

“I'm different, yeah I'm different”: Classifying Rap Lyrics by Artist

Stephanie Guo and Scott Khamphoune

Introduction

Rap songs can sound radically different from one another because each rapper is unique, or at least they claim to be. Because hip hop artists and rappers often rap about their personal stories, current challenges, and hopes or dreams, every rap song can be considered a personal reflection of its artist.

There has been significant work in text classification and stylometry on problems such as literary author classification and song genre classification. Our project seeks to extend off these past works to investigate author classification within a song genre, specifically in rap or hip hop. Although most humans can subjectively classify hip hop and rap songs lyrics based on knowledge of popular artists and their backgrounds, the question of whether or not this can be transformed into a text classification problem has not been widely explored yet.

Thus, the question we seek to address is: Can we design a model to predict the artist of a hip hop or rap song based on the song's lyrics?

Data Collection and Formatting

To narrow the scope of our investigation, our dataset will contain songs from the following four artists: Eminem, Nicki Minaj, Kanye West and Nas.

We chose these four artists because of the topics they tend to rap about tend to be distinct from one another. Eminem talks about his negligent mother, his love for his daughter, and how his race has distinguished him in the rap world. Kanye West, on the other hand, raps about a broad range of topics, from dealing with fame and fortune to themes of race and consumerism. Nas often raps in the form of first-person narratives, focusing his experiences growing up surrounded by gang violence, drug use and poverty. Finally, Nicki Minaj often touches on her gender and feminism in the hip hop/rap community and her relations with other female rappers.

We obtained approximately 30-35 songs per artist, evenly distributed across each artist's album. All song lyrics were obtained from RapGenius.com, a popular crowdsourcing platform for posting, correcting, and annotating song lyrics. Each song lyric was preprocessed so that lyrics from a featuring artist and annotations indicating a repetition or hook were removed. We formatted the text by tokenizing across whitespace and lower casing all words.

For testing, we randomly chose 5 songs from each artist to be used for testing and used the

remaining songs as training examples.

Artist	Number of Training examples	Number of Testing examples
Eminem	26	5
Nicki Minaj	31	5
Kanye West	31	5
Nas	27	5
Total	115 training examples	20 testing examples

Our feature vector consisted of key words and terms resembling each artist. We used two different approaches to generate the feature vector:

- In our first approach, we used a bag of words model on our dataset to create a frequency table for each artist. Using the frequency table, we selected 16 of the most frequently used words for each artist. If a word was used by multiple artists (as is most often the case), the word would only count towards the artist who used it the most. We called these our “objective feature words.” In total, we had 64 objective feature words.
- In our second approach, we used our common knowledge to generate a list of significant words frequently used by each artist. This approach was motivated by the fact that there was a significant amount of overlap among most frequently used words across artists in our first approach. The bag of words model also many words that were not very descriptive, and used by all artists, such as “never” and “don’t”. For each artist, we had approximately 16 of their most frequently used/significant words. We called these our “hand selected feature words.” In total, we had 64 hand selected feature words.

Models Used

We used the 3 following models (courtesy of the scikit machine learning library):

1. Multinomial Naive Bayes. Naive Bayes is widely regarded as the most efficient and effective machine learning algorithm for text classification. Despite its untrue assumption of conditional independence, it is still able to perform exceptionally well [1].
2. Support Vector Machine. SVMs work well with text classification problems because they acknowledge particular properties of text, including high dimensional feature spaces, few irrelevant features, and sparse instance vectors [2].
3. Decision Trees. Lastly, we chose decision trees because we thought it would be interesting to compare an algorithm not traditionally used for text classification with

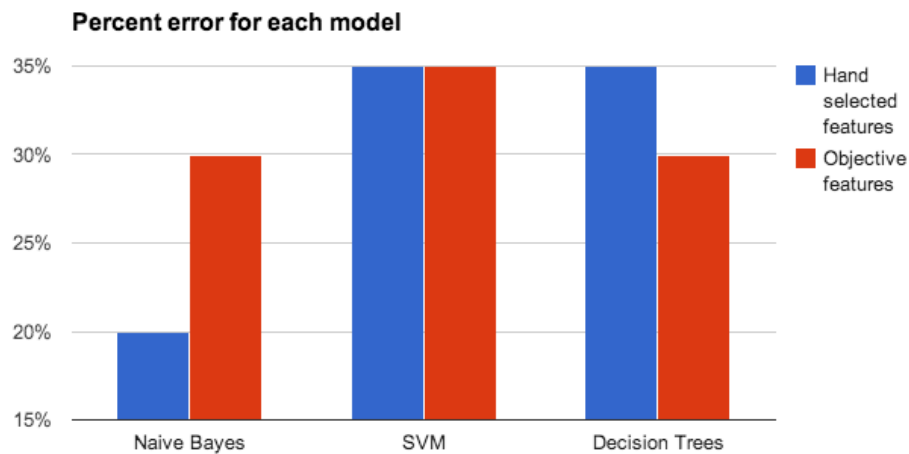
two algorithms that are known to perform well on text classification problems (Naive Bayes and SVM).

