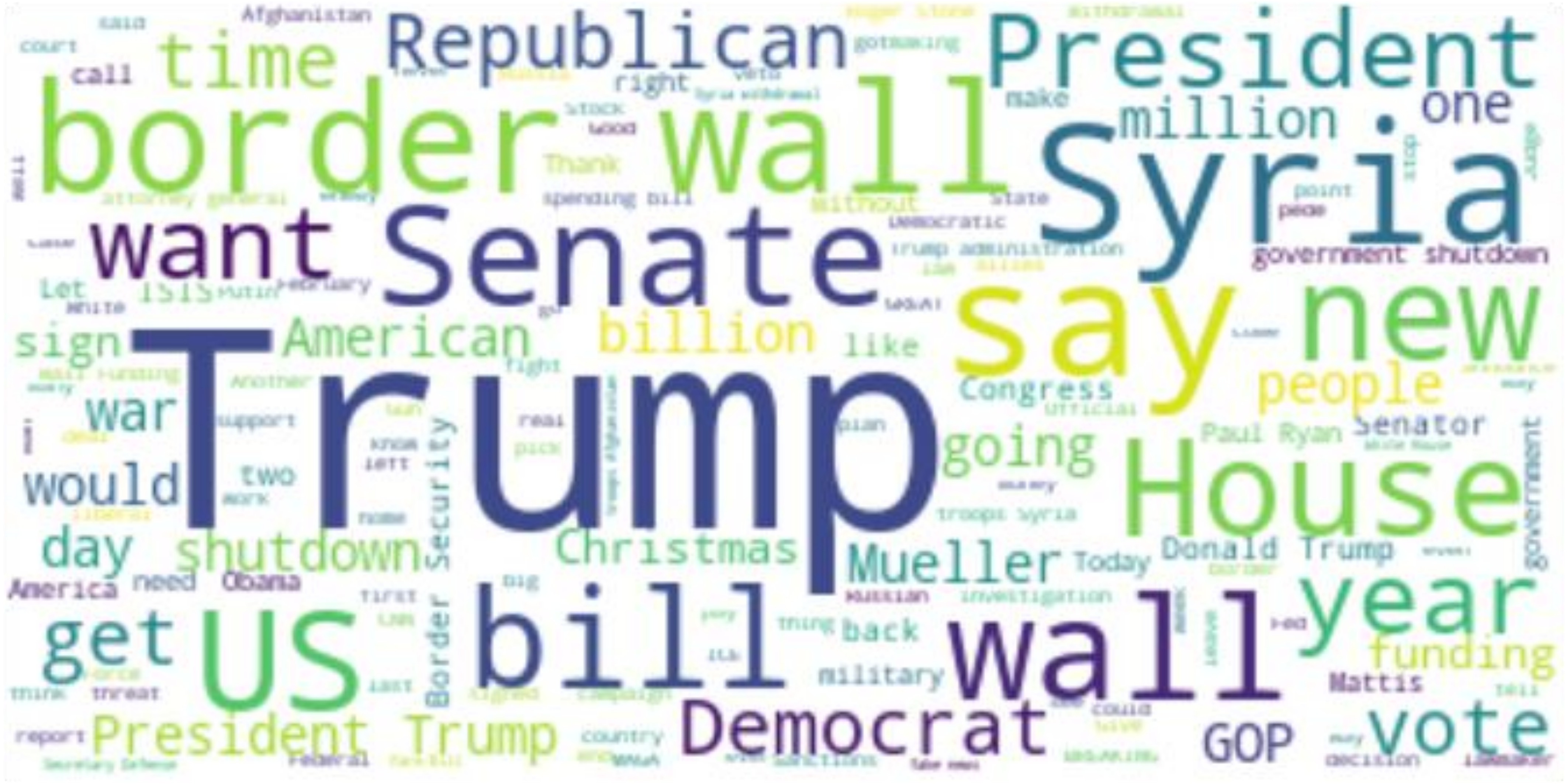
