Benjamin Lin

Logic, Young Sinatra, or Bobby Tarantino?

Project Design

My project focused on musicians and their alter egos. We have seen this concept all throughout time - from Prince and Camille to Eminem and Slim Shady. For my particular project, I am focusing on the lyrics of Logic, and his two alter egos, Young Sinatra and Bobby Tarantino. My first objective is to use unsupervised machine learning to run topic analysis and determine how his lyrics are bucketed into different groups. My second objective is to build a model that can predict if a phrase most likely came from Logic, Young Sinatra, or Bobby Tarantino.

For my unsupervised modeling, I am mainly going to use PCA and LDA. For the supervised modeling features, I am going to experiment with different vectorizers (BoW, tf-idf, W2V, etc) and NLTK packages. In terms of the actual models, I am planning on using 3 algorithms: SVC, Multinomial Naive Bayes, and KN Classifier.

Tools

As for tools, I used thecypher (a built-in package I found on Github) to scrape some of my data and further used pandas and numpy to clean my data. For the unsupervised ML, I used PCA and LDA.

For my features, I experimented with: Spacy, CountVectorizer, tf-idf, Word2Vec, and several permutations of those.

For my models, I used many of sklearn's tools. Specifically, they were: gridsearchCV, pipeline, several metric calculators, and several built in models and algorithms (more information in the Algorithms section).

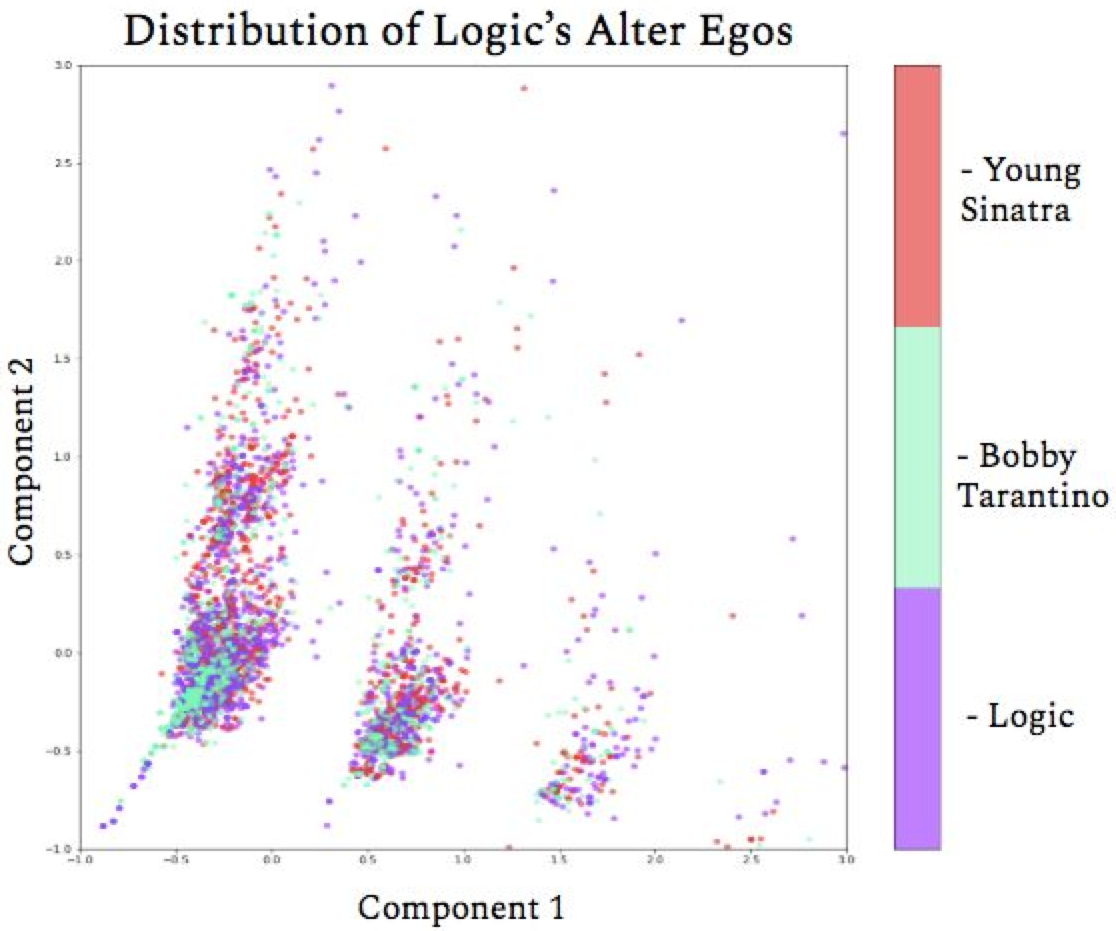
Data

For my data, I used lyrics from six albums, two under each persona. In my dataframe, each row was a line of that particular song. There were approximately 2,000 rows for each persona, so I did not have to address any class imbalance.

For my data acquisition, I mainly used a package called thecypher, which scrapes lyric data from lyricwiki.com and formats it into a pandas dataframe. However, thecypher wasn't able to retrieve all the data, so I also had to manually scrape lyrics from the missing songs.

