Capstone #2 -- Examining and Predicting Fake News

Introduction

The problem of fake online news is a persistent concern in contemporary society, impacting politics and society at large. While the Internet enables access to a wealth of information, it is also a medium by which disinformation can be easily spread. In particular, major websites with user-generated content have been met with harsh criticism and calls for legal action due to fake news being circulated on their platforms. Large websites with user-generated content can make use of machine learning to quickly identify sources as being potentially suspect or reliable.

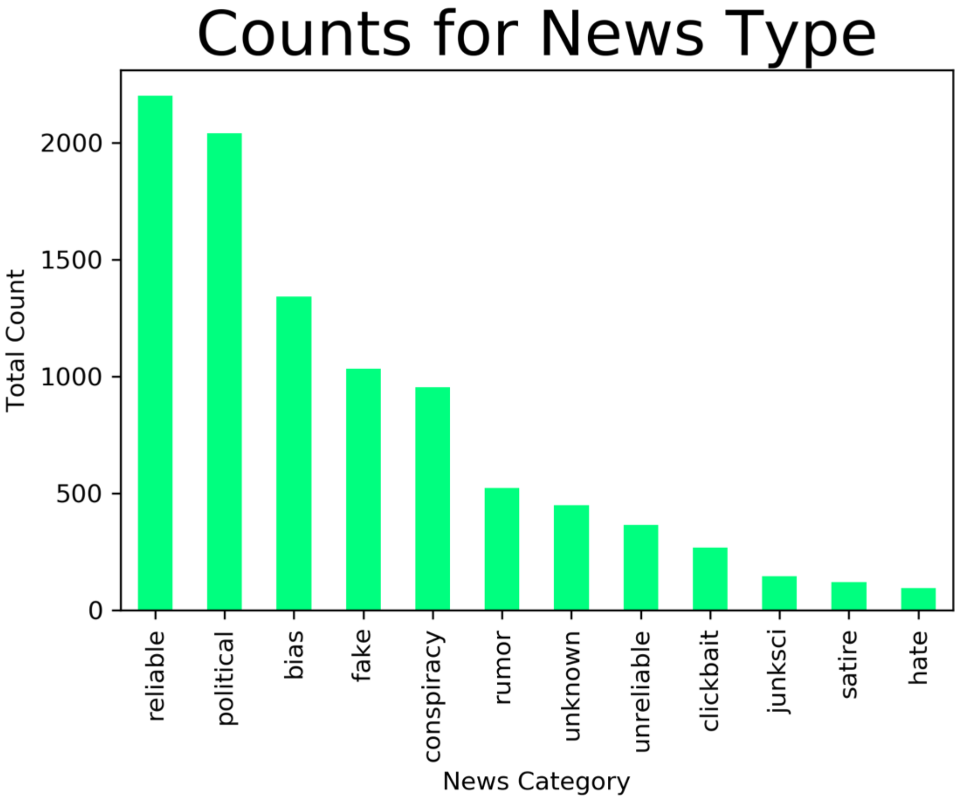
Many NLP and machine learning scholars have researched the increasingly important problem of online disinformation and have developed highly-technical, insightful approaches and analyses. In this project, I examine the problem of fake news classification, by analyzing a large dataset of scraped news articles using various Python libraries. Using the large fake news dataset scraped by Maciej Szpakowski available at https://github.com/several27/FakeNewsCorpus , in this project a subset of news articles are sampled from the corpus and text analysis is performed on them.

Data Cleaning and Pre-processing

Initial sampling and EDA

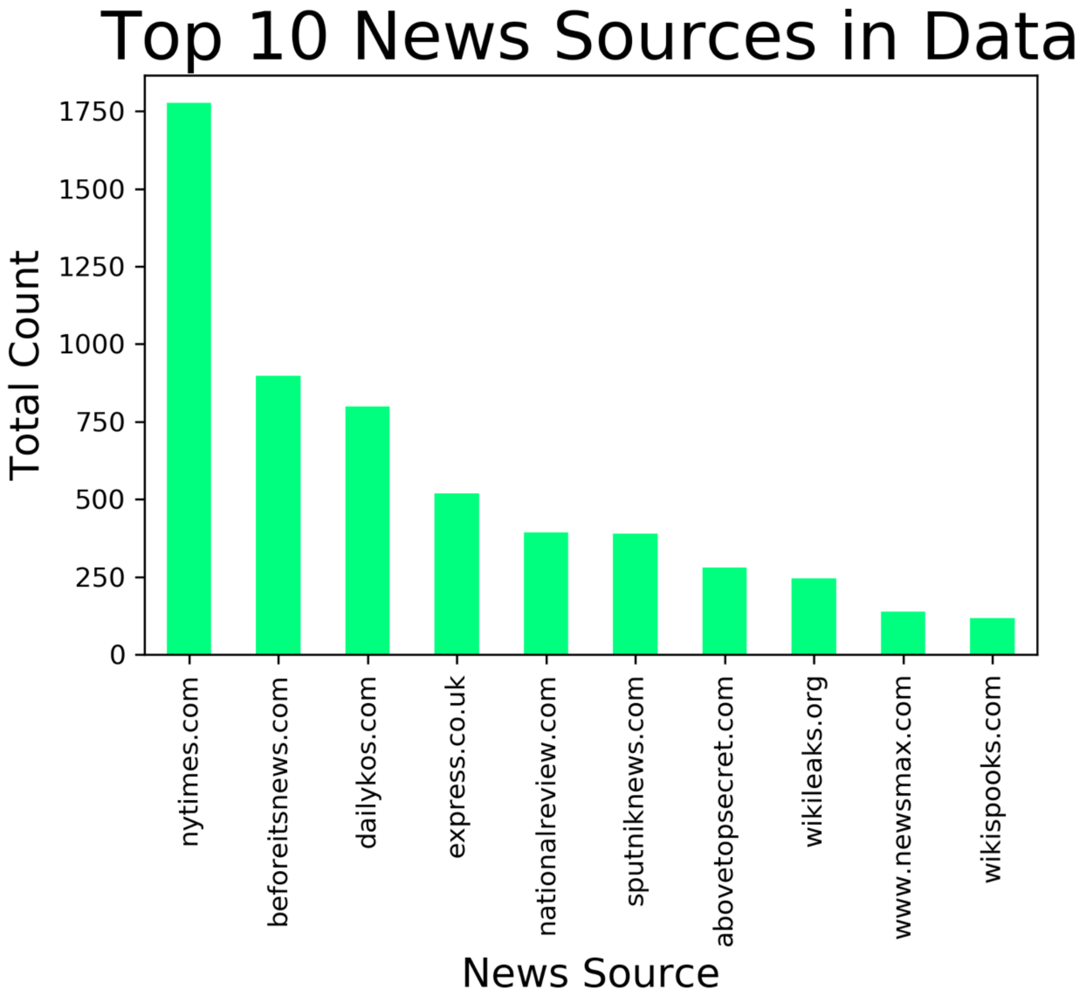
The corpus used includes over 20 million articles. Because of considerations regarding time and processing power, I first decided to sample 10,000 articles for my analysis. While performing initial EDA on the corpus, it became apparent that all articles were categorized into 12 different news types (Fig. 1).

Fig. 1



Upon inspecting each of these news types, some issues began to emerge. Namely, all articles from any given source were given a particular label without consideration of individual articles. For example, all articles from nytimes.com were labelled as “reliable,” all articles from “beforeitsnews.com” were labelled as ‘fake’, and all articles from “sputniknews.com” were labelled as “bias”. Figure 2 shows the top ten news sources from particular web domains represented in the dataset.

Fig. 2



While the vast majority of articles from The New York Times can likely be considered reliable, some of the labelling of other various news sources seems to present some issues in the data. For example, all articles from ‘dailykos.com’ were labelled as ‘political’ (Fig. 5), all articles from ‘express.co.uk’ were labelled as ‘rumor (Fig. 6)’, and all articles from ‘sputniknews.com’ (Fig. 7) were labelled as ‘bias.’ Is every single article from each of these sources inherently more political or biased than articles from sources labelled as ‘reliable’? Because of the issues presented with the labelling, I decided to resample from the original dataset.

Fig. 3

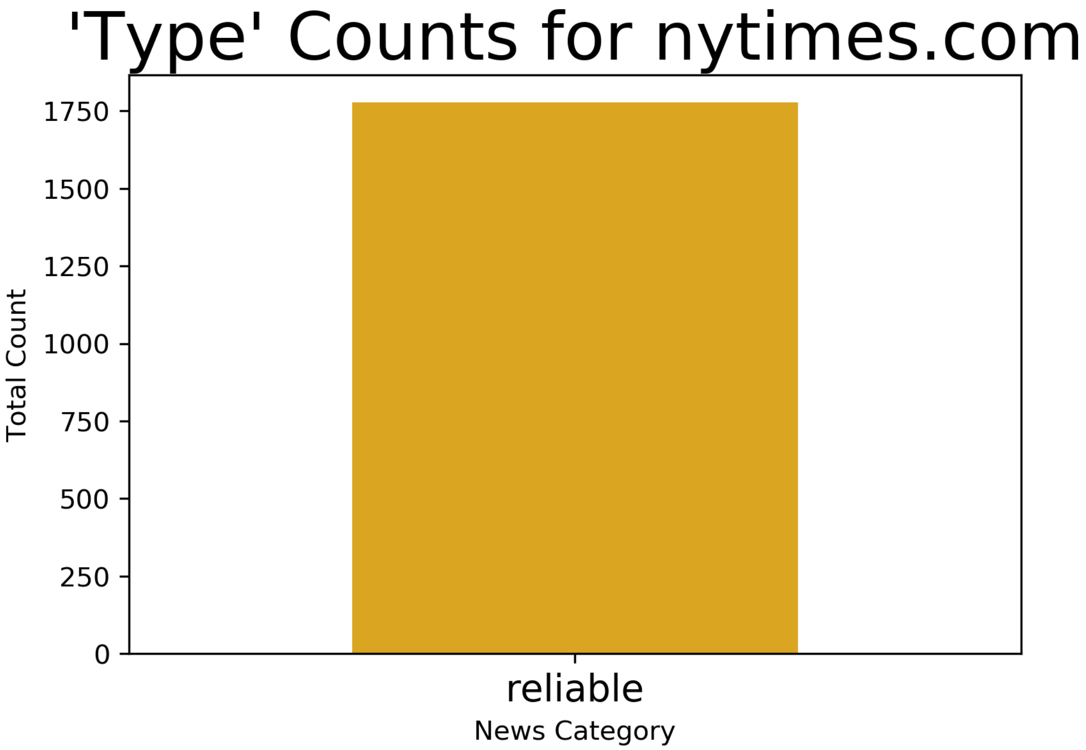

'Type' Counts for nytimes.com
Total Count
0
250
500
750
1000
1250
1500
1750
reliable
News Category

