Predicting the Title, Artist and Genre of a Song

Boris Borisov - i6211839
Joan Botzev - i6219289
Bas Göritzer - i6225799
Stela Topalova - i6208317
Kristian van Kuijk - i6214757

Abstract

Song's lyrics are known to be the best approach for artists to express their emotions (Swaminathan and Schellenberg, 2015). This paper examines a personalised and adapted approach to analysing and classifying lyrics, to an artist's name and a musical genre, independently from one another, and present in the datasets. We also propose a model to predict a song title that suits the given lyrics. We find that despite the small size of the datasets, the different prediction model manage to achieve great results for certain artists, genre and predicting title, while some models fail poorly on specific artists. This paper also gives insight to the lyrics difference between different music genres, specially between nowadays the two most streamed one worldwide: pop and hip-hop.

1 Introduction

Music is everywhere. People are listening to it in their home, at their office, on the street while taking a walk (Najafov, 2019). In the field of Natural Language Processing, a text can be generated and then specific values can be predicted and compared how accurate those values are. Our general purpose of this project is to perform an analysis, different prediction models and other techniques that seem to be appropriate for this task. This paper will concentrate on three topics: Artist, Title and Genre of a specific song. Different language models, preprocessing operations, machine learning models are covered further in this paper.

2 Related Work

A lot of research has already been done in text classification, regarding genre or emotion classification among many others. Machine learning model include Naive Bayes, SVM, Random Forest or Logistic Regression on text or sometimes audio. More recently, deep learning techniques have been investigated covering CNNs and LSTMs (Wang et al., 2016) (Mathur et al., 2018). Furthermore, many approaches were shown to improve emotion classification performance on lyrics, such as feature extraction (Yang et al., 2008). Moreover, many procedures have promising results, including multi-modal approach to emotion prediction, where (Strapparava et al., 2012) introduced a novel corpus of both music and lyrics.

3 Datasets

Finding the datasets Having a valuable dataset is essential. We wanted one that was in a format such that we could use the pandas library (in python) to have meaningful insights, and that we could use to model our three predictions. Unfortunately, we did not manage to find one to match all of our criteria. Nevertheless, we used the following one for the Title and Artist prediction: Song Lyrics Dataset: A dataset containing song lyrics of various artists, and Song Genre Classification Data Set for the genre prediction.

Insights From the first dataset (Title / Artist) we retrieved 5315 songs from 20 different artists (as shown in table: 1, where the year column highlights the year in which the artist released the most songs presented in the dataset. The genre was judged by the authors of this paper.).

In this first dataset, we notice two different groups: pop and hip-hop artists. An extensive investigation on the two different groups showed that hip-hop songs are longer (They have on average 574.81 words compared to 328.78 words for pop songs. Hip-hop songs in the dataset have 75% more words than pop songs).
Besides, the most frequent words are very different between the two genres. As shown in the word cloud 1, the most used words for writing a pop

Table 1: Dataset 1 - Summary

Artist	Genre	N°of Songs	Year
Ariana Grande	Pop	308	2018
Beyoncé	Pop	406	2019
Billie Eilish	Pop	145	2017
Cardi B	Pop	75	2016
Charlie Puth	Pop	75	2018
Justin Bieber	Pop	347	2020
Dua Lipa	Pop	246	2020
Ed Sheeran	Pop	294	2011
Drake	Hip-Hop	464	2009
Rihanna	Pop	397	2011
Maroon 5	Pop	197	2017
Post Malone	Pop	147	2016
Katy Perry	Pop	324	2020
Coldplay	Pop	333	2018
Taylor Swift	Pop	477	2020
Khalid	Pop	64	2019
Selena Gomez	Pop	174	2013
Eminem	Hip-Hop	521	2020
Nicki Minaj	Pop	321	2012

songs are related to sentiments ("love", "baby") or are repetitive syllable that can easily create a melody that the listener can remember ("oh oh"). However, for Hip-Hop songs, artists use more often rude words, or even tell their own name (as shown in the word cloud 2).

