en.wikipedia.org/wiki/Death_metal
[2] https://en.wikipedia.org/wiki/Pirate_metal
[3] https://tinyurl.com/2p8cnx8u Photo via Metal Blade Records
[4] http://thyrfing.com/shop.html

8 References

[1] Raghunathan, Divya. "NLP in Python- Vectorizing", https://towardsdatascience.com/nlp-in-python-vectorizing-a2b4fc1a339e

[2] Janakiev, Nikolai. "Practical Text Classification With Python and Keras", https://realpython.com/python-keras-text-classification/

[3] Blanco, José. "Hacking Scikit-Learn's Vectorizers", https://towardsdatascience.com/hacking-scikit-learns-vectorizers-9ef26a7170af

[4] Wikipedia article about Death Metal, https://en.wikipedia.org/wiki/Death_metal

[5] Wikipedia article about Pirate Metal, https://en.wikipedia.org/wiki/Pirate_metal

[6] Wikipedia article about Power Metal, https://en.wikipedia.org/wiki/Power_metal

[7] Wikipedia article about Viking Metal, https://en.wikipedia.org/wiki/Viking_metal

[8] Patrick Galbraith and Nick Grant. https://mapofmetal.com/

[9] Crauwels, Kwinten. https://www.musicmap.info/