Detecting Fake News with Machine Learning

Bryson Hogsed 730520041

Kensho Pilkey 730448185

Gregory Glasby 730567386

William Zahrt 730490857

Abstract

The rise of the internet has made the creation and dissemination of fake news easier than ever, contributing to widespread misinformation. This project seeks to identify the most effective machine learning techniques for detecting fake news by training and testing various models on labeled datasets. By comparing these models, we aim to determine the most accurate and reliable approach for identifying false information. The resulting model will empower users to input articles for verification, helping to curb the spread of misinformation and enable informed decision-making.

1 Introduction

The rapid growth of online media platforms has made people more connected than evert. While this expanded reach has many benefits, it also poses significant challenges, particularly in the form of fake news. Defined as misinformation or intentionally misleading content presented as factual reporting, fake news has the potential to influence public opinion, shape political discourse, and erode trust in legitimate sources of information[4].

Addressing this problem requires robust tools capable of accurately differentiating between legitimate and falsified content. Recent advancements in machine learning (ML) have shown promise in automating the detection of fake news, offering scalable solutions that can analyze large volumes of text efficiently. However, determining which technique or combination of techniques is most effective remains an open question.

In this paper, we explore a range of commonly used ML methods for fake news detection, including logistic regression [3], Naive Bayes, decision trees [2], random forests [1], among others. By evaluating these models on established benchmarks and varying types of misinformation, we aim to identify the strongest approaches for accurately classifying news articles. Our results seek to provide guidance for researchers and practitioners, contributing to the ongoing development of tools that help maintain the integrity of the information ecosystem.

Cleaning The Data

The data we chose for this project we obtained from Kaggle and it consisted of two data sets: one with real articles and another with fake articles. The articles contained columns for the title of the article, the text making it up, the subject of the articles and the date it was published. For the purposes of this project, we dropped the columns pertaining to the date, subject, and title. We added a new column "class" which provided information on whether the article was a fake one or a real one. A "0" was given to the fake articles and a "1" was given to the real articles. Afterwards, we had to clean up the text and standardize it for the different models we were about to run the subjects through. To do this, we removed any links in the text as well as html tags, spaces, and any other unwanted criteria such as emails.

After modifying the two tables and combining them into one, we then split up our features and our targets. For this project, X encompassed the preprocessed text and our target variable y was the class

I Can't Believe It's Not Better Workshop III at NeurIPS 2022.

variable that told us whether an article was fake or real. The data was split randomly into a training set and a testing set with 80 percent of the data in the former and 20 percent of the data in the latter. The X train and y train sets were used to fit a model while the X test and y test sets were used to evaluate the performance of the model.

Methods

When determining whether news is fake, it is important to find the most accurate method. We use binary classification for Logistic Regression, Naive Bayes, Decision Trees, and Random Forests to train and evaluate models. We also note that more advanced methods, such as deep learning-based approaches, have been applied successfully in text classification [5] and may be considered in future work.

1.1 Logistic regression

Logistic Regression is a widely used statistical method for binary classification tasks [3], making it well-suited for distinguishing between fake and real news articles. Unlike linear regression, which predicts continuous outcomes, Logistic Regression estimates the probability that a given input belongs to a particular class—in this case, fake (0) or real (1) news.

Mathematically, Logistic Regression models the relationship between the dependent binary variable and one or more independent variables by applying the logistic (sigmoid) function. This function maps any real-valued number into the range between 0 and 1, effectively representing probabilities. The decision boundary is determined by a threshold (commonly 0.5), where inputs with predicted probabilities above the threshold are classified as one class, and those below as the other.

Logistic Regression was chosen for its simplicity, interpretability, and efficiency, making it a strong baseline model for comparison against more complex algorithms like Decision Trees and Random Forests. Its ability to provide probabilistic outputs also offers valuable insights into the confidence of predictions, which is beneficial for applications requiring transparency and accountability.

1.2 Naive Bayes

To train a model for Naive Bayes, you first take all the words that occur in a real news section. You count the frequency of a given word in the real news section and divide it by the total number of words. This gives the probability for this word given that it is real news. You do this for all words in the news dataset. You do the same process for the fake news dataset. This gives us a list of probabilities for each word in the real news data set and the fake news data set. We will calculate a P(R) and a P(F) with these being for real and fake respectively. We will take the number of real news pieces and divide this by the total number of news pieces to get P(R). To get P(F), we do P(F) = 1 - P(R). Next, we multiply the P(R) by all the given probabilities we calculated before. You will multiply these given probabilities as many times as they occur in the given message. You do the same process for the fake news. If the number for the real news is higher, then we would classify it as real news. If the number for the fake news is higher, then we would classify it as fake news.

1.3 Decision trees

Another method we employed was decision tree classification. In essence, decision tree classification works upon the principle of splitting data into smaller subsets based on a specific feature [2]. This involves first having a root node and then looking at features like article length or specific words and then determining where to make a split at that root. These features are chosen according to some criteria such as entropy or Gini impurity. Exapnding upon that, the feature is chosen based off of whether it provides the highest information gain or it produces the least impurity. This is a recursive method that ends once all leaf nodes belong to a class which in our case is whether or not the news is fake “0” or real “1”. Using this method we were able to predict what articles were fake news and which were real news with an impressive accuracy of 99.59.