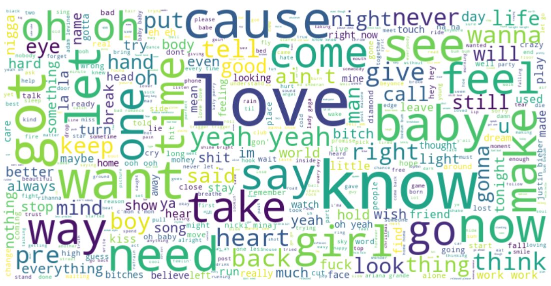
Figure 1: Word cloud from Pop Lyrics

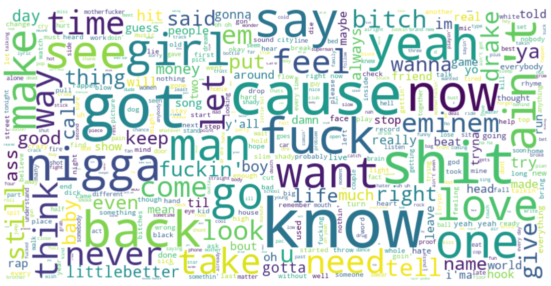
Figure 2: Word cloud from Hip-Hop Lyrics

For the Genre prediction, a different dataset was used. It contains columns such as Index, Song, Year, Artist, Genre and Lyrics. Some of the columns will be presented in a table.

Table 2: Dataset 2 - Summary

Song	Artist	Genre
Ego-remix	Beyonce	Pop
Mightier-than-sword	Borialis	Rock
Honesty	Beyonce	Pop
Love-el-amor	Etin	Other
Love in the afternoon	Gene-watson	Country
Diary-of-werrwoulf	Fang	Rock
Floating-pebbles	Chant	Other
Atmosphere	Funkadelic	Rock
Final	Children-18-3	Rock
Waiting	Taubenfeld	Rock

Furthermore, the table consists of more than 350.000 songs with different artist and genres. The best way to show the data is to visualize it. Two Word Clouds have been created such that you can explore them. The first one is about the frequency of words in lyrics. (See Figure 3). A lot of lyrics

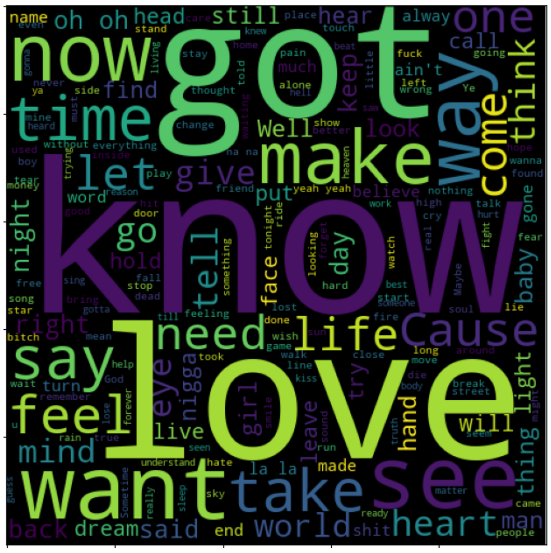
Figure 3: Word cloud for most frequent words in lyrics

contained words such as "love", "know", "heart" from which we can conclude that the main theme in these lyrics is the love and love feelings.

The second one is about frequency of genres. (See Figure 4). The conclusion is that multiple genres such as Rock, Pop, Hip Hop, Metal and Country are some of the main genres in this data set.

Country Country
Pop Pop
Rock Rock
Metal Metal
Electronic Electronic
Hop Hip
Jazz Jazz
Folk Folk
Hip Hop
Indie Indie

Figure 4: Word cloud for most frequent genres

4 Model implementation

4.1 Title Model

Preprocessing For preprocessing of the titles couple of methods are used. First lowercasing is applied and punctuation is removed. Also the data contains remixes and descriptions of titles, which are also being removed by searching for characters like ”/” or ”( )”. Finally ”u” in the titles is being changed to ”you”, as most lyrics contain the full word as opposed to the title.

Machine Learning Model The main model to predict a title only by the lyrics n-gram. The length of the n-gram is created according to a training set, which finds what is the average length title that an artist uses. As we can see in table 3 the average lengths of titles varies from one to three, with two being the most common. To determine the predicted n-grams lengths, those averages are used and a margin of two was added to make it possible to find different sizes of titles. Also there are the percentages for the chance of a title to be in the lyrics, which shows the maximum possible accuracy that can be reached.

