NLP Project Report

Hein Huijskes (s2386836)
Tibet Tugay (s2489384)
Group 43
November 2025

1 Abstract

This paper explores the classification of songs according to their genres based only on their lyrics. Research shows that song genre classification soley based on lyrics is not as thoroughly investigated as auditory (and lyrics) based song genre classification. However, what is examined is the effect of sophisticated features and their effect on the performance of the classifier model. Out of all the sophistichated features discussed in the literature, this paper explores the contributions of song length and semantics to the classifier. Moreover, we inspect and review the change in model performance with varied Natural Language Processing techniques both within preprocessing and model parameterization. These techniques include normalization, tokenization, stopword removal, infrequent word removal, and n-grams. The result is a discussion on a model that utilizes a multistage pipeline and its limitations. The final model proves to perform well on the dataset (around 78% accuracy) though proves to be overfit (achieving 69-75% accuracy on validation data).

2 Introduction

This project is aimed to investigate the area of genre classification for songs based on their lyrics only. Even though this NLP task may present itself as a relevant and interesting topic, many of the research in this area revolves around the classification of songs based on audio features, mainly tested on the community standard GTZAN dataset[1]. In contrast to the audio features of songs, this project focuses purely on the lyrics of the songs, even excluding the structural information presented in the lyrics (as discussed in Section 5.1).

The exploration of song classification based on lyrics was conducted through multiple lenses during this project. This paper outlines both the machine learning aspects of the project and the sophisticated features used to boost the machine learning methods. The sophisticated features discussed are inspired by the research presented in Fell et al. [2]. Through experimentation and discussion, this research will attempt to answer the following research question: "Which combination of models, Natural Language Processing techniques, and sophisticated features yield the highest accuracy for classifying music genres based on song lyrics?"

Answering this question is done in the following parts: The Related Work considers the relevant research preceding this project. The Dataset section showcases the dataset and its details. The Methodology elaborates on how the experiments were conducted while the Results section illustrates the findings of this research. Lastly the Discussion will reflect on the process as a whole and conclude with the lessons learned.

3 Related Work

Fell et al. [2] is the main inspiration behind this project. The paper investigates the so called "sophisticated features" of the song lyrics in its dataset to great extent. The authors of the paper have collected their own dataset with 400.000 English songs from a combination of 7.200 artists. The main difference between their dataset and the dataset from this paper is that theirs has 9 major genres while our dataset has 4. Another point of discussion regarding the datasets is that they

[1] https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

have combined the Pop and Rock genres into one class while we have removed the Pop genre all together.

The paper explores the sophisticated features through the following sections: Vocabulary (type-token ratio, non-standard words), Style (POS and chunk tags, length, rhyme, echoisms) Semantics (imagery), Orientation (past tense verb forms, pronouns), and Song Structure (repetitive structure, chorus, song title). However, the methodology of Fell et al. [2] was vague regarding how they have generated these features for the song lyrics. While the paper mentions the main idea behind all of the features, the practicalities of how the actual results were obtained was left out. Though it is known that Fell et al. [2] have used an n-gram model for their model, which is similar to the model explored in this project.

Lastly, Figure 2 from Fell et al. [2] was of real significance for this project. The figure showcases the feature contribution each feature had for their model. This figure was the main criteria while finding the second feature to use alongside Imagery for our "sophisticated feature" investigation. Semantics was already selected as it was the most indicative feature for the "meaning" of the lyrics.

4 Dataset

The dataset used is the song_lyrics dataset[2]. It consists of 3 columns: song lyrics ("lyrics"), genre ("tag"), and song title ("title"). The dataset has 4.39 million entries, including 6 genres: Pop, Rap, Rock, Misc, R&B, and Country. See their distribution in Figure 10.

The dataset was initially selected due to its large size, though was later reduced to 10.000 songs per genre since more songs barely impacted the model. Genres Misc and Pop were also excluded from testing and training. This was done since Pop and Misc are very difficult to distinguish from other genres. This intuitively makes sense, since Pop (meaning "popular") could refer to any of the other genres. The same goes for Misc ("miscellaneous"). A limitation of the resulting dataset is that there are only 4 genres to classify. Although these are less classes than the original paper this project set out to investigate, it still presents a thorough examination of the concepts this project set out to explore.

Testing also showed that training on more than 10.000 songs (e.g. 80.000 or 100.000) did not significantly improve the model (later shown in Figure 8). Most initial tests for the model were done using only 1000 songs, since this followed a similar trend to 10.000 songs, and was much faster to train.

5 Methodology

The general methodology used can be viewed in Figure 1. This pipeline shows the data preparation, with added "Sophisticated features" (RID and Length). It also shows that at the end, results from testing are incorporated to refine and improve the whole process. Since this can lead to overfitting, an extra test with an unused dataset was performed to obtain the accuracy of the final model.[3]

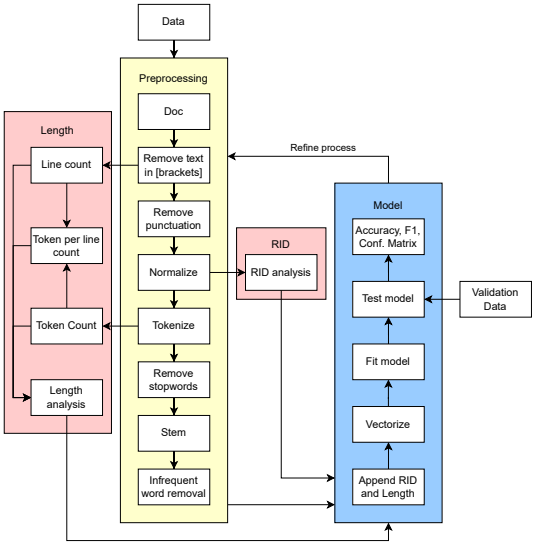

Figure 1: Model Pipeline

5.1 General preprocessing

First, for all documents text in brackets (e.g. "[Chorus]" or "[Kanye West]") is removed. These pieces of text contain additional information about the song that is not purely in

[2]https://huggingface.co/datasets/amishshah/song_lyrics

[3]All code and gathered data can be found at https://github.com/HeinHuijskes/NLP-Project

the lyrics, which is not the goal of this project nor the paper it was based on [2], so it is removed. Then each document is normalized by lowercasing all words. The normalized words are tokenized[4]. Stopwords are removed[4] using a list of English stopwords. Stemming is used to reduce words to their root (this worked marginally better than lemmatizing, see also Figure 11). Finally infrequently occuring words are removed.

