Fake News Detection In The Absence of Transformers

Cassady Shoaff, Raz Besaleli, Iyanuoluwa Shode
Montclair State University
1 Normal Ave
Montclair, NJ 07043
{shoaffc1,besalelir1,shodei1}@montclair.edu

Abstract

Fake News, Naive Bayes, LinearSVC, Logistic Regression

In this paper, we compare the performance of several primitive classification algorithms, including Bernoulli Naive Bayes, LinearSVC, and Logistic Regression, in automatically detecting fake news with minimal preprocessing. Although we find that Logistic Regression classification is ostensibly most effective in detecting fake news, which agrees with previous findings, there are some interesting results that might make Naive Bayes or LinearSVC that might make them appropriate candidates in domain-specific settings.

1 Introduction

Social media platforms such as Twitter, Instagram, Reddit and many other socializing applications have become everyday sources of local news. However, these platforms have hugely propagated widespread fake news that is inevitability causing havoc in our societies by spreading misinformation. Unfortunately, news consumers have little or no time to investigate the authenticity of the news presented to them before digesting the news, thereby consuming false information.

Because manual classification and selection in order to detect the credible news out of the legion of fake news is rather too tedious, time-consuming, energy tasking and can eventually lead to bias, it seems necessary to employ classification models that are faster, void of human bias, and accurate. In order to correctly classify the news, we are using the machine learning text classification using several different models. In order to do this, we extract features from a given set of labelled data and then placed into classifiers that can accurately tell the category the news belongs to. For our investigation, we experimented using multiple classifiers such as the probabilistic machine learning classifier Naive Bayes, Logistic Regression and, Linear Support Vector Classifier (LinearSVC) and determine their performances based on their levels of accuracy in detecting the fake news.

Classification methods such as the ones we experiment with are easily accessible for researchers and can be utilized for fake news detection quickly via common Python packages, for example, scikit-learn. Transformer-based models are an alternative to these methods, but may be less accessible because of the lengthy request process to gain access to the models and checkpoints. In particular, transformer-based model GROVER (Zellers et al., 2019) is specifically designed to tackle the task of fake news discrimination, but the checkpoints for this model are locked away behind several layers of bureaucracy in order to prevent bad actors from using it to create fake news. While this human-verified retrieval process may protect such models from misuse, it hampers availability to legitimate researchers. For these reasons, this project compares Bernoulli Naive Bayes, Linear SVC, and Logistic Regression.

2 Related Work

To detect false news, researchers use a variety of algorithms. According to (Wang, 2017), detecting fake news is a difficult task. (Zhou et al., 2019) discovered that false news is becoming more prevalent over time.

Machine learning classifiers were utilized by researchers in (Mahir et al., 2019) to detect false news. According to the researchers' tests, the SVM and Naive Bayes classifiers are the most effective in detecting false news. Based on accuracy, these two are superior to other classifiers. A more accurate classifier is regarded to be a better classifier. The accuracy supplied by any classifier is the most important factor. (Kudarvalli and Fiaidhi, 2020) claim that false news identification is essential since many people propagate fake news on social media to deceive the public. They suggest that machine learning is extremely beneficial in this area. They tried other machine learning algorithms and discovered that Logistic Regression is a classifier that performs better with more accurate results, though what this qualitatively means is unclear.

3 Data

3.1 Sourcing

The dataset we used, created by Ruchi Bhatia and shared via Kaggle, consists of 2095 cleaned and annotated news articles in English and German, collected from online sources. In it are 801 real news articles, and 1294 fake articles. Its annotations include whether the articles are real or fake, publication date, author, source, website URL, language, type (conspiracy theory, pseudoscience, etc.), and processed news article text with minimal stopword removal.

3.2 Preprocessing

In order to preprocess the data, it was loaded into a Pandas DataFrame, which uses underlying NumPy arrays to make managing the data more efficient than Python's default arrays. After randomizing the data, a column was added to the DataFrame that indicated whether a given article was real using Boolean values (True for a real news article, and False for a fake news article). The news article texts were tokenized with a regular expression tokenizer by NLTK, embedded into vectors using count vectorizer by scikit-learn, and run through a more comprehensive stopword removal algorithm than originally provided that dramatically improved the performance of our selected models. In the raw dataset, there is one row containing null values, which was dropped from the table before proceeding with the analysis.