Table 3: Title statistics related to each artist

Artist	Av.Title Size	Title in lyrics %
Ariana Grande	2	0.67%
Beyoncé	2	0.74%
Billie Eilish	3	0.75%
Cardi B	2	0.75%
Charlie Puth	3	0.76%
Coldplay	3	0.56%
Drake	2	0.69%
Dua Lipa	3	0.73%
Ed Sheeran	3	0.71%
Eminem	3	0.56%
Justin Bieber	2	0.68%
Katy Perry	2	0.64%
Khalid	1	0.68%
Lady Gaga	2	0.80%
Maroon 5	3	0.84%
Nicki Minaj	2	0.63%
Post Malone	2	0.48%
Rihanna	2	0.78%
Selena Gomez	3	0.78%
Taylor Swift	3	0.67%

4.2 Artist Model

Preprocessing After removing the punctuation (without any libraries) and transforming all the lyrics to lowercase, we create a n-gram matrix of token counts from our training data (the training data contains 80% from the dataset, the test data the remaining 20%) based on either words, or characters using the *CountVectorizer* from the scikit-learn library. Then this *CountVectorizer* is applied to the training data and test data. The next step is to encode our labels, transforming our categorical data to integers.

Machine Learning Model Finally, we feed our training and test data and labels to a machine learning model (*Naive Bayes, Random Forest, and Logistic Regression* are compared in section 5). *Naive Bayes*, for example was chosen as our first baseline model due to its widespread applications in text classification. *Random forest* is investigated as it is known to remove biased (our dataset is not equilibrated as we have more pop songs than hip-hop songs, while we have proven that those two have very different lyrics in section 3).

4.3 Genre Model

Preprocessing For the preprocessing, it was decided that we should mainly focus on the lyrics of every song. The process includes removing rows with duplicate lyrics, removing non-alphabetic characters, lowercasing every word, tokenization and removing stop words. Due to time constraints and efficiency of the algorithm, it has been decided that these steps will be enough for generating the most optimal results. Further possible improvements in the algorithm might involve preprocessing operations such as removing non-English words, stemming/lemmatisation and checking the spelling of every word.

Machine Learning Model Firstly, a preprocessing and formatting of the dataset was performed. The unnecessary columns were removed from the data during the formatting so that data is kept simple and it is easier to work with less columns. Moreover, the null values were removed. After these steps, the data was split in training and testing data(80 % training data and 20% testing data). The training was done in lyrics and genre, respectively. Furthermore, the testing was performed, firstly in lyrics, and the in the genre. CountVectorizer classifier from scikit library was used in order to train and test the provided lyrics. Then, the labels were encoded and the categorical data was transformed to integers. Finally, with different classifiers such as Linear Regression, Multinomial NB, Ada boost, KNeighbours and Random Forest. The training data was fitted and accuracy score was provided. On top of that, a neural approach has also been implemented. To do this, pytorch lightning has been used. The neural network exists out of either two or three layers. After passing these layers, the output gets fed to the softmax function. The input of this neural network is constructed the same way as previously described, creating a bag of words. However, due to time constraints, the preprocessing step of removing stop words has been skipped. This should not be a problem though, as the model itself will learn which words are more important and which words carry less information. The loss function that has been used is the cross entropy loss function. There have been several experiments with different layers, sizes and activation functions. Whilst doing these experiments, the training and validation accuracy and loss have been logged throughout the learning process. When training has been completed, the neural network gets evaluated with a test set. The training, validation and test set are random partitions of the dataset of size 80%, 10% and 10% respectively.

5 Experiments and Discussion

5.1 Title Model

On the test set all the n-gram counts are stored. However, to improve the model's accuracy, weights are introduced. These are used to increase the importance of bigger n-grams. Finally, to check how many titles are possible to be predicted, the training set of title occurrences is used as seen in table 4. This increases the accuracy percentage since part of the titles can not be found in the lyrics. Another model is needed to handle that, however, is not implemented here. To measure the accuracy, a custom predictor is created. It would check if a title is fully matched. Also, it adds a second accuracy, which is based on partial matching of the title. As seen in the results in table 4 full match is accomplished with 10.8% accuracy and partial match is reached in 35.7% of the cases.

