Real or Fake News

Tyler Anderson

Bellevue University

DSC 680: Applied Data Science

Professor Catie Williams

November 10, 2020

Abstract

I will be examining two datasets that have news that was determined to be false and news that was determined to be true. One dataset will have all false news and the other having all true news. I want to analyze this dataset to try and see if I can predict the accuracy of a news being true or false based on the datasets provided. I hope to gain some insights on what makes some news false and others true and see if the discrepancy of true or false is a fine line between them. I also want to see if the dataset is biased as many of the datapoints in the datasets come from twitter and fake or real news to some may be subjective.

For this project I will have to do some text analysis and try to find a way to determine fake or real news using text from twitter and official news sources. I might be able to try to determine the accuracy of fake or real news based on one or a couple words. I can use logistic regression to try to accomplish this. My main hypothesis is that the data is biased in some way and I can try to prove that by using a inconspicuous word like “the” and get a high accuracy by using regression models.

The main issue I can see is the potential bias of the datasets and an issue where it’s possible I could get too high of a score for example 99-100% accuracy. This might point to the dataset not being good at all. If there is bias, I want to see if it skews to fake or real news. Overall, I expect to have some interesting results as the current political climate has exasperated fake news and could possibly make it harder to predict. I expect to see some differences in the text analysis of the real and fake news datasets.

Meaning of Variables

Both datasets here have the same variables, so the short list of variables listed here are for both the Fake and Real CSV files.