4 Models

We compared the ability of several models to detect fake news. Among these were Bernoulli Naive Bayes, LinearSVC, and Logistic Regression, whose implementations were all provided by scikit-learn.

4.1 Bernoulli Naive Bayes

Bernoulli Naive Bayes (BernoulliNB) is a Naive Bayes classification algorithm that is designed to handle data that fits into a multivariate Bernoulli distribution. It is built off the following decision rule:

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i)$$

4.2 LinearSVC

Linear Support Vector Classifier (LinearSVC) is a classification algorithm that partitions data using linear support vectors obtained by supervised training.

4.3 Logistic Regression

Logistic Regression (LR) is a statistical model that fits a binary variable onto a logistic function, which may be used to determine (or, classify) the probability of a given binary state. Its use may be extended to a classification technique.

5 Procedure

5.1 Train-Test splitting

After preprocessing, the data was partitioned into train and test sets (the training set was allocated 3/4 of the dataset, and the test set was allocated the remaining 1/4 of the dataset). For the sake of replicability, this was done using scikit-learn, with $random\_state = 5$.

5.2 Training

Using scikit-learn implementations, each model was trained on the train set with the following parameters:

5.2.1 BernoulliNB

Alpha smoothing was set to 2.

5.2.2 LinearSVC

Default parameters were used.

5.2.3 Logistic Regression

C was set to 2. Maximum iterations was set to 1000. Number of jobs was set to -1 to avoid convergence failure due to training set size.

5.3 Evaluation

After being trained on the train set, each model was evaluated on the test set. This evaluation included a confusion matrix of true and false positives and negatives (Table 1). From this, accuracy, precision, recall, F1, and support were calculated, with both raw values (Table 2), and macro/weighted averages (Table 3) being returned.

	T Neg	T Pos	F Neg	F Pos
BernNB	0.557	0.899	0.266	0.088
LinSVC	0.509	0.205	0.150	0.135
LR	0.535	0.205	0.150	0.109

Table 1: Raw Error Data of Models

6 Observations

After running the given models through our evaluation metric described above, it seems that logistic regression ostensibly had the best performance, with an accuracy rate of 74%. However, there were some interesting features of other models that are worth discussing.

6.1 BernoulliNB

Although most other metrics were mediocre, the Bernoulli Naive Bayes classifier's ability to detect true negatives (i.e., labelling truly fake news articles as 'fake') was exceptional, with a raw True Negative rate of 0.557 (Table 1), and a False recall score of 0.86 (Table 2), which was the highest of the three models. Its raw False Negative rate was also exceptionally high, at 0.266, which may indicate that the model consistently overshoots and prioritizes catching fake news over catching authentic news, which would agree with the previously discussed metrics. Its rate of catching false positives was also exceptionally low, at 0.088. That said, its metric averages (Table 3) were consistently lower than the other two models' metric averages.

6.2 LinearSVC

The LinearSVC classifier's performance was in the middle of the BernoulliNB and Logistic Regression classifiers. It has no metrics that stands out, though if it had a lower computational load than either of the two, it would hold an advantage.

6.3 Logistic Regression

Again, the Logistic Regression classifier's performance was considerably higher than the other models, with its metric averages being consistently the highest. Like the others, it still overshot by having a significant number of false negative errors, but has a lower number of false positives than LinearSVC, though higher than BernoulliNB.

7 Analysis and Interpretation

Because each model seems to have certain metrics that stand out, as described above, it is difficult to say that one is objectively superior than the other.