Table 4: Accuracies based on possible predictions

Total labels	1161
Possible predictions	842
Correctly predicted	91
Partially predicted	301
Possible max accuracy	72%
Full match accuracy	10.8%
Partial match accuracy	35.7%

This is because the titles some times contain words, which are a lot more repeated than the whole title itself. That causes the algorithm to match only partialy. Also not all titles are perfectly preprocessed and this further drops the accuracy. In conclusion, the title is the most difficult to predict because every song has a different title and therefore the number of titles equals the number of labels, which explains why the accuracy is so low.

5.2 Artist Model

Words and Character ngram To build the best model, we need to investigate which is the best method to preprocess the lyrics. In our preprocessing step, we construct a n-gram matrix of token counts as explained in section 4. We start by comparing the performance of a character bigram model against a word bigram model on the different machine learning model. A summary of this experiment is provided on table 5, where *ML Model* refers to Machine Learning Model, and *Char Bigram* refers to Character Bigram. We notice the

Word bigram outperforms the character bigram on each machine learing model, by at least 12%, up to 30.5%. This is the case as two words give more context that just two characters.

Table 5: Word bigram compared to Character bigram f1-score

ML Model	Word bigram	Char Bigram
Naive Bayes	0.61	0.41
Random Forest	0.51	0.44
Logistic Regression	0.60	0.45

Unigram, Bigram, Trigram and 4gram Our next task is to compare the performance of different word grams on our classification task. This is done on table 6, where NB stands for Naive Bayes, RF for Random Forest and LR for Logistic Regression. It appears bigram performs the best among all the compared n-grams, where it reaches 0.61 of f1-score with the *Naive Bayes* model.

Table 6: Unigram, Bigram, Trigram and 4-gram f1-score

ML Model	Unigram	Bigram	Trigram	4-gram
NB	0.55	0.61	0.57	0.55
RF	0.50	0.51	0.48	0.44
LR	0.59	0.60	0.54	0.49

The Performance on Different Artists We already explained our dataset is biased (we have more pop artists than hip-hop ones). This is clearly shown on the results on table 7 as for instance Eminem is clearly better classified than other artists (0.89 f1-score, while the general accuracy f1-score is of 0.61). This is explained by hip-hop lyrics having a lot more words than pop lyrics, and a completely different vocabulary. The word ”Eminem” even often occurs in Eminem lyrics, which would allow a human to directly map such lyrics to Eminem. On the other hand, pop artists vocabulary is more similar, which makes it hard to differ Charlie Puth from Ariana Grande for example.

Table 7: F1-score on different artists based on the Naive Bayes Classifier

Artist	f1-score	N°songs in test set
Ariana Grande	0.39	60
Beyoncé	0.60	78
Billie Eilish	0.70	33
Cardi B	0.45	17
Charlie Puth	0.22	16
Justin Bieber	0.56	70
Dua Lipa	0.71	51
Ed Sheeran	0.64	64
Drake	0.61	103
Rihanna	0.69	61
Maroon 5	0.62	37
Post Malone	0.49	29
Katy Perry	0.45	71
Coldplay	0.67	79
Taylor Swift	0.61	104
Khalid	0.31	9
Selena Gomez	0.58	28
Eminem	**0.89**	101
Nicki Minaj	0.64	58
Accuracy	*0.61*	*1133*

5.3 Genre Model

Firstly, different classifiers were tested and experimented. These are Logistic Regression, MultinomialNB, Ada Boost, KNN and Random Forest. A table will be provided where the reader can analyze the results from the different classifiers.(See Table 8)

Table 8: Different results with several classifiers

Type of Classifier	Accuracy
Linear Regression	0.60
Multinomial Naive Bayes	0.54
AdaBoost	0.547
KNeigbours classifier	0.475
Random Forest classifier	0.573