Fig. 4

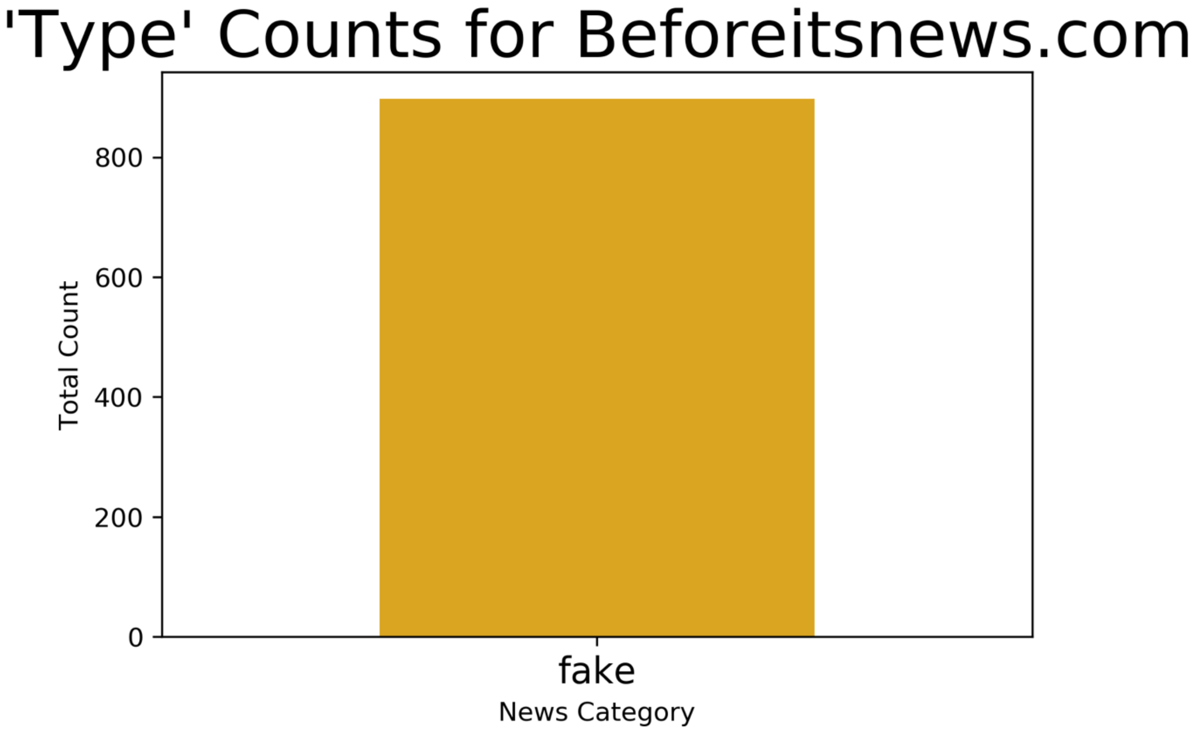

'Type' Counts for Beforeitsnews.com
Total Count
0
200
400
600
800
fake
News Category

Fig. 5

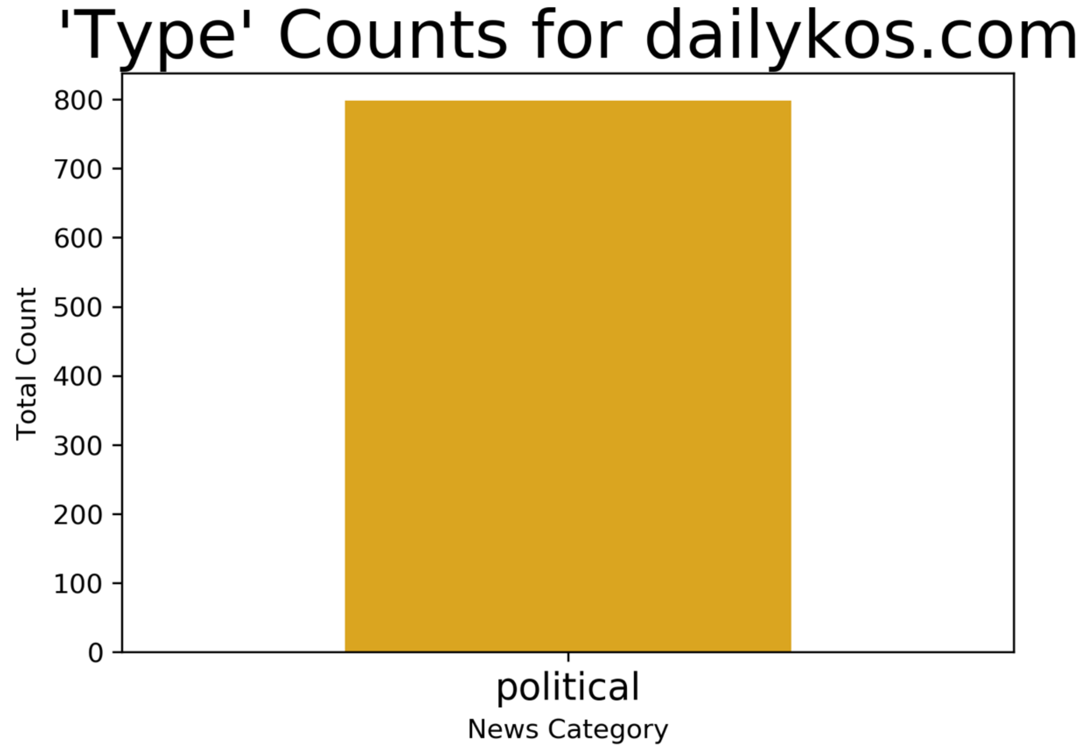

'Type' Counts for dailykos.com
Total Count
0
100
200
300
400
500
600
700
800
political
News Category

Fig. 6

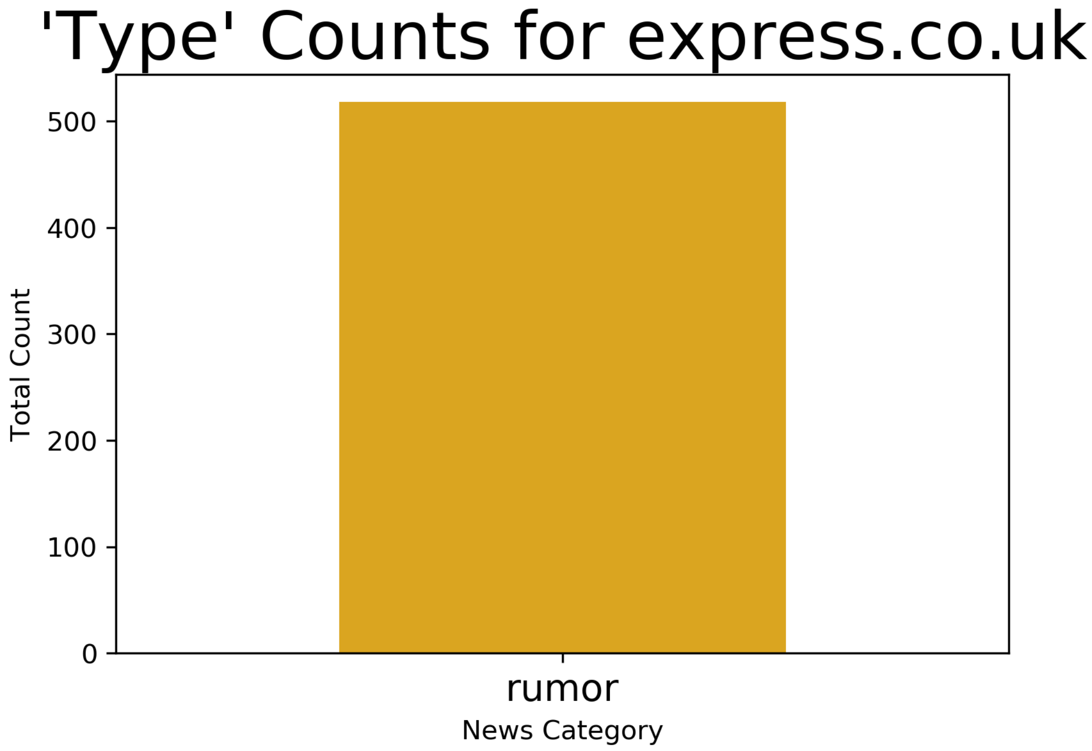
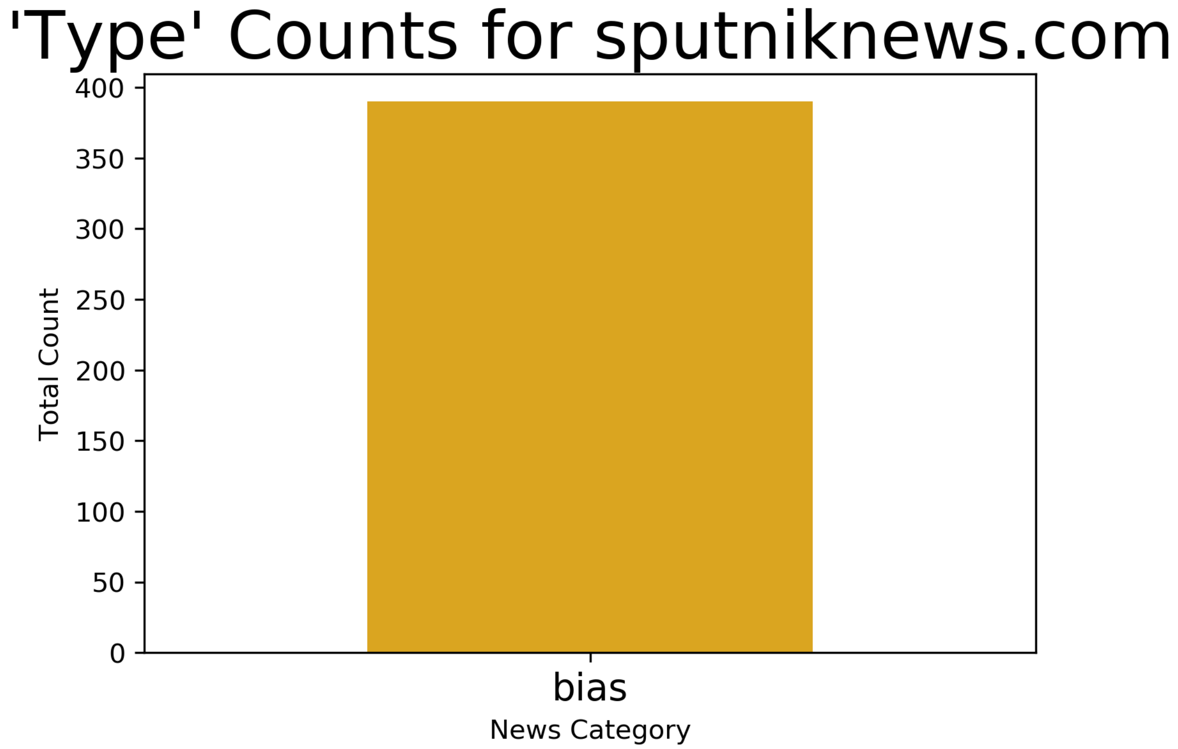

