Comp562 Final Project — Comparison of Naïve Bayes, Logistics Regression, and Random Forest in Fake News Classification

Siyuan Li Yifan Zhou Yiyin Gu Kaiwen Lan

May 1 2022

1 Introduction

With the rapid development of internet nowadays, people can retrieve information fairly conveniently online. However, also due to the fast-spreading of information on the internet, more and more fake news are being intentionally made up and thrown into our vision. The misinformation can have dramatic negative impacts on the society. Therefore, how to classify whether a piece of news is fake given its content is a problem that is worth studying. Specifically, having learnt several classification algorithms in class, we were interested in studying the difference in their performance when applied to the same dataset from Kaggle. The following sections of this report discusses in detail of our approach.

2 Related Works

Before starting the project, we surveyed some former approaches to help us better understand the problem. We found that scholars have applied multiple machine learning algorithms to resolve fake news classification. [1] [2] shows that models such as Naïve Bayes and Random Forest tackles the problem with realatively high accuracy, and there are also a lot of approaches using Deep Learning, such as the one discussed in [3]. Among the variety of algorithm choices, we are most interested in comparing the performances between the non-neural-network algorithms. In specific, we decided to focus on Naïve Bayes, Logistic Regression, and Random Forest, which are all widely used algorithms in classification problems.

3 Methodologies

3.1 Dataset

The dataset used in this project is the Fake or Real News dataset taken from Kaggle(link available at https://www.kaggle.com/datasets/jillanisofttech/fake-or-real-news/code). The dataset consists of 6335 samples with 4 attributes: number, title, text, label.

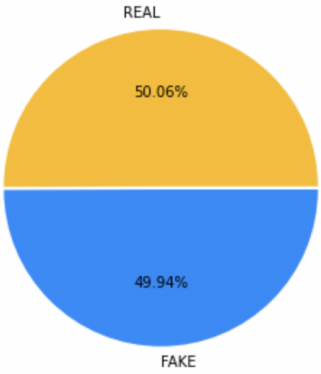
3.2 Data Pre-processing and Data Cleaning

The number attribute is not helpful for our prediction, so a dataset without this attribute is obtained after it is read in using pandas. Figure 1 shows what the dataset looks like at this stage. We also check that there are no NAN values in this dataset that need to be filled. Figure 2 demonstrates that the dataset is almost balanced.

Figure 1: Head of the Dataset

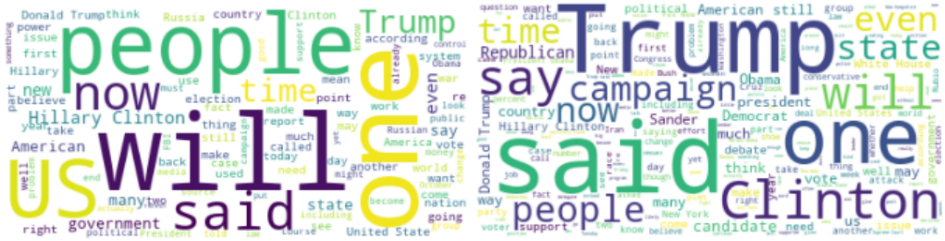
	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

Figure 2: Ratio between real and fake news



To give ourselves some sense of what words are most common in both types of news, we also created two word maps illustrated in Figure 3.

Figure 3: Most Common Words in Fake News(left) and Real News(right)



Before feeding our text data into machine learning models, we implemented a data_cleaning function that does the following:

- Convert the text to lowercase
- Remove the punctuation and special characters from the text
- tokenize and lemmatize each word(i.e. transform words into their base form, "feet" to "foot", "flew" to "fly", etc.)

The first few rows of the dataset after applying the function is shown in Figure 4.

Figure 4: Dataset after cleaning

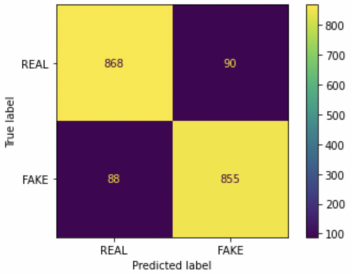
	title	text	label	cleaned_text
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	daniel greenfield shillman journalism fellow f...
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	google pinterest digg linkedin reddit stumbleu...
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	u secretary state john f kerry say monday stop...
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	kaydee king kaydeeking november lesson tonight...
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	primary day new york front runner hillary clin...

After data cleaning, we forward to model training.

