Classification of Rap Artists by Lyrics

INM430 Coursework

Thomas Martin

16th December 2018

Notebook: https://smcse.city.ac.uk/student/aczd005/inm430_notebook.html

I. Introduction

Domain

This project aimed to create a model to accurately classify tracks by artists' lyrics. Analysis of lyrics is a popular research topic in natural language processing, as they have characteristics different to prose such as a greater emphasis on rhyme, structure, and repetition.

Related Work

Numerous authors have performed the task of classifying track lyrics by metadata, such as by artist, release period, or genre [1,2]. Previous studies do not generally limit themselves by both music genre and release date as this project does: I only considered rap artists with a charting release since 2000, simply due to my interest in the domain.

Questions

- Can the model match the reported results of other authors?
- How does the choice between text representation affect the accuracy of the model?
- How does the choice between classifier affect the accuracy of the model?

Objectives

- Accurately classify songs by artist using lyrics
- Determine effectiveness of text representations
- Find set of features for an effective classifier
- Get comparable performance to benchmark

II. Approach

Data Collection

To find the set of relevant artists, `get_charting_albums.py` found albums to have featured on the Billboard Rap Albums chart [3] from January 2000 to November 2018. For a set of the 10 most prolific artists in this period 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 artists, I found the matching lyrics for each song from Genius.com [4] using `get_song_lyrics.py`. In total, the lyrics 679 tracks for 10 artists were found.

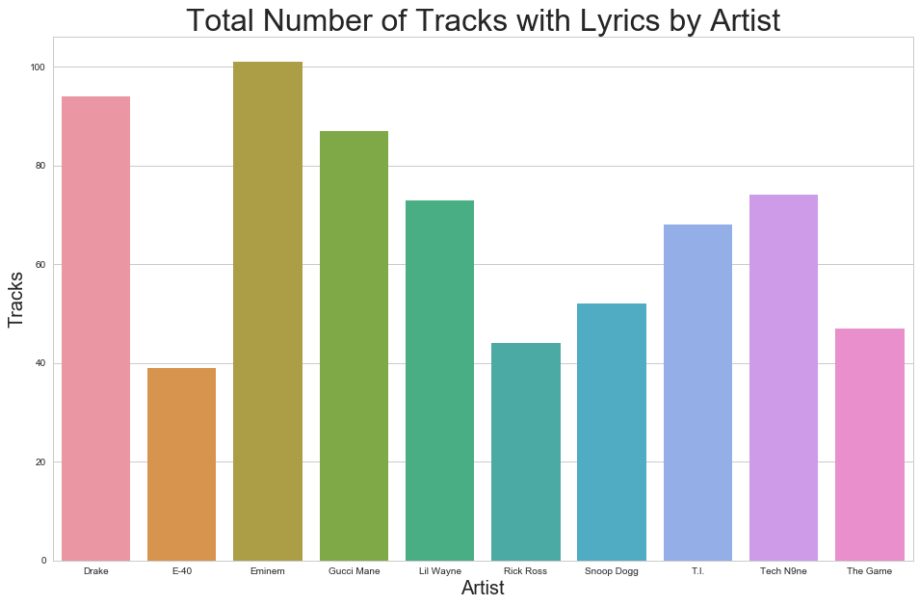

Figure 1: Total number of tracks with lyrics by artist

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

Analysis Strategy

I pre-processed the lyrics following typical NLP techniques: remove stopwords, extract tokens, remove non-alphabetic characters, and remove punctuation.

I performed feature engineering to extract features from the source data. These combined features other authors considered, as well as what I thought was appropriate. Features considered for each track,

- Vectorised track lyrics: For text representations given below strategy. This is main feature in the project.
- Track line count
- Average line length
- Unique word proportion
- Song structure

See section 5. of the notebook for details

The text representations considered were, bag-of-words, TF-IDF, Doc2Vec. These are typically used in classification tasks like this, although Doc2Vec is infrequently referenced in song classification.

The classifiers considered were logistic regression and linear SVM. These are both often cited in multi-class classification tasks. Given these are linear models, it is easier to infer what they are doing behind-the-scenes.

The evaluation metrics I used are precision, recall, and f-measure. These are used for tasks that deal with unbalanced, multi-class classification (Figure 1.). This is because they can indicate how well a model identifies true positives, while keeping false positives and false negatives to a minimum.

For each text representation, I used grid search-cross validation to find the best classifier and hyperparameters using a training dataset. For the best performing model, I tested against a previously held out dataset.

