Predicting Factuality of Facebook Posts

Shota Yasunaga, Justin Lauw, Madison Hobbs

Overview

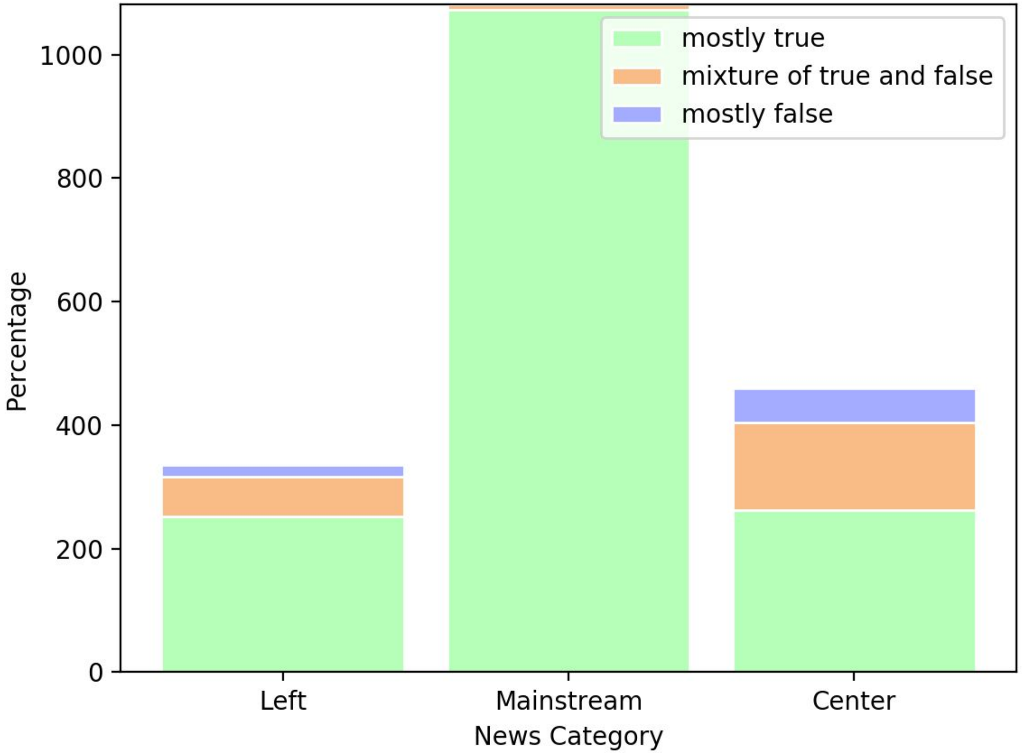
- ***Motivation***
- Data Exploration & Insights
- Pre-Processing
- Results
- Conclusion

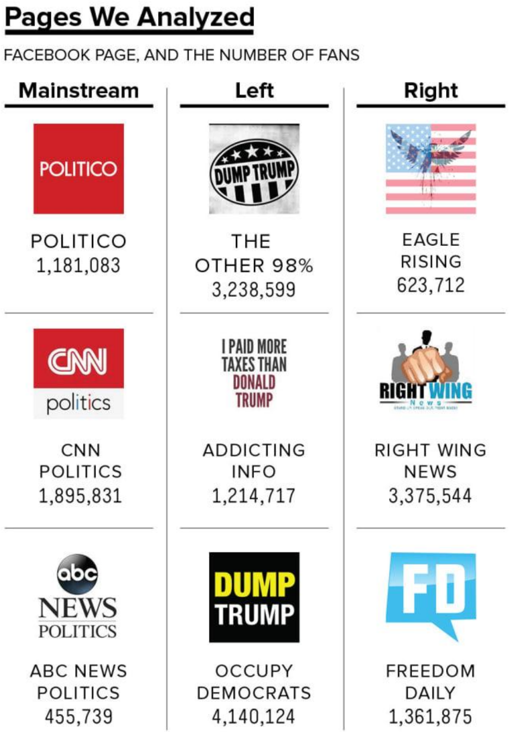
Motivation

Overview

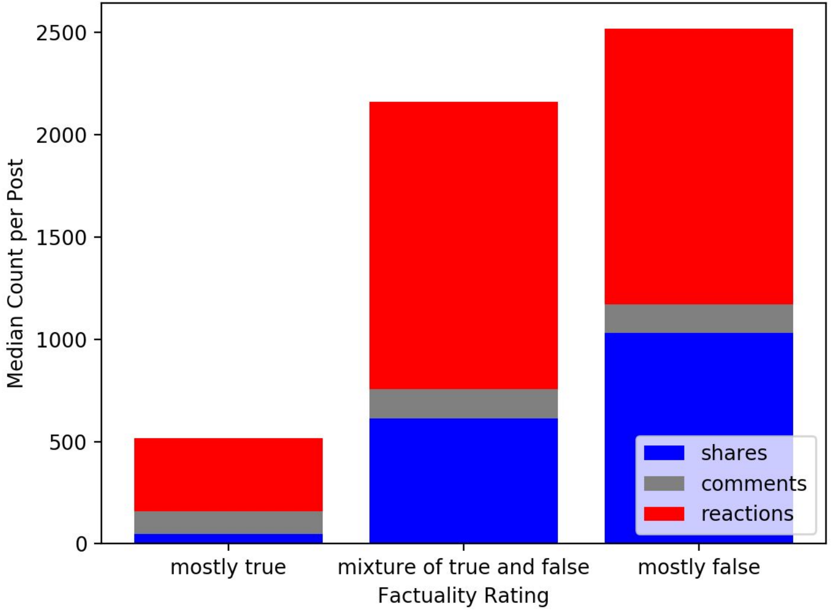
- Motivation
- ***Data Exploration & Insights***
- Pre-Processing
- Results
- Conclusion