1.4 Random Forests

Random Forests is an ensemble learning technique that combines the predictions of multiple decision trees to improve accuracy and reduce overfitting [1]. By averaging the predictions of several decision trees, Random Forests better generalize unseen data. Each decision tree in the forest is trained on a different bootstrap sample of the training data, and at each split in the tree, a random subset of features is considered for selection. The randomness makes sure that each individual tree in the forest is diverse, reducing the likelihood of overfitting to the training data.

Random Forests also provides feature importance metrics, which can help identify which features are most significant for distinguishing between fake and real news articles. This interpretability is an added advantage for understanding the model's decision-making process.

For this project, the Random Forests classifier was trained using the preprocessed dataset. The ensemble approach led to the model achieving a 99.7% accuracy.

Results

1.1 Logistic Regression Model Performance and Accuracy

The Logistic Regression model was trained to classify news articles as fake or real. Below are the performance metrics for the model:

Accuracy: 98.70%

Classification Report:

	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	4733
1	0.98	0.99	0.99	4247
Accuracy			0.99	8980
Macro Avg	0.99	0.99	0.99	8980
Weighted Avg	0.99	0.99	0.99	8980

Table 1: Logistic Regression Classification Report

The Logistic Regression model achieved an impressive accuracy of 98.70%, indicating a high level of correctness in its predictions. Both precision and recall metrics for the fake (0) and real (1) classes are above 0.98, demonstrating the model's ability to accurately identify and differentiate between fake and real news articles. The F1-scores, which balance precision and recall, are both 0.99 for both classes, further confirming the model's robustness and reliability in this binary classification task.

1.2 Naive Bayes Model Performance and Accuracy

Accuracy : 93.03%

Classification Report:

	precision	recall	f1-score	support
0	.93	.94	.93	4733
1	.93	.92	.93	4247
accuracy			.93	8980
macro avg	.93	.93	.93	8980
weighted avg	.93	.93	.93	8980

The Naive Bayes obviously performs poorly relative to the other techniques when distinguishing between real and fake news with an accuracy of 93.03%. This is likely because this technique does not take context into account as it looks at words individually without looking at the context of the word in the sentence.

1.3 Decision Trees Model Performance and Accuracy

Accuracy : 99.59

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4733
1	.99	1.00	1.00	4247
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980

The Decision Tree model achieved an impressive accuracy of 99.59%, making it the second best model to decipher between real and fake news. Both precision and recall metrics for the fake (0) class are at 1.00 while the F1 score is at 1.00. The recall and F1 score for the real (1) class are both at 1.00. The only part this model lacks in is its precision for the real (1) class as it is at a score of 0.99. This means that there was a minuscule amount of case(s) where the model made a positive prediction for a real (1) article but it actually ended up being a fake article. With that being said, 0.99 is still an impressive score for the precision of the model predicting real (1) news articles.

1.4 Random Forests Model Performance and Accuracy

Accuracy : 99.7

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4733
1	1.00	1.00	1.00	4247
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980

The Random Forests model achieved an impressive accuracy of 99.70%, outperforming Logistic Regression and Naive Bayes and closely matching the performance of Decision Trees. The precision, recall and F1 scores were perfect for the fake (0) and real (1) classes, demonstrating the model's ability to classify news articles correctly. The ensemble approach effectively mitigates overfitting while maintaining high accuracy, making Random Forests one of the most reliable techniques for fake news detection.

Conclusion and Future Work

In this paper, we investigated several machine learning algorithms—Logistic Regression, Naive Bayes, Decision Trees, and Random Forests—for the task of detecting fake news. Our results showed that while Logistic Regression and Naive Bayes performed well, Random Forests and the Decision Tree approach achieved near-perfect accuracy. These findings suggest that tree-based methods, particularly ensemble techniques, are highly effective at distinguishing between real and fake articles.

While our dataset was sufficient for an initial comparison, future work could focus on expanding the variety of news sources, incorporating multimodal data (such as images or metadata), or employing more sophisticated language representations. Methods like deep neural networks [5] or transformer-based models (e.g., BERT, GPT) may capture contextual nuances better than traditional approaches, potentially improving performance and robustness.

Finally, deploying these models in real-world systems requires considering their performance over time, domain adaptation to new topics, and resilience against adversarial attempts to evade detection. Addressing these challenges is essential for developing truly reliable, scalable solutions that help mitigate the spread of misinformation and maintain the integrity of our information ecosystem.

References

[1] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324. URL: http://link.springer.com/10.1023/A:1010933404324 (visited on 12/13/2024).

[2] Leo Breiman et al. *Classification And Regression Trees*. 1st ed. Routledge, Oct. 19, 2017. ISBN: 9781315139470. DOI: 10.1201/9781315139470. URL: https://www.taylorfrancis.com/books/9781351460491 (visited on 12/13/2024).

[3] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression*. 2nd ed. New York: Wiley, 2000. 1 p. ISBN: 9780471722144 9780471654025 9780471673804 9780471356325.

[4] Kai Shu et al. *Fake News Detection on Social Media: A Data Mining Perspective*. Sept. 3, 2017. DOI: 10.48550/arXiv.1708.01967. arXiv: 1708.01967. URL: http://arxiv.org/abs/1708.01967 (visited on 12/13/2024).

[5] Xiang Zhang and Yann LeCun. *Text Understanding from Scratch*. Apr. 4, 2016. DOI: 10.48550/arXiv.1502.01710. arXiv: 1502.01710. URL: http://arxiv.org/abs/1502.01710 (visited on 12/13/2024).