**Artist Classification via**

**Unsupervised Learning from Lyrics**

Clyde Clarke

**I. Introduction**

Motivation

The amount of raw data available online has increased dramatically over the past few years; in order to maintain the usability of this data we must develop effective ways to efficiently and automatically organize it. For this unsupervised learning capstone I am choosing to employ NLP of rap lyrics from 10 artists. The corpus consists of 10 songs per artist. The primary focus is on developing lyric-specific features that would allow a classifier to easily distinguish between songs from different artists.

**Project Description and Implementation**

I decided to develop a classifier that can distinguish between ten different artists from the rap/hip-hop genre. To collect training and testing data for our classifier, the rapgenius API was used to download 100 songs, 10 songs for 10 specified artists. The artists are Kendrick Lamar, Eminem, GZA, Ghostface, Andre 3000, Nas, Yasiin Bey,Aesop Rock, Pharoah Monch and MF DOOM. The 10 most popular songs for each artist were chosen and downloaded into a single dataset. After generating datasets (approximately 10 songs artist) and developing using tfidf vectorization, KMeans clustering and Mean Shift clustering are used to try to separate artist by choice of either a line of text from an artist which is taken from the test set or a complete song.

Experimental Method 1

The first method consists of creating a dataframe from the corpus of songs and separating each song line into a line in the dataframe tagged with the artist name. The resulting dataframe is then cleaned to remove text that doesn’t add to the analysis. The dataframe is then vectorized (with the vectorized document consisting of sentences from the various artist’s songs).An elbow plot of the vectorized features is then used to obtain the optimal number of clusters for the K means and Mean shift clustering methods. K means clustering and mean shift clustering is performed on the sentences with the optimal number of clusters. The resulting performance of the clustering methodologies are then obtained using the silhouette method.

Experimental Method 2

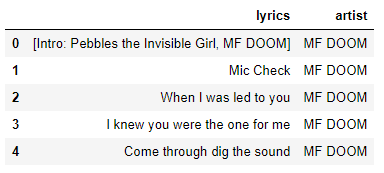
The second method obtains bag of words for each sentence in the dataframe generated in method 1. The resulting bag of words is combined with the vectorization output to create a new feature set which is then used as input to the K Means and Mean Shift Methods.

Experimental Method 3

The document to be vectorized is modified from a sentence to an entire song. The analysis and performance of this choice of analysis is equivalent to method 1.

**Analysis**

Below is a sample the text dataframe



The above is just a sample of an artist and the associated sentences spoken by the artist for each line. During text cleaning the titles of the songs are removed. The sentence count for each artist is shown below the give a sense of the size of the corpus for each artist.



The vectorization processes produced approximately 7000 features. The features consist of words with their associated tfidf scores. Below is a sample 10 of the features obtained from the vectorization process.

|  |  |
| --- | --- |
| 'elvis': | 0.06465733785800441 |
| 'considered': | 0.06465733785800441 |
| 'glad': | 0.05900433356187969 |
| 'jaw': | 0.0549934631837209 |
| 'fabulous': | 0.06465733785800441 |
| 'kicked': | 0.0549934631837209 |
| 'crap': | 0.0549934631837209 |
| 'trapped': | 0.0549934631837209 |
| 'realized': | 0.0549934631837209 |
| 'instruments': | 0.06465733785800441 |

Overall most features occur frequently since the tf-idf scores range between 0.05-0.07. Therefore there is very little to distinguish words and as a result classify artists based on their tf-idf scores gives poor performance. This analysis holds up as a silhouette analysis of the performance of k means with 10 clusters. Since the silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). Low silhouette scores indicate that the object is not well matched to its own cluster and poorly matched to neighboring clusters. The scores below indicate for 10 clusters the silhouette score for the vectorizer training set and test set provide poor performance

|  |  |
| --- | --- |
| X\_train | 0.09368876339338758 |
| X\_test | 0.09741740396840247 |

