Russian Troll Tweets: A Kaggle Dataset

Chris Johnston & Sarah A. Cantu

https://github.com/NumbersAndStuff/STAT689-Project

Troll Tweets During the 2016 Election Campaign

NBC recovered about 200,000 tweets from alleged Russian troll accounts that were tied to “malicious activity” during the election.

NLP: Text Classification

NLP - Natural Language Processing

Supervised Techniques:

- Naive Bayes
- Bag-of-Words
- Decision Trees
- Random Forests

Bag of Words

Called such because it essentially creates a bag of words

Cleaning and Tokenizing the Datasets

- Python packages nltk and SpaCy were used to clean and tokenize the data

```
The bright example of our failing education https://t.co/DgboGgkgVj
['The', 'bright', 'example', 'of', 'our', 'failing', 'education', 'https://t.co/DgboGgkgVj']
['the', 'bright', 'example', 'of', '-PRON-', 'fail', 'education', 'https://t.co/dgboggkgvj']
```

BoW Example

A standard JSON example of BoW

1. Original text
2. All position information is lost when tokenizing and counting
3. Final output for use as input or analysis

```
(1) John likes to watch movies. Mary likes movies too.
```

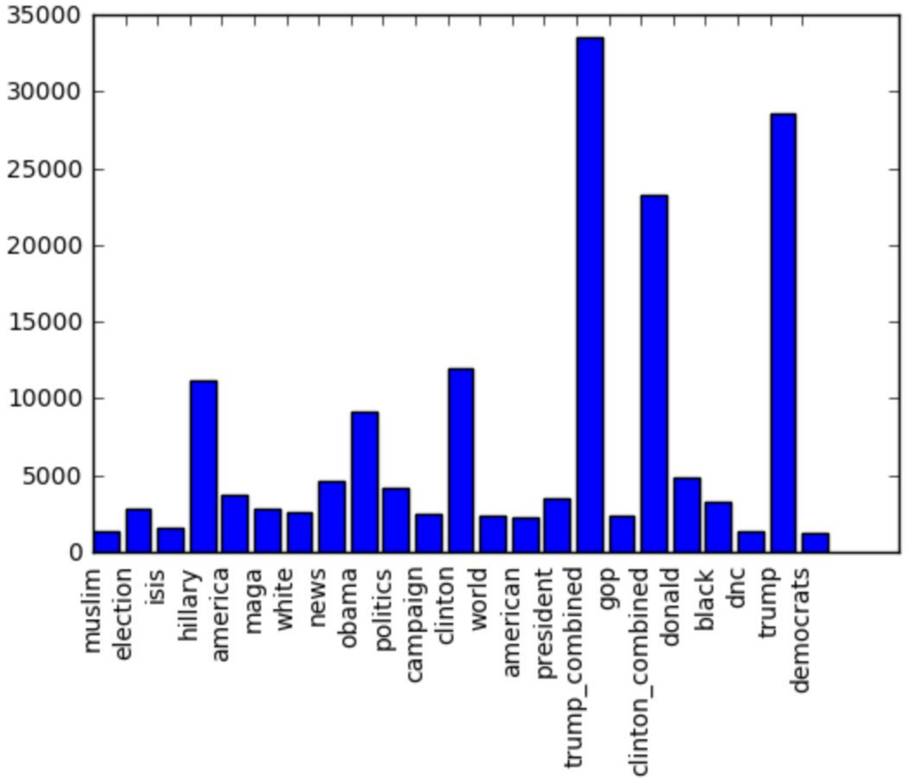
```
(2) John also likes to watch football games.
```

```
BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

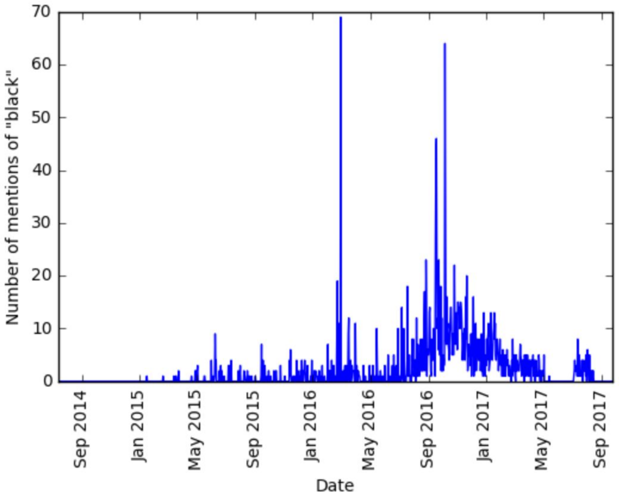
```
(1) [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]
(2) [1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
```

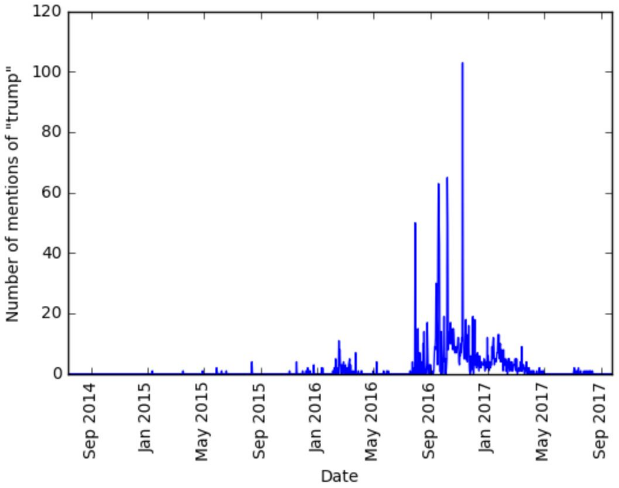
Images from Wikipedia

All the tweets as one document and each tweet as one document

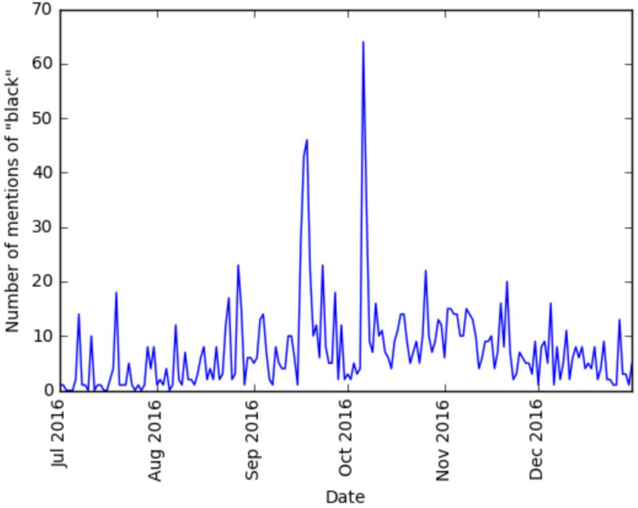


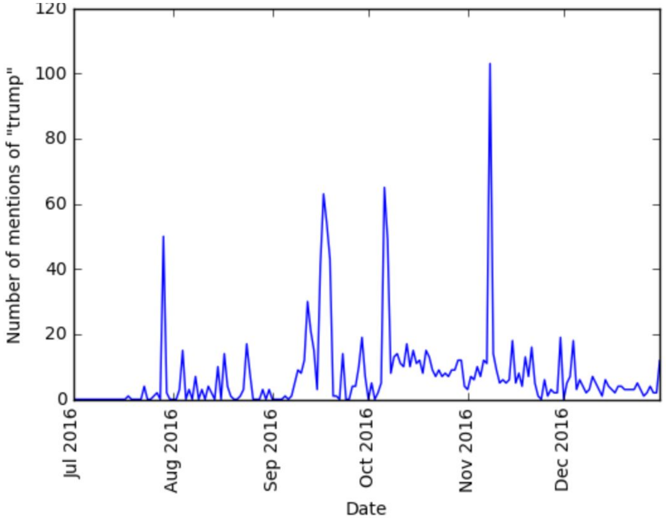
- Highest occuring words over the whole dataset were found
- The most interesting words were isolated - 23 words chosen
- These words and the tweets containing these words were looked at vs. time





