The Divide Between Falsity and Honesty
A Classifier for Fake News

Rolando Franqui and Connor Brown

Introduction:

In an age of social media and the ability to go viral at the touch of a phone screen, anyone and anything can make the news, inform others, or simply make someone laugh across the world. However, this comes with caveats; people have the power to spread misinformation, rally groups of people politically or radically, and/or create division between them. Over the past several years, this understanding has taken the majority of media by storm, leading to what is known now as 'fake news.' This term was coined to describe the specific nature of "propaganda that is intentionally designed to mislead the reader, or…be designed as 'clickbait' written for economic incentives" (Desai et. al., 2021), and, as is clear from its description, is entirely problematic and relevant to the future of media. But what differentiates 'fake news' from a more realistic, unbiased source of information? The answer is in the question: bias. An all-too-common source of misinformation is bias, whether it be politically-motivated, for economical gain, or even poor training; it is also very difficult to defend against (Desai et. al., 2021). Despite this, it is still possible to understand where it comes from, and what signs serve as alarms for its presence. This study works to analyze 'fake news,' using a dataset from Kaggle, built with around forty thousand articles with around half being 'fake', based on biased, unreliable and/or politically-motivated sources, and half being 'true', based on unbiased articles from Reuters that serve to offer knowledge rather than giving interpretation (Bisaillon, 2020). With this data in hand, we hope to find solutions to the problem, "what features and characteristics of an article showcase dishonesty, malintent, or even truth?" With these solutions, we plan on building a classifier to showcase the effectiveness of these features as predictors for any article that may come to be, to determine whether it brings about knowledge or conspiracy. In our current societal climate, we see day after day across social media and news sites, a plethora of information; all it takes is one viral headline to skew society's view and turn people against each other. There must be ways to differentiate the two sides of this problem, and we hope to find one.

Dataset:

In order to fully understand fake news, we must first find a reliable, hand-picked division between fake and true news. This is fulfilled by a dataset found on kaggle.com, a reputable site that serves as a platform for publicly available datasets, coding information, etc. The 'fake and real news dataset,' posted by Clément Bisaillon, offers us a plethora of material, with 21,417 'true' articles and 23,481 'fake' articles. The true articles are sourced from

Reuters.com, a news agency that publishes articles sometimes seen as 'just the facts,' given their initiative to simply sell articles to anyone, regardless of political agenda, motivation, or otherwise. The fake articles are collected from unreliable sources that are flagged by Politifact, which is a fact-checking organization famous for their 'Truth-O-Meter" (Bisaillon, 2020). Within this dataset, each record is given as a specified article, which has corresponding metadata: a title, unformatted text content, a subject (defined by the dataset itself), and a publishing date. In further work with the dataset, we were able to extract more specific aspects of the data. In learning about the 'text' metadata, we found that the fake news articles tended to be a bit longer on average than true articles, by a factor of two hundred tokens. In addition, the fake articles have a far lower sense of originality. In calculation, we found that the fake article subset had 53.3% uniqueness, in correspondence with the proportion of those that have no duplicates; this is not surprising, given the circumstance of fake news' existence. In addition, they have only 74.3% of documents being 'distinct,' which is the percentage of those that are significantly different from each other. Whereas, the true article subset had a distinction of 98.9%, and a uniqueness of 98%. Even just in understanding the dataset, the bias is already quite foreshadowed. As for the 'title' metadata, we have similar statistics: fake articles have far longer titles with average 278 tokens, as well as a distinction of 76.8% and a uniqueness of 54.5%, while true articles have an average token length of 107, a distinction of 97.2% and a uniqueness of 94.8%. As for the 'subjects,' we found slightly less useful metadata in the grand scheme of our assessment. True articles had two possible subjects: either 'politicsNews' or 'worldnews;' fake articles had *five*: 'News,' 'politics,' 'left-news,' 'Government News,' and 'US_News.' And finally, as for dates, all articles were published between 2015 and 2017, with each subset having the highest frequency in 2016. These values were all calculated using Pandas-Profiling, which afforded us a straightforward way of understanding our dataset (Brugman et. al., 2021).