Title: The title of the article

Text: The text of the article

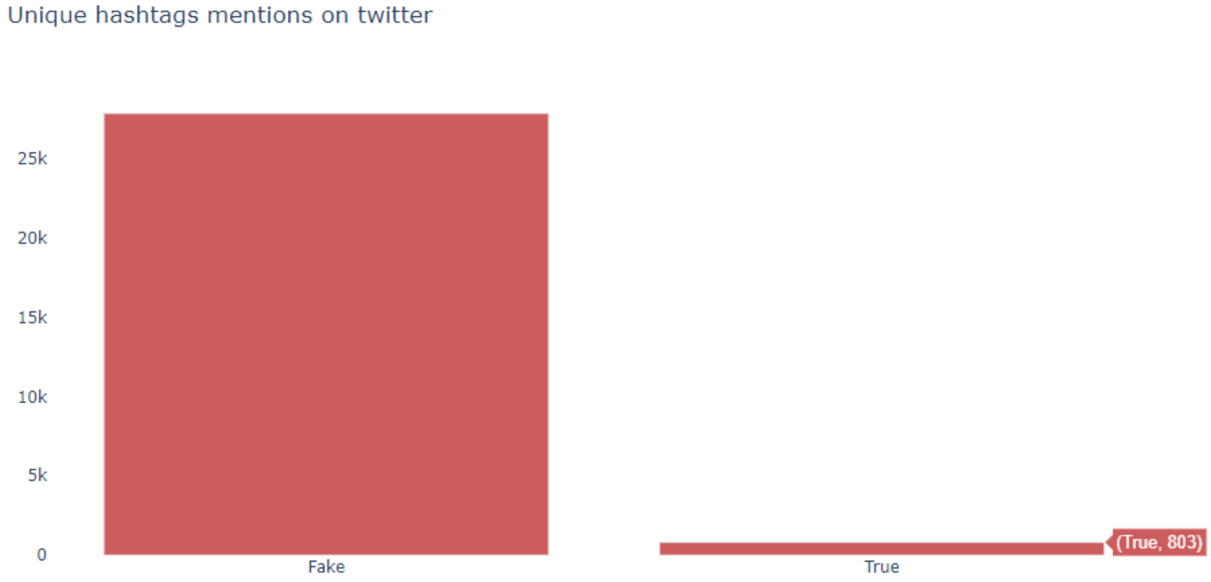
Subject: The subject of the article

Date: The date at which the article was posted

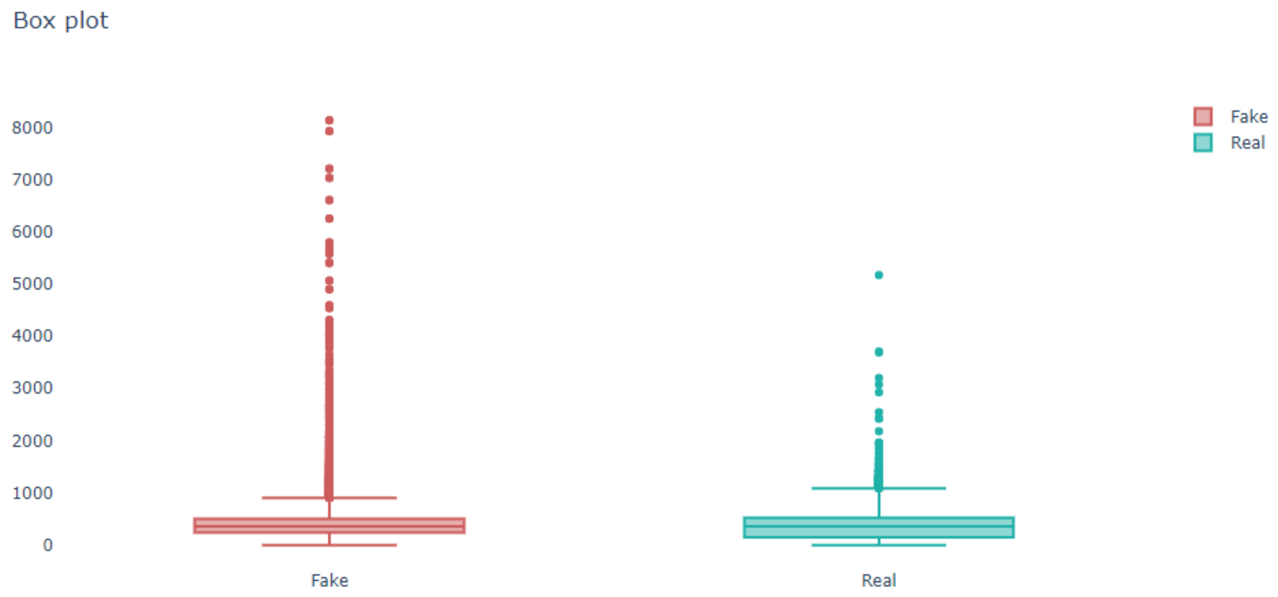
Analysis

This dataset is complete with no missing values or variables that are in the wrong format making this dataset easier to start working with. I first displayed the text of the first fake article and noticed that fake news has many mentions of quotes from twitter users. To see the differentiation between true and false news I raised the number of quotes in false and true news.

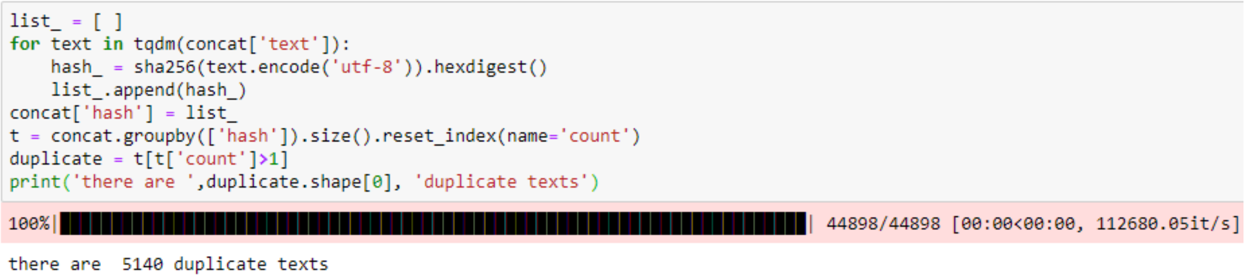
This led me to a unique visualization of hashtag mentions in twitter posts.



We can clearly see that fake news has a much greater presence of @ mentions on twitter, the sources from which the real news was collected are from news articles mixed with twitter posts. This already leaves me to believe that the dataset is biased. However, the fact that @ mentions of users would not be enough to affirm that the data is not suitable for identification of fake news, since any preprocessing that removes @ would be able to decrease the bias. So, I went in search of another aspect like the text size, to see if there is a lot of discrepancy between the texts.



After making the above plot fake news in general has a lot more words than real ones, which is kind of weird assuming real news has a tendency to bring details about events to inform the reader, however as noted the fact that fakes news is a mix of twitter posts and news this may justify the fact that they have more words. However, this might demonstrate the bias of the dataset as well as this can be resolved given that we would only work with fake news that was less than or equal to the news with more tokens, so this problem could be managed hypothetically. There may be more to this like the presence of duplicate news. This might be something that is frequent in the data when the data is split since we would have both in the training and in the test, so it is possible to present the same samples for the model. After checking for duplicates we can clearly see some here.