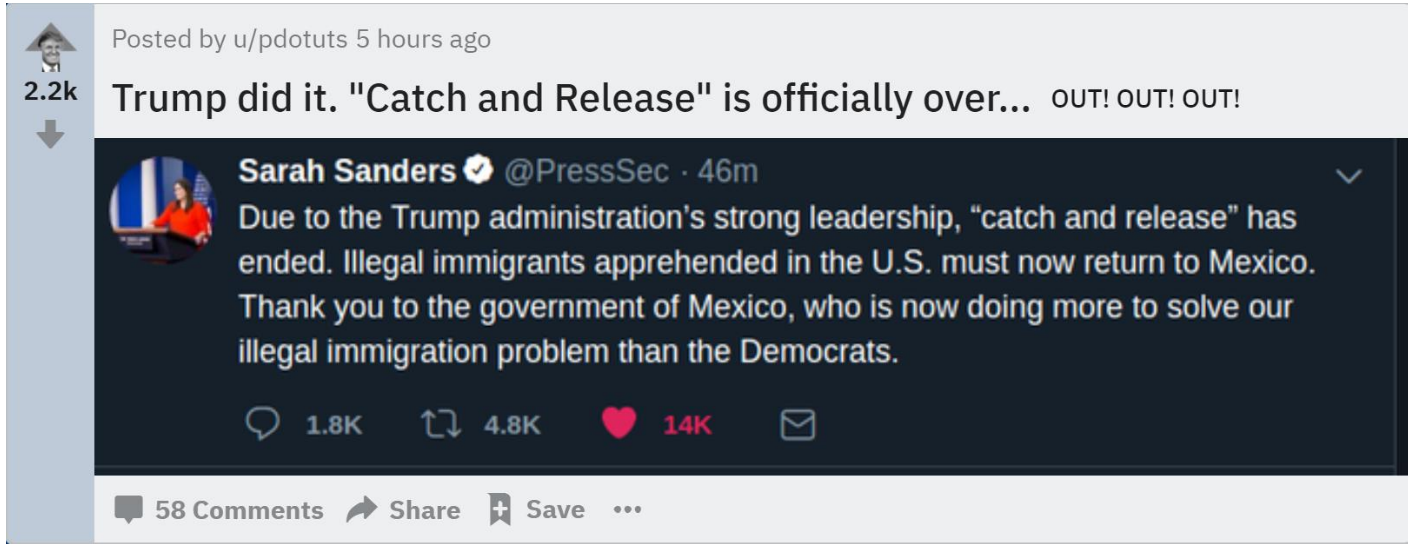
Sub-Reddit Classification using NLP



