**Creating Natural Language Processing Models for Detecting Fake News with Limited Data**

Lakshya Mehta

Summer Ventures in Science and Mathematics

The University of North Carolina at Charlotte

Ethical Data Science

Dr. Marco Scipioni

July 29, 2022

# Abstract

Fake news is often spread on social media sites at a volume that requires automation to moderate. Failing to do so undermines trust in all information, leading to further insulated online echo-chambers, exacerbated health crises, and denial of democratic processes. A common approach is employing natural language processing (NLP) to detect fake news from the data about the articles, usually with deep-learning models. The research aims to identify an accurate non-deep model, using only the text and titles of an article. After preprocessing a labeled dataset of article titles, multiple algorithms were trained: multinomial naive Bayes (MNB), support vector machine (SVM), logistic regression, passive aggressive classifier (PAC), random forest, and extreme gradient boost (XGB). The models were tested on a subset of the training data and the titles and text of an article from another isolated dataset. The models were compared using the binary classification reports. Across the board, the models struggled with accurately predicting based off of the titles of the test data, but performed competently on the texts, with the best algorithm being PAC with an accuracy of 96.084%.

*Keywords*: natural language processing, classification report, fake news, non-deep model

**Creating Natural Language Processing Models for Detecting Fake News with Limited Data**

# Introduction

Information is an important resource, and the manipulation of information using fake news is a potent weapon. This power is especially exploited for political and financial gain, specifically targeting and convincing older populations (Zhang & Ghorbani, 2020, 102025). This ability to convince has been debated to have affected the outcome of the 2016 presidential election, and further exacerbated the COVID-19 pandemic. Because of the threat to public safety and democratic processes, fake news must be automatically moderated. Effective content moderation of fake news can help prevent insulated echo chambers that reject truth and divide people. Using natural language processing (NLP), this study aims to create a non-deep machine learning model to identify articles as real or fake based only on the text or title. Since this data is easily available for any article, a successful model can be widely deployed to moderate content. The code associated with the study is linked at <https://github.com/LMehta0210/Fake-News-Research>.

# Literature Review

Multiple studies have been conducted on detecting fake news with machine learning and NLP. The majority of these papers focus on training deep-learning models, which are more computationally intensive than the lighter non-deep models. One study by Kumari et al. (2022) attempted sentiment analysis, interpreting an article as positive or negative, combined with assessing an article’s novelty to better predict an article as fake or not. Other research focused on parts of the preprocessing, especially the pre-processing involved with vectorizing text, as done by Suhasini and Vimala (2021) and Nagy and Kapusta (2023).

## Algorithm Selection

The algorithms chosen within this study are based on the algorithms tested by Joyce George (2020), which includes the multinomial naive Bayes (MNB), support vector machine (SVM), and passive aggressive classifier (PAC) algorithms (George, 2020). The performance of these models provided a baseline from which to observe the improvements when adding stop words processing and N-grams vectorization, both of which are further explained as part of the methods. The other algorithms are commonly used for classification tasks. A K-nearest neighbor model was attempted but proved unable to test itself within eight hours of processing, and was thus abandoned.

# Materials & Methods

## Dataset Description

Using Python Jupyter notebooks, multiple machine learning algorithms were trained on one dataset, and tested on an independent dataset. The two datasets are sourced off of kaggle.com. The training data has nearly 70,000 labeled titles. There are ~34,000 fake news titles sourced from right-wing extremist sites, the EUvsDisinfo project, and a previous study (Steven, 2022). Approximately another 35,000 real news articles sourced from the *Washington Post, New York Times*, and Reuters (Steven, 2022). The balanced nature of this data ensures limited bias within the model and that the metrics are representative of real-world performance. The test data is also a 50-50 split between fake and real news, roughly 6000 values in total. The data itself is from an open source website and includes both the titles and the text of the labeled articles. (Jillani, 2022).

## Preprocessing Methods

The key to successful NLP models is the processing of the naturally written words into numerical data for a machine learning algorithm to train with. Generally, this requires homogenizing the text from an article by removing punctuation, and lowercasing the letters. After, the sentences must be tokenized (split into individual words), before “stop words” can be removed— words (the, of, a, etc.) that are part of grammatical writing but hold no meaning. The tokenized and condensed text is then lemmatized, turning all words into their root word, as opposed to stemming which simply deletes suffixes (Shetty & Koss, 2022). These methods outlined by Shetty and Koss were used to help count the text in the training data to visualize the most commonly used words. The training data still held words that held limited context, so they were added to the list of stop words for the preprocessing algorithm to filter.