At multiple points the pipeline splits, to obtain the necessary information about songs for Length and RID analysis. This is done to prevent further preprocessing from making some words unrecognizable or removing parts of the sophisticated features.

5.1.1 Infrequent word removal

Words that occur infrequently provide little value to classifying a text, while exponentially increasing the feature space of a model. Testing showed that removing all words that occur 10 times or less did not significantly impact model performance (Figure 14) while drastically reducing feature space (Figure 15). This is additionally relevant for RID (5.3.1). Thus a requirement of words occuring at least 10 times across the training set was added to the model.

5.2 Model

Fell and Sporleder use a model called "topK" [2], which is only ever referred to as "an n-gram model". Therefore we opted to instead use a simple Multinomial Naive Bayes classifier[5], although some testing was also done with Logistic Regression[5], which proved to be worse in performance (see also Figure 12).

5.3 Sophisticated Features

The features "Length" and "RID" were chosen since they appear to have a relatively high impact compared to other features (see Figure 14 of the reference paper [2]). Furthermore they seemed feasible to implement, and described clearly enough.

Both features are included in documents by adding a unique string to the end of documents. For instance for a document tagged with the RID category "Emotion", the string "EMOTION" is added to the end of a document a number of times, changing its feature vector.

5.3.1 RID

As discussed in [2], Regressive Imagery Dictionary is a psychological work that classifies a passage into the 3 main Imagery categories. "RID classifies words as belonging to the separate fields 'conceptual thought' (abstract, logical, reality-oriented), 'primordial thought' (associative, concrete, fantasy), and 'emotion'" [2]. Thus, RID allows the model to interpret the most likely semantic category of the text it was given. The model was trained with the addition of the RID flag at the end of the song, allowing a possibility to change the importance of the RID flag compared to the rest of the words in the lyrics of any given song.

A coding scheme[6] was used to implement RID. This coding scheme is a terminal application that given the text, returns the most likely imagery out of the 3 aforementioned imagery categories. Some of the methods of this scheme was repurposed to instead run RID as part of the Model Pipeline as shown in Figure 1. See Figure 2.

5.3.2 Length

Three "Length" features are also discussed in [2], under the Style feature. The features are "lines per song", "tokens per song", and "tokens per line". Further explanation past the names of these features was not provided however. An issue of these features is that even just for a thousand songs we find significant variation in the amounts, making it difficult to capture them in a compact vector size (see Figures 16, 17, 18). For this reason we group length features in **quantiles**, and we optimize the amount of quantiles by comparing performance for a wide range of quantiles for each feature consecutively. For example, grouping lines into 4 quantiles (also called quartiles), we could find the quantile boundaries: $[0, 46, 64, 90, 389]$. If a song now has 42 lines, it falls in quantile 1 (since $0 \leq 42 \leq 46$), which will be added to its feature vector. The same is done for tokens per song, and (average)

[4] Tokenization and stopword list provision, stemming and lemmatizing are all done using NLTK [1]

[5] For both the Multinomial Naive Bayes and Logistic Regression classifier we used the scikit-learn library [3]

[6] https://pypi.org/project/regressive-imagery-dictionary/#files

tokens per line per song. The quantiles are only tested starting at 2 quantiles and above, since 0 or 1 quantiles grant no new information about the data.

5.4 N-grams

Various N-grams were used in the training of the classification model. The expectation was to see an increase in the performance metrics with the usage of n-grams as they would supply more context for the model throughout the lyrics. However, the initial training data showed a decline on the accuracy and f-score as the n increased in the n-grams used. Therefore, this decline was examined thoroughly by training the model with mixed n-grams, allowing this experiment to display the various combinations of n-grams to induce context for the model. This unfortunately still showed a consistent decrease in performance (see also Figure 13).

6 Results

Most of the testing and hyperparameter tuning done for this research is not relevant to its main goal and comparison to the reference paper. Therefore only results of the "Sophisticated features" and the final model are shown here. The used performance metrics are accuracy, F1 score, and for the final model a confusion matrix.

6.1 Sophisticated Features

The sophisticated features are incorporated into the feature vector of each document by extending a document at the end of preprocessing, and including (repeated) words corresponding to the value of the added feature. Because of this, the size of the feature space could decrease the impact of the added features. For this reason different tests were run for "cutoff" rates, where a cutoff of 20 means that any word occuring 20 or less times across the whole corpus is removed from the vocabulary (see Figure 14 for the impact of removing up to 50 words). Results were obtained for the effect of different cutoff amounts on the performance of sophisticated feature RID.

6.1.1 RID

Figure 2 shows the results of tuning RID, for different amounts of cutoff. We can see that cutoff 20 consistently gives the best result (which seems in line with the results in Figure 14). There is however not a lot of consistency in improvement for repetitions of RID. Here repetitions mean that the value for RID is simply repeatedly added to documents, to increase its impact. If we focus only on cutoff 20, then more repetitions do seem to increase accuracy, peaking at 19 repetitions. We take this as our tuned hyperparameter, though with some caution since the result does not seem generally conclusive.

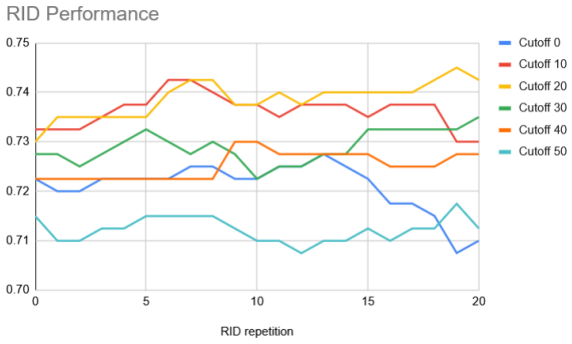

Figure 2: RID Accuracy With Different Cutoff

6.1.2 Length

Figure 3 shows the impact of Line Count on performance for different quantiles. Here we can see an interesting pattern emerge: the performance seems to peridically go up and down as the amount of quantiles increases, until it drops off. This could indicate that the quantiles are having a significant impact, and that pushing some songs from one quantile to another by shifting the amount impacts the performance. We take the peak at 10 quantiles as the optimum and continue.