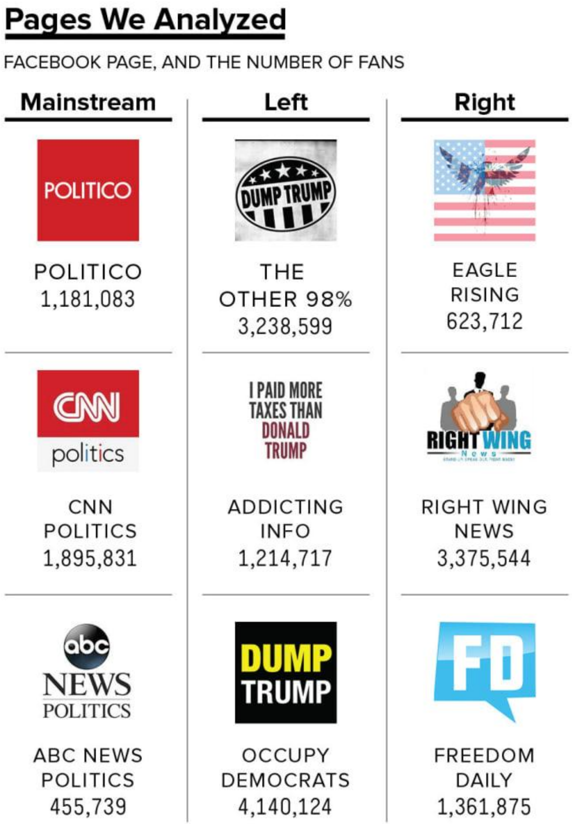
Mainstream news is mostly true
Dataset is unbalanced





False Posts are Popular

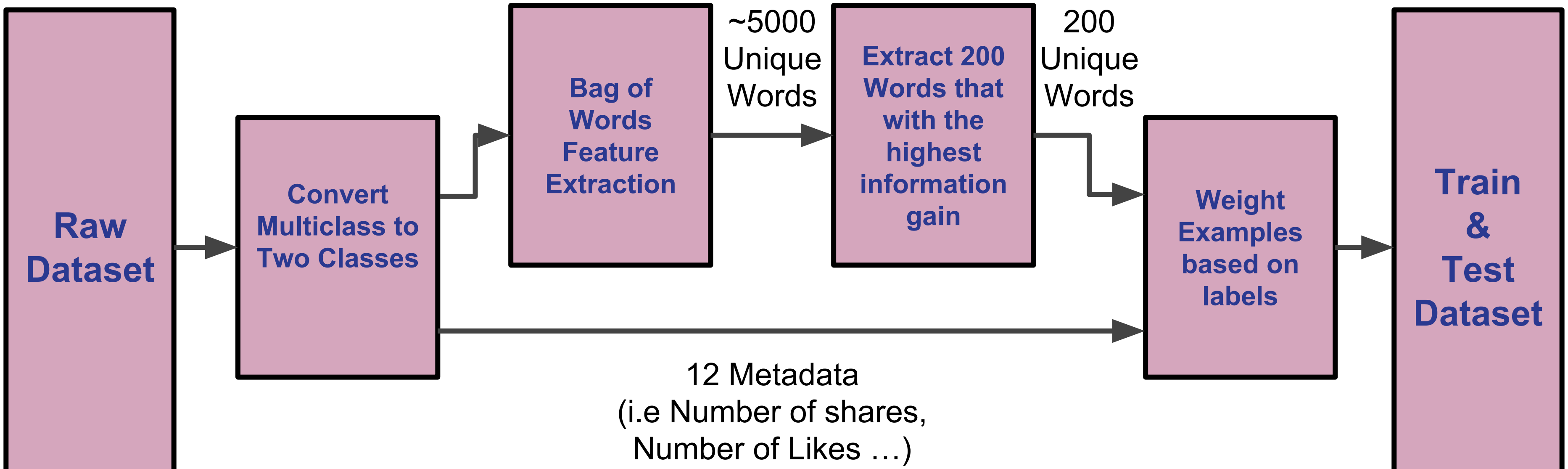




Overview

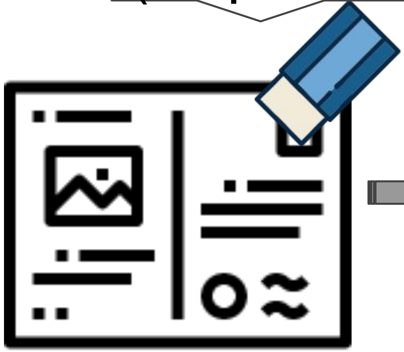
- Motivation
- Data Exploration & Insights
- ***Pre-Processing***
- Results
- Conclusion

Data Preprocessing (Feature Selection)