Initial experiments for the neural network were performed with two layers, excluding the input layer. The input is of size 159.773 (the vocabulary size) and the output size is 11 (the number of genres in the dataset). The hyperparameters which were experimented with initially were the hidden size (size of the first layer) and the learning rate. Hidden layers of size 50, 100 and 200 were tried, with learning rates varying between 0.01, 0.001 and 0.0001. There has also been experimenting with three layers. The activation function used was either the ReLU function or the Sigmoid function. These experiments all yielded the same results: the training accuracy and loss centred around 0.46 and 2.1 respectively almost immediately after training had started. The validation accuracy and loss followed this trend and the test accuracy and loss ended up with those numbers as well. This all indicated that, with these parameters, the model does not learn much. The training accuracy curve seems to be awfully similar for all previous experiments, as seen in Figure 5. The same holds for the training loss graph in Figure 6 (The x-axis of this sort of graphs indicates how many batches have already been trained with). The model seems to get stuck in a certain learning sequence, even with different initial hyperparameters. This might be explained by Figure 7. Since rock is the most occurring genre in the dataset, always picking it will yield larger accuracy than randomly guessing. This is why these models all create a bias towards rock. Apparently, straying away from this bias does not decrease loss for these models.

As these activation functions did not seem to work, a different one was picked: The leaky ReLU function. Models with this function do not get stuck in the rock bias. Their accuracy curve climbs further and their loss continues to decrease (see Figures 8 and 9). The experiments with leaky ReLU were done with hidden sizes of [100, 25], [200, 50] and [400, 100], where the first list item corresponds to the first hidden layer size and the second list item to the second hidden layer size. The results for the two smallest sizes both have training accuracy of about 0.62 and test accuracy of 0.57. For the larger sizes ([400, 100]), training accuracy is 0.66 and test accuracy is 0.59. This difference in accuracy is probably because a larger hidden size makes the model more expressive, making it fit better. In Figure 10, it is apparent that there is still bias involved. However, now the model is better at predicting the other common genres of the dataset.

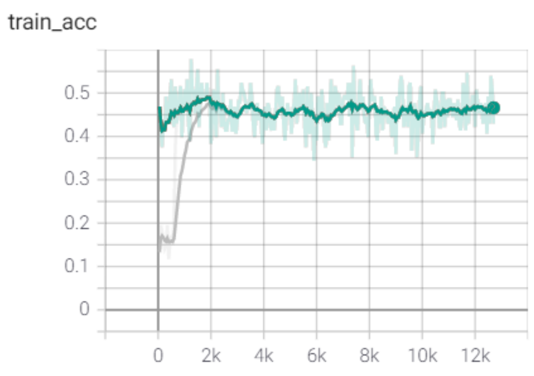

Figure 5: Graph of accuracy curve throughout training with ReLU or sigmoid function

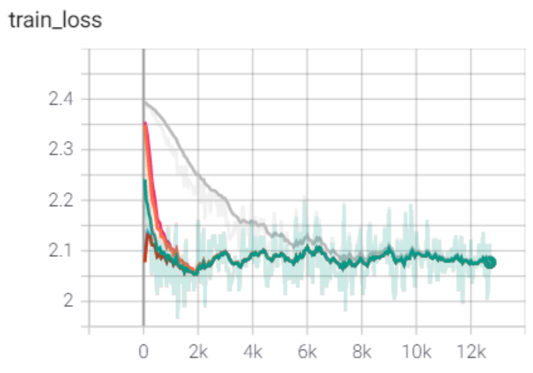

Figure 6: Graph of loss curve throughout training with ReLU or sigmoid function

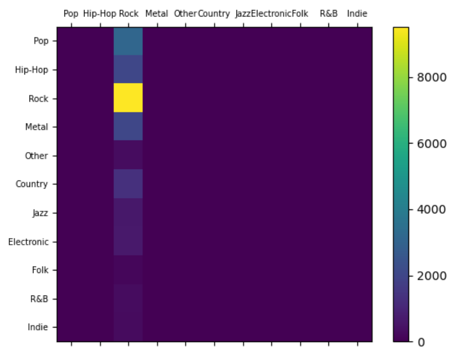

Figure 7: Confusion matrix corresponding to experiments done with ReLU and sigmoid