Figure 4 shows the impact of Token Count on performance for different quantiles. The periodic movement seems less clear here, instead we see a clear peak at 18 quantiles, which we take as our optimum.

Figure 5 shows the impact of Token per Line Count on performance for different quantiles. Dividing the data in 2 quantiles has the highest performance, and after 5 quantiles the performance drops and stays around 74%. 2 quantiles is our optimum.

Figure 6 shows the impact of a different number of repeats of the length feature on performance. The 3 Length features were arbi-

trarily tested at 20 repeats, which is the RID optimum. Instead we now try to find the optimum for the Length feature, which we take at 16 repeats. Here we can also clearly see that more repeats do increase the performance of the model, suggesting Length has a positive impact on the classification.

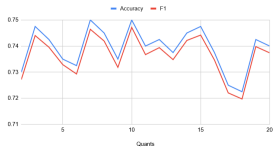

Figure 3: Line Count Performance With Different Quantiles

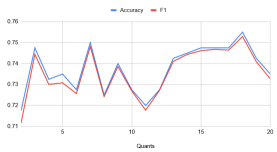

Figure 4: Token Count Performance With Different Quantiles

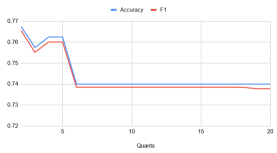

Figure 5: Token Per Line Count Performance With Different Quantiles

Figure 6: Performance For Different Repeats Of Length Feature

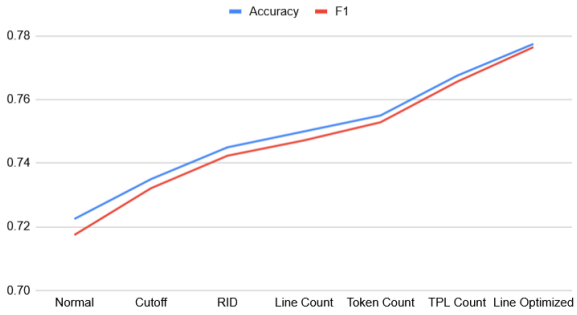

Figure 7: Model Performance With Each Improvement

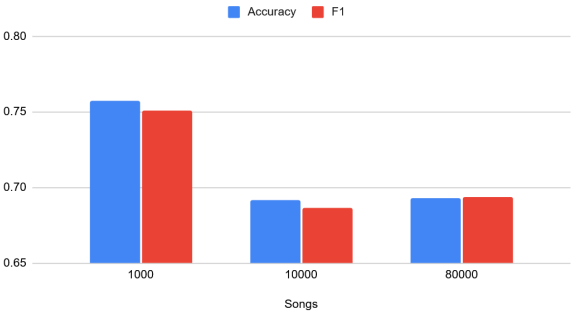

Figure 8: Results Of Final Model On Validation Data

6.2 Final Model

Following each step that was taken to improve the model, we can see in Figure 7 that the model consistently improved after each step. This does however pose a significant risk of overfitting, especially since the intermittent test results are based on a model using only 1000 songs (Note that for all listed models 10% of the data is used as validation, so 100 test songs and 900 training songs here). Indeed, performing validation on different songs, we can see that the performance drops to about 75.8% for 1000 songs and much more for 10.000 (69.2%) and 80.000 songs (69.3%) (shown in Figure 8). Looking at the confusion matrices (Figures 9, 19, 20) we can see that Rap and Country are consistently identified correctly, while Rock and R&B are predicted less consistently, often being identified as country instead.

7 Discussion & Conclusion

7.1 General performance

The final model clearly seems overfit. The found parameters work quite well in testing, increasing the accuracy up to 77.8%, but testing on different and larger data immediately plummits the performance. Still, the model is decent at classification, and looking at the confusion matrices we can see that a lot of classification is going well.

Rap and Country seem to fare well in classification. For Rap this intuitively makes sense, since it contains a lot of (slang) words that are unique to rap only (see also Figure 4 of [2]). Country instead seems to perform well partially because the model is biased towards it, since both R&B and Rock have a tendency to be classified as Country instead.

7.2 Sophisticated Features

7.2.1 RID

RID seems to have had a limited final impact. The performance only increases marginally, and testing shows that the improvement is highly inconsistent across different settings of cutoff.

7.2.2 Length

The Length feature showed more impact than expected. Combining all three length subfeatures we saw the most clear improvement

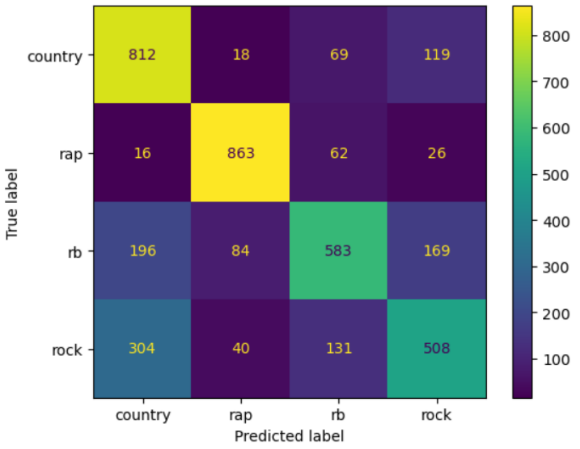

Figure 9: Confusion Matrix For Model Trained On 9000 Songs

of performance. This is similar to the reference paper, where Length was also the most impactful sophisticated feature.

7.3 Limitations & Future Work

7.3.1 More Sophisticated Features

Only two of the Sophisticated Features listed in the original paper were implemented here. In future research, more of them could be implemented for potential increase in performance and generality of the model.

7.3.2 RID Subcategories

Only the main three categories for RID (Emotions, Primary, Secondary) were used in this project. Instead their many subcategories could also be considered and added, which could improve performance. Similarly only the highest present category is currently being added, meaning that if Emotion has 54% and Primary 46%, Primary is not considered. This could be changed by including all categories instead of just the highest one, for instance using the bucket/quantile approach used for the Length features.

7.3.3 Data quality

Another possible limitation is the lack of information regarding the subgenre or release year of the song in the dataset. While this information is not necessary for lyrics classification and would not even be fed to the model if they were present, they do present patterns the model could learn from, which would result in a slightly more overfit model. This would be less of an issue if the whole of the dataset or even the 100.000 entries per class dataset was used. However, with the current 10.000 songs per class setup and considering the first 10.000 from each genre was fed into the model instead of a random 10.000 chunk, this could present some overfitting.