Since these results were not satisfactory I chose to implement further features into the model. Below is a section on feature selection which details the reasoning behind choosing additional features to try to improve the accuracy of the method

II. Feature Selection

**Bag of Words**

Song lyrics are usually relatively short in length and are constructed from a relatively limited vocabulary. Thus, the selection of words in a song becomes one of its most important characteristics. There are two types of words that are encompassed by this analysis- content and

function. First, past studies on authorship attribution have shown that common function words such as ‘of’ can be an effective marker of author style. Holmes [2] suggests that they may characterize an author’s writing so effectively because they are not entirely under his or her control. Tao et. al. also use them in their study of musical artist style [1]. Content words, on the other hand, allow us to extract semantic meaning from the song. Words such as ‘life’ or ‘love’ can be strong indicators of the song’s topic. By using these words, we can compare tendencies between different genres to choose different topics. a relevant bag-of-words was created for each artist. The bag of words dataframe was then combined with the vectorized tf-idf dataframe to create an overall dataframe of over 7000 features.

A bag of words whose importance was greater than a cutoff, determined to limit the size of the bag to approximately less than 500 words per artist. This choice of bag size was chosen to limit computation time.

Below is a bag of words for the artist MF Doom

['tell','-PRON-','come', 'know', 'man', 'DOOM', 'right', 'need',' go',

'yeah', 'get', 'let', 'rhyme', 'shit', 'be', 'time',' Super’, ‘food',

'friend', 'feel', 'leave' ,'thing', 'ya', 'year', 'sound',]

An analysis of bag-of-words overlap between artist shows that there is significant overlap if not complete inclusion of sets of words between artist. This means the artist use the same set of words and have very little unique words used specifically by artist. For example the subtraction of the sets of bag of words between MF DOOM and all the other artists returns an empty set. Little hope that the bag of words will make a difference.

**Lemmatization**

Lemmatization was done on the dataframe one sentence at a time to reduce the variability in the number of words and to obtain a consistent meaning to similar words. The data was then appended to the bag of words for each sentence.

**POS**

Parts of speech were then obtained from the dataframe and also appended to the feature matrix.

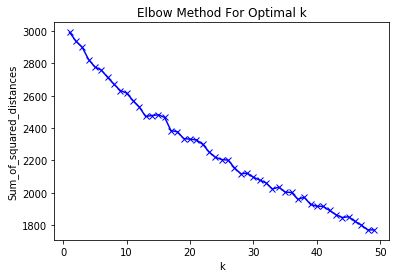
Once all the features were obtained vectorization was done on the text portion of the feature dataframe and the tfidf obtained from the vectorization was appended to the lemmatization, bag of words and parts of speech dataframes. The resultant dataframe was used to fit a k means clustering technique. The following is the result of the new fitting on a sentence by an artist from the dataset.

**Results**

A sample of the combined feature matrix is shown below. It includes all the features mentioned above



The elbow plot of the data for the combined features is given below. As can be seen from the plot there is no definitive elbow and the mean square error continues to drop as the number of clusters increases. However k-means does not do well with large numbers of clusters.



The combined features provide even less consistency within cluster and less separation outside of the cluster than just the raw text dataframe used in method 1

|  |  |
| --- | --- |
| X\_train | 0.043547605658033936 |
| X\_test | 0.02736313448621765 |

The method with larger corpus also returned bad results for the silhouette method of 10 clusters

|  |  |
| --- | --- |
| X\_train | 0.015463134658033936 |
| X\_test | 0.015803373486210765 |

Conclusions

The analysis shows that using unsupervised learning techniques for artist classification is a hard problem. The clusters that are generated from the generated features perform poorly with in cluster and out of cluster performance via the silhouette method. Future work could include additional features. However the main issues with this type of classification is that the features are too similar to each other as rappers use the same type of vocabulary words according to bag of words analysis. An additional methodology could include using artists from different genres to improve performance. On a sidenote cross validation on random forest, logistic regression and gradient boost algorithms were able to obtain satisfactory performance as can be seen in the jupyter notebook associated with this project.