Icebreaker

Match Them Up: r/Politics or r/The_Donald ?



2.2k

Posted by u/pdotuts 5 hours ago

Trump did it. "Catch and Release" is officially over... OUT! OUT! OUT!

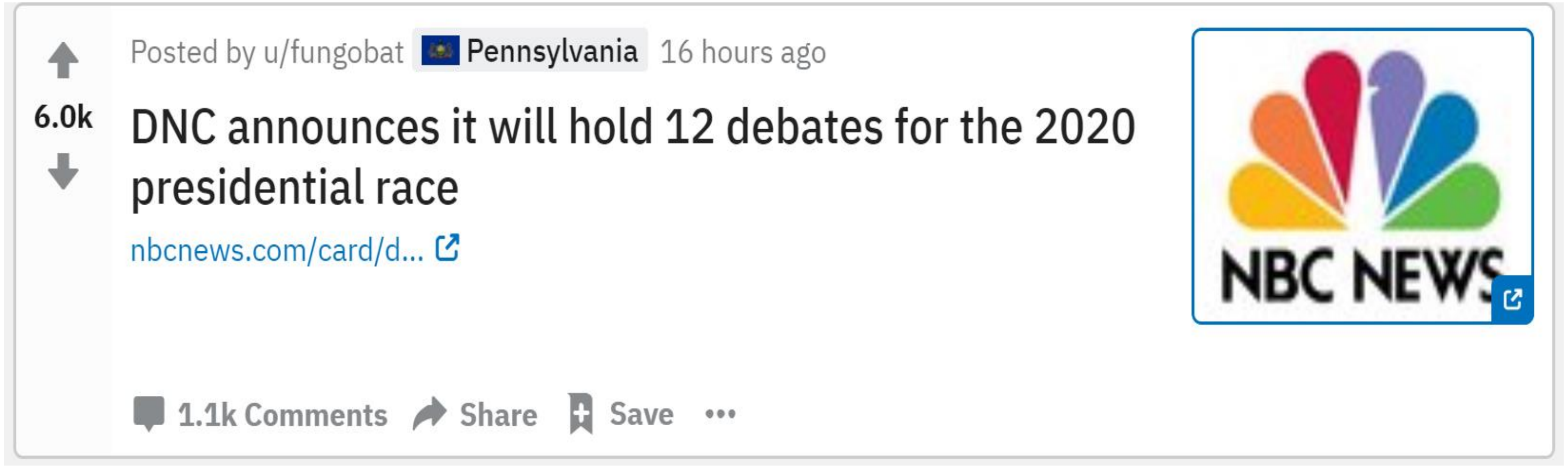
Sarah Sanders @PressSec · 46m

Due to the Trump administration's strong leadership, "catch and release" has ended. Illegal immigrants apprehended in the U.S. must now return to Mexico. Thank you to the government of Mexico, who is now doing more to solve our illegal immigration problem than the Democrats.

1.8K 4.8K 14K

58 Comments Share Save ···

vs.



6.0k

Posted by u/fungobat Pennsylvania 16 hours ago

DNC announces it will hold 12 debates for the 2020 presidential race

nbcnews.com/card/d...

