Classification of Rap Artists by Lyrics

INM430 Coursework

Thomas Martin 16th December 2018

Total 500 words

I. Introduction

Domain

This project aims to create a model to accurately classify tracks by artists using their lyrics (or features generated from their lyrics) alone. Analysis of song lyrics is a popular research topic as they form an interesting area of natural language processing (NLP) research, as they have characteristics different to prose such as a greater emphasis on rhyme, structure, and repetition.

Research tends to focus on one of two tasks,

- Supervised: can songs be accurately categorised by a given piece of metadata e.g. artist
- Unsupervised: can the relationship between songs be characterised by a given piece of metadata e.g. genre

Related Work

This paper is falls into the former category. In the supervised domain there have been a number of attempts to classify song lyrics at the level of artist, release period, or genre. In all cases the

Classifiers typically used include logistic regression

However, previous studies do not generally limit themselves by both music genre and release date as this project does.

Frequently follows an unsupervised task resulting in classification of artists

Questions

This project will specifically address the following questions,

1. Can the model match the reported results of other authors?
2. How does the choice between text representation affect the accuracy of the model and why?

3. How does the choice between classifier affect the accuracy of the model any why?
4. Are some artists

List of objectives

- Accurately classify song by artist using lyrics
- Effectiveness of word embeddings vs bag-of-words
- Find set of (derived?) features for an effective classifier
- Get comparable performance to benchmark

II. Approach

Data Collection

The data for this project was taken from a couple websites. To find the set of relevant artists, the script `get_charting_albums.py` finds all albums to have featured on the Billboard Rap Albums chart between January 2000 and November 2018. From this, I wanted to find a set of the top 10 most prolific artists in this time period, this is simply to ensure that I have sufficient number of tracks for each artist. To do this I found the full track list for each album with `get_tracklist_for_albums.py`, and grouped the total number of released tracks by artist. For these ten artists, I found the matching lyrics for each song from Genius.com using `get_song_lyrics.py`. This set of lyrics was further refined as not all lyrics were successfully returned from the website following this procedure due to parsing errors (In the notebook, I filter out row from the dataframe with null lyrics). In total, the lyrics 679 tracks for 10 artists were found.

The number of tracks for artist ranged from 39 for E-40, to 101 for Eminem

All scripts for the data collection process are in the `scripts` folder.

Analysis Strategy

With a data collected, I planned to do the following,

- Pre-process the lyrics following typical NLP techniques. Using Sklearn

As part of the pre-processing process, I needed to go through a process of feature engineering

Features considered for each track

- Song vectors
- Total line count
- Average line length
- Unique word proportion

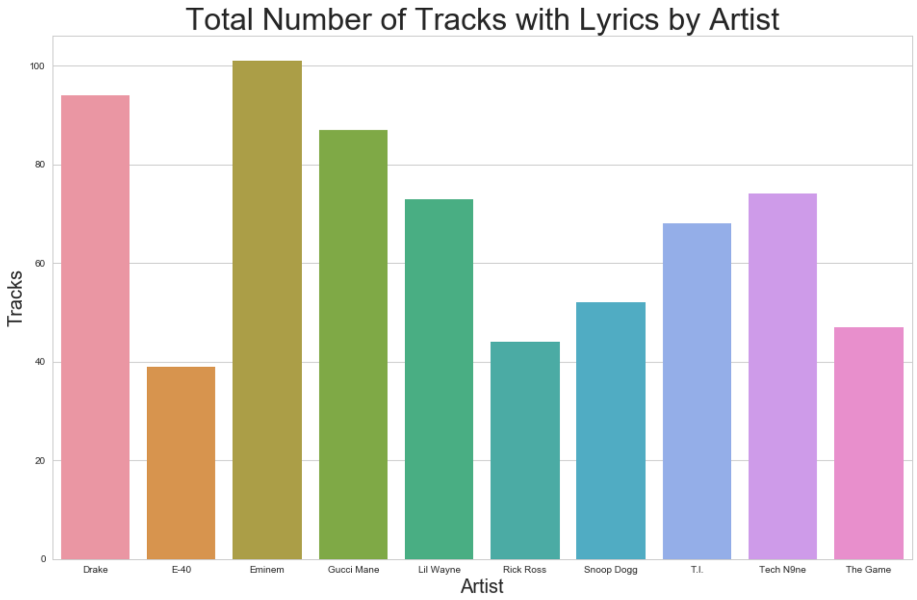

Figure 1: Total number of tracks with lyrics by artist

To get the raw text data into a usable format I used one of the following processes to get the lyrics into a vector format.