Algorithms

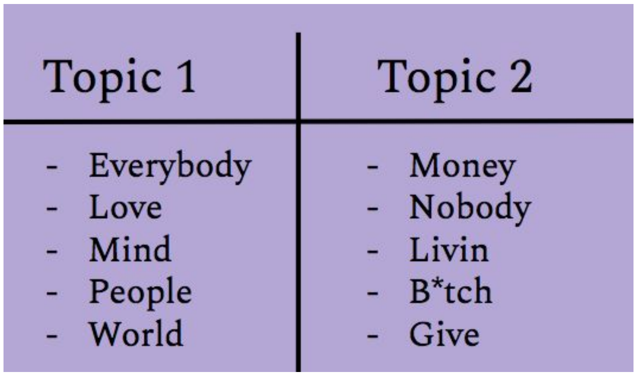
I first looked at the distribution of the 3 alter egos in terms of their lyrics using PCA. Here is the visualization:



Although his different alter egos do have different distributions (ex: you can see that Bobby Tarantino has a smaller range of y compared to the other two), it is ultimately pretty random. The point of this graph is to show that the lyrics under each alter ego actually isn't that distinguishable.

The next thing I was interested in was looking at topic modeling using LDA. I wanted to see if the topics that LDA created would be synonymous with the topics under his alter egos. As a quick primer, here is what each of his alter egos is associated with: (Logic: peace, love, and society), (Young Sinatra: success, motivated, vengeful), (Bobby Tarantino: outspoken, turn-up, carefree).

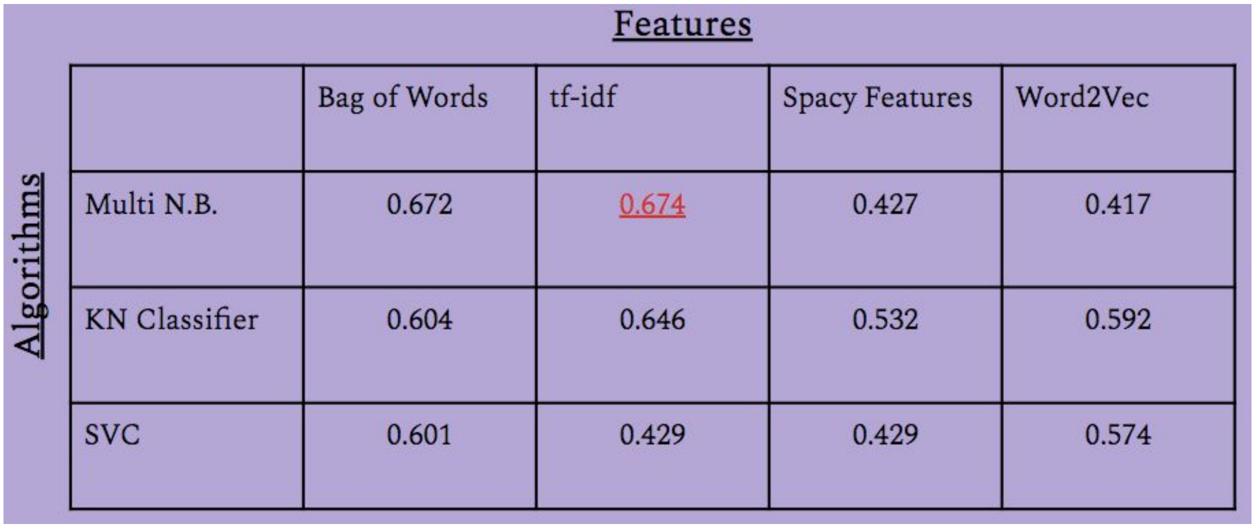
After running LDA, it most sensically broke into two topics, instead of three like I had wished. The top terms under each topic was:



Topic 1	Topic 2
- Everybody	- Money
- Love	- Nobody
- Mind	- Livin
- People	- B*tch
- World	- Give

Topic 1 is extremely synonymous with Logic and what he represents. On the other hand, Topic 2 is more synonymous with both Young Sinatra and Bobby Tarantino. When I tried to create 3 topics, instead of 2, the different topics became abstract and unclear. So my conclusion with LDA is that it clearly separated topics for Logic Vs. Young Sinatra/BT. However, it had difficulty differentiating between Young Sinatra and Bobby Tarantino.

My next step was to create a model that can accurately predict which alter ego a phrase came from. Here is a snapshot of the different features and models I tried (the metric is the accuracy score on a hold out test set):



Features

Algorithms

	Bag of Words	tf-idf	Spacy Features	Word2Vec
Multi N.B.	0.672	0.674	0.427	0.417
KN Classifier	0.604	0.646	0.532	0.592
SVC	0.601	0.429	0.429	0.574

As you can see, using tf-idf and the multinomial Naive Bayes model gave me the best accuracy score of 0.674. My intuition for why this model performs best is because tf-idf calculates the relative weights that evaluate how important a word is.

Conclusion

If I could go back and redo this project again, I would put more time and emphasize on cleaning the text as much as I can. I did basic cleaning but I feel like I could have increased the accuracy on my model by making the text cleaner and more uniform. On top of that, I would have loved to try even more algorithms, such as decision trees and random forests. Additionally, playing around with a deep learning model would have been really interesting. I would have also liked to run PCA on something other than just a bag of words dataframe - and see if the distributions would be more separable or not.

Ultimately, I really enjoyed working on this project because Logic, Young Sinatra, and Bobby Tarantino have a special place in my heart. While I wish my accuracy score was higher and there was more separation in my LDA, it was very interesting to compare my intuition and preconceived thoughts with what the data actually showed.