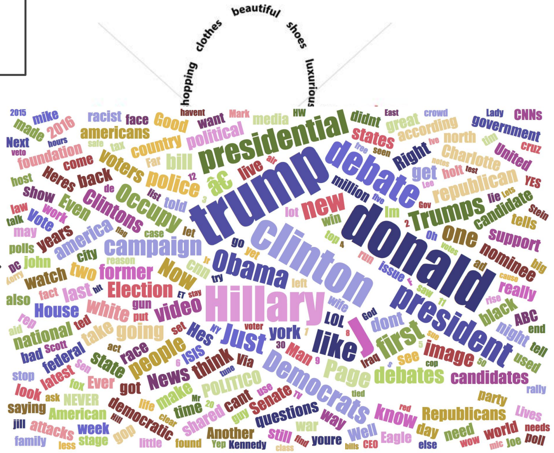


Feature Setup

, . an, a, in
(stop words)



Removed Punctuation
stop words

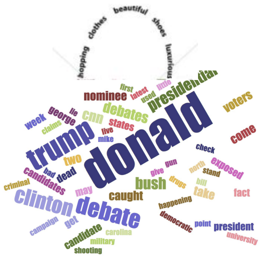


Bag of words

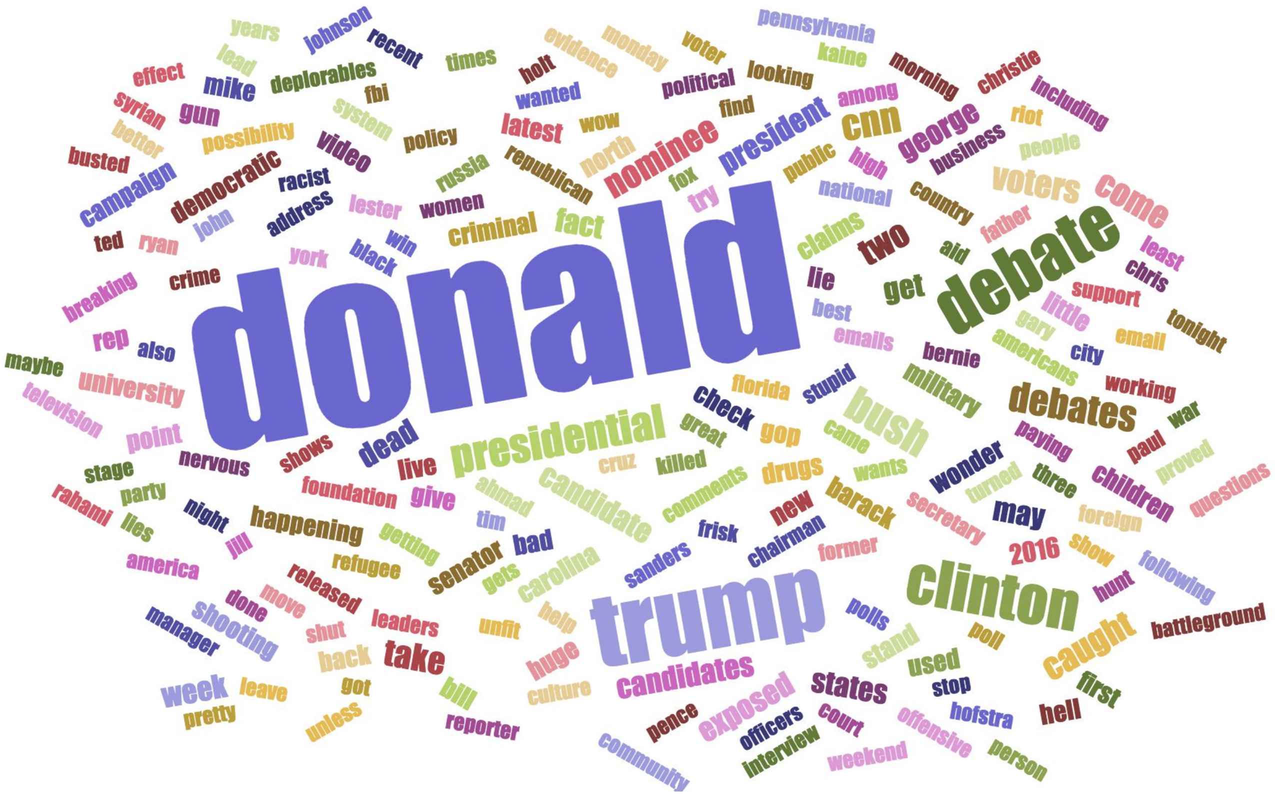
Choose 200 words based on entropy

Meta Data

+



Feature Importance (1st order information gain)



Depends on frequency of appearance

Alternative: 200 most common words, but choosing features this way performs better

Class Weights & Preprocessing

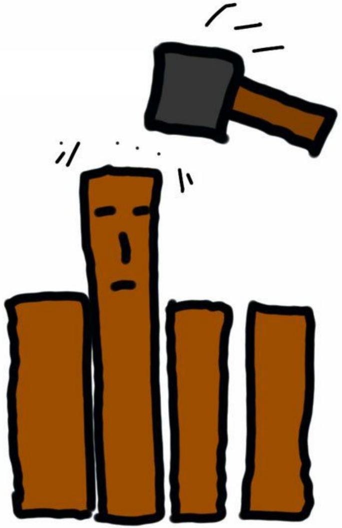
5 1

Mostly true

Mostly false
Mixture of true/false

Class weights

Feature Normalization



Overview

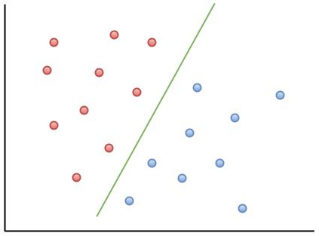
- Motivation
- Data Exploration & Insights
- Pre-Processing
- ***Results***
- Conclusion