## Training Methods

The homogenized and lemmatized text still cannot be understood by a numerical algorithm, so the words must be vectorized. Using pipeline model code from Joyce George, the models could be trained using the relatively basic count vectorizer, more commonly known as the “bag of words” method. The counted words would then be transformed using a term frequency-inverse document frequency (TF-IDF) transformer to provide importance to words inversely to their frequency within the data (George, 2020). To increase the accuracy of the models on the testing texts, the vectorizer was changed to an N-grams method, which can look at all occurrences of any consecutive set of words within the dataset (Shetty & Koss, 2022). The program uses N = 2, so every consecutive set of two words within a sentence. Since the N-grams vectorized data is still fed through the TF-IDF transformer, this work is similar to that outlined by Suhasini and Vimala, who proposed a joint N-grams and TF-IDF processing method. Processing the data this way should allow for a mild contextual understanding, since the stop words processing should limit the sentences to the keywords, and therefore the barebones of some ideas will be correlated with each other. This N-grams vectorizing and TF-IDF transforming method was used in every pipeline, and the only difference was the type of model used. The algorithms trained are MNB, SVM, logistic regression, PAC, random forest and XGB.

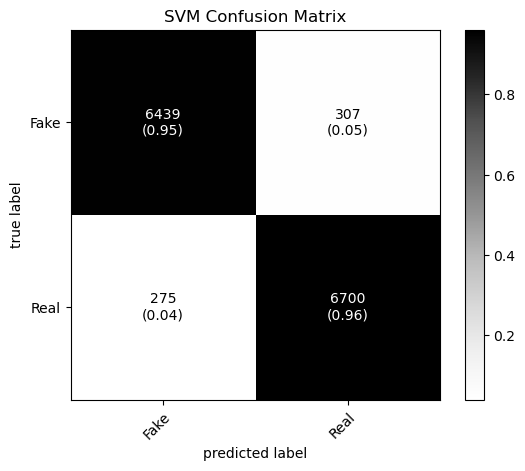
# Results

## Testing Methods

Each model was tested against a 20% subset of the training data set. The text and the titles from the test dataset from Jilani were extracted and used to independently test the models. This was to ensure the models were not overfitted and could generalize well enough to be deployed into the real world. Competency in either the external titles or texts would allow for successful deployment, even if processing a whole text will take more time. Therefore, performance on these testing subsets is the most important metric for evaluating the success of the models.

**Figure #1**

*SVM Confusion Matrix for the Training Dataset*



*Note.* Reading left to right, top to bottom, the quadrants are true negatives, false negatives, false positives, and true positives. This confusion matrix shows a strong model since most data is predicted correctly.

## Confusion Matrix Data

The models were assessed on the three testing subsets using classification matrices, such as figure 1, which map the predictions of a model versus the actual binary labels of the data. All the confusion matrices are in the GitHub repository linked at the end of the introduction. Overall competency was measured using an accuracy score, combining the true positives and true negatives and dividing by the total.

**Table # 1**

*Accuracy Scores for Various Algorithms across all Test Subsets*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Subset | Algorithms | | | | | |
| MNB | SVM | Logistic Regression | PAC | Random Forest | XGB |
| Training Data subset | 91.305% | 95.578% | 93.412% | 96.028% | 93.186% | 91.670% |
| Test Dataset Titles | 69.997% | 53.245% | 53.117% | 55.195% | 49.760% | 49.888% |
| Test Dataset Texts | 92.935% | 95.668% | 86.765% | 96.084% | 92.919% | 77.206% |

*Note.* Every model performs far better on the external texts than on the external titles.