III. Analysis of Results

The text representations considered were,

- Bag-of-Words (BOW): Represents each track by a vector where each vector value is the frequency of each token identified from the source text.

- TF-IDF: Short for text frequency-inverse document frequency. This weighs the BOW representation by the inverse frequency with which a token appears in all tracks. This means that tokens that are frequently found across all tracks are penalised, whereas tokens found in only a few tracks are promoted.
- Doc2Vec: This approach predicts a given word in a track using both the set of surrounding words as well as the document feature vector.

Classification Report by Best Performing Model

Text Representation	Classifier	Precision	Recall	F Measure
BOW	Logistic Regression	72.3%	71.1%	70.7%
TF-IDF	Linear SVC	78.8%	77.9%	77.4%

Text Representation	Classifier	Precision	Recall	F Measure
Doc2Vec	Linear SVC	75.6%	75.0%	74.6%

Both TF-IDF and Doc2Vec show improvement over the BOW approach across all evaluation metrics although the improvement is not very great, TF-IDF does slightly better than Doc2Vec. The difference between the text representations is even smaller if we were to consider the same classifier - see the performance against the training set.

Training Set Accuracy for Logistic Regression Classifier

Text Representation	Accuracy
BOW	74.5%
TF-IDF	79.6%
Doc2Vec	76.8%

The small differences between the results indicate BOW characterises the tracks fairly well. To compare with TF-IDF, we see that the BOW vectors have a dimensionality of 13,488 whereas TF-IDF vectors are of 2,677 as both the 30% most frequent tokens, and terms occurring in fewer than 5 tracks were dropped. In the case of BOW especially, this indicates a large degree of linear independence between the feature space of the extracted tokens. This is probably joint result of not stemming tokens, as well as not filtering out infrequently occurring tokens. For example, vocalisations related to "ah" included "ahh", "ahhh", "ahhhh", "ahhhhh", " 'ahhhhhhh' ". Additionally, the small improvement of TF-IDF over BOW may be due to the feature scaling that the former introduces: frequencies can be scaled to close to zero, aiding the classification process, without changing the dimensionality of the feature space.

Doc2Vec works very differently to the other two, as rather than map all tracks to the same vector space, it finds a distributed representation of lyrics in the track. This means the track is represented in a feature space embedded to represent word similarity. In theory this means that Doc2Vec better general document structure. However, the lower performance compared to TF-IDF is likely due to the inconsistent structure of songs, as well as the relatively small corpus of tracks considered. There is also the issue that a cutoff like min_df/max_df was not used as per TF-IDF as I was unsure how to do this.

For both representations, linear SVM outperforms logistic regressions, for instance test set accuracy for Doc2Vec reported 79.6% and 76.8% respectively. The small difference indicates that either linear model performs well. The small difference is likely due to the different loss functions used.

The most successful trained models only ended up using song vectors as their sole feature, which may be a consequence of these features being heavily correlated with the song vectors. Also due to constraints of time, it was difficult to exhaustively explore permutations of features in the cross-validation step.