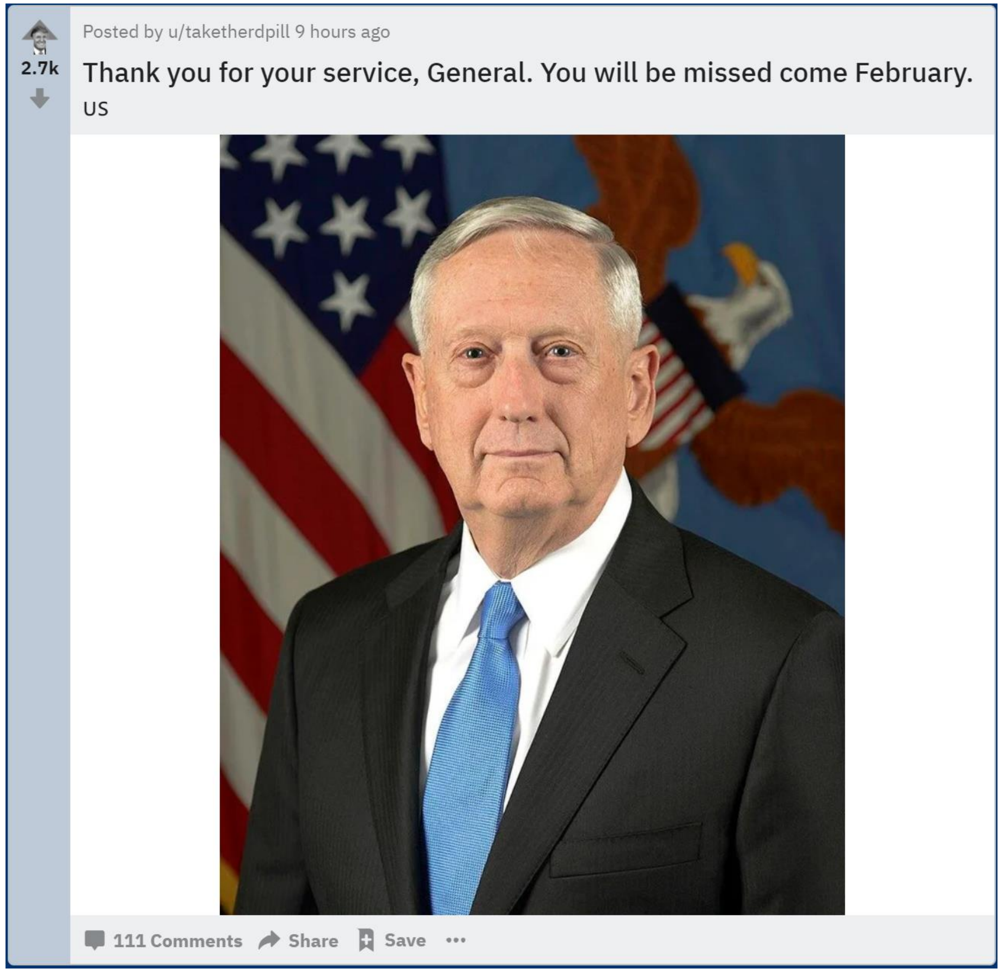
NBC NEWS

1.1k Comments Share Save ···

GENERAL ASSEMBLY

Icebreaker

What about this?


Posted by u/taketherdpill 9 hours ago

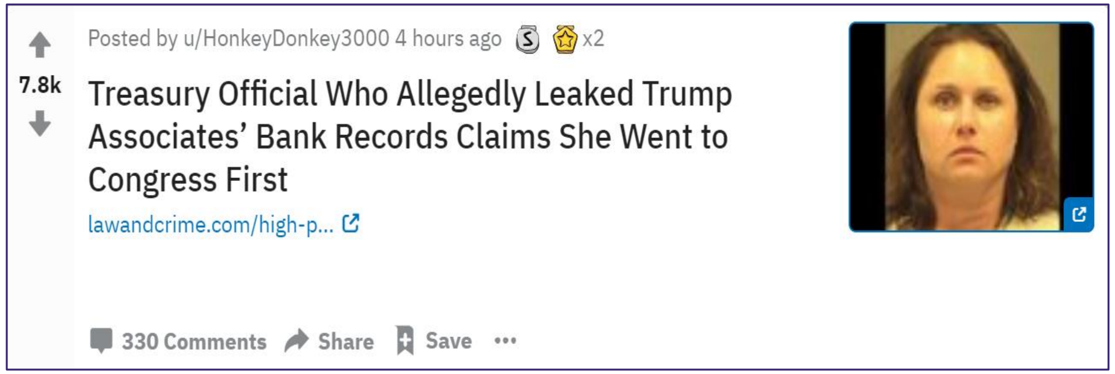
2.7k

Thank you for your service, General. You will be missed come February.

US

111 Comments Share Save •••

vs.


Posted by u/HonkeyDonkey3000 4 hours ago x2

7.8k

Treasury Official Who Allegedly Leaked Trump Associates' Bank Records Claims She Went to Congress First

lawandcrime.com/high-p...

330 Comments Share Save •••

How accurately can we classify a given post into the right sub-Reddit?

The Issue in Context

- **Moral of Exercise**: manual classification is not always quick and easy!
- **Applications:** managing the aftermath of human error / bugs that misclassify posts

The Approach

• **Collection:**	Webscrape sub-Reddits with Reddit API (500 posts each)
• **Cleaning:**	Filter out irrelevant posts; deal with strange characters.
• **Word Vectorizers**:	Count Vectorizer (EDA), TF-IDF Vectorizer (Preprocessing)
• **Classification Models**:	(1) Random Forest
	(2) Logistic Regression
	(3) Support Vector Machine

The Evaluation

• **Train/Test Split:**	Train model on 70% of all posts; testing on remaining 30%
• **Key Metric**:	Accuracy to determine the best of the three
	…vs. Baseline Accuracy (~50%)