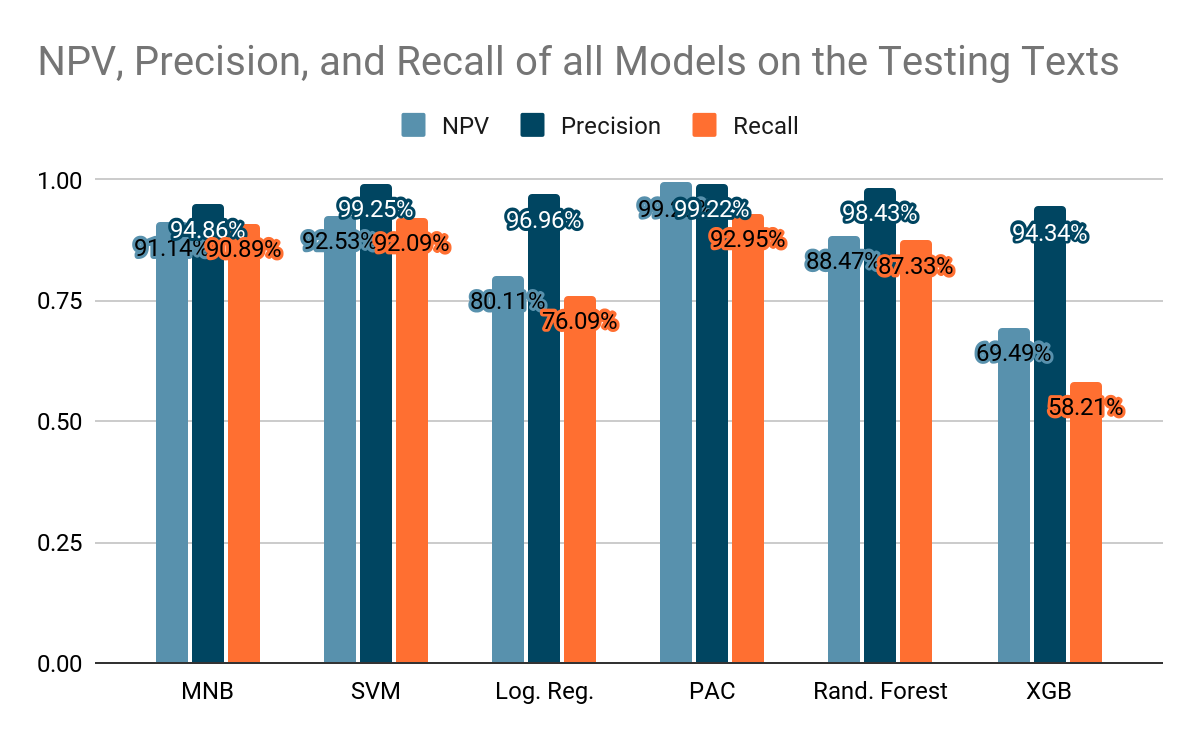
As seen above, the models all performed competently on the subset of the training dataset, but failed to extrapolate for predicting titles not part of the training dataset. All the models except the MNB model have accuracies near 50%, meaning they are essentially guessing on whether an article is fake or not. However looking at another two metrics, the negative predictive value (NPV) and the recall, will give us further insight. The NPV measures how often a model correctly classifies a datapoint given that the datapoint was predicted as negative. The recall measures the chance the model correctly classifies the data given that the datapoint is positive. The NPV when predicting the news as fake for the five models was also near half, and the recall rate was always under 15%. This means the five models were always guessing the external titles as fake news, the most egregious model being random forest with a NPV of 50% and a recall rate of nearly 0%.

# Analysis

Another common metric for binary classification is the precision which is the number of true positives divided by the predicted positives. Figure 2 shows all the metrics on the text

**Figure # 2**

*Precision, Recall, and NPV of all Models on the Testing Texts*

**

*Note.* PAC performs the best overall, matching its higher accuracy, while XGB and logistic regression perform the worst.

Using these metrics on the testing texts of the articles allows us to better see how the model performs than just a single accuracy number. In the context of detecting fake news, the NPV demonstrates if a model will blanket too many articles as fake, so a higher NPV results in a lower likelihood. The precision shows how accurate a model will be with its real news guesses, and the recall shows how a percentage of the truly real news that the models will classify as real. These metrics allow us to see how successful a model is at allowing correct information to be sent out into a social media platform. Based on these metrics and the original accuracy score, the passive aggressive classifier algorithm was the best in this case at identifying an article as fake or real news.

# Discussion & Conclusion

The MNB, SVM, PAC, and random forest algorithms were competent at detecting fake news, but only specifically in identifying the articles based on the text of an article. This result is likely from the whole text simply having more data for a model to work with before making a prediction. However, the idea that training an algorithm on smaller data to process larger pieces of data is important for the deployment of NLP as a whole. A lighter training dataset will directly correlate with a faster training time, allowing for algorithms like these to quickly adapt as people weaponize fake news to target specific groups of people. Needing less training time, companies can feasibly retrain on a subset of the data as people see shifts in the targets of political pundits. Although having a competent model on the test dataset titles would have been optimal for minimizing the time for a prediction, the text of an article is still easy to extract from a link being shared on a social media site, so content moderation is still efficient. Especially since the data is only the texts, and not anything about the subject classification or political leaning of the article, these models can provide predictions for many more articles.

## Ethical Considerations

The raw text input into the models also limits biases that may arise from having information about other categories. For example, right-leaning outlets weaponize misinformation more than left-leaning outlets, but without that information, models can’t learn to generalize about right-leaning outlets. Instead the text of an article has to act as proxy for political affiliation, but the model will still allow accurately written stories through.

