**Abstract**

It has been well [documented](https://www.theatlantic.com/politics/archive/2016/08/donald-trump-twitter-iphone-android/495239/) that there are multiple users for Donald Trump’s twitter account based on what device a tweet was made from. The question is can we try to predict who is posting based on the contents of the tweet? That question was what sparked this project, which uses an [archive](https://www.thetrumparchive.com/) of Trump’s twitter profile now that his profile has been banned.

**Design**

After pulling tweets from a specific period (more on that later), I put the data set through EDA added onto it by conducting a sentiment analysis. I then put it through some text preprocessing and topic modeling before finally running it through a series of classification models.

**Data**

The dataset is comprised of tweets made by Trump (minus retweets) from June 16, 2015 through March 25, 2017. This period is from his official announcement for the presidential campaign to the day a tweet was last made from an Android device. The raw data includes over 5,300 tweets and several other features including the date/time the tweet was made, whether it was deleted, a retweet, or flagged. Most of the native features were dropped and additional features were added including a sentiment analysis of each tweet, counts of punctuation, and existence of URLs.

**Algorithms**

For the EDA portion, the project utilized numerous panda features to wrangle the data while illustrating it through python’s matplotlib. Additionally, a Scattertext illustration was included to break out words for Trump vs. non-Trump users. Then for the NLP portion, the text was preprocessed using regex, stemming, and removal of stop words while sentiment analysis was conducted using the VADER library.

An extensive topic modeling was conducted by developing multiple doc-term matrixes using CountVectorizer and TF-IDF tokenizers. These matrixes were fed through NMF and LSA models and after an iterative process, the NMF model using the CountVectorizer doc-term matrix was chosen.

For classification, I ran the doc-term matrix through a Logistic Regression model for a baseline result then compared it to a Bernoulli Naïve Bayes model and a Multinomial Naïve Bayes model. Unsurprisingly, the Multinomial NB model performed the best:

Table

Description automatically generated

As an extra step, I took the topics as a proxy for the individual tweets and used that as a feature for a traditional classification model, along with other features I engineered. The features were tested on a variety of models with the following results:

Table

Description automatically generated

Overall, the XGBoost model actually outperformed all the other traditional classification models *and* the Naïve Bayes models.

**Tools**

Numpy, Pandas, Sciki-learn, Matplotlib, NLTK, RegEx

**Communication**

A PowerPoint was presented to show my findings. Additionally, all my work and material will be uploaded to my GitHub.