'Type' Counts for express.co.uk
Total Count
0
100
200
300
400
500
rumor
News Category

Fig. 7



Initial sampling

Due to the inherent biases in the dataset for the labelled categories ‘junksci’, ‘bias’, etc., it was initially decided to resample the dataset by including data from the two categories ‘fake’ and ‘reliable’. Although there are issues with designating all articles from certain sources with the labels ‘fake’ or ‘reliable’, the sources used with these labels seem to be more consistent than with the other labels -- nytimes.com is, for the most part, reliable, whereas, a major website represented in the data, beforeitsnews.com, is not.

EDA of Sampled Data

The sampled data contains 10,000 articles labelled ‘reliable’ and 10,000 labelled ‘fake’. The majority of these 20,000 articles were from nytimes.com (labelled as ‘reliable’) and beforeitsnews.com (labelled as ‘fake’). There were, however, 141 different online news sources represented in the data (Fig. 9).

Fig 8.

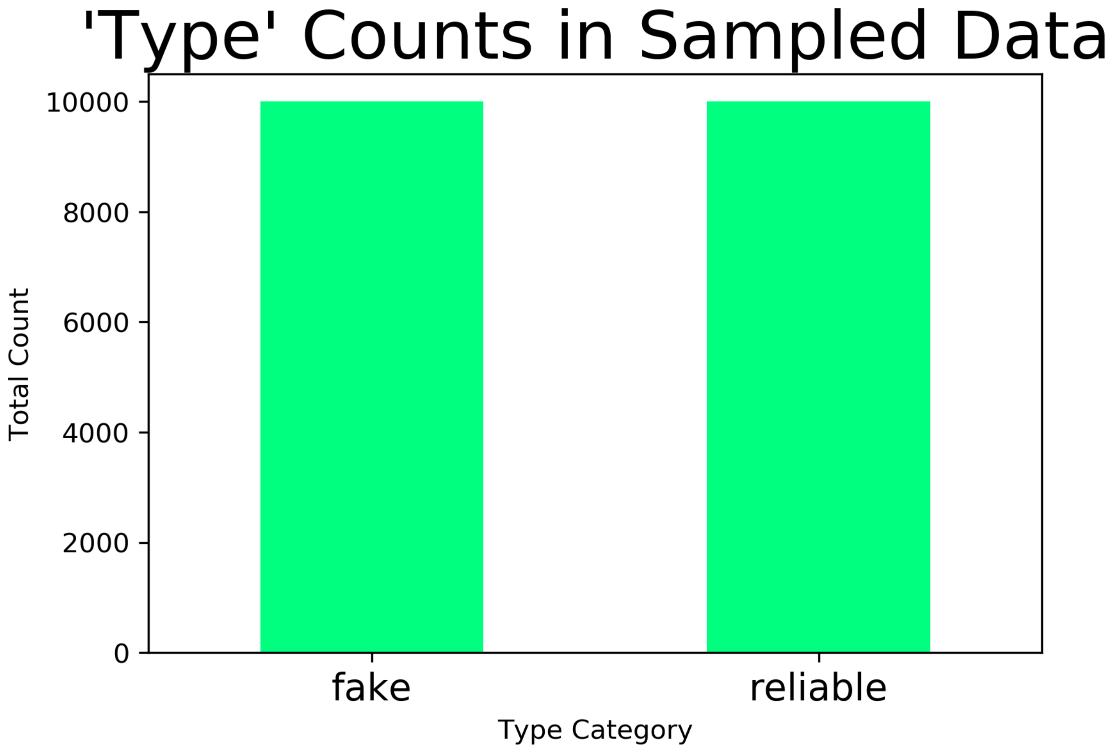
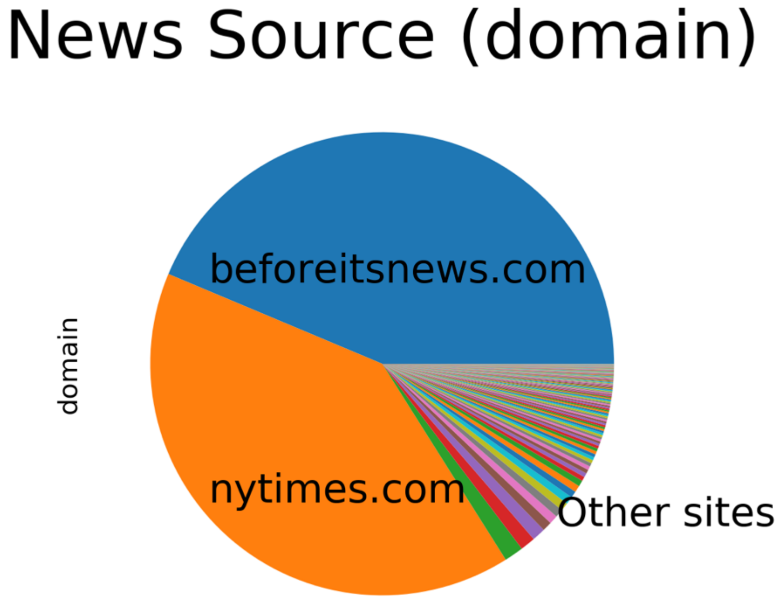

'Type' Counts in Sampled Data
Total Count
10000
8000
6000
4000
2000
0
fake
reliable
Type Category

Fig. 9


News Source (domain)
domain
beforeitsnews.com
nytimes.com
Other sites

Sentiment Analysis

The nltk Vader Sentiment Intensity Analyzer and TextBlob were used to analyze polarity in the corpora. Documents were labeled 1 (positive) if their score was greater than 0.2 (on the 1 to 1 scale used by both libraries), -1 (negative) if the score was less than -0.2, and neutral if it was between them.

Textblob was also used to calculate subjectivity scores. The scale for subjectivity scores is -1 to 1. Documents were labeled 1 (biased) if their subjectivity score was greater than 0.55, and -1 (unbiased) if the score was less than 0.45, and 0 if the score was between 0.45 and 0.55.

Table 1
Polarity scores

	Label	SIA Polarity Score	TextBlob Polarity Score	TextBlob Subjectivity
Fake News	1	5468	983	1128
	-1	3013	67	5784
	0	5013	7944	3088
Reliable News	1	6721	1175	762
	-1	2581	73	6800
	0	698	8752	2438

The averages of all the scores taken for both the ‘fake news’ and ‘reliable news’ in the sampled data are as follows:

	Textblob Average Polarity Score	Textblob Average Subjectivity Score	SIA Compound Average	SIA Negative Average	SIA Neutral Average	SIA Positive Average
Fake	0.11	0.42	0.22	0.07	0.83	0.09
Reliable	0.10	0.41	0.38	0.06	0.84	0.09

Textblob had very similar scores for both the ‘fake’ and ‘reliable’ articles. In SIA, the ‘fake’ and ‘reliable’ articles had a similar average of ‘negative’, ‘neutral’, and ‘positive’ sentiment.

Not a large difference was found between the ‘reliable’ and ‘fake’ articles. Perhaps more data needs to be used. The most significant finding was that 23% of the articles labeled ‘fake’ had a subjectivity score higher than or equal to .5, whereas only 17% of the articles labeled ‘reliable’ had such a value.