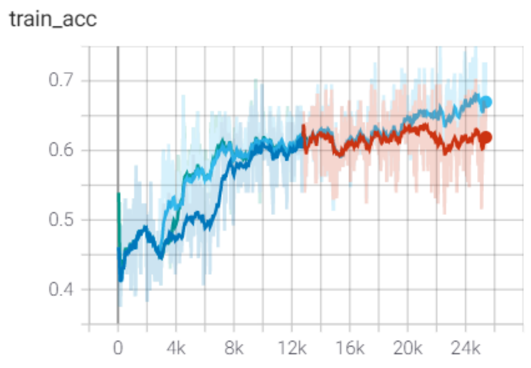

Figure 8: Graph of accuracy curve throughout training with leaky ReLU function

Technique such as Principal Component Analysis was researched and considered but it was concluded that it will not improve the results of the

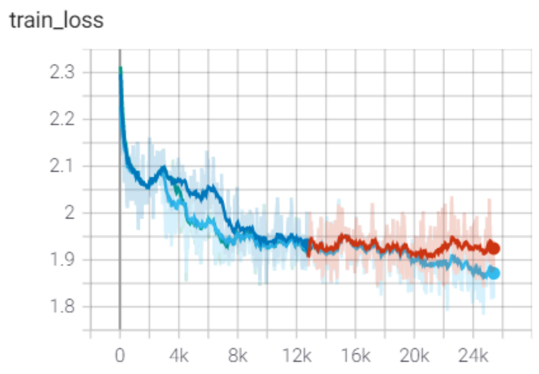

Figure 9: Graph of loss curve throughout training with leaky ReLU function

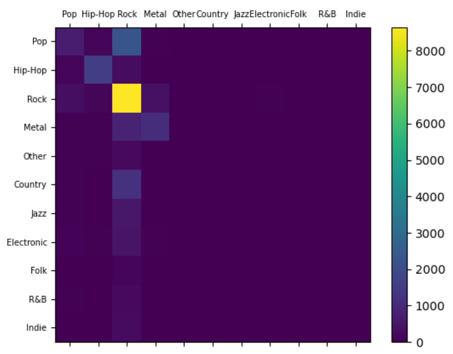

Figure 10: Confusion matrix corresponding to experiments done with leaky ReLU

model. It was proven that it is possible to train a model to correctly identify the genre of a song 60% of the time based on only scanning its lyrics. Further improvements will be to use hyperparameter estimation with Grid Search for more optimal results. The neural approach did not outperform the other classifiers. It did however almost as good as the Logistic regression model. With more experiments and longer training, this neural approach could possibly outperform other methods. Further improvements could be longer training, larger hidden sizes or using word embeddings as inputs.

6 Conclusion

In this report three main tasks were accomplished: predicting the artist, title and genre of a particular song. Two different datasets were used: one for title and artist and a different one for genre. For predicting the title n-grams were used. For each artist the average length of title is computed. Those averages were used and a margin of two was added to achieve some variation. Full match accuracy is achieved in 10.8% of the cases and partial accuracy of 35.7%. For the artist predictions, Naive Bayes, Random Forest and Logistic Regression were tested. As results indicate, the better method for artist prediction is the wordgram and the best word gram for the classification task is the bigram. When it comes to predicting the genre, CountVectorizer was used together with the classifiers: Linear Regression, Multinomial NB, Ada boost, KNeighbours, Random Forest and a neural network and an accuracy of 60% is achieved when using Linear Regression. This result is based on only scanning the lyrics.

References

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium. Association for Computational Linguistics.

Rahil Najafov. 2019. Music and society. 678:20–30.

Carlo Strapparava, Rada Mihalcea, and Alberto Battocchi. 2012. A parallel corpus of music and lyrics annotated with emotions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2343–2346, Istanbul, Turkey. European Language Resources Association (ELRA).

Swathi Swaminathan and E. Schellenberg. 2015. Current emotion research in music psychology. *Emotion Review*, 7:189–197.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 225–230, Berlin, Germany. Association for Computational Linguistics.

yi-hsuan Yang, Yu-Ching Lin, Heng-Tze Cheng, I-Bin Liao, Yeh-Chin Ho, and Homer Chen. 2008. Toward multi-modal music emotion classification. pages 70–79.