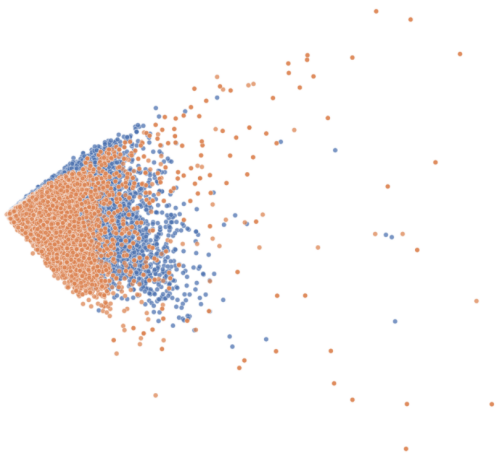
Methodology:

With a high magnitude of data, we had a lot of possible avenues to take. Broadly, we took three different approaches to our data: a feature analysis, an assessment through clustering, and the building and training of a classifier. Before tackling any of these, we utilized methods to build our data that we learned in our analytics course. This involved using Python to read in our subsets into Pandas dataframes and then collecting them into a single list. This list kept track of every title, text, subject, date, the frequencies of every word within that article, and whether it was fake or not. The frequencies in question are based only on specified tokens, tokenized using the Natural Language Toolkit (NLTK), which offers packages useful for tokenization, classification, and plethora of other corpora resources (Bird et. al. 2009). We chose to ignore all stop words and punctuation, so as to show us the words that truly define intentions of the article. We also globally kept track of the counts of all words, as well as

separated by true and fake, and the document frequencies of each word. With these ready, we tackled feature analysis by building a vocabulary with words appearing in more than two documents as a feature set, and built Numpy vectors for feature visualization. We moved on to visualize the features using TruncatedSVD, and built a basic Naive Bayes classifier to observe the accuracies of these features using Leave-One-Out cross-validation, all using Scikit-Learn. Finally, to get a better understanding of how individual words have an impact, we built contingency tables to find the Dunning G-Scores of certain words. This allowed us to see how significant of an impact each word had on its respective classification. After this, we moved onto clustering, where we observed both agglomerative and K-means clustering. We first rebuilt new, normalized vectors, and then used agglomerative clustering and plotted dendrograms of multiple, smaller subsets of the articles. For more information, we decided to take it further by trying K-means clustering, which we found to be a more useful approach. We also discovered through this that K-means is generally far less computationally-intensive and worked far better for our purposes given our large dataset. Throughout both of these methods, we made sure to take a deeper look at the cluster definitions and how our algorithms labeled each article. Finally, we created another, multinomial Naive Bayes classifier using Scikit-Learn. We used Pandas again to create a collective dataframe, combined important features into a signal column (‘text’ and ‘title’), and then cleaned our text by tokenizing and lemmatizing. We then trained our model with 40% of our data, and tested it on the other 60%, all to find the accuracies and precisions of our classifier, utilizing a Classification Report offered by Scikit-Learn’s Metrics package.

Analysis:

Through these methods, we acquired very conclusive results. Due to a wide variety of memory-based limitations, a good portion of our results were gathered through smaller subsets, however they appear to be very representative of the data as a whole. In terms of our feature analysis, our first findings appeared in our visualizations. Using TruncatedSVD, we were able to reduce the dimensionality of the entire feature set to see how related the two subsets are. **Figure 1** (*right*) shows true (blue) and fake (red) articles dimensionalized in a singular value decomposition, and there are already very visible clusters. To further defend the feature set, our Naive Bayes classifier, when given a subset of 1000 articles (500 hundred random true, 500 random fake), produced an overall accuracy for the feature set of 98.5%. This broke down further into an accuracy of 99.2% for true articles and 97.8% for fake articles. This makes sense given what we know, as fake articles have more tokens and less uniqueness. *Our contingency*



tables showed multiple, interesting finds, which will be discussed in a smaller case study in the next section. Thus, our feature analysis showed us that this feature set is indeed useful and efficient in association with the data. Clustering was a bigger learning experience for us, with both successes and failures. Agglomerative clustering served to be more of a struggle than a resolution, however we learned a lot about the process. To begin with, we went into the process hoping to see two obvious clusters of our data, to represent the two sides.