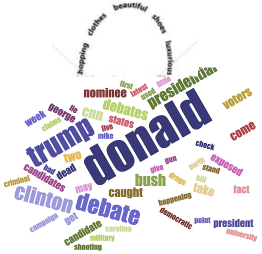
Experimental Process

1

Data Exploration

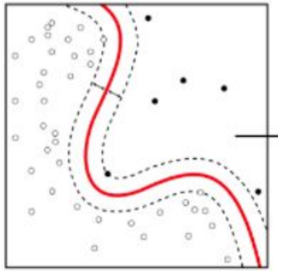
2





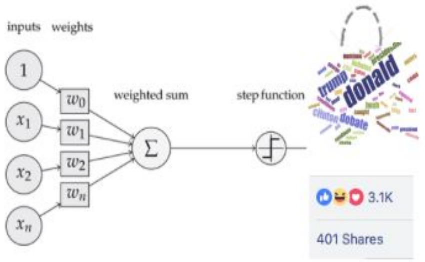
Linear SVC with bag of words
4 classes
2 classes

3



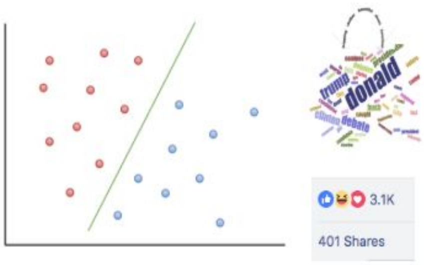
Polynomial SVC with meta data
4 classes

4



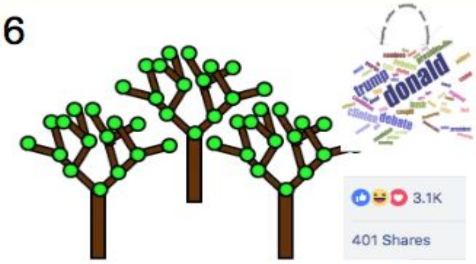
Perceptron with both
2 classes

5

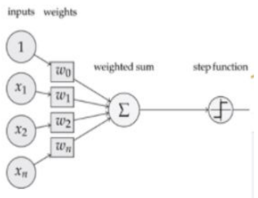


Linear SVC with both
2 classes

6



Random Forest
2 classes



Perceptron (high bias)

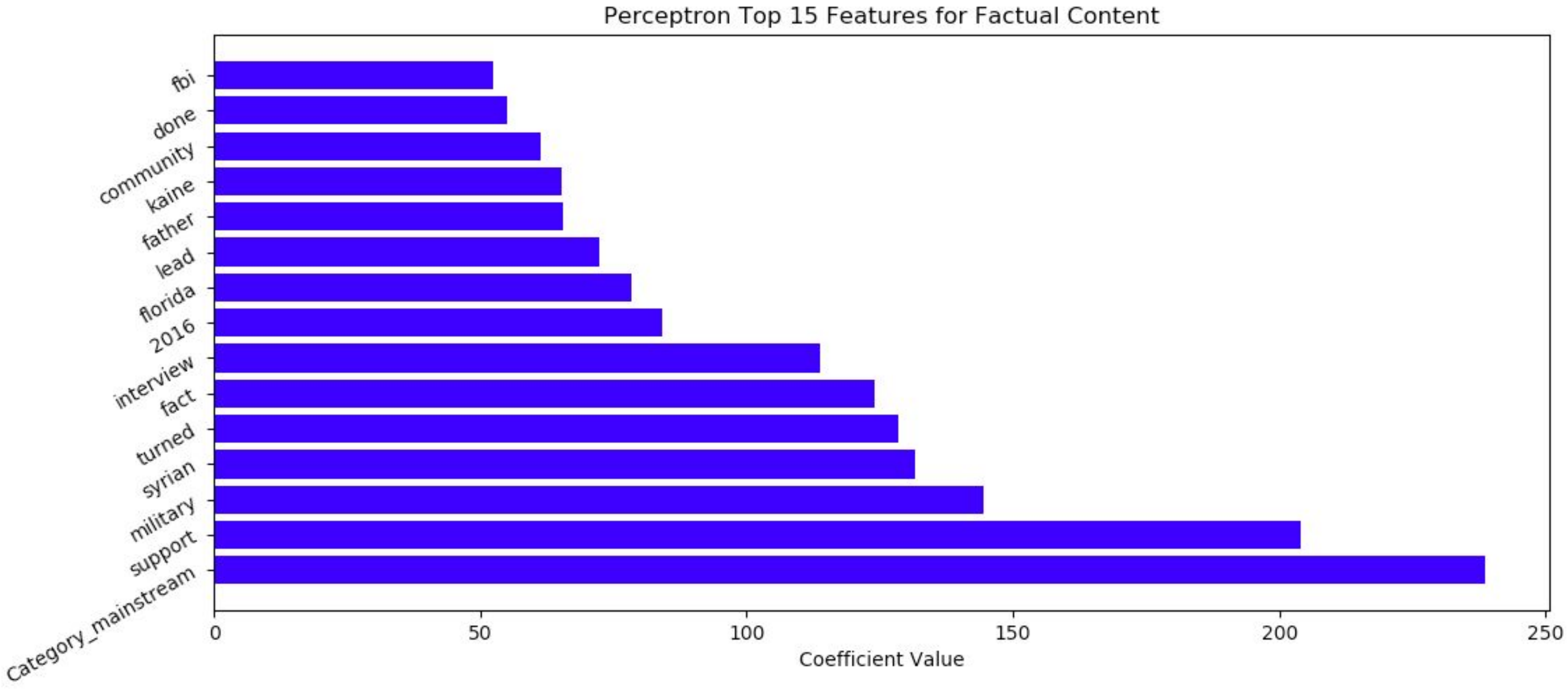
	Actually Negative	Actually positive
Predicted Negative	TN = 34	FN = 24
Predicted Positive	FP = 33	TP = 285

