Classifying Genres to Songs Based on Lyrics

Sarah Waseem
I6161055

Costanza Siani
I6153520

Abstract

This report talks about the different models that can be used to classify genres to a song based only on the lyrics. To retrieve the text the Bag of Words and TFIDF model was used and several classifiers were applied to check the accuracy. Next, the Word2Vec model was used with the XGboost and Linear Support Vector Machines classifier.

1 Introduction

Music is essential all around the world. It is a common ground for people from different backgrounds and it brings people together. Almost every culture has their own meaning for music and how it unites the people. Through lyrics, artists share their struggles, beliefs and love. Songs, like people, can be similar but also very different. Some can be cruel, hard and straight to the point, while others are more soft, loving and calming.

Due to this strong connection to music ourselves, we chose to pick this as our topic for this assignment. Classification in music is a field that is increasingly being researched. Music apps like Spotify and Apple Music classify songs by genre, artists and mood amongst other things, but they do it by using the audio of songs and comparing frequencies and different attributes in each genre. The task chosen for this project was to be able to classify a song based on just the lyrics of the song. This is a much harder classification as its hard to draw the difference between words of a particular genre. Genre classification is also very subjective as some songs can be classified to many genres.

Indeed some genres are easier to recognize such as Hip-Hop and Country as they have similar words in it or repeated lines. Therefore, we hypothesize that some genres will be more easily classified.

However, this is a very hard classification task as even humans are not able to categorize a genre based on just looking at the lyrics of a song.

2 Prior Literature

Music classification is a growing topic of research in NLP with small performance gains throughout the years. Some researchers have used the same problem which they tackle with different models and classifiers. As a consequence, this report and project lays its basis on different classification problem related works.

Genre prediction problems are mainly conducted using either only audio, or both, audio and lyrics (Dawen Liang,Haijie Gu and Brendan O'Connor, 2011). This paper uses a mix of both audio and lyrics to classify genres. The audio is used to catch the frequencies in the sound and compared with the features in the lyrics to get a more precise accuracy.

Furthermore, many other type of experiments were also carried out from the comparison of best and worst songs to approximating the year of release (M.Fell, C Sporleder, 2014). This paper also researched genre prediction but used much more features extracted from the dataset

In addition, numerous research papers with a very similar choice in models and classifiers were found and some more than others played a key role in the better understanding and analyzing of our results/experiments (D. Dutta, 2018). This paper was most similar to our problem. This only used lyrics as well but used slightly different models.

3 Dataset

The dataset chosen for this task was taken from Kaggle which consists of 380,000+ songs from Kaggle. This was most suited for our task as such datasets are not abundant and others did not have genre classified already to check the classifier against. Initially the dataset had 6 columns which

were the Index, Song, Year, Artist, Genre and Lyrics. Some of these columns are not useful to the task so they were deleted. These were Index, Year and Artist.
The data was split into a train and test set which had 80% of the original dataset for training and 20% for testing. They were also randomized so as to not get the same type of data in one set and another type in the other. To make the running time faster, initially, a sample of 25,000 was taken for the training and 5,000 for the test.
Both sets were pre-processed to remove things that were not important to the data such as punctuation, numbers, stopwords and words with contractions in them.

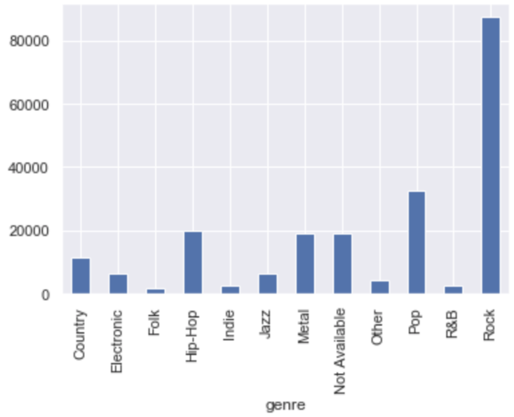

Figure 1: Training set genre distribution

As seen in the genre distribution above, some genres had very few songs while some genres dominated the dataset. Therefore, we decided to remove the less represented genres such as ”Electronic”, ”Folk”, ”Indie”, ”Not Available”, ”Other” and ”R&B”. After running some tests, we also decided to drop ”Rock” as it is so dominant in the dataset.

4 Approaches

Three different approaches were taken to solve the task. These were the Bag of Words, TFIDF and Word2Vec. Multiple different classifiers were used.

4.1 Bag of Words

The bag of words model is one the most commonly used text retrieval models because it is simple to understand and to implement. It helps to represent data by extracting features and giving a way to let the machine learning algorithms to work. Bag of Words disregards word order and only checks a words frequency. To implement this, the countvectorizer from sklearn was used. We initialize this and then fit and transform the lyrics.

