Detecting Fake News Using Machine Learning

Elsa Norman

Yorktown High School

ABSTRACT

Fake news has had a significant effect on society and politics. To aid in combating the spread of misinformation, we worked to develop a machine learning algorithm that could detect fake news based on textual data. We used a count vectorizer to vectorize our text which we then inputted into Logistic Regression, Support Vector Machine (SVM), and Linear Support Vector Classifier (SVC) models. The greatest accuracy score achieved was 99.97% with the Linear SVC. We discovered however that there was a significant difference in how the real and fake news datasets were constructed that would not translate into real life: the true news articles contained quotation marks, apostrophes, and dashes while these characters were not present in the fake news articles. Because of this, we also developed a more applicable Logistic Regression model removing these specific characters from the dataset all together with an accuracy score of 98.4%.

Introduction

Through this research, we aimed to develop an A.I. model that could classify news articles as either real or fake based on the text. In 2016 and the years to follow, concerns of fake news have spread across social media, providing users with false information [2]. Researchers at MIT found that fake news spreads to people six times faster than true news [4]. In the 2020 U.S. presidential election, President Trump posted on Twitter that it was “statistically impossible [for him] to have lost,” despite opposing evidence, leading his followers to make incorrect assumptions [5]. On January 6th, 2021, the effects of Trump’s tweets turned violent as thousands of the president’s followers protested Biden’s victory at the U.S. Capitol Building [5].

For our model, we used a dataset of 44,898 articles containing real and fake news. We took the raw language data from the text of the article and tokenized it. This was then converted into numerical data using a count vectorizer. We observed Logistic Regression, Support Vector Machine (SVM), and Linear Support Vector Classifier (SVC) models to classify the inputted contents of an article as either real or fake.

Background

In their exploration of the detection of fake news using machine learning at Chulalongkorn University in Thailand, Aphiwongsophon et al. received an accuracy score of 99.9% using a Linear SVM model and CNN model to detect fake news through twitter posts with profiled attributes [3]. The model depended on how other users classified the post to make its decisions, which may turn difficult when trying to classify a post that has yet to be viewed by others [3]. We believed it would be best for our research to base the model exclusively on text data so that fake news could be detected even before it spreads.

H. Ahmed et al. explored numerous classification techniques and both a count vectorizer and weighting metric to represent the text data. Their best accuracy was obtained through the use of Linear SVM and Term Frequency-Inverse Document Frequency (TF-IDF), which takes into account the significance of each word in the dataset [1]. They managed an accuracy score of 92% using a dataset from Kaggle [1]. The model was limited to only one dataset and Ahmed et al. hoped to improve it by expanding the training data through the LIAR dataset [1].

Dataset

Our dataset for this project was retrieved from https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset. It contained 21,417 true news articles and 23,481 fake news articles. Information about each article was given, including the title, main text, subject, and publication date. The true news was found to be sourced mainly from Reuters, while no true source was found for the fake news articles. This meant that all true news articles began with the word 'Reuters'. Because we wanted to be able to use this model on articles beyond just one source, we removed the word 'Reuters' from each of the articles when preprocessing the text.

The main text of each article was the input data used in training the model and the output would be whether the article was true or false. The subject of the article was excluded as the subject categories for the real and fake news seemed to be mutually exclusive and subject categories won't always come with an article, making our model less real-world application friendly. The dates and titles of the articles were also excluded. We tokenized the data, replacing the long string of article text with an array of the individual words.

We removed stop words, which are usually common words that tend to only provide "low-level information" about an article [7]. These include words such as 'it', 'was', and 'or' and were removed from articles so that the model would give more focus to the higher-level information. We converted all of the words to lowercase so that a word capitalized at the beginning of a sentence would be seen the same as if it were found elsewhere. We considered removing punctuation but decided against it as certain punctuation marks such as the exclamation may be more common in one type of news than the other.

We used a count vectorizer to convert the text data into numerical data. The arrays of words from the articles were replaced with arrays of numbers. The 1000 most common words from the training data were used, so that each article had a corresponding array of 1000 numbers. Each number symbolized the number of times a certain word appears in the article.