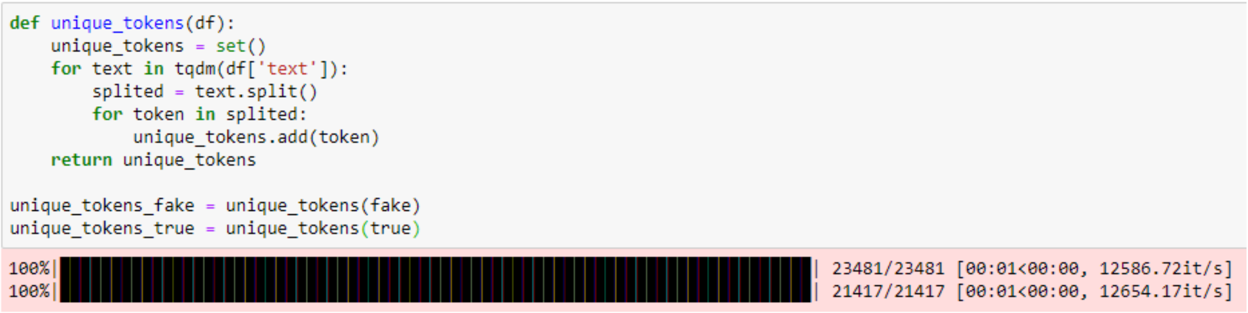


```
list_ = [ ]
for text in tqdm(concat['text']):
    hash_ = sha256(text.encode('utf-8')).hexdigest()
    list_.append(hash_)
concat['hash'] = list_
t = concat.groupby(['hash']).size().reset_index(name='count')
duplicate = t[t['count']>1]
print('there are ',duplicate.shape[0], 'duplicate texts')
```

```
100%|██████████████████████████████████████████████████████████████████████████| 44898/44898 [00:00<00:00, 112680.05it/s]
```

```
there are  5140 duplicate texts
```

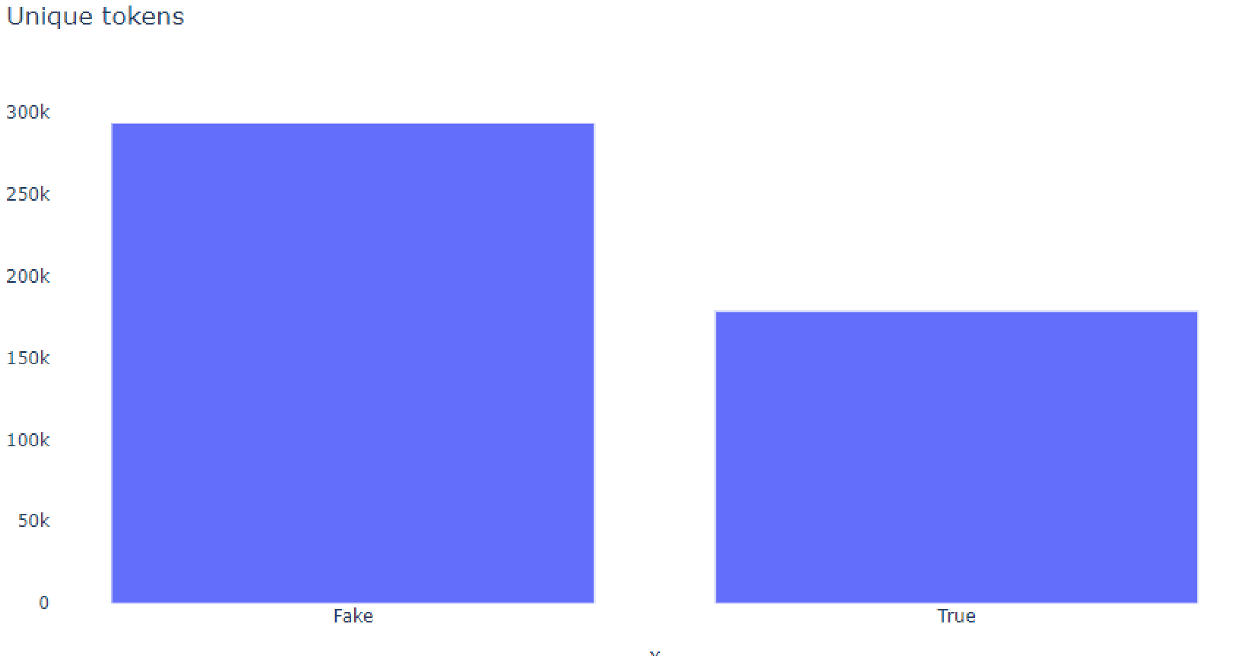
There is actually duplicate news in the dataset but the ratio is not enough to allow the models to be as accurate, but it already contributes to the bias. I now go a little deeper in the data as there must be some aspect of writing for the words that allow a clear differentiation, so I will observe the presence of the tokens in the true and false news.