For our first approach, we trained each model on the dataset using the 64-length objective feature vector and observed the percent error on the testing examples. Then we iteratively removed a certain number of feature words per artist from the feature vector and observed the resulting percent error.

For our second approach, we used forward search feature selection on the 64-length hand-selected feature vector and observed the subset of the 64 feature words that resulted in the least percent error.

Results

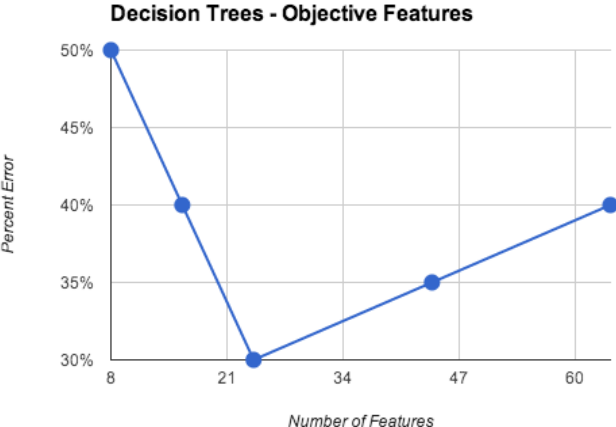
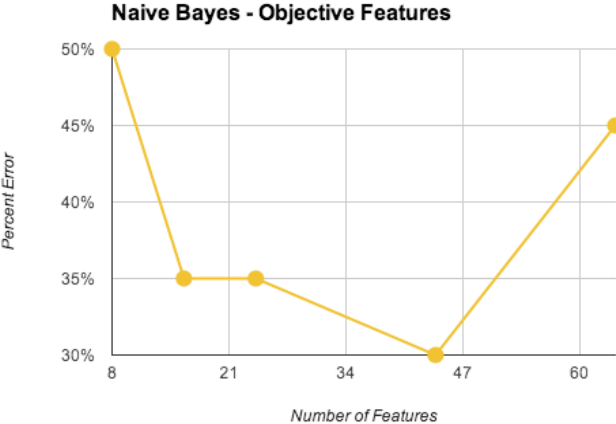
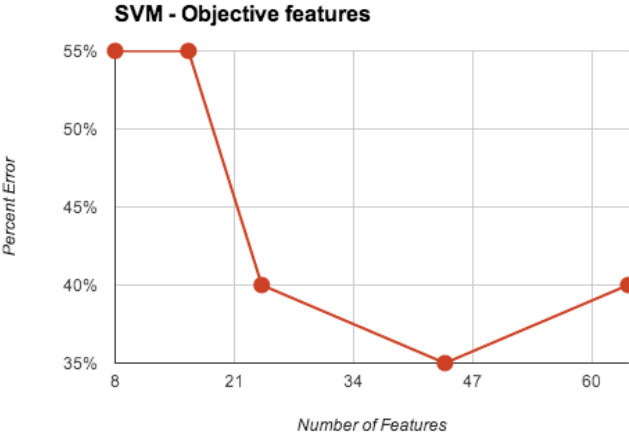
Best percent error for each feature vector for each model



Percent Error as Result of Feature Selection of Hand Selected Features

Model	# of features after feature selection	Percent Error
Naive Bayes	64	20%
SVM	62	35%
Decision Tree	54	35%

Percent Error of Objective Features



Discussion and Further Work

Our results overall were somewhat satisfactory considering the 20% error of Naive Bayes. Implementing forward search feature selection yielded the permutation of a feature set that performed the best and had the lowest error. Feature selection, however, did not improve the percent error of Naive Bayes as the subset of feature words that performed the best was the entire feature vector (all 64 feature words). Therefore, future work should expand the feature vector to include either more words or more sophisticated features for Naive Bayes.

The SVM model performed equally well for both the hand-selected feature set and the objective feature set (35%). Due to the SVMs' ability to work well in high dimensional feature spaces and fulfilling the need for feature selection [2], it follows that results for both the hand-selected feature set and the objective set were the same, as they had 62 and 64 features, respectively. Thus, feature selection did not play a significant role in improving the performance in Naive Bayes and SVM, which is more evidence that the feature vector should have been expanded.

We observed that the performance of objective features followed a consistent pattern; as the number of features were decreased steadily, the percent error would start to decrease, and then start increasing. Removing objective features steadily allowed only the most significant words to be used as identifiers, until a threshold was hit, and then the models would start to underfit, experiencing too much bias.

References

- [1] Zhang, Harry. "The optimality of naive Bayes." *Proceedings of the FLAIRS Conference*. Vol. 1. No. 2.
- [2] Joachims, Thorsten. *Text categorization with support vector machines: Learning with many relevant features*. Springer Berlin Heidelberg, 1998.