5-fold CV tuned with f1 score

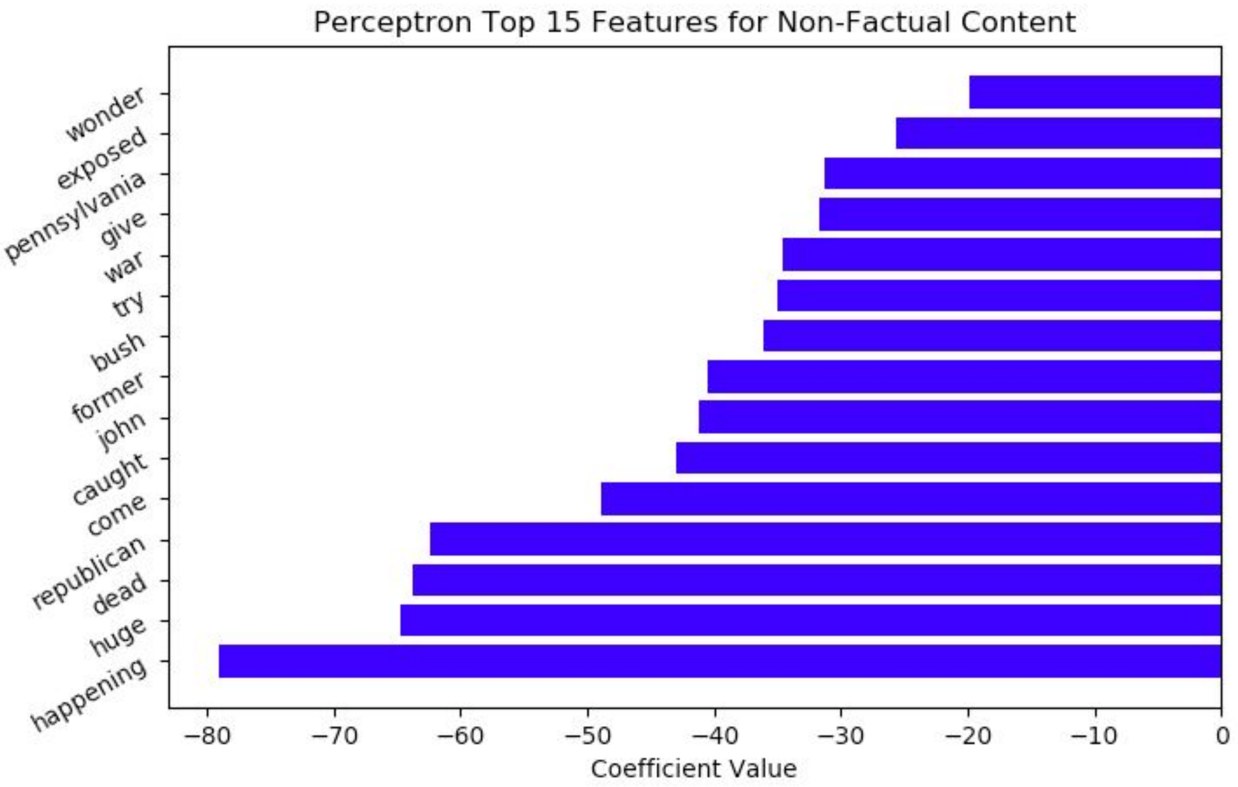
Training f1 score: 0.868
Test f1 score: 0.853

Training accuracy score: 0.867
Test accuracy score: 0.848

Perceptron Feature Importance



Perceptron Feature Importance



"Not Mostly Factual Content"	"Mostly Factual Content"
Popularity: 1. num_reactions 7. num_shares News Source: 2. Category_right 5. Category_left Negative Words: 8. war 14. lies 15. least 19. bad 25. hell Race-based: 13. Black	Popularity: 1. num_likes 3. num_angrys 4. num_loves 5. num_hahas 6. num_sads 20. num_wows News Source: 2. Category_mainstream Negative Words: 18. stop Positive Words: 9. support 15. fact 21. community

Linear SVC

(slightly lower bias)