We used a test size of 33% and stratified based on y when splitting the data into training and testing data. After splitting the data, we used a word cloud to visualize common words in the real and fake news articles in the training data. The most popular words in fake news articles seemed to be 'Trump', 'said', 'President', and 'New', while in real news they were 'said', 'Trump', 'State', and 'would'.

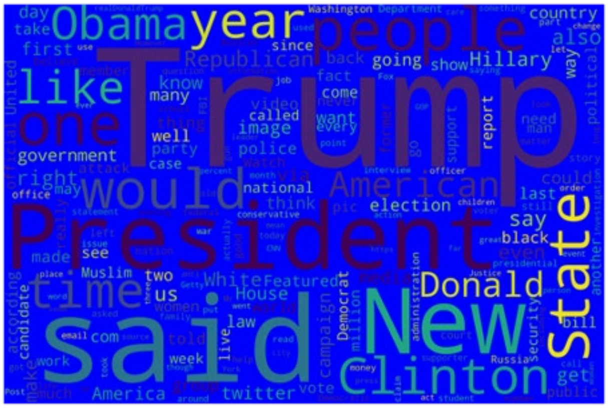

Figure 1: Word Cloud for fake news

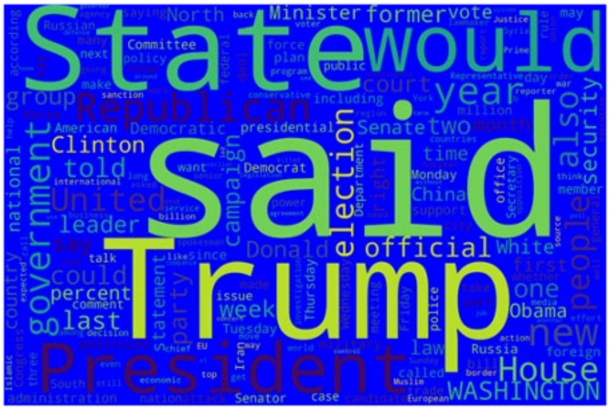

Figure 2: Word Cloud for real news

Methodology

We looked at several classification models to achieve our simple binary classification task. We tested Linear Support Vector Classification (SVC), Logistic Regression, and Support Vector Machine (SVM) models in classifying the data.

When tokenizing our data, we tested different combinations to see how the accuracy of our models differed. We tried removing all punctuation and removing stop words. We also tried using a lemmatizer, which replaced words with their stem if they were plural or used in a different tense.

While we tried removing all punctuation, as one of our tests, throughout all of our tests, we removed apostrophes, quotation marks, and dashes. This is because in our initial models that included punctuation, when looking at what words the model considered to be associated the most with real news, we noticed that quotation marks, apostrophes, and dashes had the highest ranking. When exploring the real and fake news provided in the dataset further, we noticed that punctuation marks seemed to have been excluded from the text data for the fake news articles. Because we wanted this model to be applicable outside of the dataset (and clearly fake news would not always lack punctuations like quotation marks and real news wouldn't always possess it), we removed such punctuation marks from the dataset for all of our tests.

The remaining pieces of punctuation were kept in the data, however, as we saw that punctuation marks like the exclamation point and ampersand tended to indicate that an article is fake news but could still be found in real news articles.

Results and Discussion

We tested our Logistic Regression, SVM, and Linear SVC models with different methods of preprocessing the data to see what combination would lead the models to be more accurate. We looked at the effects of removing stop words and punctuation, as well as using a lemmatizer to replace words with their stem. We converted all words to lowercase and tried removing specific words and punctuation that seemed to only appear in the real news dataset due to the differing in organization of the real and fake datasets.

With each of these adjustments, the models were run and the accuracy scores were noted. We also looked at the accuracy scores of the models without altering the data to see the effect that each change had.