Predictive Modeling

The data was divided into a training and testing set. A several classifier was trained using sckit-learn, and then used to predict the labels for the testing data. Accuracy varied depending on vectorization approach:

Table 2

		Vectorization Technique		
		Bag of Words	Tf-idf	Tf-idf with two bigrams
Classifier	MultinomialNB()	86.7%	87.2%	90.4%
	LinearSVC()	87.0%	90.8%	91.7%
	XGB Classifier()	87.5%	89.0%	83.3%

The most accurate approach was to use LinearSVC() with tf-idf classification using bigrams. There may be some issues with this model, linked with the data itself. Inspecting the most predictive features revealed, for example, that some of the most predictive bigrams for classification were ‘york time’ and ‘york citi’. This is due to the over-representation of The New York Times as news labelled as ‘reliable’ in the dataset.

In order to deal with the problem of the over representation of articles from the New York Times in the dataset, the data was resampled to get a more varied distribution of article sources.

Analysis after Resampling

The New York Times and Beforeitsnew.com were vastly overrepresented in the dataset. While there were 141 different domains represented, the vast majority of documents come from one of two domains.

Fig. 10

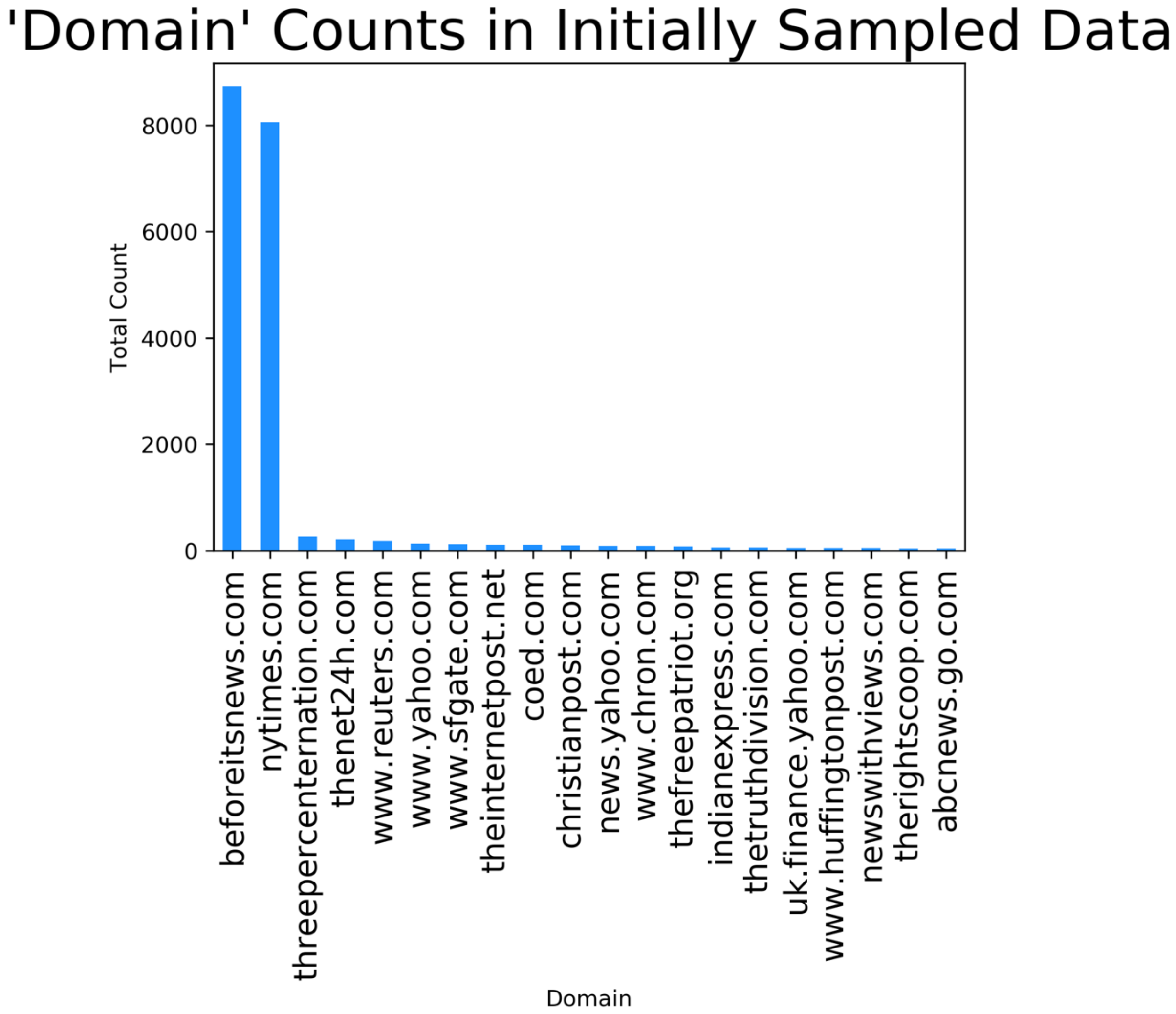
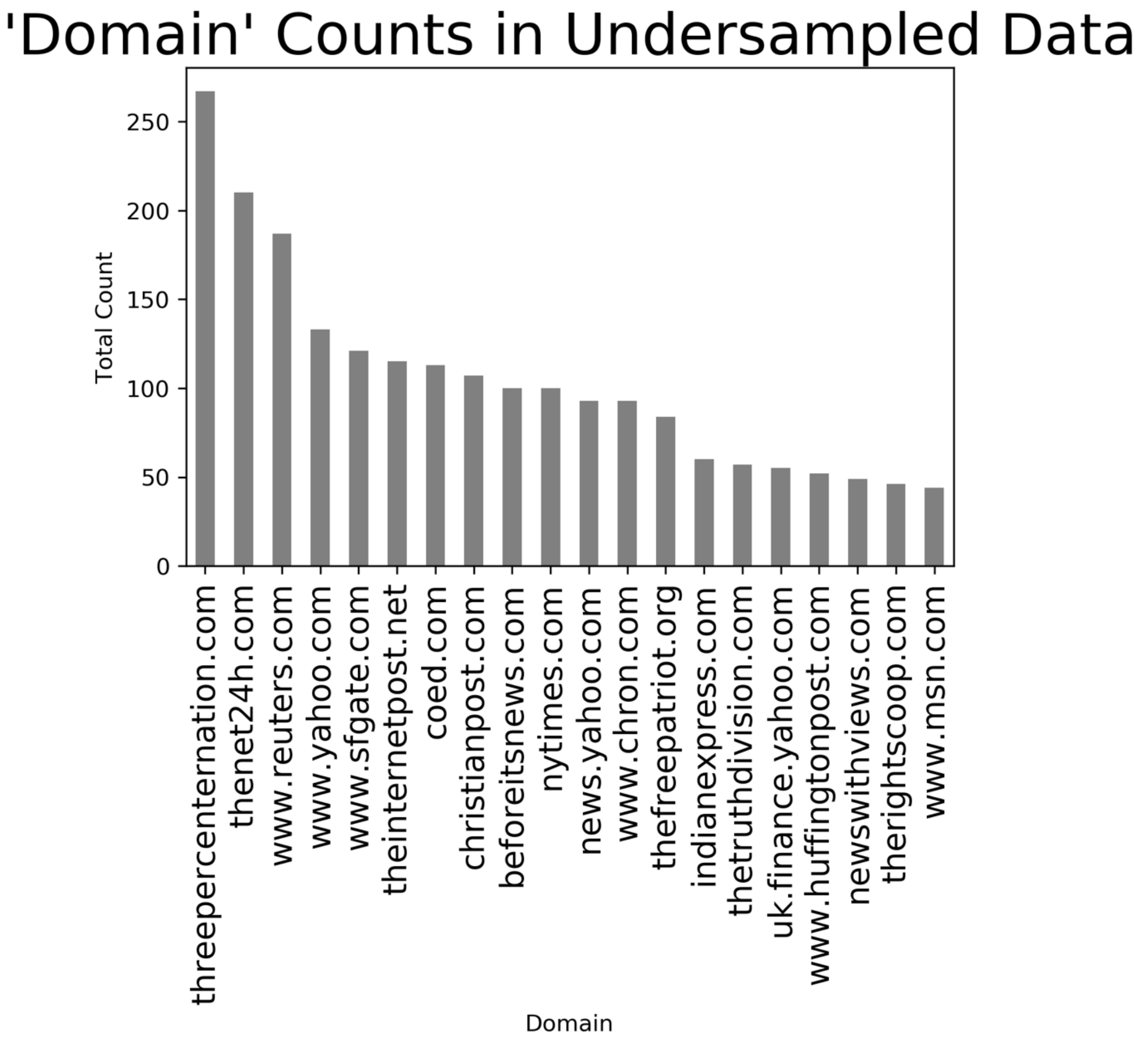


Fig 11.



Because the domains were so unbalanced, undersampling was performed to somewhat rectify this problem. The New York Times and beforeitsnews.com were undersampled to 100 random articles. After this, the distribution of domains was more balanced.

Undersampling them left the dataset with only 3410 articles, which is a potential issue. Several classifiers were trained on the newly sampled data, and new accuracies were observed. Again, accuracy varied depending on the vectorization approach:

Table 3

		Vectorization Technique		
		Bag of Words	Tf-idf	Tf-idf with two bigrams
Classifier	MultinomialNB()	78.4%	63.9%	66.0%
	LinearSVC()	80.7%	85.6%	82.1%
	XGB Classifier()	79.8%	84.6%	78.7%

It is clear from looking at the informative features that document source is still playing a large role in predicting whether a given document is ‘fake’ or ‘relaliable’. Even after resampling, some of the most informative features for td-idf with bigrams were “new york”, “associ press”, and “thomson reuter”. This indicates that document source is highly relevant in determining whether a document is predicting as being ‘fake’ or ‘reliable’.

In order to approach this issue in a different manner, words that were directly linked to article source were included as stopwords. This tactic resulted in informative features that were more reflective of journalistic topics, such as “budget forecasters” and “black slaves”; however, accuracy of the classifier (LinearSVC() with tf-idf bigrams) was significantly lower when extending the stopwords (78%).