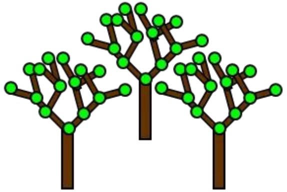
Training F1: 0.873

Test F1: 0.842

Training accuracy: 0.857

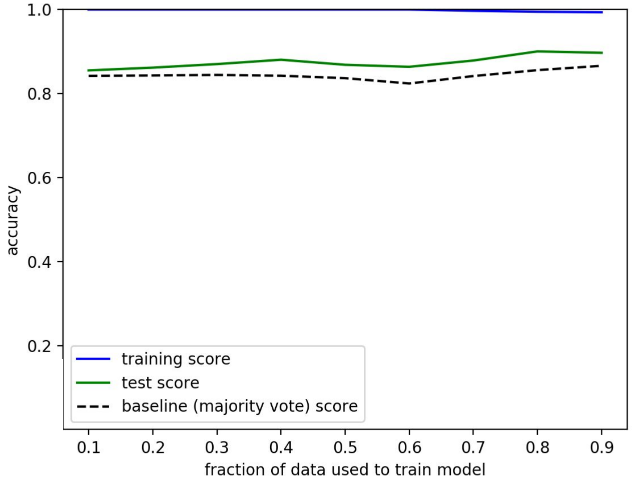
Test accuracy: 0.822

	Actually Negative	Actually positive
Predicted Negative	TN = 28	FN = 30
Predicted Positive	FP = 17	TP = 301



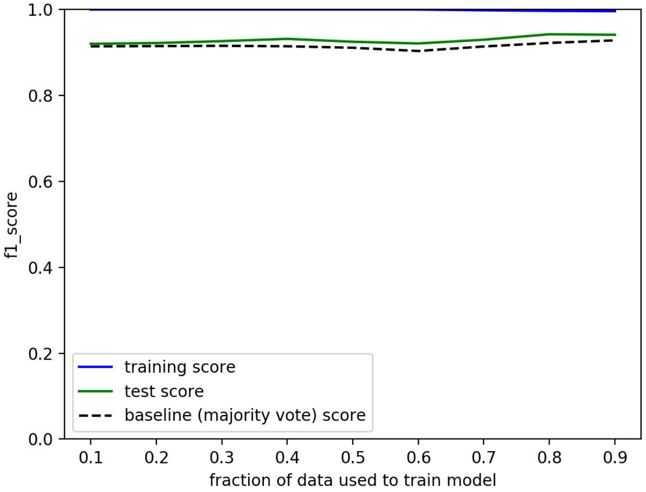
Random Forest

(high variance)



Training f1 score: 0.998
Test f1 score: 0.868
Best CV f1 score: 0.872

Training accuracy: 0.995
Test accuracy: 0.894

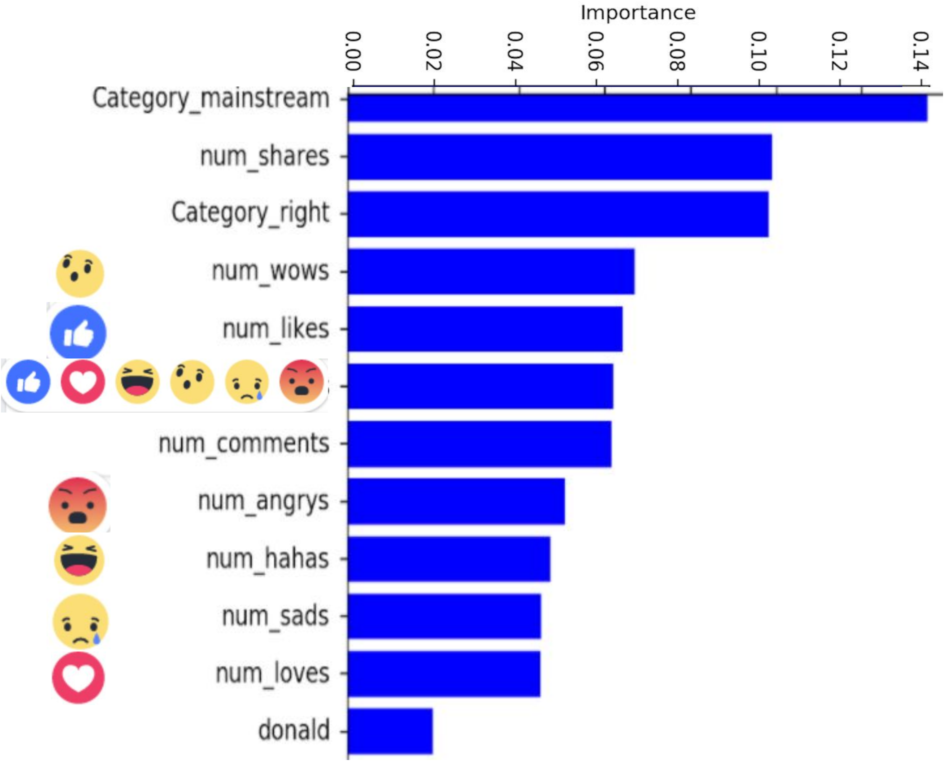


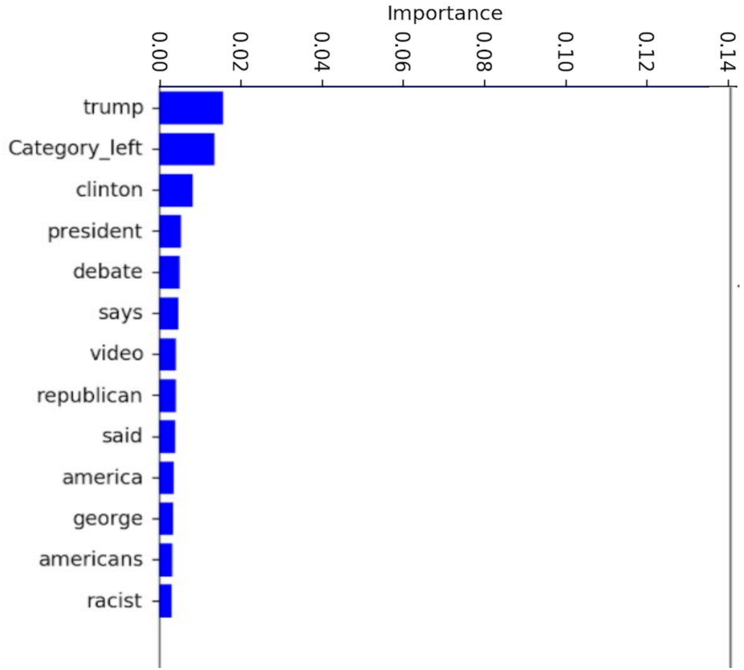
	Actually Negative	Actually positive
Predicted Negative	TN = 28	FN = 30
Predicted Positive	FP = 17	TP = 301