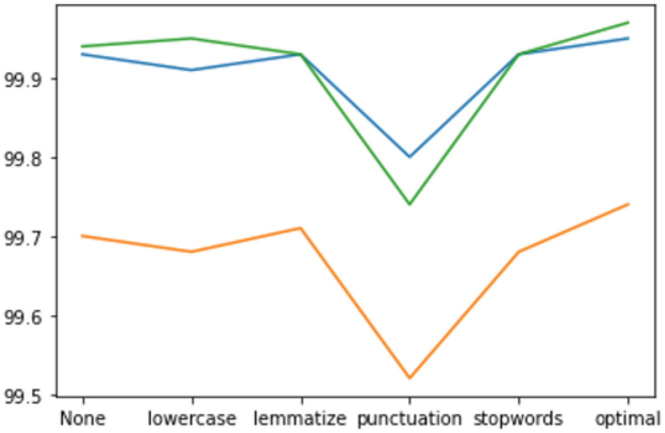

Figure 3: Accuracy rates for removing different features

The graph above depicts the varying accuracy scores of the models, the Linear SVC model's accuracy represented in green, the Logistic Regression model's accuracy in blue, and the SVM model's accuracy in orange.

Each model had an accuracy of over 99.5% with each alteration. The SVM had the worst accuracy while the Linear SVC was able to achieve the greatest accuracy score. This model tends to work well with text categorization as it can generalize datasets with large numbers of features [6]. We found that the Linear SVC was most efficient with default parameters. With the removal of stop words and the word 'Reuters', in addition to converting all words to lowercase, our Linear SVC model was able to achieve an overall accuracy score of 99.97% when determining the truthfulness of an article.

With the optimal accuracy mentioned with the Linear SVC model, we looked into changing the model's parameters to see if we could make it more accurate. We mainly looked into altering the C value but found that the default for each parameter gave the best results.

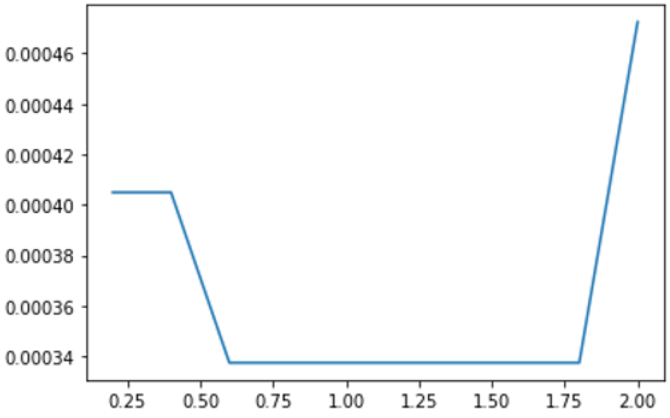

Figure 4: Error as a function of C value

The graph above depicts the effect of the C value on the mean absolute error of the model, with 99.97% still the highest accuracy the model achieved. The C parameter for the Linear SVC model had a default value of 1, which is included in values that caused the lowest error in the graph.

Continuing to work with the Linear SVC model, we further explored its accuracy. Both the true and false news achieved a precision, recall, and f-score of 0.999. We observed the ROC Curve as well, which can be seen in the figure below.

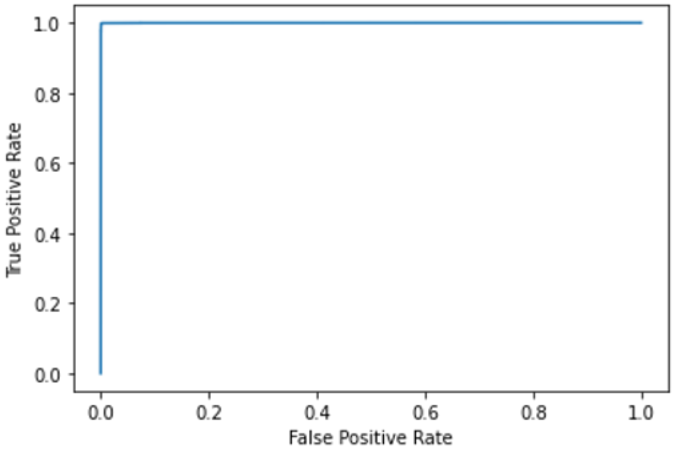

Figure 5: ROC Curve

In observing the dataset, we noticed that each true article contained quotation marks, apostrophes, and dashes, while none of the false articles contained them. In order to make our dataset more accurate outside of the dataset, we removed these punctuation marks entirely from the dataset. In doing so, our Linear SVC accuracy dropped from 99.97% to 97.89%. Both the Logistic Regression and SVM models had an accuracy of 98.4%.