Multi-class Classifier

Additionally, a multi-class classifier was built to include other classes than ‘fake’ and ‘reliable’ from the initial dataset. The categories of ‘bias’ and ‘political’ were also included, as these are the next largest types represented in the dataset. A multinomial Naïve Bayes, Linear SVC and XGBoost classifier was built, using count_vectorizer(), which yielded 68% accuracy. As the heat maps in Figures 12-14 shows, the classifier most accurately predicted the category of documents that had been labelled as “reliable”.

Figure 12

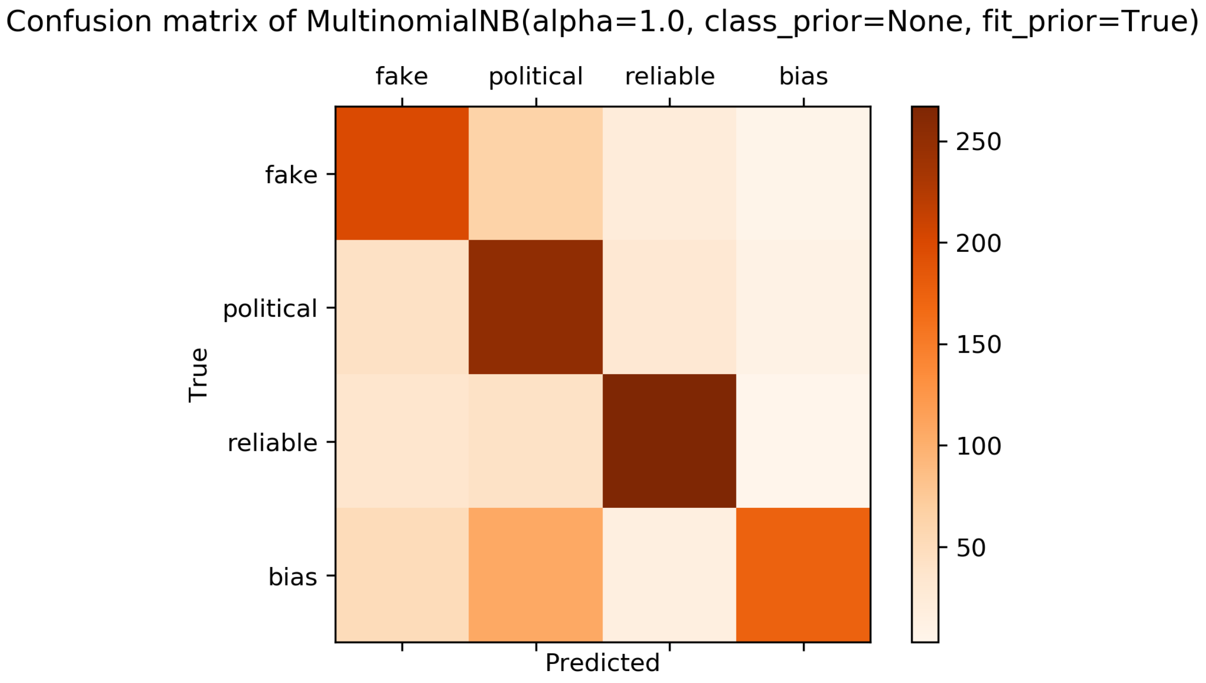


Figure 13

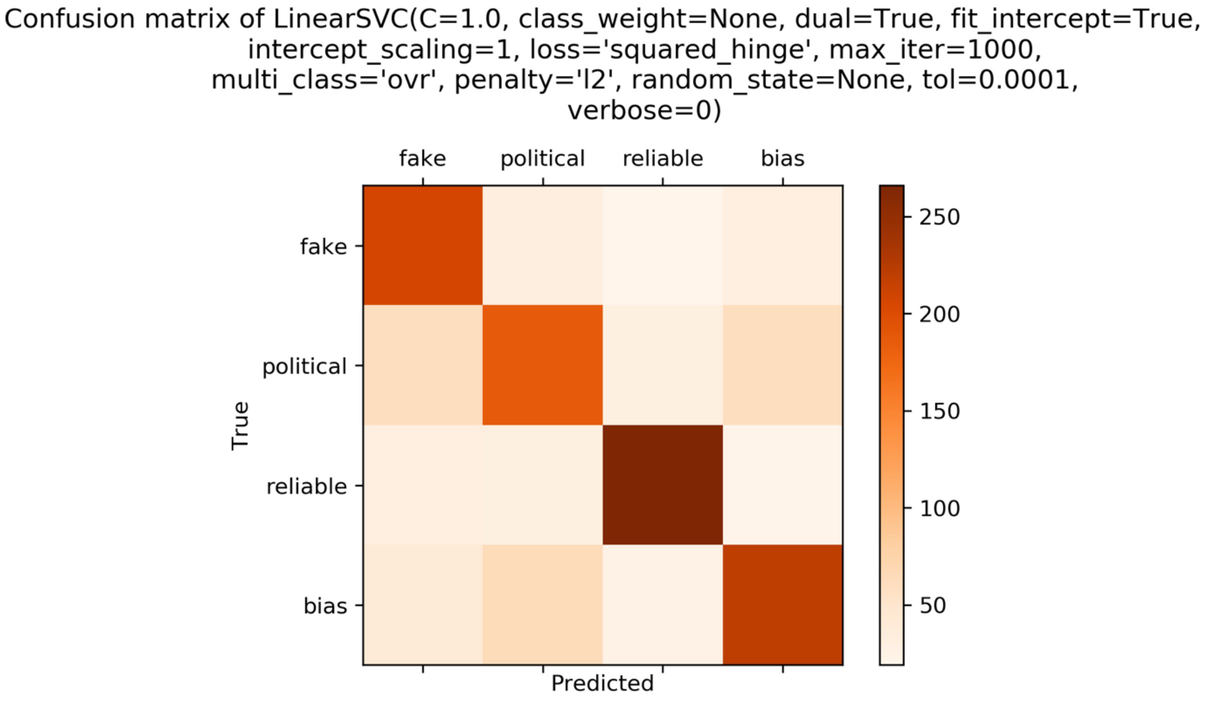
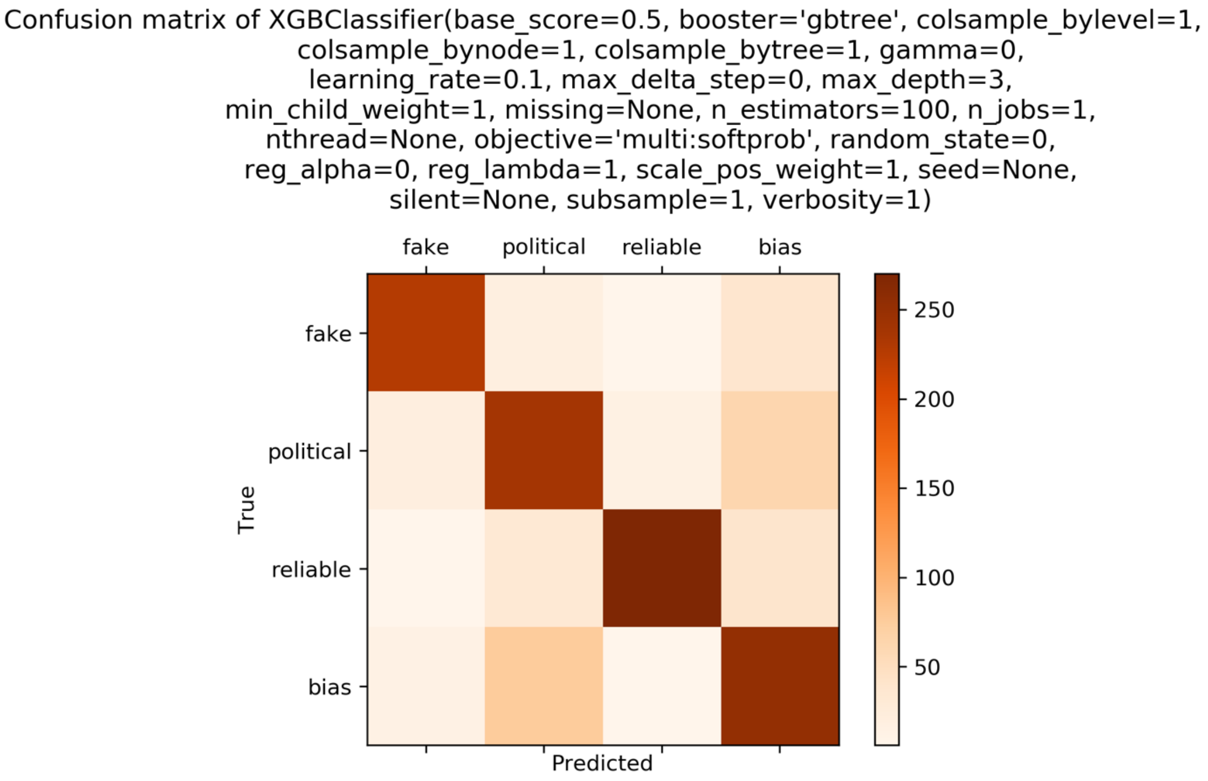