- The number of a keyword mentioned as a function of time
- A lot of the more interesting words show this trend at least a little
- Spikes can be correlated to real world events, e.g. primaries, election



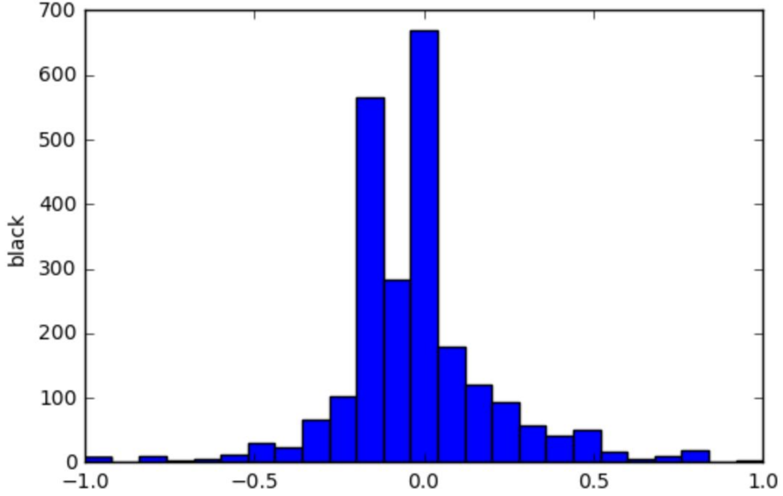


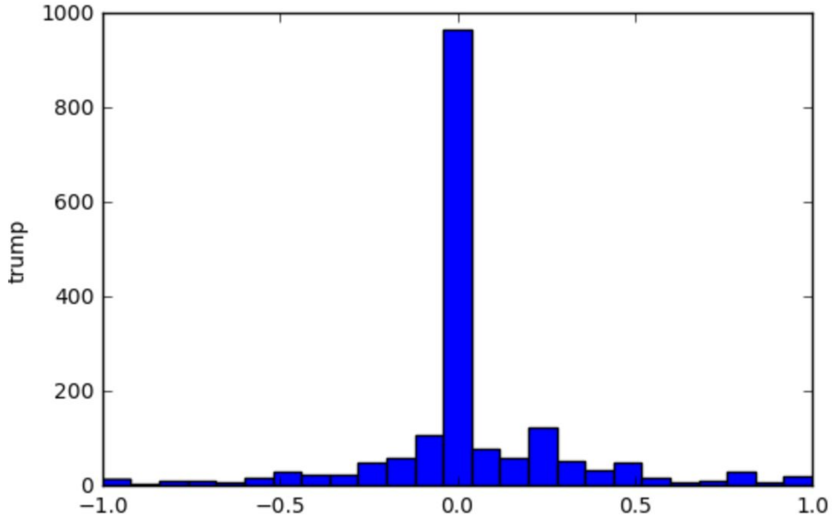
- Dates restricted to July 2016 - Dec 31, 2016
- The spikes here correlate well with the GOP and DNC conventions, presidential debates, and the election itself

Naive Bayes Analysis

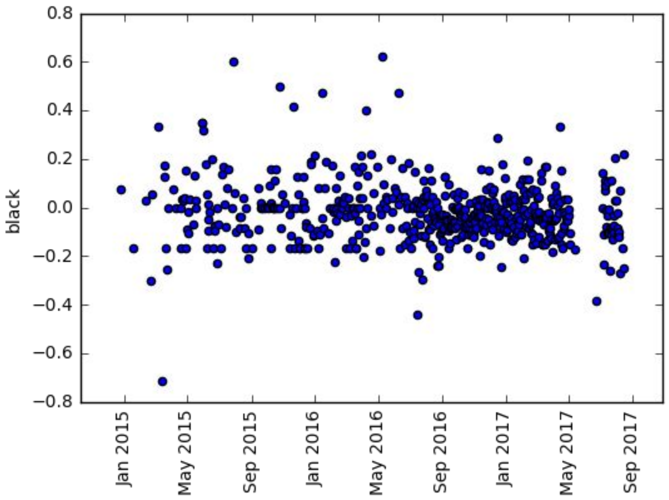
With TextBlob and

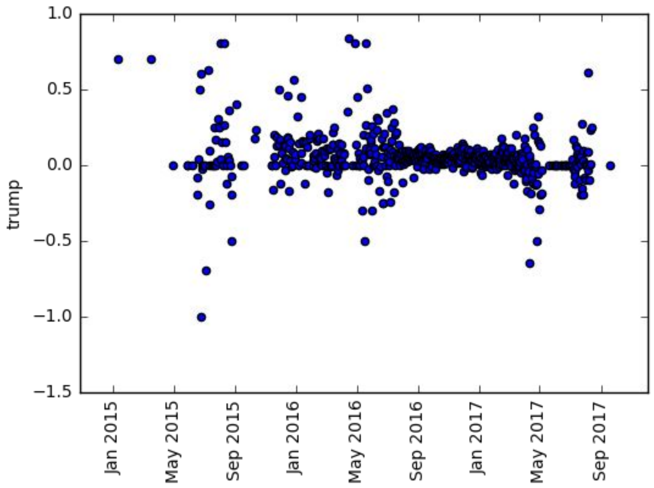
Linear Regression





- TextBlob sentiment generally shows neutrality with a slight tail
- The keyword black shows an odd distribution with negative spike





- Scatter plots of daily average sentiment were made to look for trends
- An ordinary least squares fit was attempted with the model

$$y = \beta_0 + \beta_1 x + \beta_2 i_{\text{Election Season}}$$