Methodology: Exploratory Data Analysis

High-Level Statistics on Subreddits

	/r/The_Donald	/r/Politics
No. Subscribers	690,286	4,380,649
Mean Comments/Post (% of Subscribers)	38 (0.0055%)	187 (0.0042%)
Mean Score/Post (% of Subscribers)	801 (0.12%)	2,284 (0.05%)

Metrics

- **No. of Comments** and **Scores** (= upvotes – downvotes) are the few relevant numeric features from webscraped data ⟹ indicative of subreddit activity

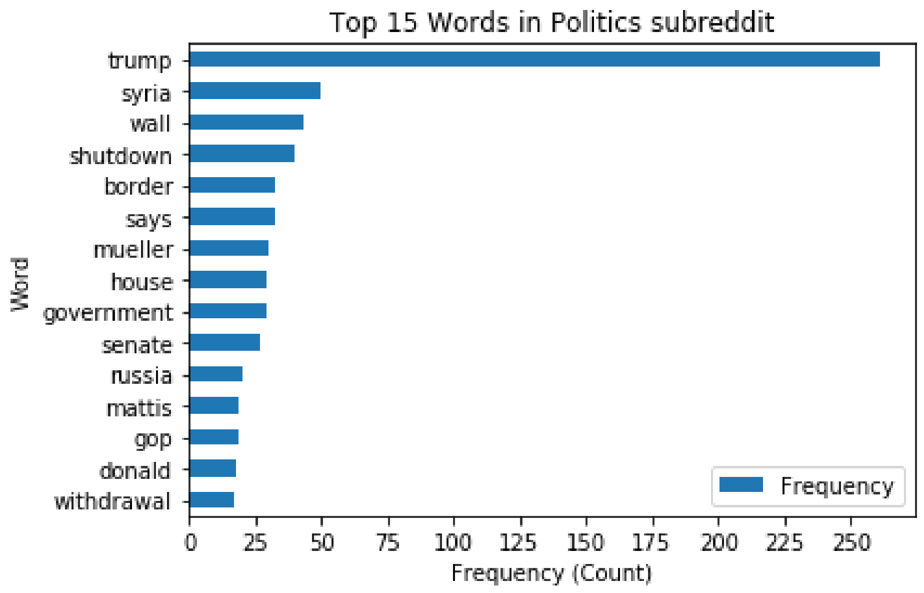
Key Insights

- **Politics** far more popular than **The_Donald** in *absolute* terms
- However, once *adjusted for size*, activity is similar…The_Donald slightly more active!
  ∴ comparable candidates to test our model on

Methodology: Exploratory Data Analysis

Examining the Distribution of Top 15 Words in /r/Politics

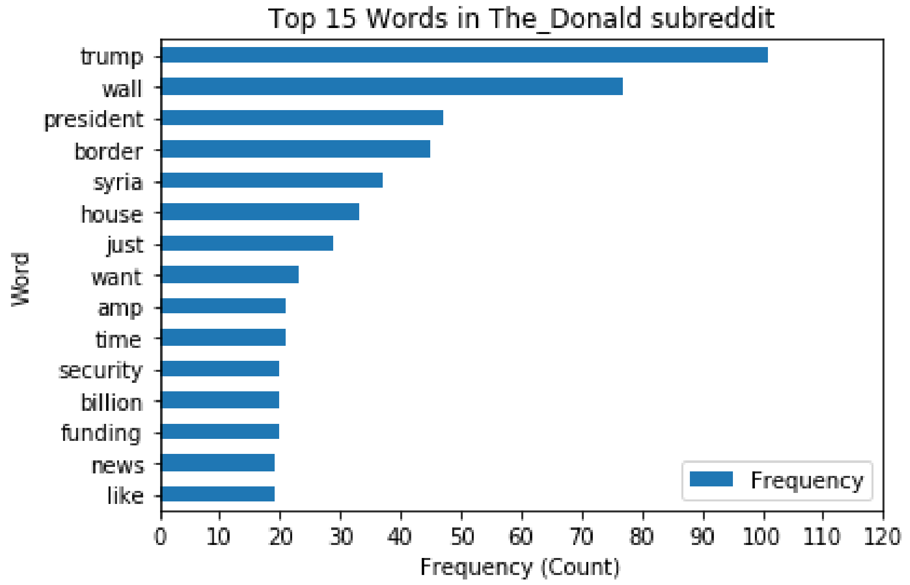
Word	Frequency in Sample
trump	261
syria	50
wall	43
shutdown	40
says	33
border	33
mueller	30
government	29
house	29
senate	27
russia	20
gop	19
mattis	19
donald	18
withdrawal	17