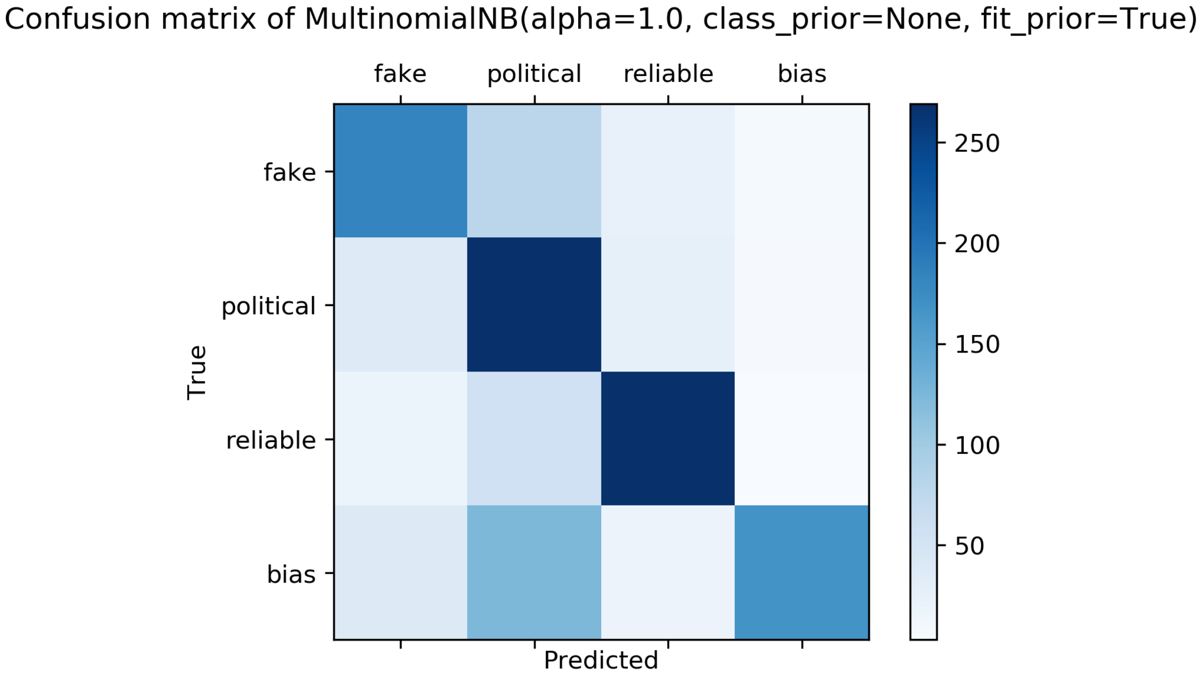
Figure 14



Confusion matrix of XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3, min_child_weight=1, missing=None, n_estimators=100, n_jobs=1, nthread=None, objective='multi:softprob', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=None, subsample=1, verbosity=1)

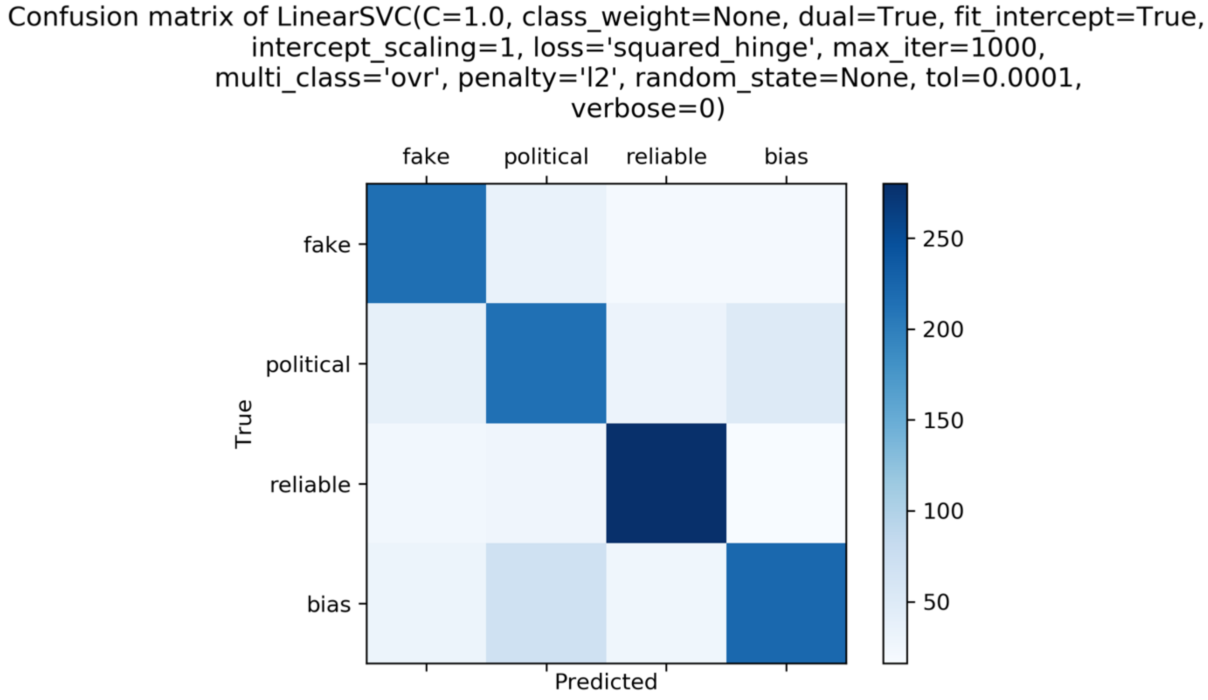
The same classifiers were then refit after using tf_idf vectorizer—the heatmaps of each are in figures 15-17:

Figure 15



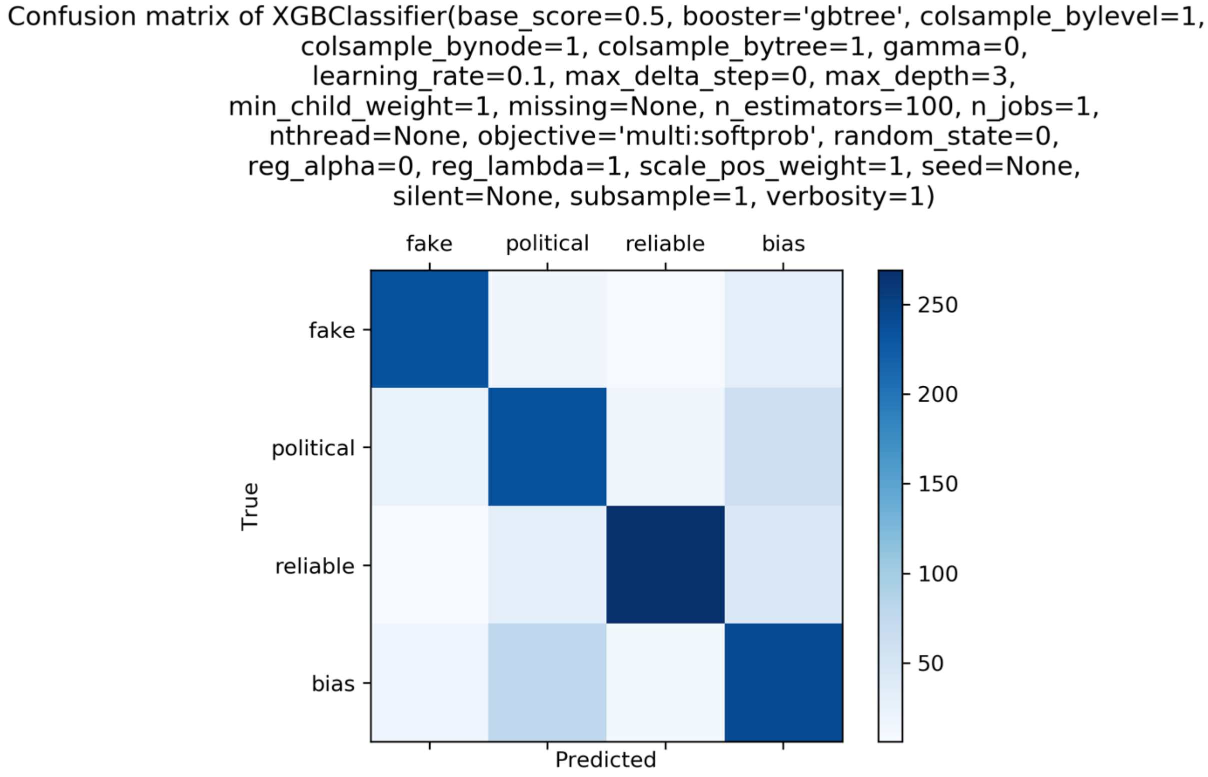
Confusion matrix of MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

Figure 16



Confusion matrix of LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='squared_hinge', max_iter=1000, multi_class='ovr', penalty='l2', random_state=None, tol=0.0001, verbose=0)

Figure 17



Confusion matrix of XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3, min_child_weight=1, missing=None, n_estimators=100, n_jobs=1, nthread=None, objective='multi:softprob', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=None, subsample=1, verbosity=1)

XGBClassifier() with tf-idf vectorization proved to be the most accurate with 75.7% accuracy.

All categories:

An additional analysis was done including all of the categories from the dataset, rather than the four most popular ones. The confusion matrices for each vectorization and classifier pair are shown in figures 18 to 23.

The results were less useful, due to the size of the sample of the dataset, as some categories had perfect precision and recall, due to their low representation in the dataset.