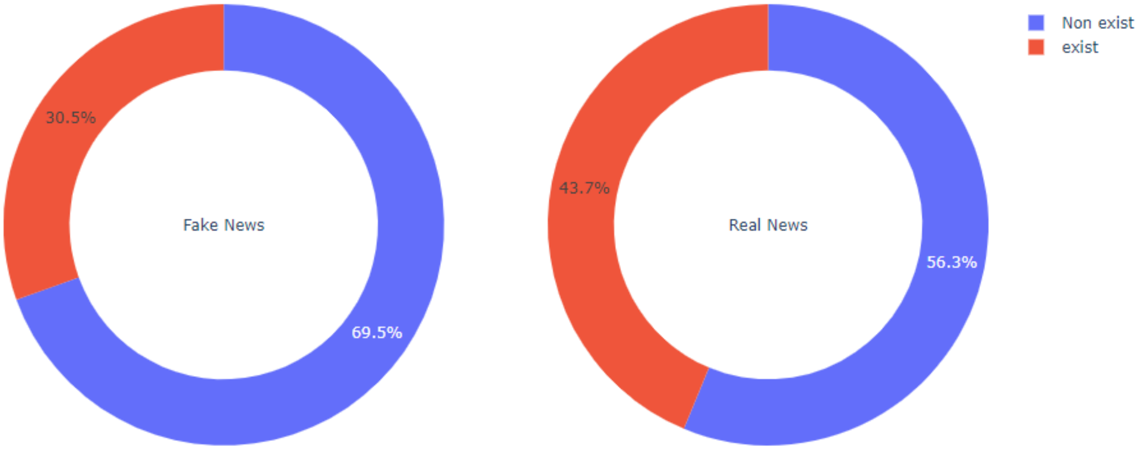
```python
def unique_tokens(df):
    unique_tokens = set()
    for text in tqdm(df['text']):
        splited = text.split()
        for token in splited:
            unique_tokens.add(token)
    return unique_tokens

unique_tokens_fake = unique_tokens(fake)
unique_tokens_true = unique_tokens(true)
```

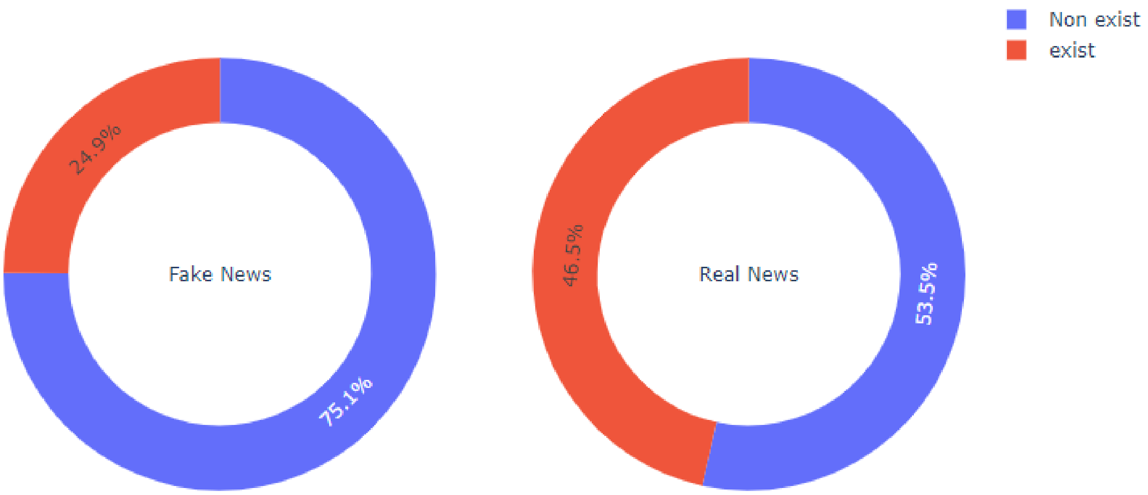
```
100%|██████████████████████████████████████████████████| 23481/23481 [00:01<00:00, 12586.72it/s]
100%|██████████████████████████████████████████████████| 21417/21417 [00:01<00:00, 12654.17it/s]
```