7.4 Ethical Implications

In the case of the deployment of the model constructed throughout this project, the rise of an ethical concern is unlikely. However, there are still biases to discuss. Since the model is trained with real world data, the usage of slang or curse words increase the likelihood of the model predicting the song lyrics as rap. This is rather correct behavior for the model since even the word clouds shown in Figure 4 of Fell et al. [2] show many of these slang or derogotary words. Regardless, this still introduces a bias in the model towards the rap genre or any song belonging to it. Other than the bias presented here, there is no inherent ethical implication regarding the model presented in this paper.

References

[1] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[2] Michael Fell and Caroline Sporleder. 2014. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 620–631.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

8 Appendix

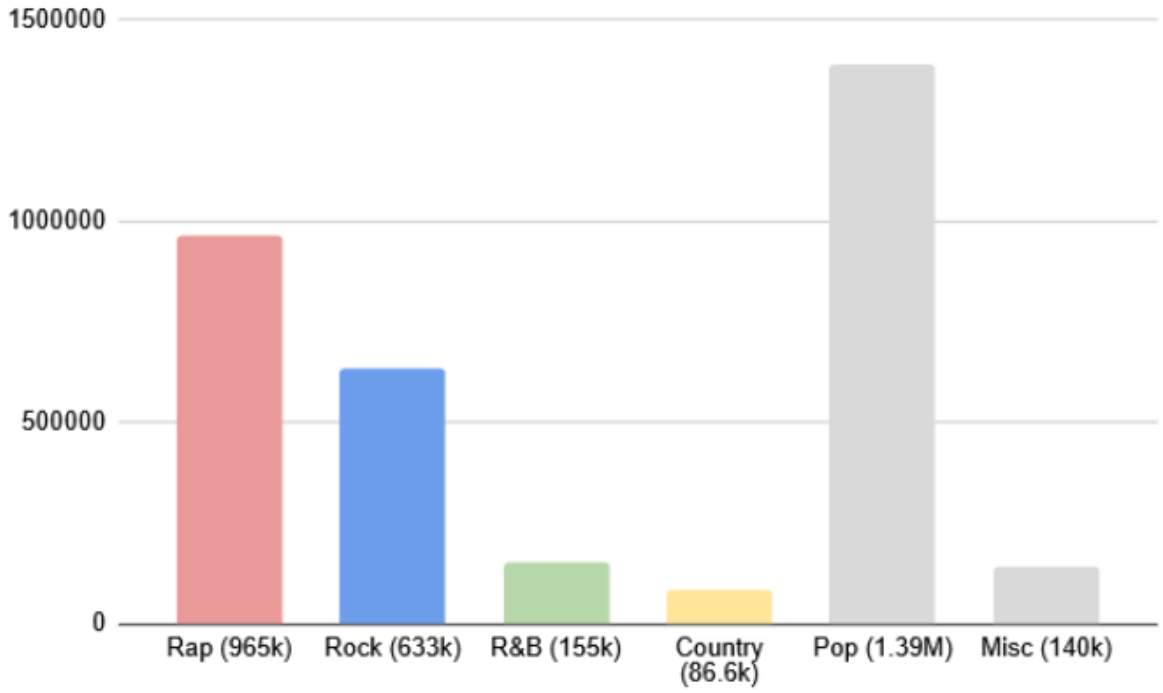

Figure 10: Song Distribution In The Dataset

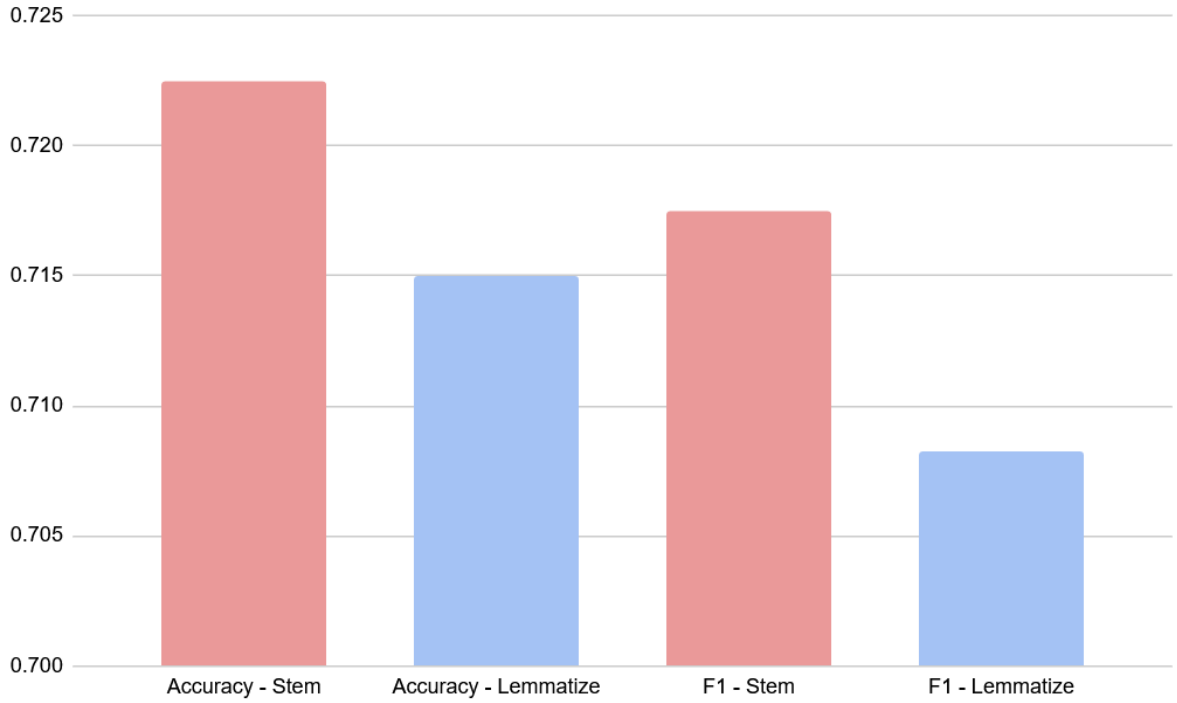

Figure 11: Stemming Versus Lemmatizing

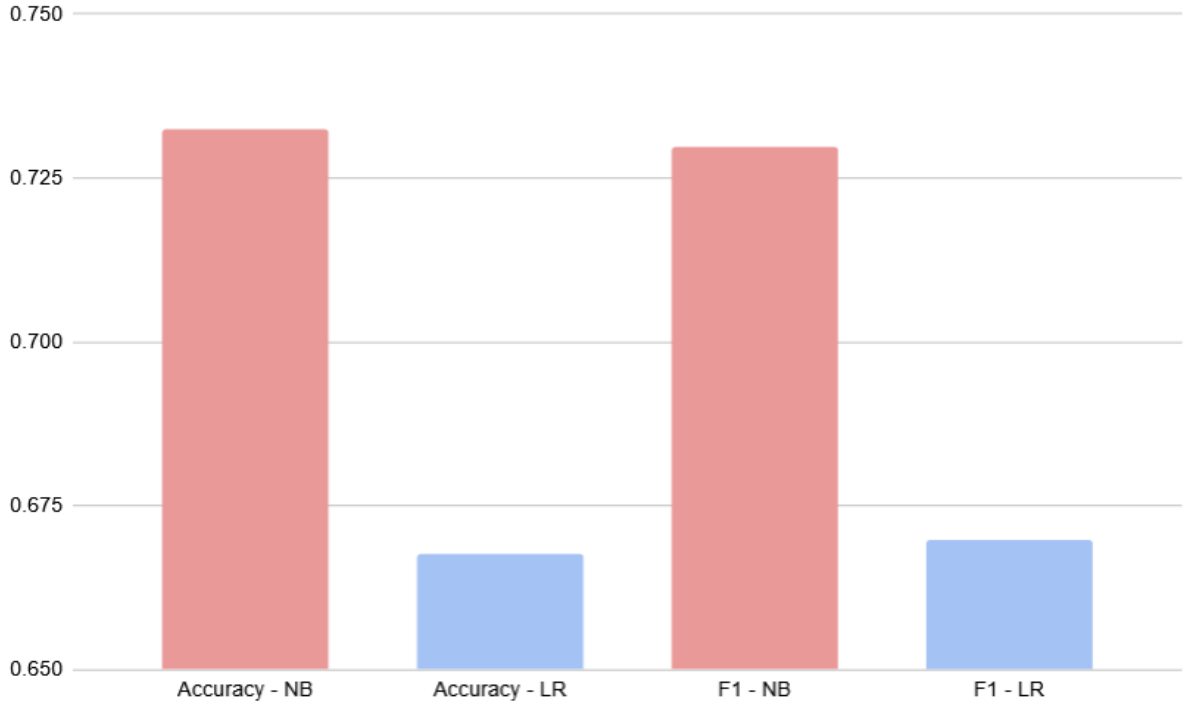

Figure 12: Multinomial Naive Bayes (NB) Versus Logistic Regression (LR)