- TF-IDF on the raw lyrics
- TF-IDF after POS-tagging of the raw lyrics
- Doc2Vec

Following the approach of other papers on this topic, I will use the following classifier for the final model,

- Logistic regression,
- SVM,
- Naive Bayes. (for naive bayes need to set doc2vec to particular setting)

(Why???)

http://billchambers.me/tutorials/2015/01/14/python-nlp-cheatsheet-nltk-scikit-learn.html

Data Preprocessing

Normalisation includes * Standardise casing * Removal of whitespace * Removal of punctuation * Removal of stopwords * (Did not perform lemmatisation - why?)

Vectorisation * TF-IDF * Doc2Vec

Other feature engineering

Why choice of models

Total 1000 words ## III. Analysis of Results

Analysis

Link to html computation notebook …

The evaluation metrics I will use are,

- Precision: the proportion of correctly classified elements for a given class, out of all the data points predicted for that class
- Recall: the proportion of correctly classified elements for a given class, out of all the relevant data points for that class
- F-Measure: a weighted average of precision and recall, over all classes.

These are typically used for tasks such as this that deal with an unbalanced, multi-class classification problems. This is because they do a job of indicating how well a model identifies true positives, while keeping false positives and false negatives to a minimum.

A big part of of this project is to compare the performance of different text representations for the classification task. The text representations considered were,

- Bag-of-Words: This is the simplest text representation considered, …
- TF-IDF
- Doc2Vec

Discuss differences between logistic regression and SVM

Best performing classifier by text representation. For both TF-IDF and Doc2Vec we see an improvement over the BOW approach by around 7% across all evaluation metrics.

Text Representation	Classifier	Precision	Recall	F Measure
BOW	Logistic Regression	73.0%	72.5.6%	72.0%
TF-IDF	Linear SVC	79.9%	79.4%	79.0%
Doc2Vec	Linear SVC	79.0%	78.9%	78.3%

The full results for the model using the Doc2Vec text representation are reproduced below.