4.2 TFIDF

TFIDF stand for Term Frequency-Inverse Document Frequency. It is similar to Bag of Words in that it counts how many times a word occurs in the document but it also checks how rare a word is. If a word occurs frequently, it is penalized and rare words are given more importance. It uses the following formula:

$$w_{i,j} = t \cdot f_{i,j} \times \log(\frac{N}{d \cdot f_i}) \tag{1}$$

4.3 Word2Vec

Word2vec is a neural network that has two layers. It processes text by taking words as an input and outputting a set of vectors with features. Word2Vec is a useful technique because it represents similar words close to each other in a vector space. By doing this it makes it readable for the computer.

4.4 Classifiers

Many different classifiers were used to test how well they do and how they differ from each other. For Bag of Words and TFIDF, the following classifiers were chosen:

- Decision Trees
- Multinomial Naive Bayes
- Random Forest
- Gradient Boosting

While the Word2Vec was used with just two of the following classifiers:

- XGBoost
- Linear Support Vector Machine

5 Results

5.1 To Rock Or Not To Rock?

Classifier	with rock	without rock
Decision Trees	53.4%	52.8%
Multinomial NB	53.1%	54.7%
Random Forest	60.4%	57.1%
Gradient Boosting	61.1%	59.9%

Table 1: TFIDF on sample of 25000.

5.2 General Results

Classifier	TF-IDF	BOW
Decision Trees	50.7%	52.7%
Multinomial NB	52.8%	57.9%
Random Forest	57.9%	57.9%
Gradient Boosting	58.1%	57.9%

Table 2: Sample of size 10.000.

Classifier	TF-IDF	BOW
Decision Trees	76.2%	76.5%
Multinomial NB	56.5%	58.7%
Random Forest	76.8%	76.8%
Gradient Boosting	61.5%	61.0%

Table 3: Full size sample.

Analyzing table 1 and table 2, it is clear that out of all the classifiers that were used, the random forest is the one that provided the best accuracy and that has been consistent in both models and on both datasets. Figure one shows the confusion matrix of TFIDF with Random Forest on the full size dataset. This shows that other than classifying Pop for most songs, Country seems to be the one most often confused. This is not surprising because, as it can be seen from the word clouds in the appendix A1, words like ”baby”,”love”,”smile”, ”God”, ”Jesus” and ” night” often appears in the lyrics of these songs.

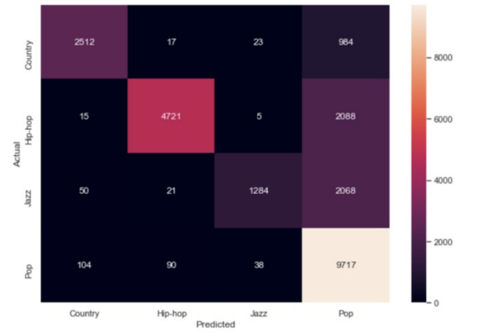

Figure 2: Random Forest with TFIDF on full size set

6 Discussion

6.1 To Rock Or Not To Rock

As mentioned in the Dataset section above, initially Rock was a part of the dataset. However, due to the fact that Rock was such a predominant genre in the dataset, the classifiers would try to classify most songs to this genre and still get a good accuracy. The confusion matrices made this clear as for some classifiers this was worse. These initial confusion matrices can be found in the appendix.

6.2 Decision Trees

Decision trees classify data by creating a sequence of rules. They are used because they are simple to understand, require little data to construct and can handle both numerical and categorical data. As seen in the tables above, it did not perform very well with a small sample size but is one of the best for the entire dataset. This could be due to the fact that decision trees are prone to overfitting.

6.3 Random forest

The Random Forest classifier makes a number of decision trees on sub-samples of the data and uses the average of the model to improve the accuracy while also preventing overfitting.
This is why Random forest works better than decision tree. It performs well on both sizes of datasets and indeed outruns the decision tree.

6.4 Multinomial Naive-Bayes (MNB)

Multinomial Naive-Bayes uses the Bayes theorem and assumes independence between pairs of features. The Multinomial Naive Bayes Classifier works very fast and work really well with small datasets. This can be seen when ran on a smaller sample set (Table 1), it indeed has one of the best performances. However, when ran on the full size set, there is an higher accuracy due to the larger dataset but when compared to the other classifiers, MNB's performance is one of the worst ones and is no longer notably better.