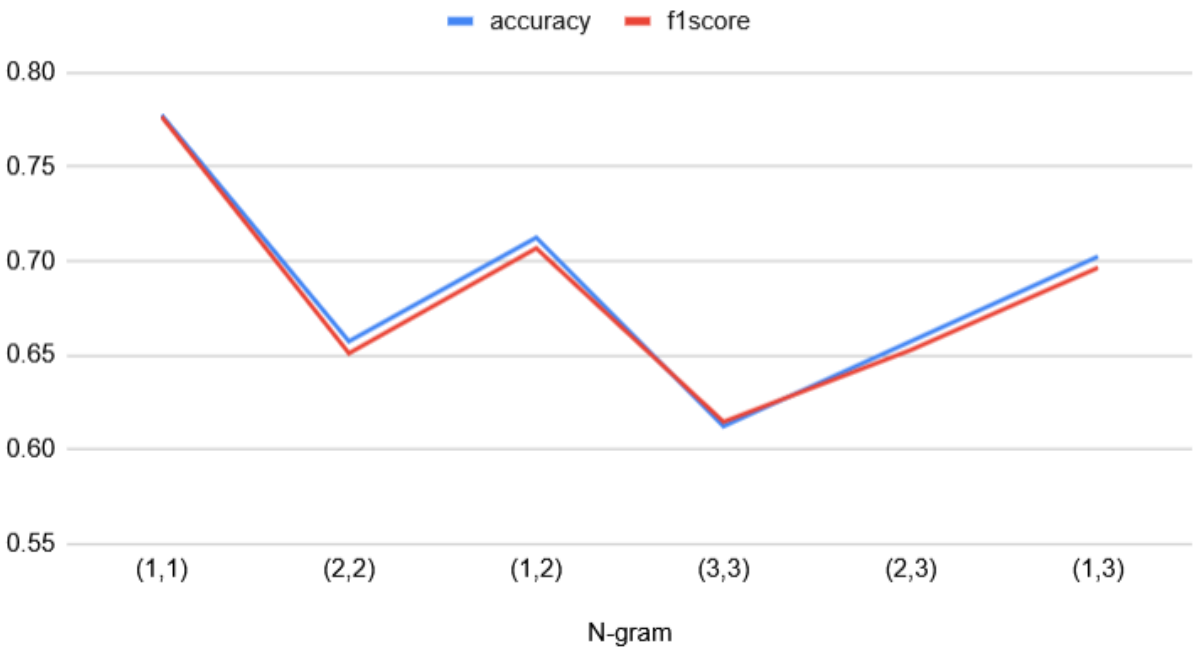

Figure 13: Performance For Different N-grams

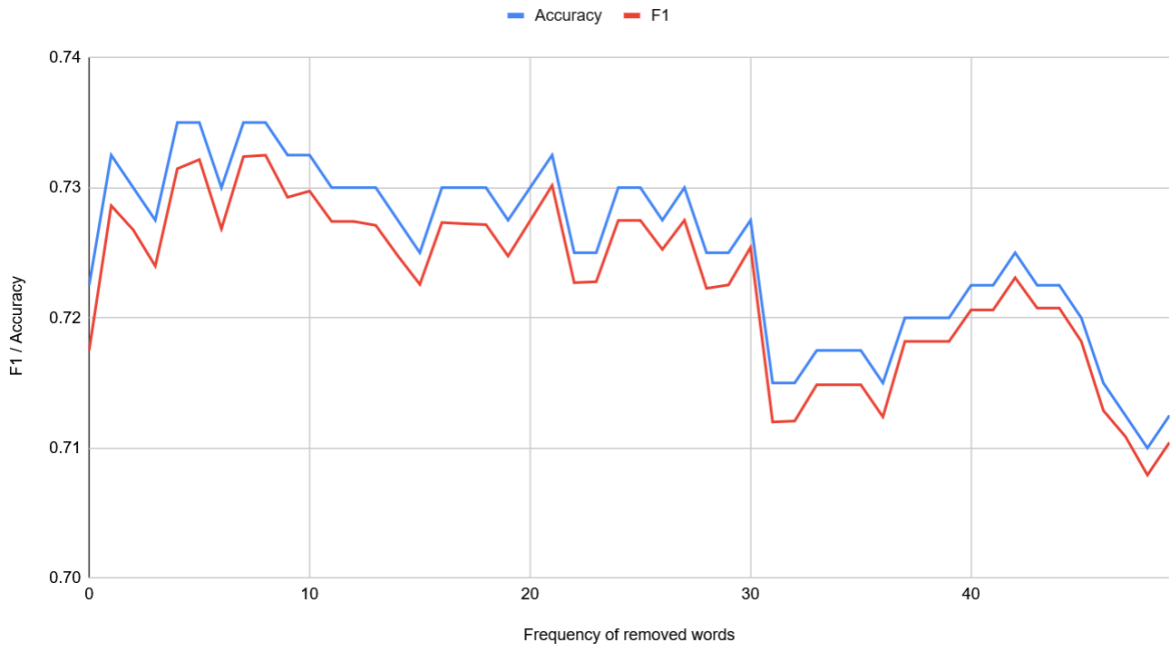

Figure 14: Infrequent Word Removal (1000 Words Per Genre)