	Categ.	Prec.	Recall	F1	Supp.
BernNB	True	0.51	0.25	0.34	182
	False	0.68	0.86	0.76	330
	Accuracy		0.65		512
LinSVC	True	0.60	0.58	0.59	182
	False	0.77	0.79	0.78	330
	Accuracy		0.71		512
LR	True	0.65	0.58	0.61	182
	False	0.78	0.83	0.80	330
	Accuracy		0.74		512

Table 2: Raw Evaluation Metrics of Models

	Avg	Prec.	Recall	F1	Supp.
BernNB	Macro	0.59	0.56	0.55	512
	Weighted	0.62	0.65	0.61	512
LinSVC	Macro	0.69	0.68	0.69	512
	Weighted	0.71	0.71	0.71	512
LR	Macro	0.72	0.70	0.71	512
	Weighted	0.73	0.74	0.74	512

Table 3: Macro/Weighted Averages of Model Evaluation Metrics

For this reason, we have focused the scope of this interpretation onto a domain-attentive discussion.

7.1 BernoulliNB

Because the Bernoulli Naive Bayes classifier appeared to focus its attention on detecting negatives rather than detecting positives, a domain-specific use may be appropriate. For example, if there were a certain population to whom fake news could be especially detrimental due to a consistent increased presence of dog-whistles, xenophobia, or other harmful rhetoric, such an overshooting classifier could be suitable.

7.2 LinearSVC

Because LinearSVC had no clear advantages, we are hesitant to say that it could work well in a specific domain. However, it seemed to have reasonable performance even with default parameters, which may be helpful if no specific feature is valued.

7.3 Logistic Regression

Because the raw error data of the logistic regression classifier seemed the most balanced out of the three chosen classifiers, it may be reasonable to say that it outperformed the other two. That said, similar to the LinearSVC classifier, it had no apparent, domain-specific advantages, so it is unclear how meaningful this is in this kind of interpretation. Its relatively high required training load may also be an impediment as far as computational load goes, which makes its specific comparison with LinearSVC especially unclear.

8 Error

Due to the slight over-representation of fake examples in the dataset (1294 fake versus 801 real news articles), this may have limited the efficacy of the classifiers. A more perfectly balanced corpus might improve performance. The classifiers also used default parameters provided by the scikit-learn package, unless something didn't work due to the size of the dataset, which may have contributed to errors in the experiment further. This may not necessarily be a source of error, but something that should have been considered was an a priori analysis of the stopword removal provided in the dataset verses the stopword removal that we used. While we observed that our stopword removal procedure yielded better results, the reason why is still unclear. This in itself seems like a worthwhile line of inquiry that should be explored further in future research.

9 Conclusion

This comparison between Bernoulli Naive Bayes, Linear SVC, and Logistic Regression classifiers demonstrates that using these models for fake news detection tasks will only reach a peak of approximately 70% accuracy with minimal, yet significant preprocessing. This lack of accuracy, while not insignificant, could be greatly improved. Each method had its own benefits. Naive Bayes was best for detecting the true fakes, although it misclassified legitimate news articles. Linear SVC had low computational load, allowing researchers with limited computational resources to use this classification method. Logistic regression had the best performance of the three, but this is overshadowed by the fact that it required the most resources to run, thereby making its implementation in a lightweight pipeline feel unjustified. Further research in this direction should prioritize investigating a different approach to classification to increase accuracy, such as transformer-based models along the lines of BERT (Devlin et al., 2018). Fake news detection to prevent the spread of misinformation is critical to the preservation of democracy.

References

Majed Alrubaian, Muhammad Al-Qurishi, Mabrook Al-rakhami, Sk Md Mizanur Rahman, and Atif Alamri. 2015. A multistage credibility analysis model for microblogs. pages 1434–1440.

Ruchi Bhatia. 2020. How do you recognize fake news? https://www.kaggle.com/ruchi798/how-do-you-recognize-fake-news/data.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aurélien Géron. 2019. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniquest to build intelligent systems (2nd ed.)*. "O'Reilly Media, Inc.".

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Harika Kudarvalli and Jinan Fiaidhi. 2020. Experiments on detecting fake news using machine learning algorithms. *International Journal of Reliable Information and Assurance*, 8:15–26.

Ehesas Mahir, Saima Akhter, and Mohammad Huq. 2019. Detecting fake news using machine learning and deep learning algorithms. pages 1–5.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *CoRR*, abs/1905.12616.

Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pages 836–837. Association for Computing Machinery, Inc. 12th ACM International Conference on Web Search and Data Mining, WSDM 2019 ; Conference date: 11-02-2019 Through 15-02-2019.