5-fold CV tuned with f1 score

```
bootstrap = True,
min_samples_leaf = 1,
n_estimators =  400,
max_features = 'sqrt',
min_samples_split = 5,
max_depth = 100
```

Random Forest Insights ~ Feature Importance





Random Forest - Metadata Only

5-fold CV tuned with f1 score

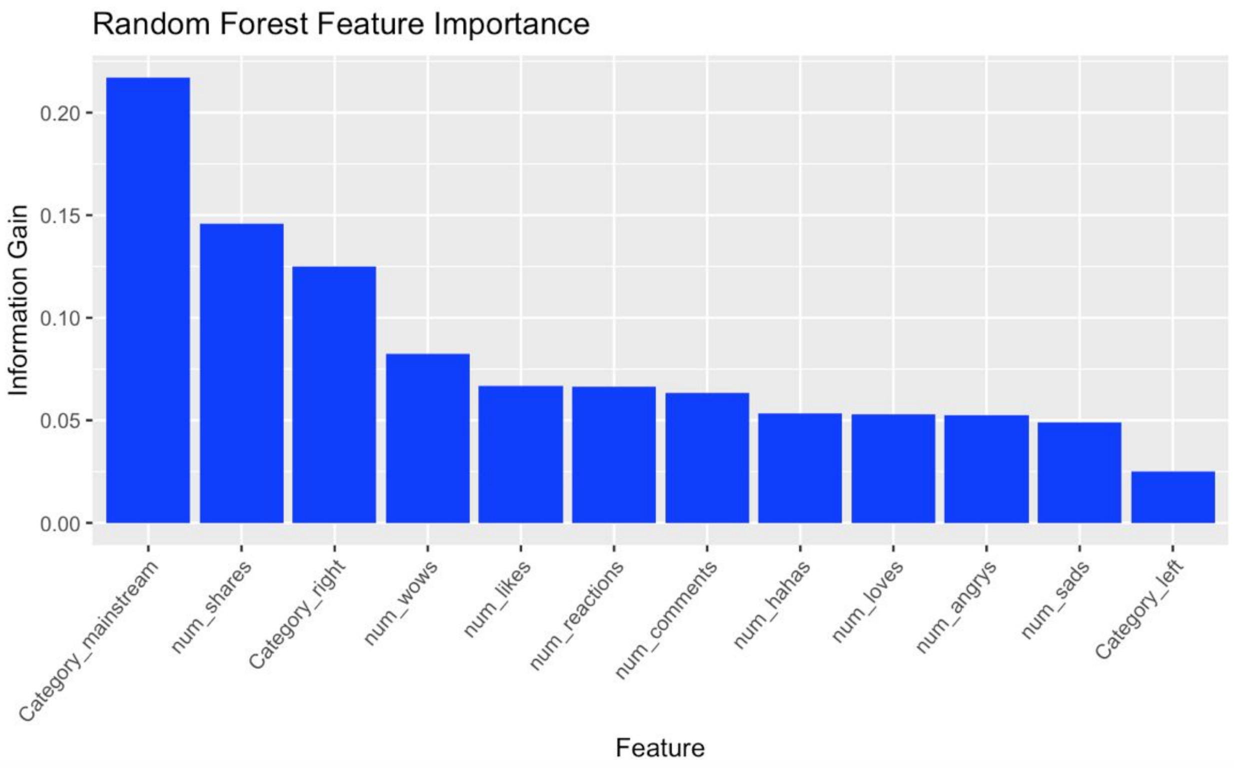
```
bootstrap = True,
min_samples_leaf = 1,
n_estimators =  1600,
max_features = 'sqrt',
min_samples_split = 10,
max_depth = None)
```

Training f1 score: 0.975
Test f1 score: 0.872
Best CV f1 score: 0.856

Training accuracy: 0.973
Test accuracy: 0.872

	Actually Negative	Actually positive
Predicted Negative	TN = 41	FN = 17
Predicted Positive	FP = 31	TP = 287

