Twitter Bot Classifier

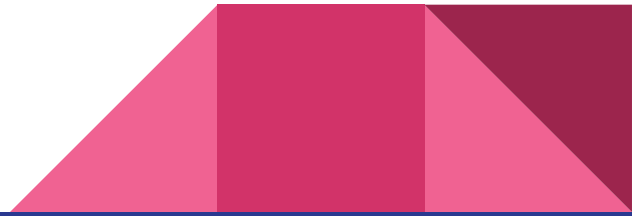
“In line with our principles of transparency and to improve public understanding of alleged foreign influence campaigns, ***Twitter is making publicly available archives of Tweets and media that we believe resulted from potentially state-backed information operations on our service.”***

Authors

Nabil Abbas & Mark Brennan

Agenda

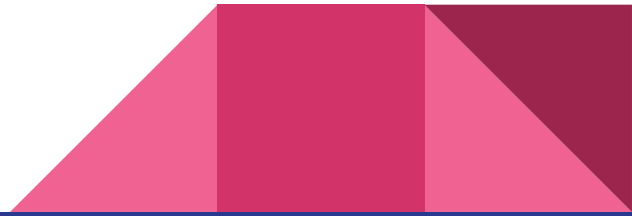
- Goals
- Strategy
- Understanding Our Data
- Our Model
- Model Performance
- Feature Importance
- Takeaways
- Future Considerations



Goals

- Use public data set provided by Twitter to create a Machine Learning Model that can classify if a tweet is from a Bot or Human