For both text pr

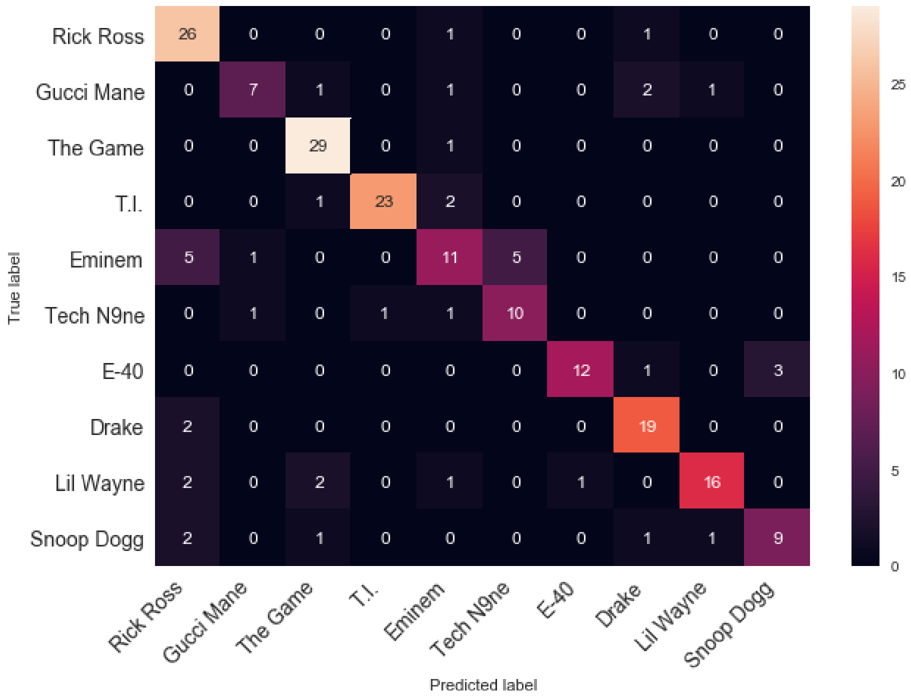

Figure 2: Confusion Matrix for classifier using Doc2Vec text representation

Doc2Vec

```
          precision    recall  f1-score   support
```

Rick Ross 0.76 0.93 0.84 28 Gucci Mane 0.70 0.58 0.64 12 The Game 0.82 0.93 0.87 30 T.I. 0.86 0.92 0.89 26 Eminem 0.76 0.59 0.67 22 Tech N9ne 0.75 0.69 0.72 13 E-40 0.91 0.62 0.74 16 Drake 0.76 0.90 0.83 21 Lil Wayne 0.80 0.73 0.76 22 Snoop Dogg 0.69 0.64 0.67 14

avg / total 0.79 0.79 0.78 204

TF-IDF

```
          precision    recall  f1-score   support
```

Rick Ross 0.70 0.93 0.80 28 Gucci Mane 0.78 0.58 0.67 12 The Game 0.85 0.97 0.91 30 T.I. 0.96 0.88 0.92 26 Eminem 0.61 0.50 0.55 22 Tech N9ne 0.67 0.77 0.71 13 E-40 0.92 0.75 0.83 16 Drake 0.79 0.90 0.84 21 Lil Wayne 0.89 0.73 0.80 22 Snoop Dogg 0.75 0.64 0.69 14

avg / total 0.80 0.79 0.79 204

We see that precision and recall are not always in agreement with each other, for instance Rick Ross has a below average precision by above average recall, and the opposite is the case for Lil Wayne. This means that for an artist like Rick Ross, the model did a good job of correctly classifying tracks to him, but also attributed more tracks from other artists to him as well. For Lil Wayne, the model frequently attributed his songs to other artists. This can be seen in the off-diagonal elements of the confusion matrix.

Due to these differences in precision and recall, it could be argued that the model was biased towards some artists more than others. However, this does not seems to be due to support i.e. the number of tracks by artist in the test set, but rather may have implications for the text representation.

It's interesting to also not the difference in the standard deviation of the two text representations. We see that the standard deviation for precision for TF-IDF is almost twice that of Doc2Vec. This also has implication for the f-measure, as f-measure is calculated from precision and recall.

Text Representation	Precision	Recall	F Measure
TF-IDF	11.4%	15.7%	11.7%
Doc2Vec	6.8%	15.0%	9.0%

It is also worth noting that the classification for either model were highly correlated. Considering f-measure only, which has a Pearson correlation coefficient of 0.88 and p-value of 0.0008, meaning that there is a large, positive correlation between the two sets of results, which is statistically significant.

IV. Conclusions

Reflections

Given the data collection process for this project, there are some concerns as to the direct reproducibility of this study. Discussed above, the lyrics were taken from a third-party website and assume a standard format for the convenience of the web scraper script. In addition, as the lyrics are entered via user submission they are not necessarily reliable in their content or format. Relatedly, although the tracks investigated were chosen specifically because only the target artists was credited as the featured artists, there were a number of instances of other artists appearing on the track without credit.

- Dataset contained unbalanced classes - is this evident in wrongly classified artists?
- Overall size of dataset may be an issue?
- Not directly comparable to benchmarks due to scope of study: just one genre, only songs from 2000
- What features were not considered? Is this important? Why/why not?
- Only considered unigrams

Relate to original objectives and motivations

I believe this project has shown the efficacy of classifying artists by their written lyrics alone. From a business perspective, this has utility in forming the foundation of a text based search engine for a music index. This could also form a component of a music recommendation system, with suggestions based on the similarity of lyrical content between artists.

V. References

- Hussein Hirjee and Daneil G. Brown. 2010 “Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music” Empirical Musicology Review, pp.121-145
- Rudolf Mayer, Robert Neumayer, and Andreas Rauber 2008 “Rhyme and Style Features for Musical Genre Classification by Song Lyrics” ISMIR 2008, pp. 337-342
- Adam Sadovsky and Xing Chen “Song Genre and Artist Classification via Supervised Learning from Lyrics” CS 224N Final Project
- Michael Brevard and Kyle Kenyon, “Artist Classifier”