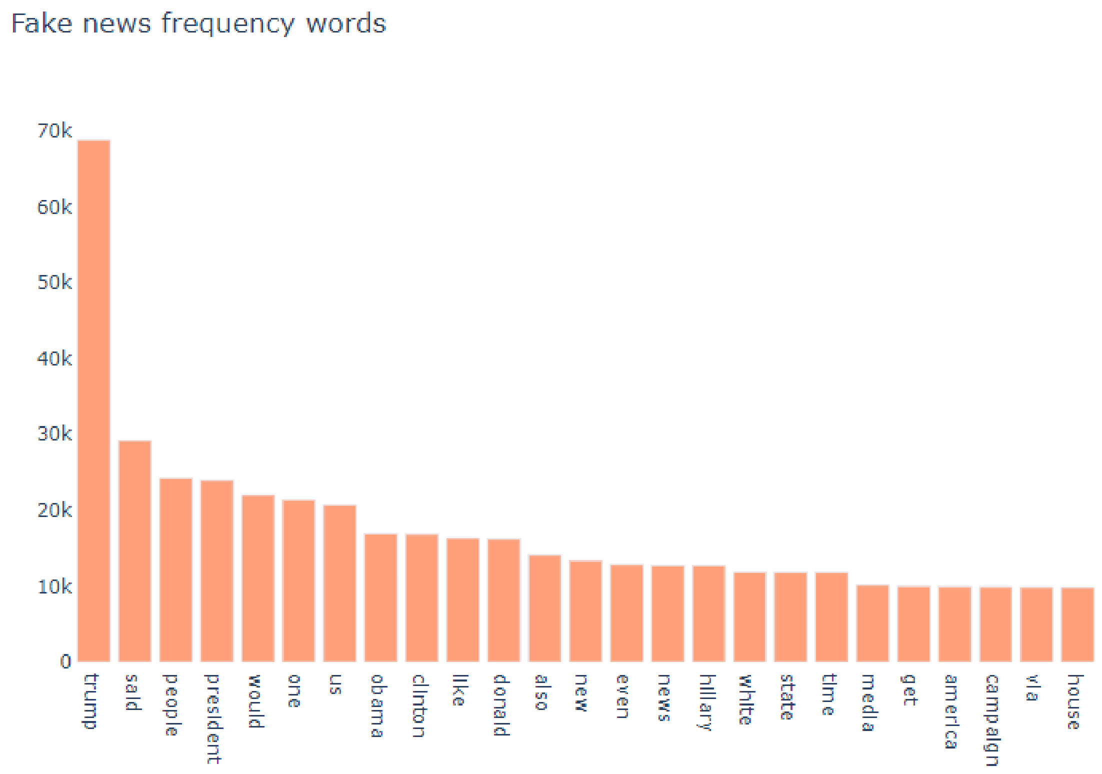
When splitting tokens we can see that fake news has many different tokens than the real ones, this was to be expected assuming that inside the fake news there are twitter posts where people use, abbreviations , slang word and language addictions in non-formal writing. I believe it is interesting to observe the occurrence of words that do not exist in the English language to see the difference between true and false news.