- Identify key terminology that may differentiate between a bot or human



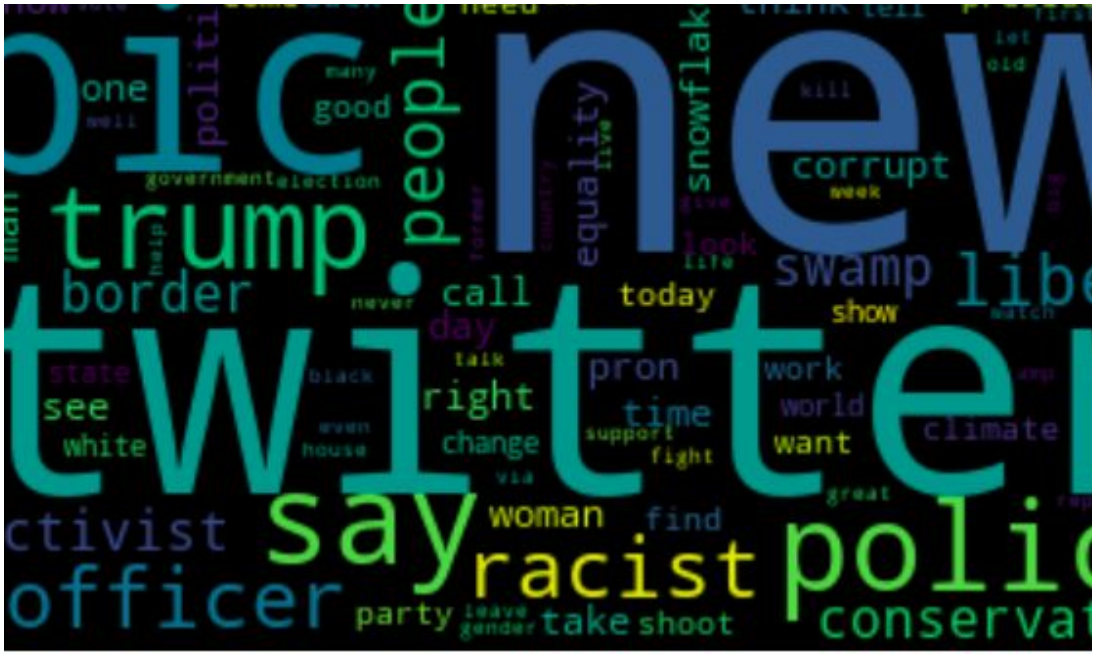
Strategy

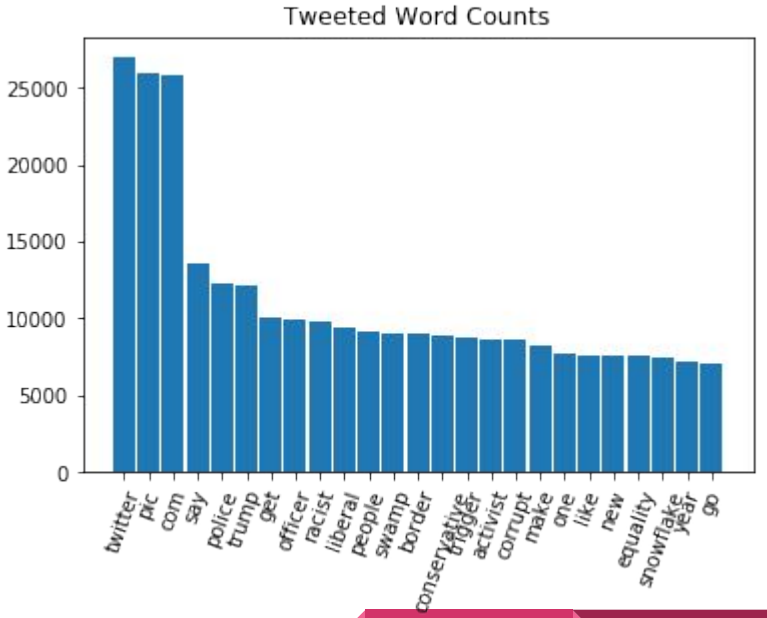
- Data Source: Twitter Election Integrity Data Set with 8,000,000 flagged, removed, and suspended tweets
  - Slice dataset to English account and English language tweets (~ 3,000,000 tweets)
  - Slice dataset to workable size (**~120,000 tweets**)
  - These tweets were in the positive class i.e. BOT classification
- Using Twint library / Twitter Api grabbed ~120,000 tweet data
  - **~120,000** tweets grabbed using terms **"equality, trigger, snowflake, swamp, border, politics, activists, liberal, corrupt, conservative, police officer "** from **verified** accounts
  - These tweets were in the negative class i.e. Human classification
- Final dataset contains ~240,000 observations (tweets)

Data Cleaning and Preprocessing

- Cleaned all tweets - removed:
  - “RT” (retweet flag)
  - @
  - #
  - URLs
  - Emojis
- Tokenized all tweets
- Remove “nonsense” tokens and non-unique tokens
  - Tokens must appear at least 5 times over entire corpus
  - Must not appear in over 50% of documents
- Removed stop words and punctuation
- Lemmatized all tweets (using spaCy)

Understanding Our Data



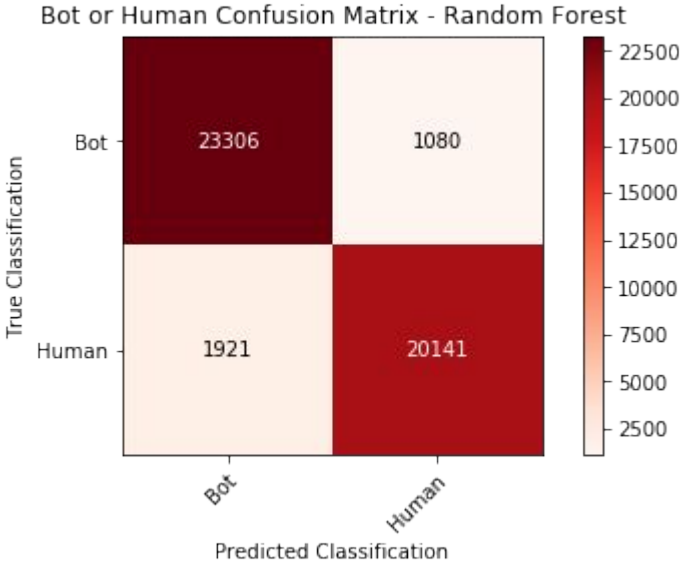


Our Model

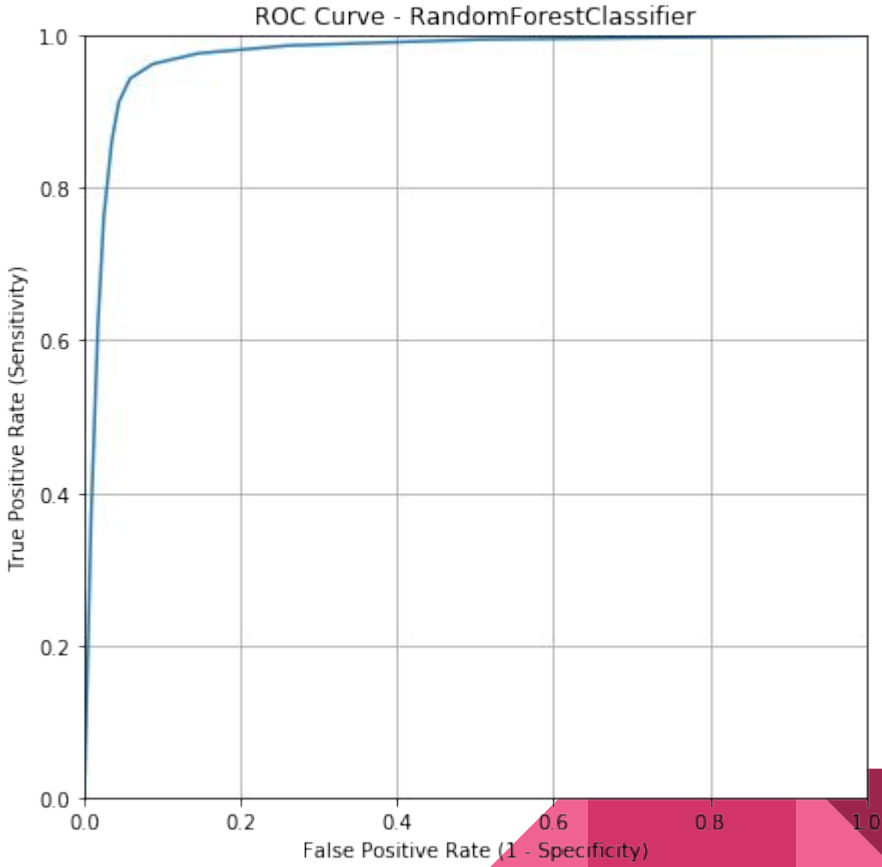
- Scikit-Learn Dummy Classifier gave us a baseline:

  Precision Score: 0.477
  Recall Score: 0.476
  Accuracy Score: 0.504
  F1 Score: 0.476
- Multi-nomial Naive Bayes was run on data vectorized using TF-IDF vectorizer and had reasonable results (precision: .91, recall: .78, accuracy, .86, F1, 84)
- Using Scikit-Learn's TruncatedSVD vectorizer ***we reduced our feature space from 19K+ dimensions to 100!***
- Decision Tree fitted on this SVD data yielded great results, but Random Forest was best.

Model Performance

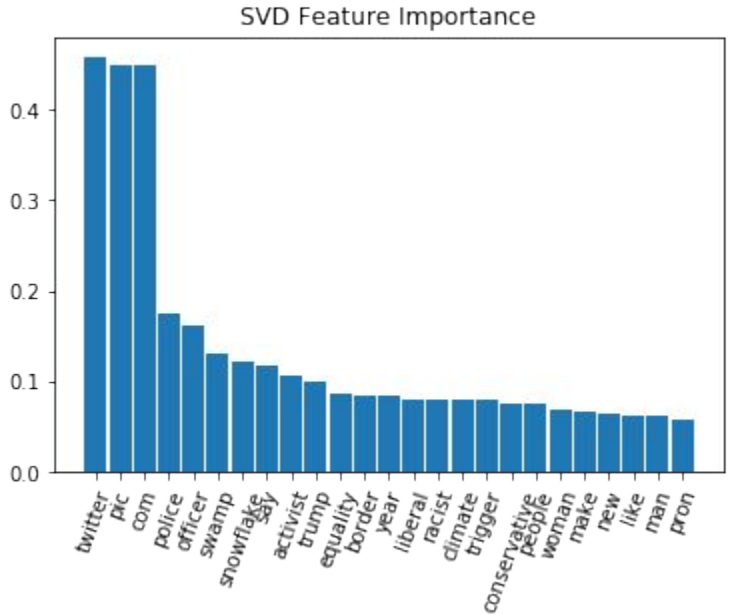


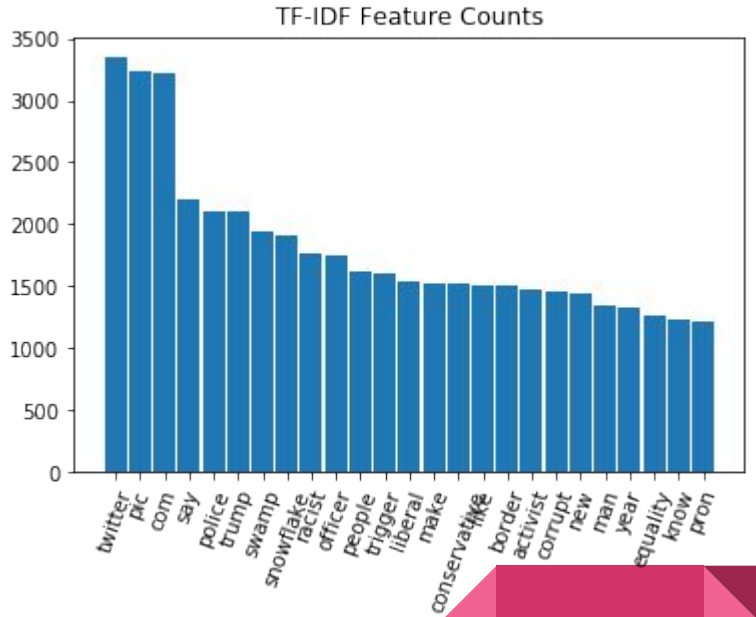
Precision Score: 0.956
Recall Score: 0.938
Accuracy Score: 0.949
F1 Score: 0.946



AUC (pred_proba): 0.973

Feature Importance





Word Importance after SVD

Takeaways

- Found key words that help classify Bot or Human tweets
  - Requires further exploration to analyze negative or positive affect in classifying

- Classifier performed at a promising 95 % accuracy

Future Considerations

- Consider **varying opinions** of our approach to see if there are any overlooked factors that led to the success of our classifier.
- Kernel kept crashing when trying to visualize distribution of **key words** over the two classifications. Visualize on a more powerful machine.
- POS tagging may yield additional insights.
- Explore options to include tweets from “non-verified” twitter users
- GridSearchCV to optimize parameters
- Test additional classification models

Questions?