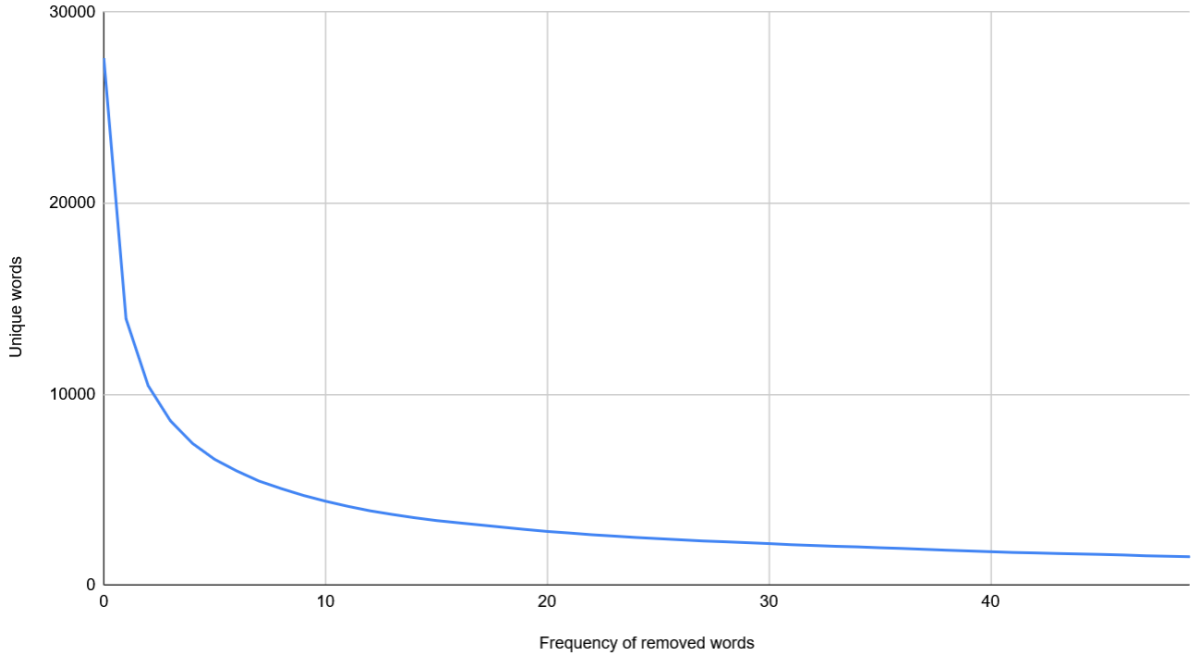

Figure 15: Feature Space (1000 Words Per Genre)

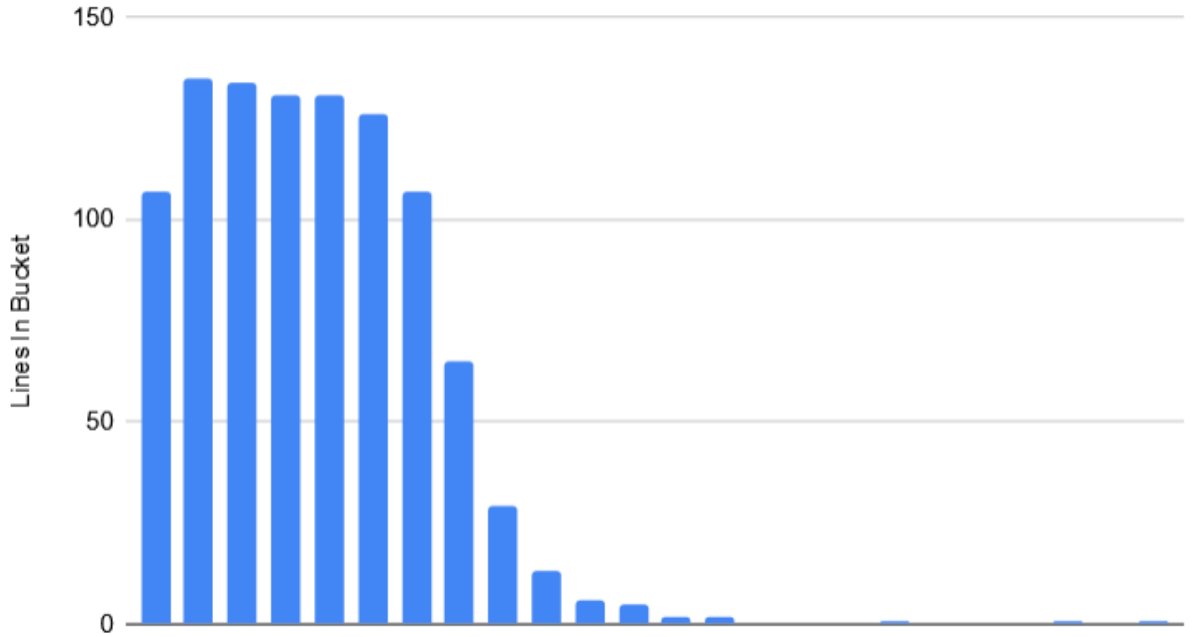

Figure 16: Histogram Of Lines Per Song (25 Buckets, 1000 Words Per Genre)