Methodology: Exploratory Data Analysis

Examining the Distribution of Top 15 Words in /r/The_Donald

Word	Frequency in Sample
trump	101
wall	77
president	47
border	45
syria	37
house	33
just	29
want	23
time	21
amp	21
funding	20
billion	20
security	20
like	19
news	19



Troll Filter: Using “Spam Signals” to filter out irrelevant posts

Troll Filter

Based on numeric features:

• `num_reports` > 0:	remove any posts flagged/reported at least once
• `score` < 0:	remove any posts with a net negative score *keep 0 scorers – the post may be new!*

Eliminating Residual Unicode Text from Data

- Noticed “&amp;” from The_Donald EDA
- Vectorizers get rid of “&” and “;” but “amp remains!
- Replace all instances of “&amp;” with “&” before vectorizing!

Concatenating Title and Sub-Reddit Data into Data Frame

- NLP occurs on *combined body of text* from all subreddits so merge
- Binarize Target Variable (Sub-Reddit) s.t. The_Donald = 1, Politics = 0

Model 1: Random Forest

TF-IDF Vectorizer Optimized Parameters:

- max_features = 1,500
- ngram_range = (1,1)
- norm = ℓ_1

Random Forest Classifier Optimized Parameters:

- n_estimators = 20
- max_depth = 20

Model 2: Logistic Regression

TF-IDF Vectorizer Optimized Parameters:

- max_features = 2,500
- ngram_range = (1,1)
- norm = ℓ_1

Logistic Regression Optimized Parameters:

- penalty = ℓ_2
- C = 10

GENERAL ASSEMBLY

Model 3: Support Vector Machine (SVM)

TF-IDF Vectorizer Optimized Parameters:

- max_features = 1,500
- ngram_range = (1,1)
- norm = ℓ_1

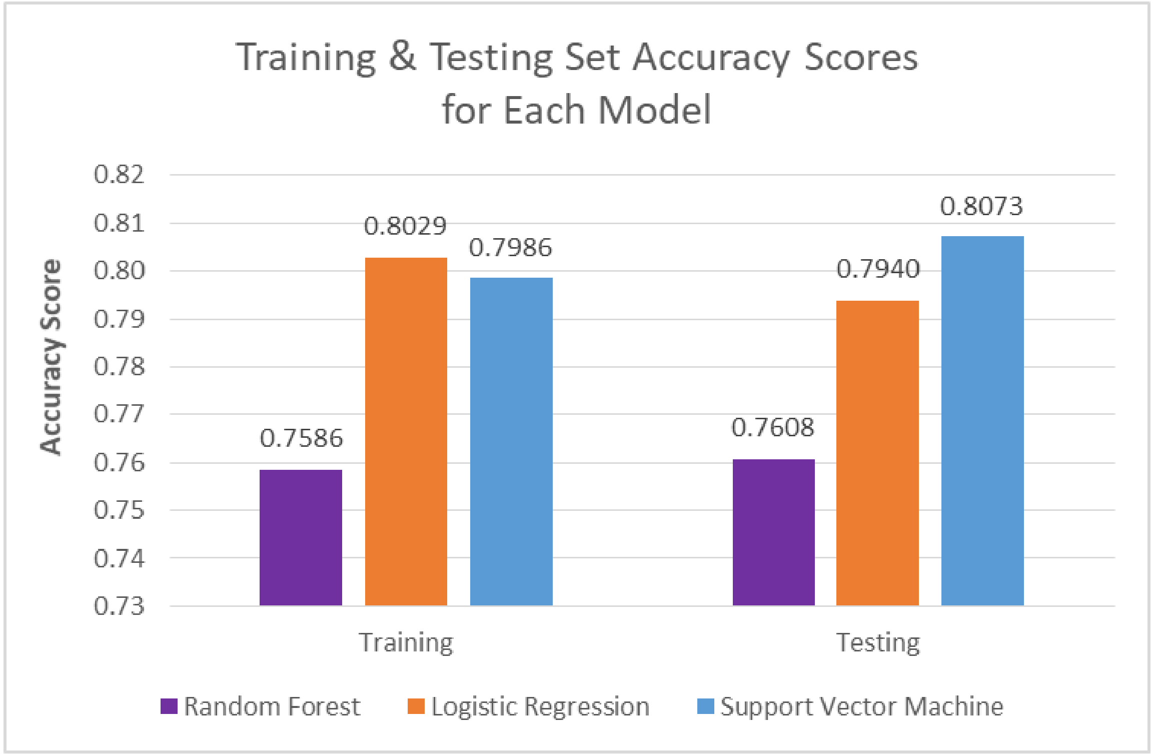
SVM Optimized Parameters:

- C = 10
- kernel = linear
- gamma = auto

Modeling: Results & Evaluation

Train Accuracy vs. Test Accuracy

	Model 1: Random Forest	Model 2: Logistic Regression	Model 3: SVM
Training Accuracy	0.7586	**0.8029**	0.7986
Testing Accuracy	0.7608	0.7940	**0.8073**
Overfit?	no	slightly	no



Key Insights

- SVM does best in Testing set
- Logistic Regression does best in Training set, though results in very slight overfit
- Otherwise, virtually no overfit in models

GENERAL ASSEMBLY

Modeling: Evaluation (Supplement)

Confusion Matrix

Random Forest	Predict Politics (y = 0)	Predict The_Donald (y = 1)
Actual Politics (y = 0)	100	51
Actual The_Donald (y = 1)	21	129

Logistic Regression	Predict Politics (y = 0)	Predict The_Donald (y = 1)
Actual Politics (y = 0)	122	29
Actual The_Donald (y = 1)	33	117

Support Vector Machine	Predict Politics (y = 0)	Predict The_Donald (y = 1)
Actual Politics (y = 0)	125	26
Actual The_Donald (y = 1)	34	116

GENERAL ASSEMBLY

Classification Evaluation Metrics

	Model 1: Random Forest	Model 2: Logistic Regression	Model 3: Support Vector Machine
Accuracy	0.7608	0.7940	0.8007
Sensitivity	0.8600	0.7800	0.7733
Specificity	0.6623	0.8079	0.8278
Precision	0.7167	0.8014	0.8169

Key Insights

- **Overall:** all three models performed well (Accuracy > 0.75)
- Support Vector Machine (SVM) outperformed all other models in terms of Accuracy, Specificity (True Negative Rate), and Precision
- Random Forest was the worst performer in all areas, except Sensitivity.

Key Takeaways

- All beat baseline accuracy (50%)
- All accuracy > 75% $\Longrightarrow$ good ability to accurately classify posts
- Clear winner: **Support Vector Machine** model (4 in 5 correct)
  - **Implication**: Reddit administrators need to review 1 in 5 misclassified posts
  - Tuning improvement possible, but SVM very computationally intensive

Further Enquiry

- Group synonyms together as features instead of individual words
- Take into consideration words that are all caps (indicate shouting/anger), but still apply .lower to words where first letter is capitalized