Figure 18

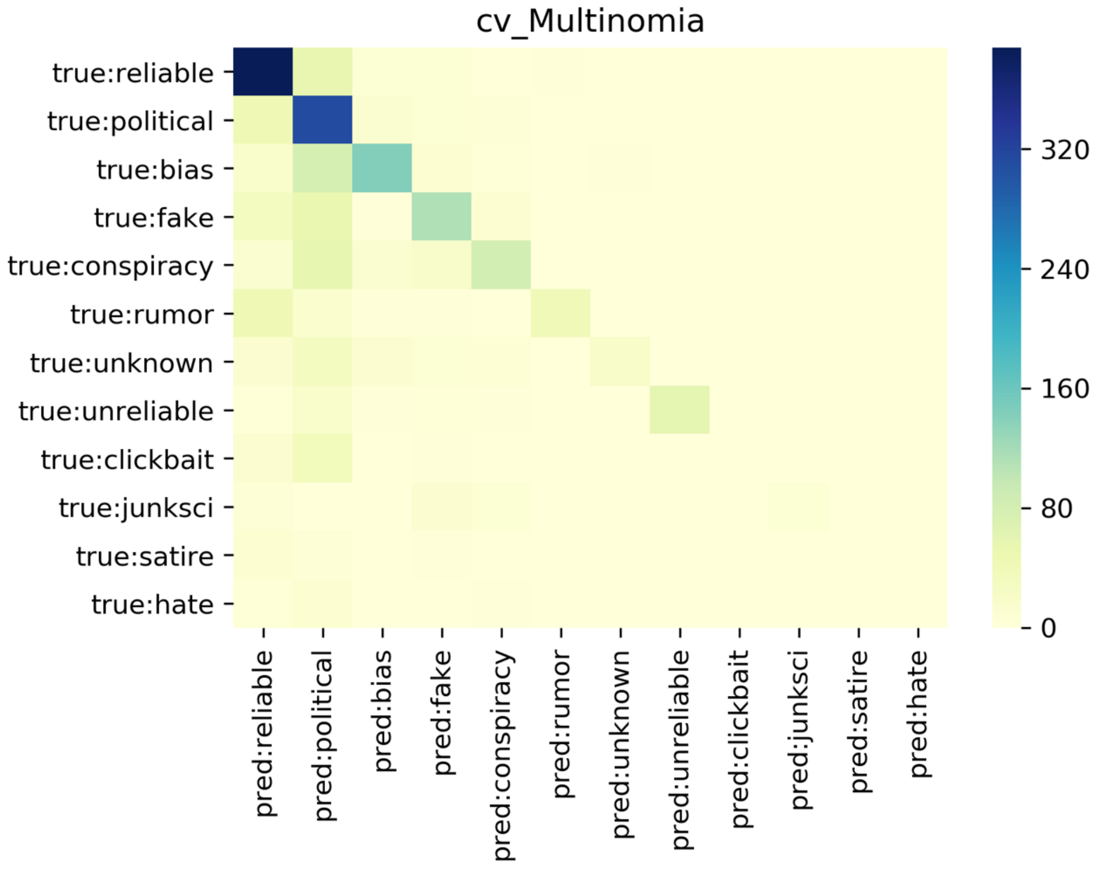


Figure 19

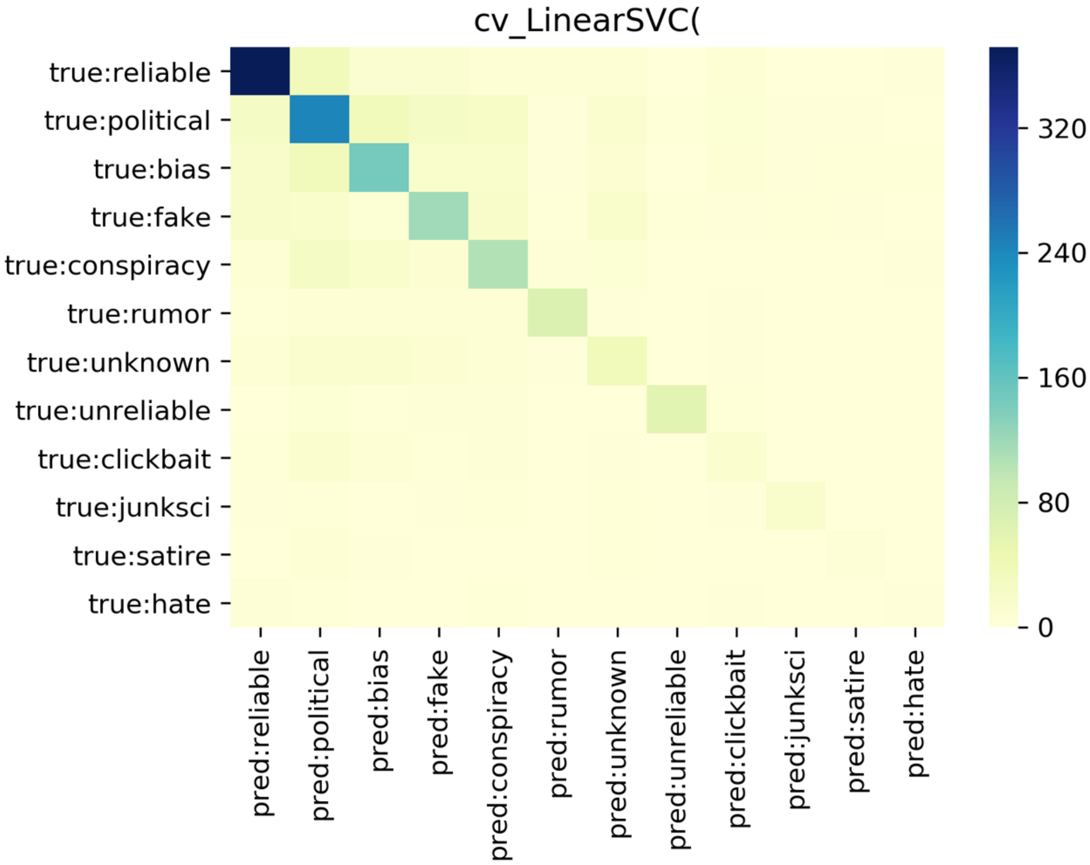

cv_LinearSVC(
true:reliable
true:political
true:bias
true:fake
true:conspiracy
true:rumor
true:unknown
true:unreliable
true:clickbait
true:junksci
true:satire
true:hate
pred:reliable
pred:political
pred:bias
pred:fake
pred:conspiracy
pred:rumor
pred:unknown
pred:unreliable
pred:clickbait
pred:junksci
pred:satire
pred:hate
320
240
160
80
0

Figure 20

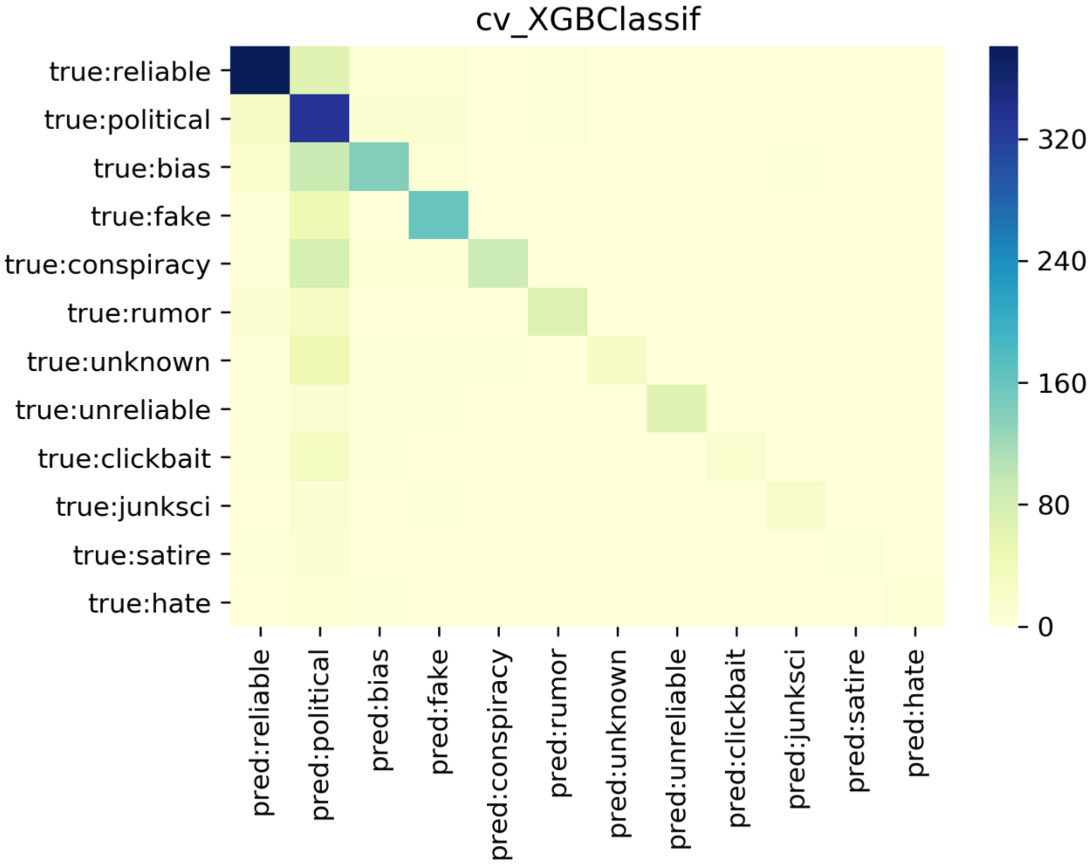

cv_XGBClassif
true:reliable
true:political
true:bias
true:fake
true:conspiracy
true:rumor
true:unknown
true:unreliable
true:clickbait
true:junksci
true:satire
true:hate
pred:reliable
pred:political
pred:bias
pred:fake
pred:conspiracy
pred:rumor
pred:unknown
pred:unreliable
pred:clickbait
pred:junksci
pred:satire
pred:hate
320
240
160
80
0