6.5 Gradient Boosting

Gradient boosting uses the objective of minimizing the loss of the model by adding weak learners using a procedure like gradient descent. This is a very powerful algorithm and can be generalized to work for many different classifications.
However, this classifier again can be prone to overfitting on large datasets that may be noisy. This could be a reason why it works quite well for the small sample but not as great on the whole dataset.

6.6 XGBoost

This algorithm is an optimized version of the Gradient Boosting designed to increase speed and performance.It is therefore a very efficient algorithm that was only applied on the word2vec algorithm. It performed with an accuracy of 60.3%.

6.7 Linear Support Vector Machine

The Linear Support Vector Machine is capable of performing multi-class classification. It implements a "One-vs-the-rest" multi-class strategy. It creates a hyperplane that is used to classify the data into classes.
This was also only applied to the word2vec model with an accuracy of 59.4%

7 Conclusion and Further Research

Running these tests gave us some insights on how and why which model works better.
Firstly, we realized that adding Rock to our dataset was making it biased. Therefore, we removed it to get higher accuracies. Secondly, from the confusion matrices we could see that although it was still trying to classify most songs for Pop, as this was now the highest occurring genre, it also did quite well with others. Genres such as Hip-Hop and Country were classified correctly many times as well. This is due to the similar style in Hip-Hop and the repeated chorus style in Country.
It can also be seen that in a way the model does not work that bad as it accurately classifies some of the less represented genres well too. This is already better than humans who also have a hard time classifying these songs based on just the lyrics.
Therefore, all models performed quite well depending on the classifiers used and the best classifier to use is the Random Forest Classifier.

8 References

- Lyrics based Music Genre Classification, aut. D. Dutta. - 2018

- Lyrics-based Analysis and Classification of Music, aut. M.Fell, C Sporleder. - 2014

- Music Genre Classification with the Million Song Dataset 15-826 Final Report aut. Dawen Liang,Haijie Gu and Brendan OConnor. - 2011.

A Appendices

A.1 Word clouds

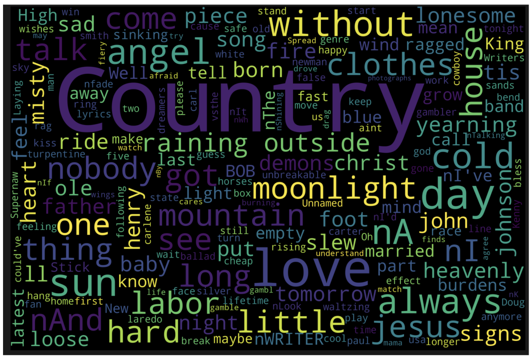

Figure 3: Country

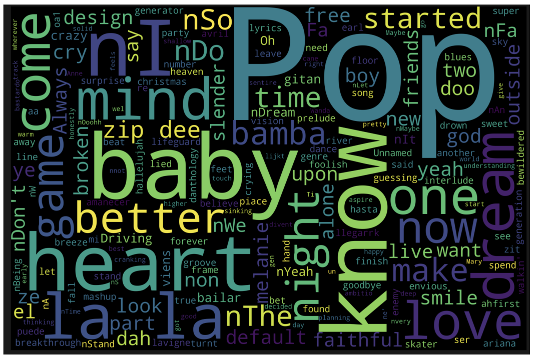

Figure 4: Pop

A.2 Confusion Matrices BOW

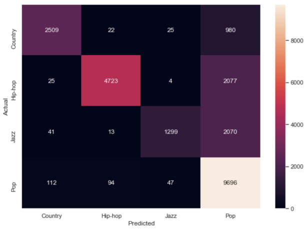

Figure 5: Random Forest

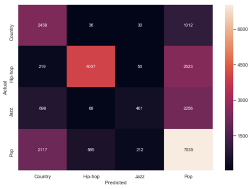

Figure 6: Multinomial NB

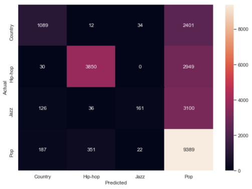

Figure 7: Gradient boosting

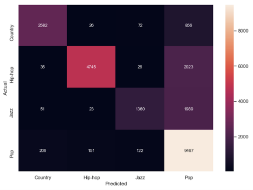

Figure 8: Decision trees

A.3 confusion matrices TFIDF

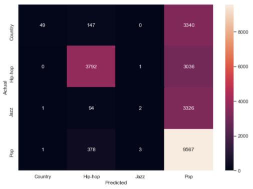

Figure 9: Multinomial NB

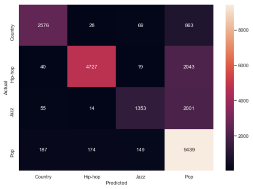

Figure 10: Decision Trees