Inherently, some discussion must occur about the censorship issues associated with having a machine moderate content. Because of the complexity of NLP algorithms, they are essentially black boxes to the users, where we put in data and extract predictions with no understanding of what the algorithm considers as an indicator of an article being misinformation. However, I think a larger concern is the control any company has of deploying a machine learning content moderation model. Any company will have control over the data, which can be filtered to achieve a desired outcome from a machine learning model. Social media companies would have the ability to censor information from specific groups of people and about specific topics, allowing for the spread of prejudiced information. Again, the limited information available to the models in this research has to act as a proxy, which can help limit biases that could arise from directly having access to information such as the race or gender of a user posting an article. Understanding how companies can manipulate these models is important, since companies can more easily dismiss unpopular choices made by an algorithm, even if an algorithm is trained to make such choices.

## Limitations of the Study

One major limitation of the study is the subpar reliability of the data, being sourced from kaggle.com with limited description about the sourcing of the test data. Better data would be from a research institution or scraped from the Internet within the study. The models could be further tested by actively scraping the Internet for articles and extracting the titles and texts, and comparing the predictions to judgements from reputable sites on fake news detection, including Snopes, PolitiFact, and FactCheck.org. Another place the study can be improved is by performing model optimization methods such as cross-validation and grid-search. Cross-validation would train the algorithms multiple times with different training data subsets to decouple the specific selection of the training data subset from the model’s performance. Grid-search would try different combinations of the hyperparameters fed into the algorithms to get the highest score.

## Questions for Future Research

Specifically with this study, there are no features extracted from the text that is fed into the algorithm, in this case by design to limit the preprocessing. However, feature extraction can still provide the benefits listed above regarding only needing text, but may allow for processing only the titles or processing shorter texts. For example, including information about the frequency of capitalized letters or the average length of words may help create correlations that a model can use to better predict an article as fake news just based on the title.

For classifying fake news as a whole, further research must be completed in actively fact-checking information within an article. Developing a model that understands logical fallacies and how statistics can be misrepresented to sway opinion is required to detect malintent, which is what differentiates an incorrectly written article and an article built to spread misinformation. Another important point is that text is not the only medium through which misinformation can be spread. Developing algorithms that can detect artificially generated or tampered audio and images is required as well, especially since they are very convincing forms of media.

# References

George, J. A. (2020, June 14). *Fake News Detection using NLP techniques | by Joyce Annie*

*George | Analytics Vidhya*. Medium. Retrieved July 19, 2023, from <https://medium.com/analytics-vidhya/fake-news-detection-using-nlp-techniques-c2dc4be05f99>

Jillani. (2022). *Fake or Real News*. Kaggle. Retrieved July 18, 2023, from

<https://www.kaggle.com/datasets/jillanisofttech/fake-or-real-news>

Kumari, R., Ashok, N., Ghosal, T., & Ekbal, A. (2022, January). What the fake? Probing

misinformation detection standing on the shoulder of novelty and emotion. *Information Processing & Management*, *59*(1), 102740. Science Direct. <https://doi.org/10.1016/j.ipm.2021.102740>

Nagy, K. S., & Kapusta, J. (2023, May 25). TwIdw—A Novel Method for Feature Extraction

from Unstructured Texts. *Appl. Sci.*, *13*(11), 6438. <https://doi.org/10.3390/app13116438>

Shetty, B., & Koss, H. (2022). *NLP Machine Learning: Build an NLP Classifier*. Built In.

Retrieved July 18, 2023, from <https://builtin.com/machine-learning/nlp-machine-learning>

Steven. (2022, July). *Misinformation & Fake News text dataset 79k*. Kaggle. Retrieved July 17,

2023, from <https://www.kaggle.com/datasets/stevenpeutz/misinformation-fake-news-text-dataset-79k>

Suhasini, V., & Vimala, N. (2021, June 7). A Hybrid TF-IDF and N-Grams Based Feature

Extraction Approach for Accurate Detection of Fake News on Twitter Data. *Turkish Journal of Computer and Mathematics Education*, *12*(6), 5710-5723. ProQuest. <https://www.proquest.com/docview/2640416454?parentSessionId=OHIdFJoiZTk8KnB2WfNLzuxlhJtoU6i6EgoQhqQvDdY%3D>

Zhang, X., & Ghorbani, A. A. (2020, March). An overview of online fake news:

Characterization, detection, and discussion. *Information Processing & Management*, *57*(2), 102025. Science Direct. <https://doi.org/10.1016/j.ipm.2019.03.004>