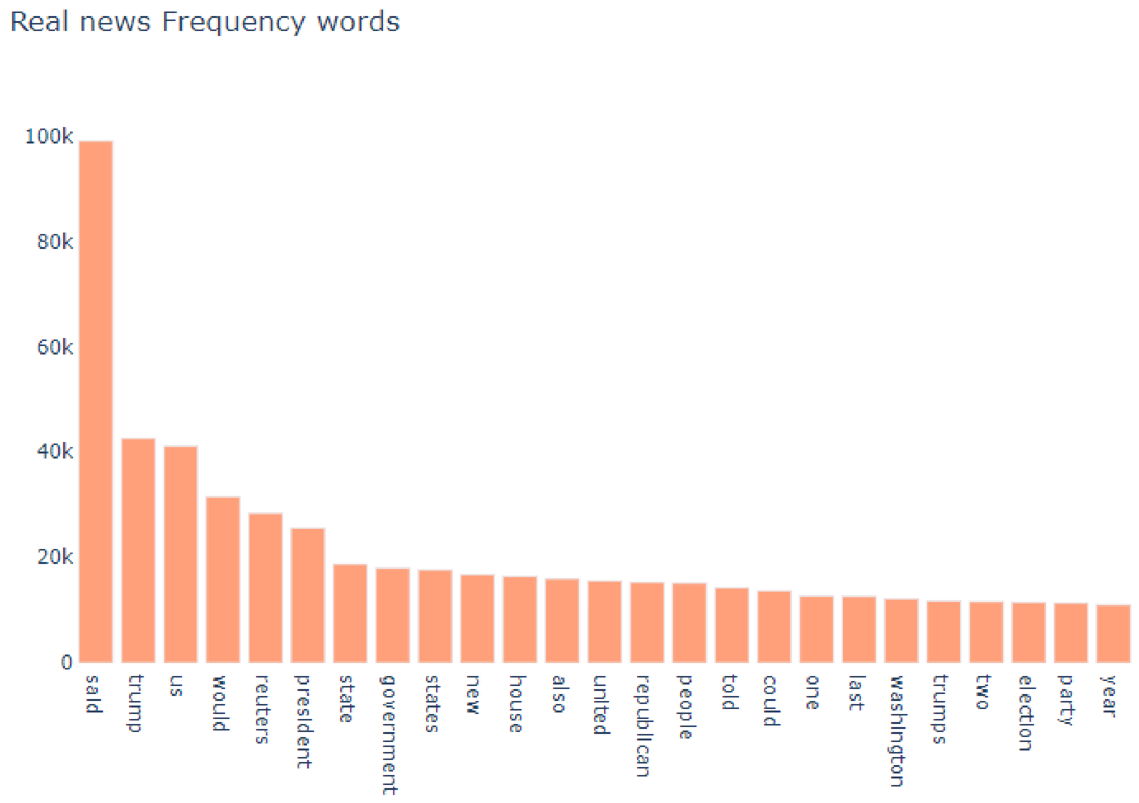


In this plot more than 70% of the words in the news fakes were not found in the dictionary used for verification, it is important to make it clear that it is not a perfect dictionary but that it already brings this section that many words are really misspelled but so far everything that has been done has been in the data without any preprocessing, so let's apply a preprocessing that clears some characters and normalizes the text so that we can compare again.

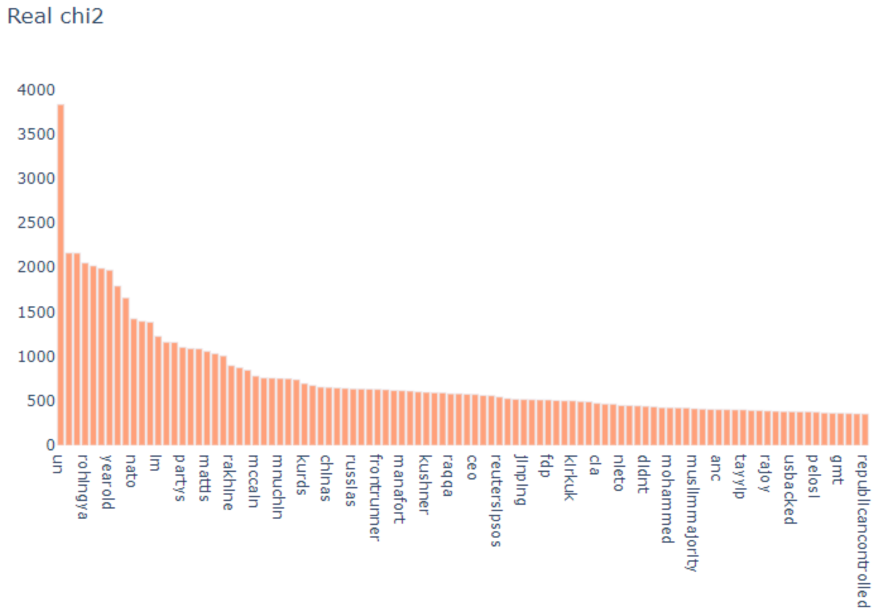


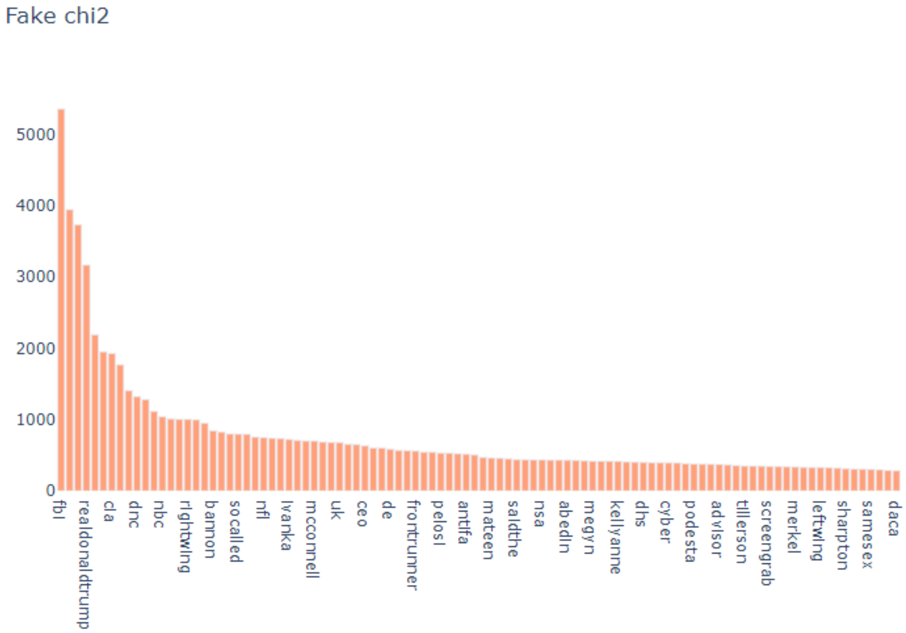
After applying some preprocessing it only showed the difference in the aspect of writing false news in relation to the real ones. Since the words are so impactful let's see if there are any that stand out in relation to the others, raising the most relevant words of each type of news.