Random Forest - Metadata Only

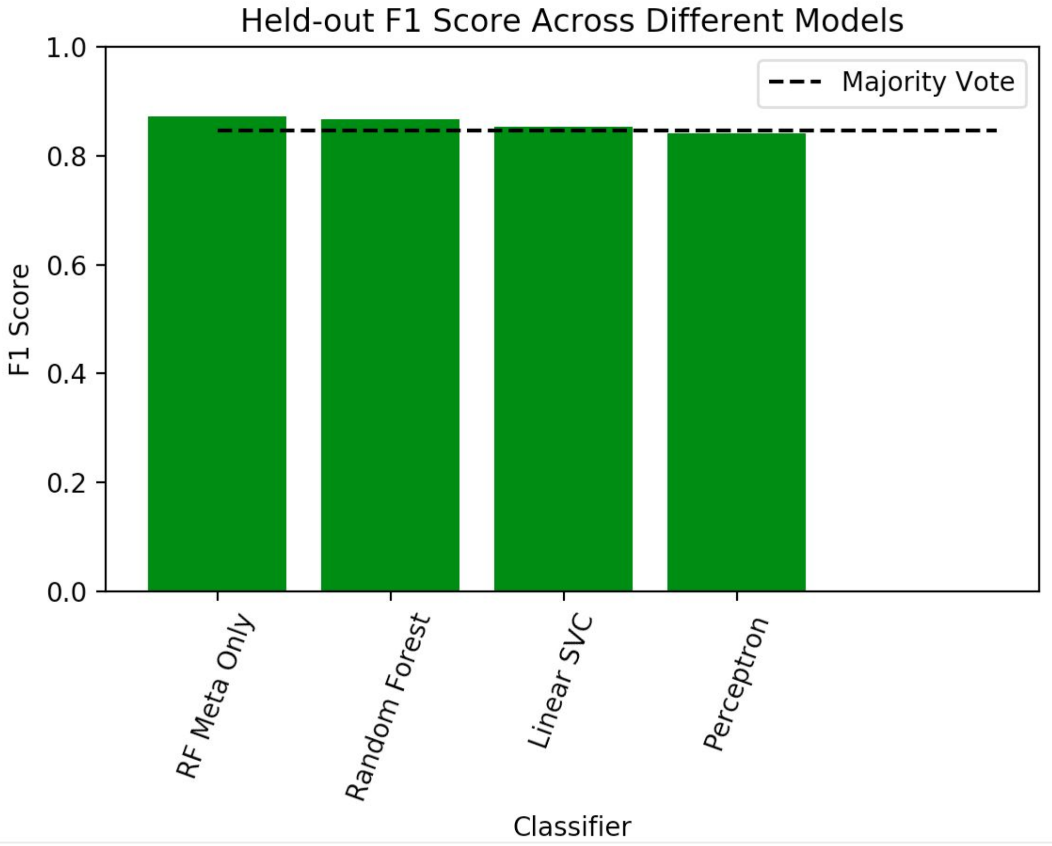


Overview

- Motivation
- Data Exploration & Insights
- Pre-Processing
- Results
- ***Conclusion***

Discussion

- High Bias Models > High Variance Models
- Meta-data > Bag of Words
  - Probably because factuality detection based on words is a lot more complicated than what we can do with 2000 examples
- Popularity & Source's Political Leaning are Most Important Predictors



Limitations	Mitigation
Unbalanced Data	Get more examples of far left and far right posts
No stemming used in bag of words	Use stemming
5-fold CV	Multiple trials of 10-fold CV
Only used some ML models	Try other models
Small data set (<2000 posts)	Get more data
Restricted time-frame: seven weekdays (9/19 - 9/23 and 9/26 - 9/27 2016)	Gather posts from wider time range
Buzzfeed source/labeling factuality and news source category	Label posts ourselves or get unbiased panel

Future Work

- Perfect a broader model trained on more representative data
  - Equal number of left, right, and mainstream posts
  - Wider time frame or sampled across multiple time frames
- Leverage the full capacity of NLTK
- More exhaustive model search
  - Finer CV grid and mutliple runs of tuning
- Focus on specificity (we don’t want to true to be classified as false)
- Predict more than two classes
- Predict percentage of factual content in a news post
- Multilayer Perceptron
- Naive Bayes
- Metadata with KNN

Questions?

How did Buzzfeed select the posts?

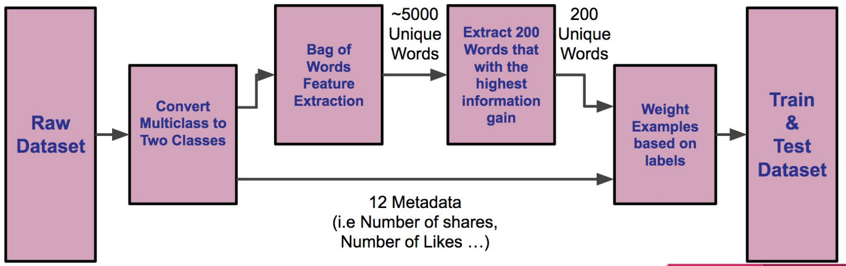
What train-test split was used?

Why not return to poly and rbf SVC?

How did you choose those machine learning modells over others?

Did you use stratified sampling?

Data Preprocessing

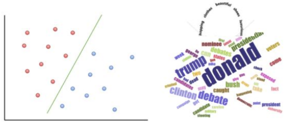


Experimental Process

1

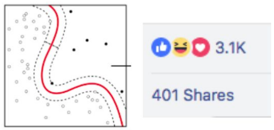
Data Exploration

2



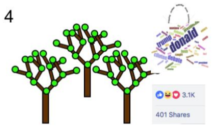
Linear SVC with bag of words
4 classes
2 classes

3



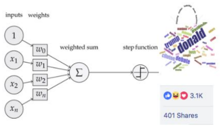
Polynomial SVC with meta data
4 classes

4



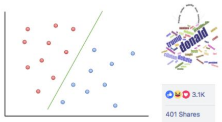
Random Forest with both
2 classes

5



Perceptron with both
2 classes

6



Linear SVC with both
2 classes