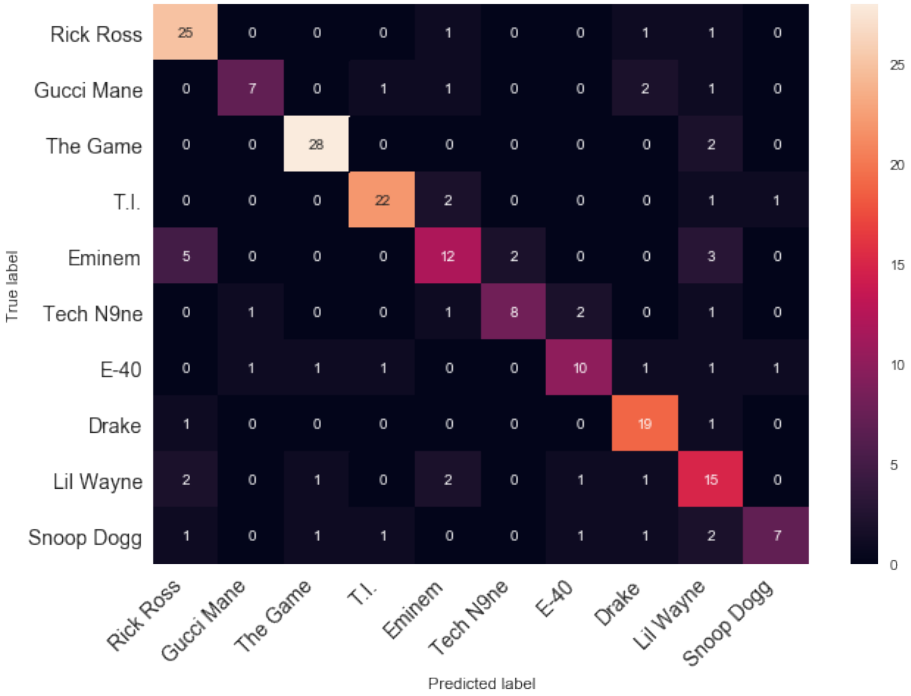

Figure 2: Confusion Matrix for classifier using Doc2Vec text representation

Full Classification Report for Doc2Vec

Artist	Precision	Recall	F-Measure	Support
Rick Ross	74%	89%	81%	28
Gucci Mane	78%	58%	67%	12
The Game	90%	93%	92%	30
T.I.	88%	85%	86%	26
Eminem	63%	55%	59%	22
Tech N9ne	80%	62%	70%	13
E-40	71%	62%	67%	16
Drake	76%	90%	83%	21
Lil Wayne	54%	68%	60%	22
Snoop Dogg	78%	50%	61%	14

Full Classification Report for TF-IDF

Artist	Precision	Recall	F-Measure	Support
Rick Ross	69%	96%	81%	28
Gucci Mane	78%	58%	67%	12
The Game	90%	90%	90%	30
T.I.	89%	92%	91%	26
Eminem	65%	59%	62%	22
Tech N9ne	75%	69%	72%	13
E-40	79%	69%	73%	16
Drake	70%	90%	79%	21
Lil Wayne	82%	64%	72%	22
Snoop Dogg	89%	57%	70%	14

Examining the results for Doc2Vec, Rick Ross has a below average precision but above average recall, and the opposite is the case for Snoop Dogg. 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 Snoop Dogg, the model frequently attributed his songs to other artists. This can be seen in the off-diagonal elements of the confusion matrix (Figure 2.). Due to these differences in precision and recall, it could be argued that the model was biased towards some artists more than others.

Comparing text representations: Considering f-measure, has Pearson correlation coefficient of 0.92 and p-value of 0.0002, meaning that there is a large, positive correlation between the two sets of results, which is statistically significant. It's hard to characterise the differences between the two approaches considered. Looking at an artist like Snoop Dogg, who sees a 9% increase in f-measure from Doc2Vec to TF-IDF, this may be due to the smaller vocabulary used in the latter as well due to the fundamental differences between the text transformations. The Doc2Vec vocabulary contains all terms in the TF-IDF vocabulary, having a size of 3,448 vs 2,677.

Mean and Standard Deviation for each Metric by Text Representation

Text Representation	Precision	Recall	F-Measure
TF-IDF	79%±9.0%	78%±15%	77%±9.5%
Doc2Vec	76%±11%	75%±16%	75%±12%

The standard deviation for TF-IDF is consistently lower than Doc2Vec across all metrics, meaning the former produces more consistent results on average. Overall, I think there is a good grounds to suggest that TF-IDF outperforms

Doc2Vec for this task.

IV. Conclusions

In terms of the overall validity of this project, it should be considered that the lyrics were taken from a third-party website. Lyrics are entered via user submission they are not necessarily reliable in their content or format. Although 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. This would probably not have an impact considering the number of tracks considered, but might mean certain words are picked up that do not relate to the target artist. The sampling method is biased towards older, more established artists, as they are more likely to have produced a large enough number of tracks within the time period considered.

With regards to the text representation used, only unigrams were considered, which may not capture repetitive elements used with a song such as hook. There is also the question as to whether stemming would help performance, by helping to reduce the overall feature space.

Related to original objectives, this project was able to reproduce results similar to those reported by other authors and find an effect model to classify artists by their written lyrics. Overall, I believe this project has shown the efficacy of classifying artists by their written lyrics alone. From a business perspective, this has utility in 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

1. Hussein Hirjee and Daneil G. Brown. 2010 “Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music” Empirical Musicology Review, pp.121-145
2. Adam Sadovsky and Xing Chen “Song Genre and Artist Classification via Supervised Learning from Lyrics” CS 224N Final Project
3. Billboard Rap Albums Chart, https://www.billboard.com/charts/rap-albums, last accessed 12th December 2018
4. Genius, https://genius.com/, last accessed 12th December 2018