- No significant trend was found

Sentiment Analysis with Decision Trees and Random Forest Classifiers

Decision Tree vs Random Forest

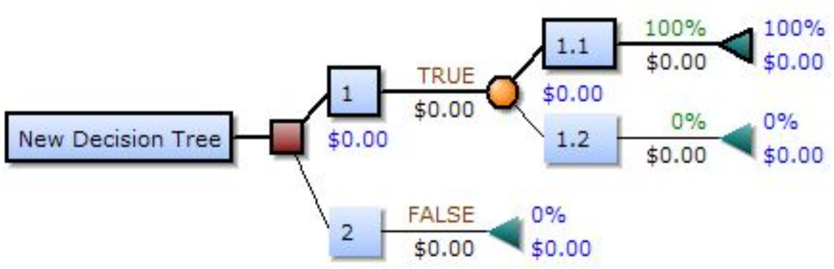
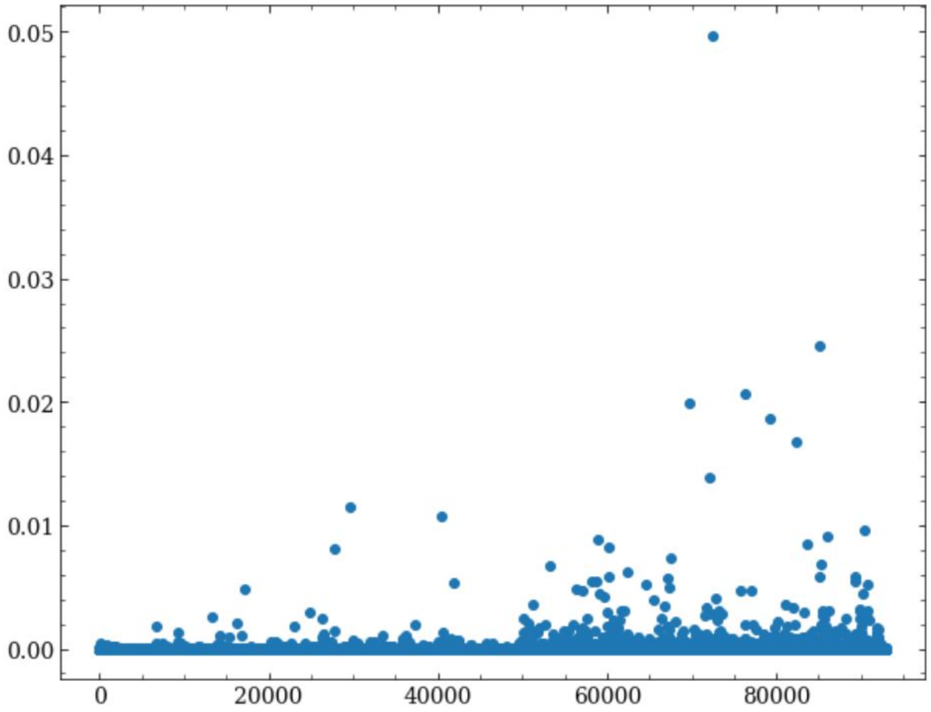


Image from Wikipedia

- Easy to understand
- Easy to look under the hood
- Trained on arbitrary amount of input features and target
- Can take a BoW input

Decision Tree Feautres



Features

- An example of feature importance in a decision tree
- Pulled out the top features and mapped them back into keyword space
- Top features: not and thank
- Average precision: 0.68

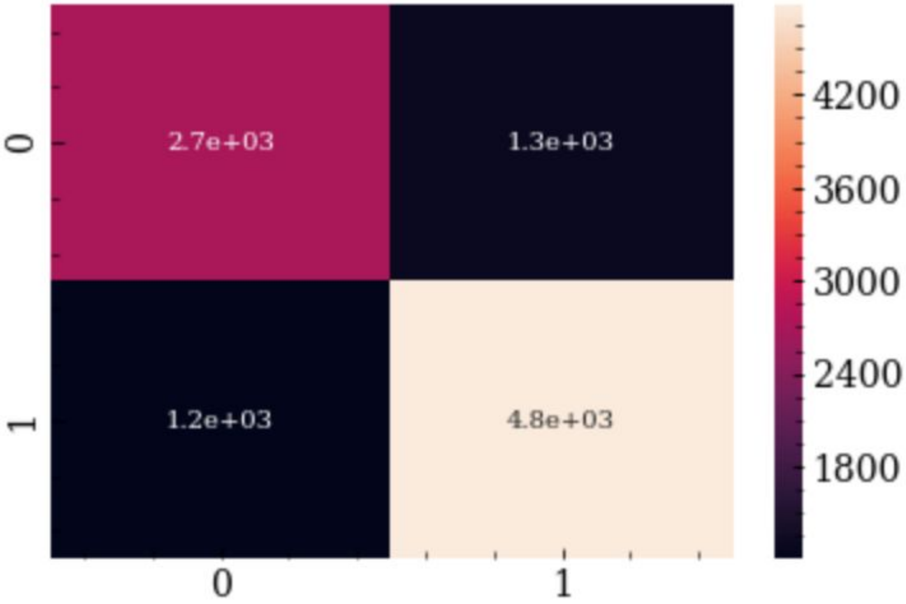
Decision Tree vs Random Forest

	precision	recall	f1-score	support
0	0.59	0.63	0.61	4001
1	0.74	0.70	0.72	5988
avg / total	0.68	0.68	0.68	9989

	precision	recall	f1-score	support
0	0.69	0.68	0.68	4001
1	0.79	0.80	0.79	5988
avg / total	0.75	0.75	0.75	9989

- Without pruning, decision trees can overfit
- RF create many shallower DT and combines them
- Bias is not reduced, but variance is
- Average precision: 0.75
- Time to run:
  BoW/NB - immed
  DT - 5 min
  RF - 10 min

Random Forest Confusion Matrix



- Diagonal contains true values
- Off-Diagonal False pos/neg
- Lack of neutral training set

Conclusion

- Sentiment analysis on tweets is difficult
- Training data was not ideal - the political nature of the analysis makes it difficult to assess the sentiment
- Weighing the accuracy/precision vs time of Naive Bayes vs Random Forest, in this case Naive Bayes is easier implementation and may be more useful

Next...

- More preprocessing - messy data leads to very slow analysis
- To compare the accuracy of Naive Bayes more quantitatively
- To tweak the Random Forest parameters and see if time and/or accuracy can be improved
- To create an algorithm that runs the Random Forest only on tweets that contain words of interest