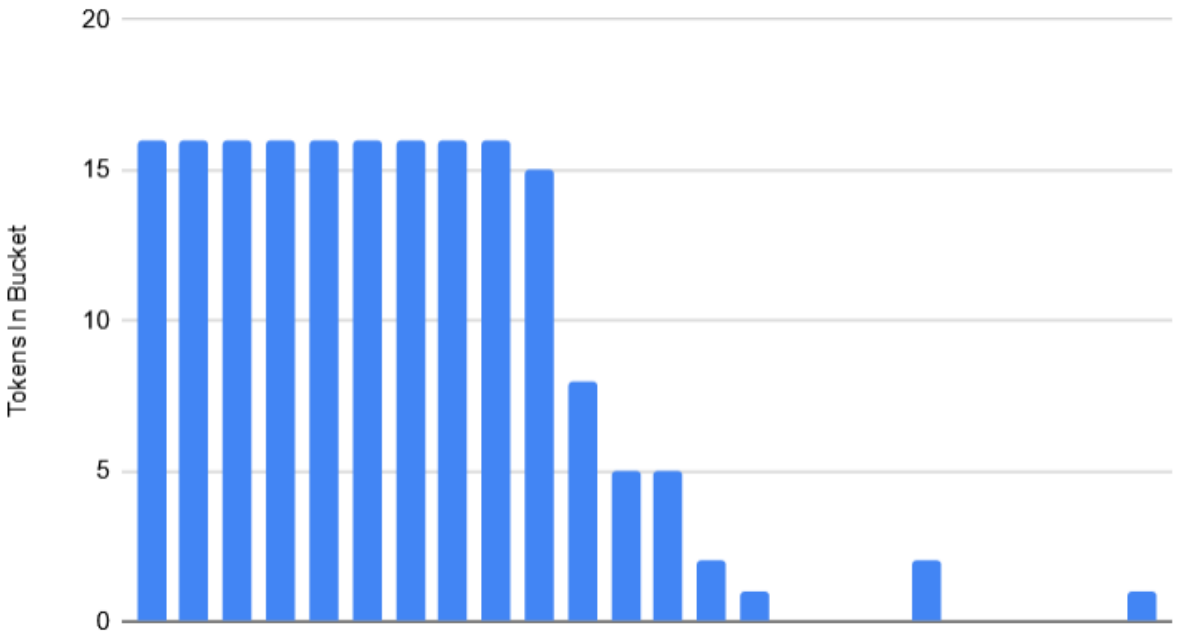

Figure 17: Histogram Of Tokens Per Song (25 Buckets, 1000 Words Per Genre)

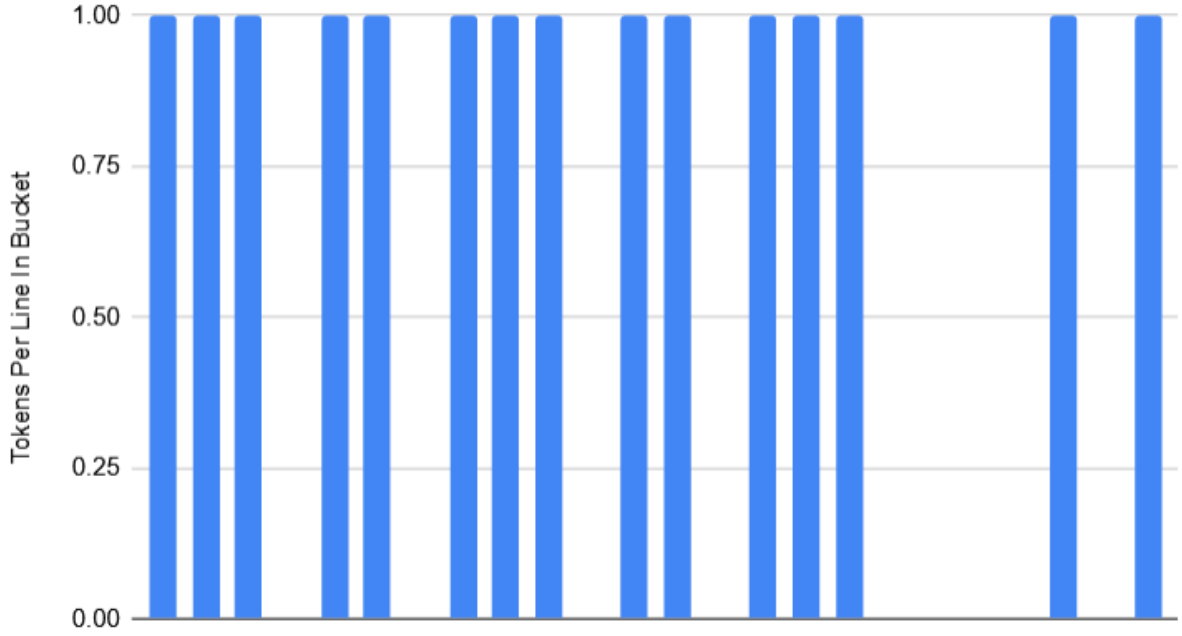

Figure 18: Histogram Of Tokens Per Line (25 Buckets, 1000 Words Per Genre)