We decided it best to use this as our final model as it is most applicable with articles outside of the dataset. This allows the model to detect fake news without getting messed up by the presence of certain punctuation marks.

The limitations of the dataset turned out to be the most debilitating in the development of our model. Because the true news articles all came from one source, it is more difficult to accurately apply our model to articles from other sources.

In the future when continuing our model, we plan to use more datasets with a wider variety of sources so that our model can work with multiple news sources. This could include the LIAR dataset that was mentioned in our literature review.

Conclusion

In this project we explored journalism and worked to develop a model that could detect fake news based on the text of an article. False information is shared quickly, especially on Twitter, where it can be spread up to 20 times faster than real news [4]. This rapid growth then creates widespread misinformation particularly in politics, influencing presidential elections such as that in 2016 (Allcott, Gentzkow). With our model, we hoped to be able to spot fake news and stop it from being shown to others.

We were able to get an accuracy of 99.97% with our Linear SVC model by removing stop words, converting words to lowercase, and removing the word ‘Reuters’ in the articles. We tokenized the data and used a count vectorizer to convert the article text into numerical data. Despite the high accuracy score, we realized that our model only worked well within the given dataset. We removed punctuation marks that only appeared in real news, as that seemed to be causing our model to make mistakes with articles outside of the dataset. With this setup, the highest accuracy we obtained was 98.4% from our Logistic Regression model. In continuing our research, we hope to use more data in training our model from a variety of sources to make our model more applicable and accurate when classifying other articles.

Acknowledgements

Thanks to Jane Boettcher for help and guidance throughout this research project.

References

[1] Ahmed, H., Traore, I., Saad, S. “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques.” https://www.uvic.ca/ecs/ece/isot/assets/docs/Detection%20of%20Online%20Fake%20News%20Using%20N-Gram.pdf?utm_medium=redirect&utm_source=/engineering/ece/isot/assets/docs/Detection%20of%20Online%20Fake%20News%20Using%20N-Gram.pdf&utm_campaign=redirect-usage.

[2] Allcott, H., Gentzkow, M. “Social Media and Fake News in the 2016 Election.” https://web.stanford.edu/~gentzkow/research/fakenews.pdf

[3] Aphiwongsophon, S., Chongstitvatana, P. “Detecting Fake News with Machine Learning Method.” https://d1wqtxts1xzle7.cloudfront.net/59012493/Detecting-Fake-News-submit20190424-97672-6rhwzo-with-cover-page-v2.pdf?Expires=1656512062&Signature=YDwzCpIwNJzqCViEMp~j-OvEYgY11u-F-49RDTNiph9OQ10xTgEnHPPpUXmo2I6d3sO2isDxeyvn5QmLVZB-CalmLpmwsPOhOCsFjR06~VwlgW8nyg94t-49T91wErp0FNKhdEJJaGFkbMlG28Qup419mRYO-6cBQIkLqSRLyc3pEEsnx1XP-Wp19UqW~RySlW0EyGeMtyZ5dxQnxn-zQZ56FiaNo26dmaiHmmSzlizS3bHW7d70DuFqXxPCNt~oijx~HpKHgwsZtGnUud4mPvekbnZdE-yUHyKT04jWhPXBwFHHeElUB5srMQD38Rs-KTr49WhKXKDwksW8ueiwSg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA.

[4] Dizikes, P. “Study: On Twitter, false news travels faster than true stories.” MIT, 8 Mar. 2018, https://news.mit.edu/2018/study-twitter-false-news-travels-faster-true-stories-0308.

[5] Dreisbach, T. “How Trump's 'will be wild!' tweet drew rioters to the Capitol on Jan. 6.” NPR, 13 Jul. 2022, https://www.npr.org/2022/07/13/1111341161/how-trumps-will-be-wild-tweet-drew-rioters-to-the-capitol-on-jan-6.

[6] Joachims, T. “Text Categorization with Support Vector Machines: Learning with Many Relevant Features.” https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf.

[7] Khanna, C. “Text Pre-Processing: Stop Words Removal Using Different Libraries.” Towards Data Science, 10 Feb. 2021, https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a#:~:text=Stop%20words%20are%20available%20in,focus%20to%20the%20important%20information.