Fake news frequency words

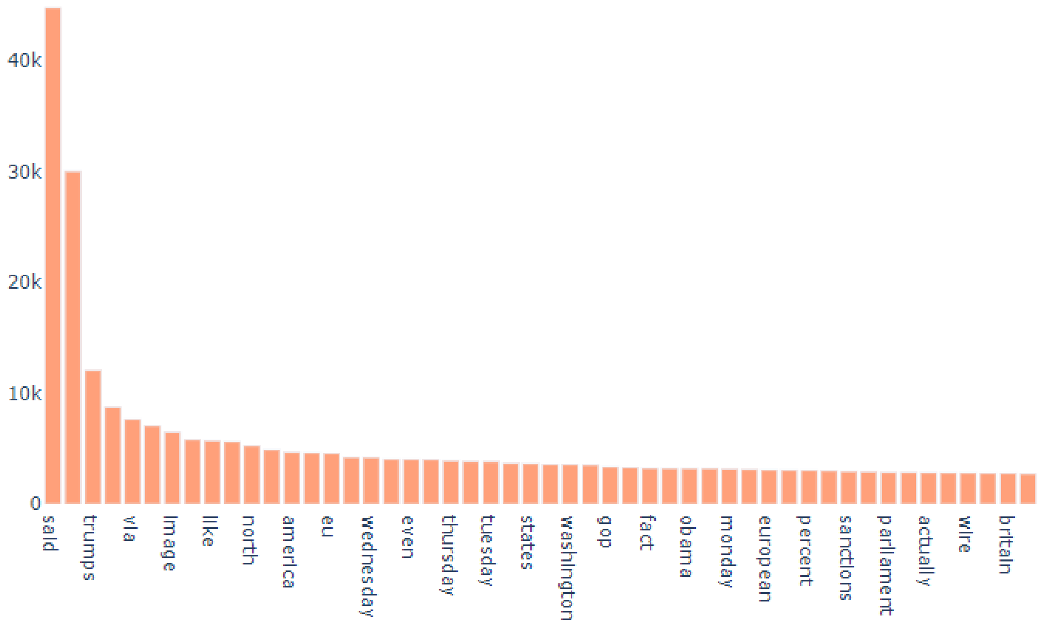

Real news Frequency words

The most frequent words of both types of news is not very different, only the frequency is not enough to differentiate and demonstrate the bias of the data it is necessary to establish some order of importance, and for that I will use the chi2 hypothesis test to raise the most relevant words in the dataset. But first we will see to do this in relation to the words that are misspelled to evaluate their impact on the news.



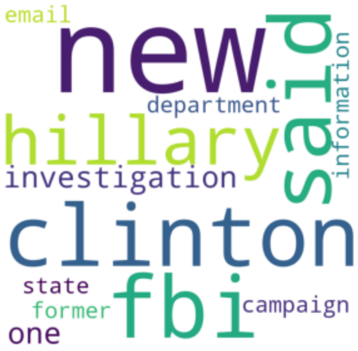


From the histogram here we can see the words raised in general are acronyms that are not present in the dictionary and words that are incorrect after preprocessing. Now we will analyze chi2 in all dataset to see which words impact the data more in relation to its relevance.

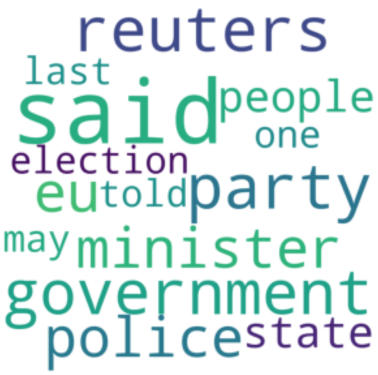


The words ‘said’ and ‘reuters’ are words that have a great prominence in the data set. To better understand how the contexts of true and false news are, I will do the modeling by topics to understand if these words of greater prominence are present in well-defined topics. I can do this by generating multiple word clouds for fake news topics and real news topics.