3.3 Naïve Bayes

According to Bayes' Theorem, probability of Y given X is given by $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$. Naïvely assuming that all the inputs are independent, we can derive a classifier $P(Y|X) = \frac{P(Y)\prod_i P(X_i|Y)}{P(X)}$. We use Multinomial Naïve Bayes which is widely applied in NLP. We use Tfidf Vectorizer to vectorize our text into word count vectors to fit with the model. When vectorizing, we filter out words that appear in more than 90% as they are not informative and those less than 0.3% since they might lead to overfitting. We use a 70%:%30 training-validation ratio to train and test the model, and obtained the following confusion matrix(Figure 5). The model achieves an accuracy of 88.32%, with 11.51% false positive rate and 11.87% false negative rate.

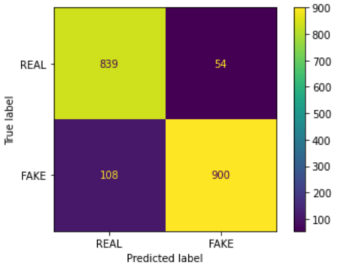
Figure 5: Multinomial Naïve Bayes Confusion Matrix



3.4 Logistic Regression

Logistic Regression is also a commonly used classification algorithm. Logistic Regression uses the sigmoid function: $sigmoid(z) = \frac{1}{1+e^z}$ to convert response variables into 0 1 scale. We use the same vectorization and test-validation split strategy and feed the input to the Logistic Regression model. The confusion matrix is demonstrated in Figure 6. The model achieved an accuracy of 91.48%, with 10.71% false positive rate and 6.05% false negative rate.

Figure 6: Logistic Regression Confusion Matrix

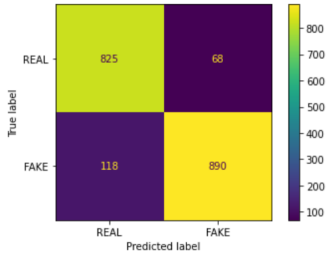


3.5 Random Forest

Random Forest uses tree-based algorithms to classify samples. It applies multiple decision trees and classifies a sample based on the class selected by most trees. By using the pre-implemented Random Forest algorithm from scikit learn, and using the same vectorization and test-validation split strategy discussed above, we obtained the following

confusion matrix(Figure 7). The overall accuracy of the model is 90.22%, with 11.71% false positive rate and 7.61% false negative rate.

Figure 7: Random Forest Confusion Matrix



4 Discussion & Conclusion

To compare the performance of the models, we built the following table which includes the accuracy, true positive count(TP), true negative count(TN), false positive count(FP), and false negative count(FN).

Model	Accuracy	TP	TN	FP	FN
Naïve Bayes	88.32%	787	892	116	106
Logistic Regression	91.48%	839	900	108	54
Random Forest	90.22%	825	890	118	68

By comparing the data in the table above, we can see that when applied to a same set of 1901 test samples, Logistic Regression outperformed other two algorithms with a 91.48% accuracy, and Naïve Bayes had the lowest accuracy of 88.32%. However, in terms of False Negative rate, the three models have fairly close performance. Namely, the other two models are marginally better than Naïve Bayes in correctly labelling real news. Since Naïve Bayes is under the assumption that all input features are independent, our input feature(word occurrence counts) are not guaranteed to be independent, and that is a possible reason for Naïve Bayes to have lower performance in this context. Also, when vectorizing the inputs we filtered out words that occurred in more than 90% of the texts assuming that they may not be informative, but if there were too many such words filtered out, the ratio between features and number of samples may decrease, and hence influence the performance of our models.

Overall, all three models we applied provided relatively accurate results in classifying fake news, and there are slight differences in their performances. We also found that both cleaning textual data and training them using machine learning models took a good amount of time and memory even on a dataset of size less than 10,000. As Internet continues to progress, news, as well as fake ones, will probably be created and spread even faster than today. With that in mind, developing faster and more efficient algorithms is a critical job for the development of text classification.

References

[1] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017, pp. 900-903, doi: 10.1109/UKRCON.2017.8100379.

[2] Saqib Hakak, Mamoun Alazab, Suleman Khan, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Wazir Zada Khan. "An ensemble machine learning approach through effective feature extraction to classify fake news," Future Generation Computer Systems, Volume 117, 2021, Pages 47-58, ISSN 0167-739X.

[3] Monika Choudhary, Satyendra Singh Chouhan, Emmanuel S. Pilli, Santosh Kumar Vipparthi, BerConvoNet: A deep learning framework for fake news classification, Applied Soft Computing, Volume 110, 2021, 107614, ISSN 1568-4946.