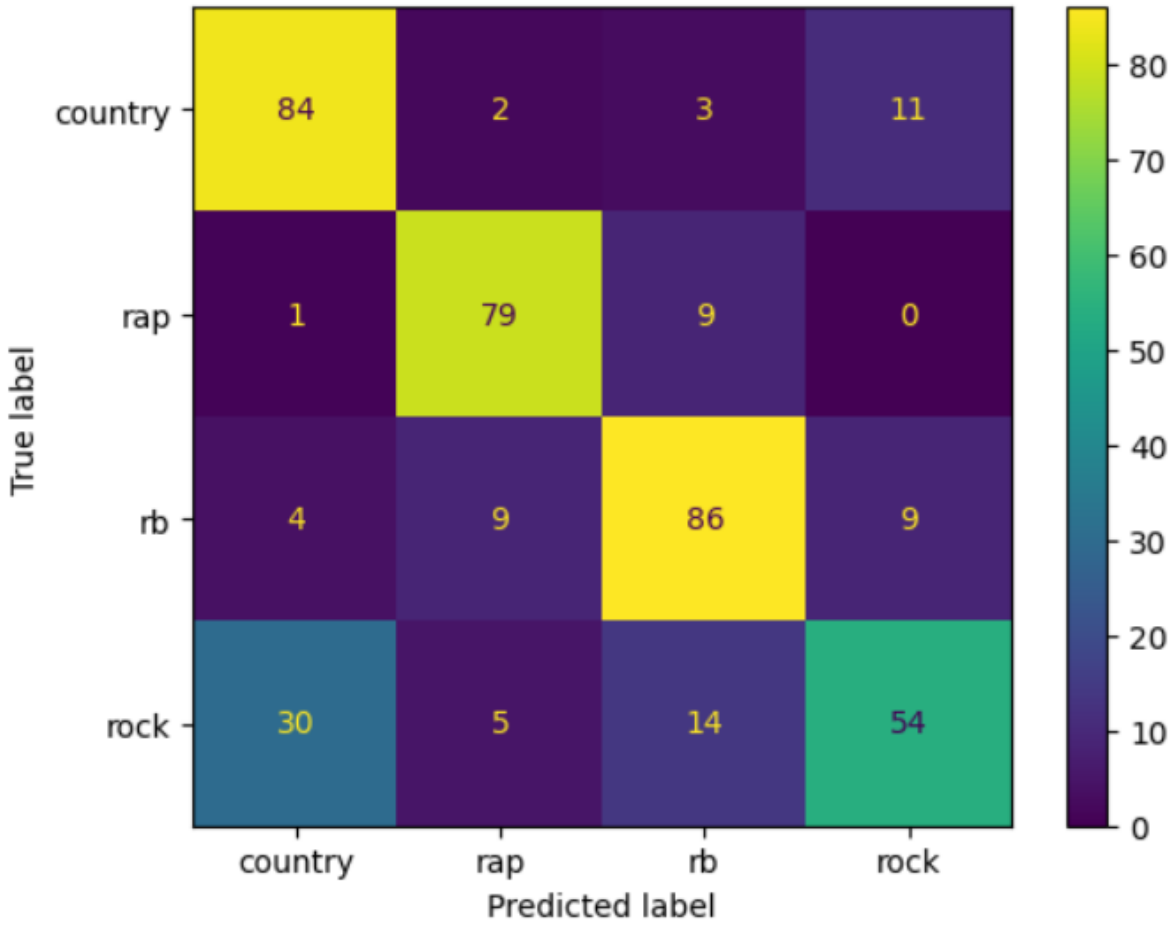

Figure 19: Confusion Matrix For Model Trained On 900 Songs

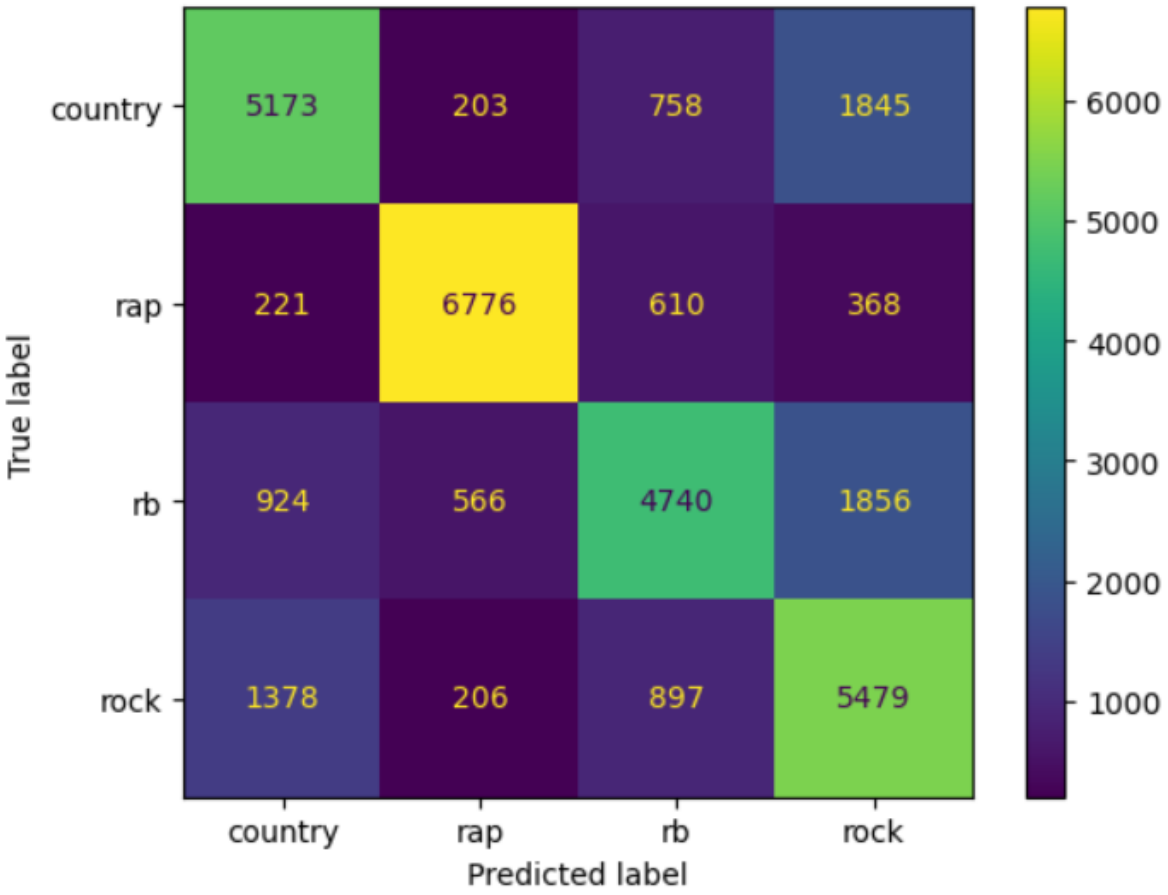

Figure 20: Confusion Matrix For Model Trained On 72000 Songs