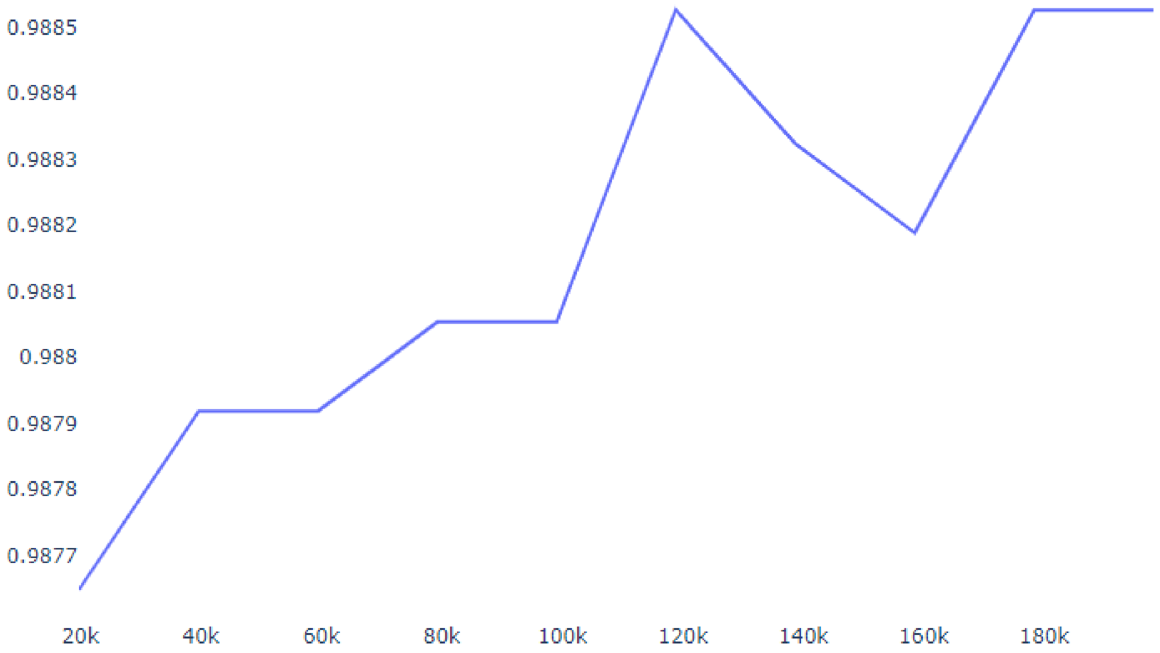
Fake:



True:



It’s interesting how the topics created do not portray misspelled words or twitter users, demonstrating that at least the subjects covered can be useful for an eventual identification of fake news. To analyze the words and their impact on the classifier, I will select every 10% of the total of features and evaluate if reducing the number of words the classifier loses performance.



Even with only 10% of the features, the classifier already shows a result of 98%, so there must be a set of words that make identification easy, to perform a more crunchy test I will only raise one feature according to chi2 and observe how a classifier based on this word behaves. The selected word was 'reuters', I will make a classifier based on whether the reuters is present in real news and if it is not false. If we remove the reuters word and replaice it with a inconspicuous word like ‘said’ we reach a 70% accuracy. This shows how biased the fake news data is.

Conclusion

The dataset has many features that point to bias towards fake news, apparently a poorly structured dataset that does not allow you to raise more consistent information in relation to real characteristics that fake news can present. The fact that fake news is mixed with twitter posts points out a lack of care for the data, generating a dataset that is extremely simple to be able to accept high results without much engineering. Resulting in the end the use of a single word can allow almost 100% what is true or false news.

Works Cited

geeksforgeeks. (2020, April 18). *Generating Word Cloud in Python.* Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/generating-word-cloud-python/

Li, S. (2017, Sep 28). *Building A Logistic Regression in Python, Step by Step.* Retrieved from towardsdatascience: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

Lisa , C. A. (2020, November 13). *Getting started with text analysis in Python.* Retrieved from towardsdatascience: https://towardsdatascience.com/getting-started-with-text-analysis-in-python-ca13590eb4f7

Pythonspot. (2020). *Tokenizing Words and Sentences with NLTK.* Retrieved from pythonspot: https://pythonspot.com/category/nltk/

Singh, T. (2019). *Natural Language Processing With spaCy in Python.* Retrieved from realpython: https://realpython.com/natural-language-processing-spacy-python/#:~:text=spaCy%20is%20a%20free%2C%20open-source%20library%20for%20NLP,use%20and%20provides%20a%20concise%20and%20user-friendly%20API.

Steen, D. (2020, Sept 26). *Progress bars for Python with tqdm.* Retrieved from towardsdatascience: https://towardsdatascience.com/progress-bars-for-python-with-tqdm-4dba0d4cb4c#:~:text=tqdm%20is%20a%20Python%20library%20that%20allows%20you,be%20installed%20using%20the%20pip%20install%20tqdm%20command.

tutorialspoint. (2020). *Python - Regular Expressions.* Retrieved from tutorialspoint: https://www.tutorialspoint.com/python/python_reg_expressions.htm

Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).