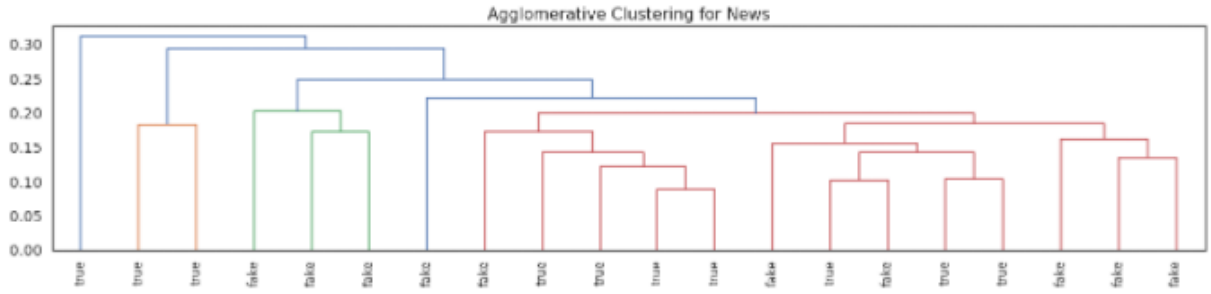


Figure 2 (*above*) is a dendrogram of a small subset of just 20 articles, which shows us the biggest struggle we had with agglomerative clustering: seeing the clusters. From this small perspective, it already appears that there are far more than our expected two clusters. Looking at the labels and working with the cluster number only brought more confusion; the results were incredibly inconsistent and mixed. This problem persisted when applied to a 2D graph, pictured in **Figure 3** (*right*). Despite these problems with agglomerative clustering, we found far greater successes in the visualization of K-means. Going into this clustering method with the mentality we used in agglomerative resulted in a visualization that proved to be much more useful. Pictured in **Figure 4** (*left*), a sample of 1000 articles (even split) showed an incredibly obvious two clusters for the data. Using centroids, we were able to also find a common central point for all of the data. When looking further at the labels this clustering created, we saw just how effective this method was, seeing a vast majority of true and fake articles being clustered correctly. Finally, we prepared and trained our classifier. Using a training model of a

random sample of 40% of the collective dataset, we were able to test on the remaining 60% a Multinomial Naive Bayes classifier. This produced great results, displayed in **Figure 5** (*below*).

	precision	recall	f1-score	support
Fake	0.95	0.96	0.95	8659
True	0.96	0.95	0.96	9301
accuracy			0.95	17960
macro avg	0.95	0.95	0.95	17960
weighted avg	0.95	0.95	0.95	17960

In terms of fake news, the classifier has a precision of 95%, versus true's 96%. Accuracy in general is quite similar, with a weighted average of 95% accuracy for classifying the test set correctly. These results are very significant, and help us to understand exactly how we can use features of these articles to predict the intent of others.

Case Study: **Strength in Language**

One of the most interesting parts of this entire project surrounded the contingency tables we created to inspect specific words' connections to their article's intention. Utilizing a sample of 1000 articles (even split), we calculated multiple Dunning G-Scores to see the magnitude of certain buzzwords. To begin with, the most important word for the timeframe these articles are from is 'Trump'. When compared with the 'fake' subset, we found a G-Score of 304.8, which is incredibly high. Especially when considering that the word appears almost 3,000 times in just 500 fake articles, versus the mere ~1,000 for true in comparison, it is obvious that this name is an instant alarm for misinformation when in high volume. When using the word 'America' in comparison to fake articles, we found a G-Score of ~80. While this word does have lower frequencies in the subset, the fake news articles showcase it more than double comparatively to true articles. However, one of the biggest surprises was the word 'tax.' For reference, the word tax appeared in just 500 true articles over 1100 times; it appeared in 500 fake articles just over 100 times. This left the G-Score for 'tax' in comparison to true articles at an astonishing 948.24, the highest significant result we found. What can we presume from this? It appears as though the fake news articles are far more politically-motivated and imperialistic, as is clear in their high frequency mentions of American-forward vocabulary and the 45th president. On the opposing hand, true articles seem far more interested in world on-goings; taxes, general but non-specified politics, crime, etc. This defends fake news major motivation: bias.

Conclusion:

To answer the question that motivated us from the beginning, "what defines a fake news article and can we use it effectively," we can now answer with one word: language. While originally we proposed that bias is the most important aspect of fake news (which it still is important), it is much clearer to us now that, in fact, the words within them are far more representative. We were able to effectively analyze the vocabularies of this dataset as a collective, and found very significant support for their efficiency in representing the data. In addition, we were able to visualize just how separated these articles are based on their vocabularies, which is clear in the clustering we performed. With this in mind, our classifier performs fantastically, allowing us to better utilize these analyzed features to differentiate a piece of public information from a misinformed, malintensive interpretation. Had we had more time on our hands, as well as far more memory to handle such calculations and visualizations, we would like to try analyzing other types of features. One feature we had hoped to work with was sentiment, which could help us see bias through the eyes of emotion. Given the motivations of fake news, there is no doubt in our minds that sentiment plays a huge role in such pieces of work. We learned a lot, however, about the process of analysis, data visualization, clustering, machine learning, and in general, society. Social issues plague our world every day, and having a better understanding of one just gives us a little more hope for what the future could bring us; or what people *say* it will bring us.

References:

Bird, S, Loper E, Klein E. (2009). Natural Language Processing with Python. O'Reilly Media Inc. https://www.nltk.org

Bisaillon, C. (2020). Fake and real news dataset. Kaggle. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

Brugman S, et. al. (2021). Pandas Profiling. GitHub. https://github.com/pandas-profiling.

Desai S., Mooney H, Oehrli JA. (2021). "Fake News," Lies and Propaganda: How to Sort Fact From Fiction. University of Michigan Library. https://guides.lib.umich.edu/fakenews