Figure 21

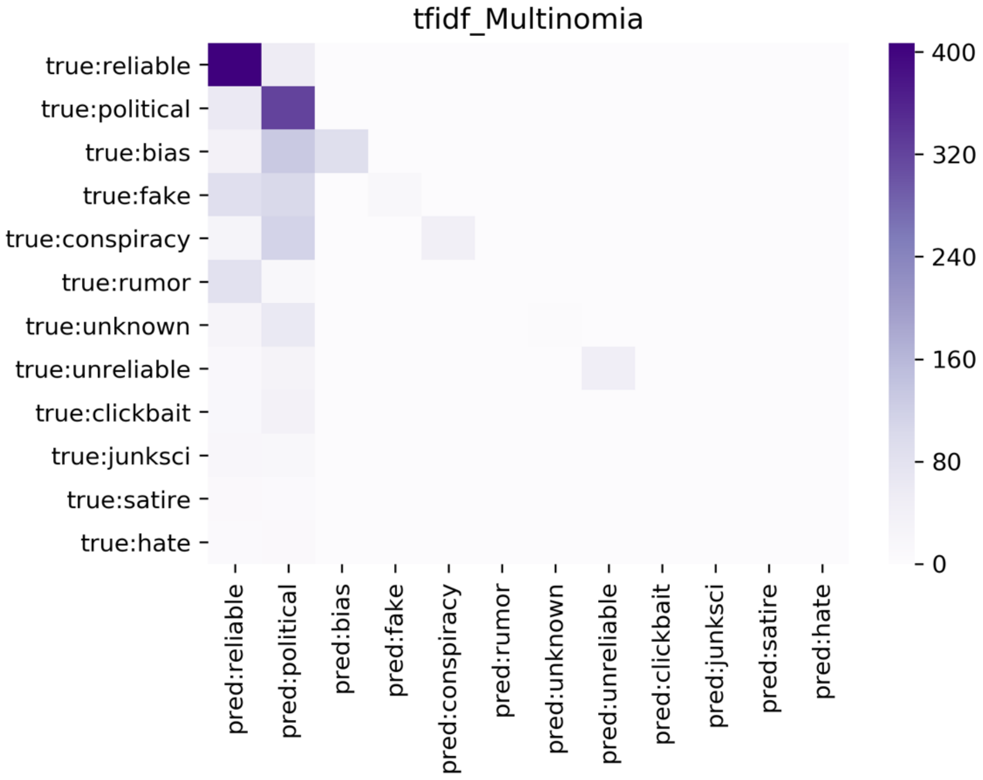


Figure 22

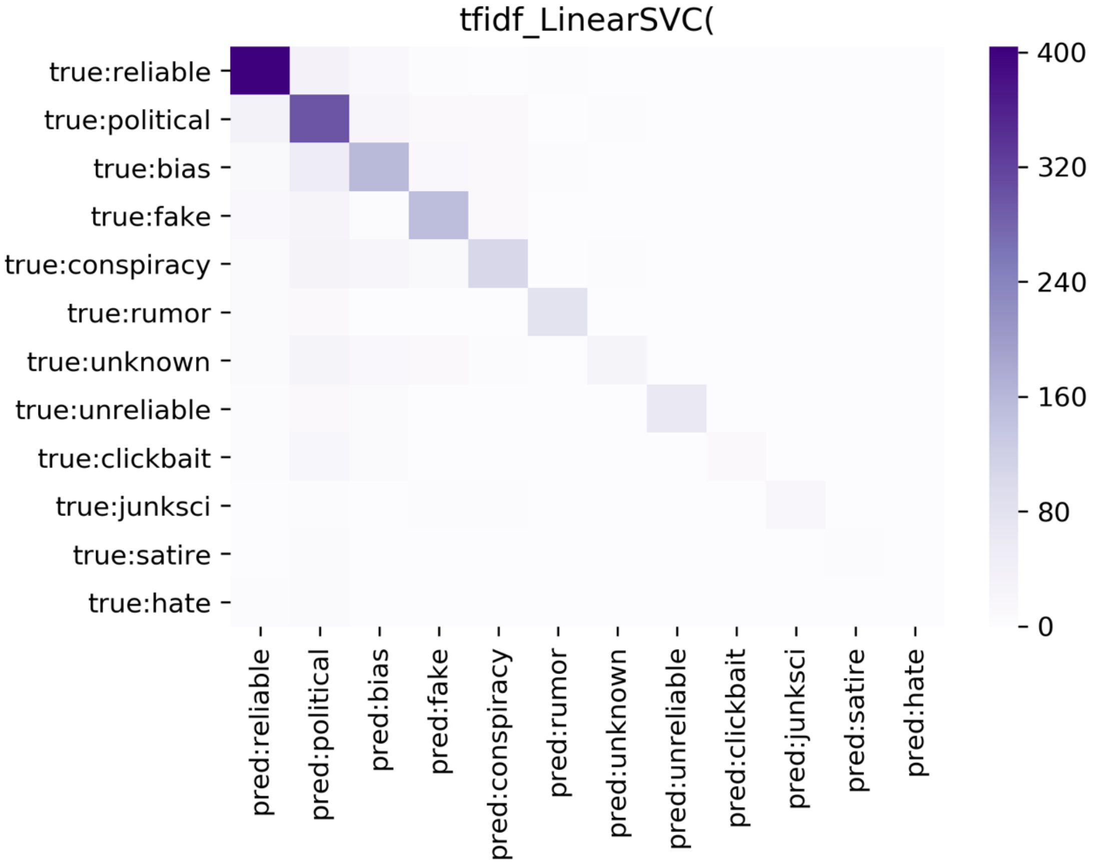
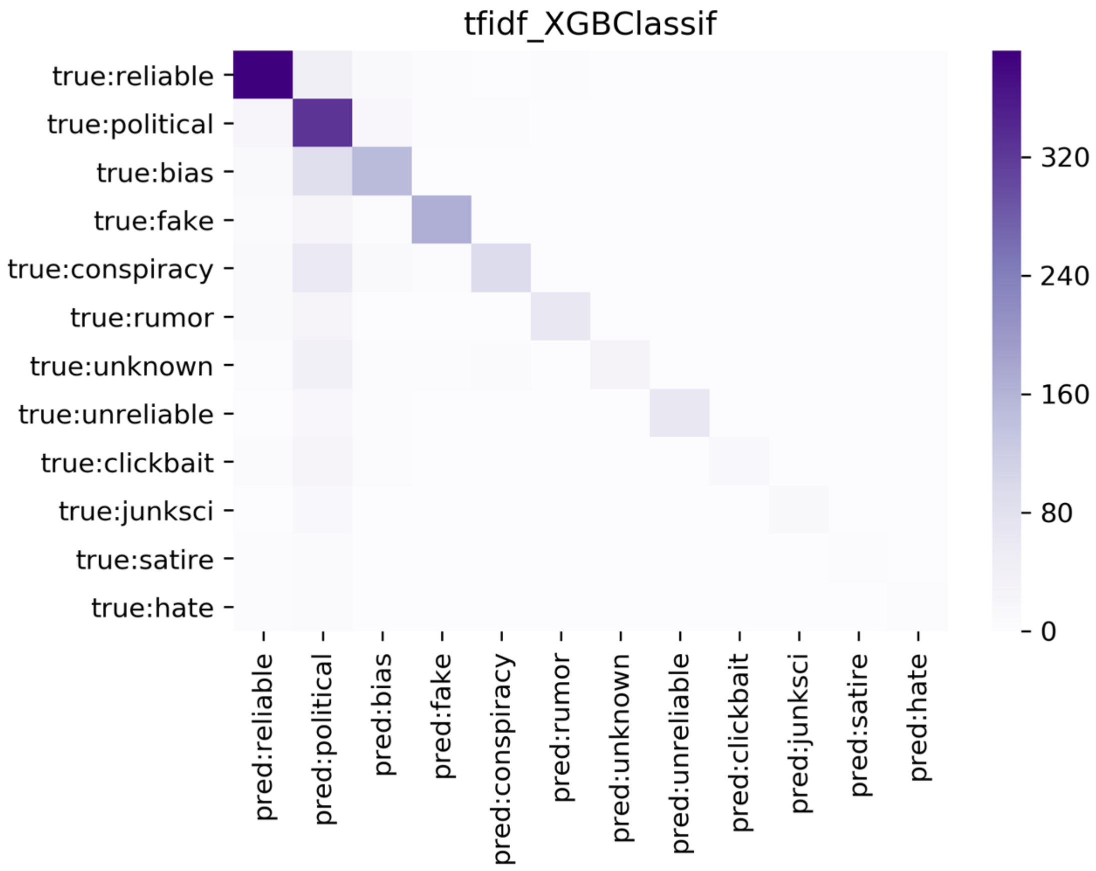


Figure 23



Conclusion

The most significant finding from the sentiment analysis of documents labelled ‘fake’ or ‘reliable’ from the dataset was that 23% of the articles labeled ‘fake’ had a subjectivity score higher than or equal to .5, whereas only 17% of the articles labeled ‘reliable’ had this high of a subjectivity score. It seemed that the TextBlob subjectivity score was more important than the polarity score in distinguishing between ‘reliable’ and ‘fake’ documents.

In the predictive analysis, various classifiers using different vectorization techniques were fairly accurate in determining whether a document from the testing data was labelled as ‘fake’ or ‘reliable’. An inspection of the most informative features indicated, however, the terms linked to document source were particularly important in determining which class a given document belonged to. This indicates that words linked to what source an article came from may be more important than other words in determining whether it will be classified as ‘fake’ or ‘‘reliable’.

Lastly, several classifiers were built to predict the four labels of ‘fake’, ‘reliable’, ‘bias’, and ‘political’, found in the dataset. The classifiers consistently performed best at predicting the ‘reliable’ label for documents in the testing data. The same classifiers were used to predict all of the labels in the sampled dataset. The results were less helpful because some of the categories were very underrepresented using previously made sample.

A next step for this project would be to consider using significantly more data. While this is time intensive in terms of processing, it may yield much more useful results for the multi-class classification.

An important takeaway of this project is that it is somewhat difficult to unlink fake news detection from human labeling. For example, even after the resampling and significant inclusion of other sources than the New York Times and beforeitsnews.com, some of the most predictive features in the most accurate classifier were “reuters”, and “ap”. Before the resampling, the most predictive bigrams were “new york”, “main story”, and other bigrams often repeated in New York Times articles. If all articles from a given source are labelled as “fake”, “bias”, or “junksci”, a machine learning classifier may actually be more or less predicting the source, if certain phrases repeat in articles from a given source, than words and phrases that are inherently distinctive to fake or reliable news.