Fake News Detection: A Machine Learning Approach

Justin Tien

04.28.2023



# Introduction

*Problem Statement*

Fake news is a growing problem in today's society. It can spread quickly through social media and can cause damage to individuals, organizations, and even countries. As the amount of digital content increases, it becomes more challenging to distinguish between real and fake news.

Therefore, it is necessary to develop an automated method for detecting fake news. The problem of fake news detection has gained significant attention from researchers, and several approaches have been proposed to address this issue.

*Motivation*

The impact of fake news has been widely known. It can mislead people and affect their decision-making process. With the rise of social media platforms, fake news can spread quickly and reach a large audience. Therefore, it is crucial to develop a machine learning approach for detecting fake news. Machine learning techniques have shown promising results in the field of fake news detection. In this study, we aim to build a model that can accurately classify news articles as either fake or real, which can be used to mitigate the spread of fake news.

*Objective*

The objective of this project is to compare the performance of several machine learning models in detecting fake news. We will use a dataset of news articles and preprocess the data by cleaning and transforming the text data. We will then explore several machine learning models, including logistic regression, decision trees, random forest, and support vector machines, to determine which model performs best in classifying fake news. We will evaluate the performance of each model using various metrics, including accuracy and precision. Ultimately, our goal is to build a comprehensive machine learning model that can accurately detect fake news, which can be used as a tool to stop the spread of misinformation.

# Data and Methods

*Data Source*

The dataset used in this project was obtained from Kaggle, a popular platform for data science and machine learning enthusiasts. The dataset consists of two separate CSV files - one containing a list of articles considered as "real" news, and the other containing a list of articles considered as "fake" news. The data is retrieved from sources during the period of 2016-2017, where the political climate was at its peak in the United States. Each dataset contains approximately 21,000 news articles and includes a number of features, such as the title of the article, the text of the article, the subject of the article and the date the article was posted.

*Data Preprocessing*

Before conducting any analysis or training any machine learning models on the dataset, a

number of preprocessing steps were applied to clean and transform the data. These preprocessing steps were carried out in a Jupyter notebook, which is an interactive python environment. The preprocessing steps included several data cleaning steps, such as removing unnecessary features, removing duplicates, and handling missing data. In addition, the text data was cleaned by removing punctuation, stopwords, and converting all text to lowercase. The preprocessing steps are designed to reduce the noise in the data. By cleaning the data and removing any unnecessary noise, the models were able to focus on the most important features of the articles. Also, concatenating the data frames was an important data integration preprocessing step, which was helped through the use of Chat-GPT. Through this step, both the true and fake data frames could be viewed in one single data frame.

*Methodology*

In this study, we used a methodology for detecting fake news using machine learning models. First, we created word clouds for both the real and fake news datasets to visualize the most frequently used words in each dataset. We used the word counter from the nltk library to

calculate the most frequently used words in the real and fake data, and then plotted the results. After the creation of word clouds and frequency analysis, the next step was to vectorize the text data using TF-IDF. This was applied to all of the machine learning models used in this project, including naive Bayes, logistic regression, decision tree, random forest, and support vector

machines (SVMs). The TF-IDF vectorization technique was used to convert the text data into a numerical format suitable for machine learning algorithms. It was used to identify the most important words or features for distinguishing between real and fake news articles.

Next, we split the data into training and testing sets using the test\_train\_split function from the sklearn library. We then used five different machine learning models (Naive Bayes, logistic regression, decision tree, random forest, support vector machine) to train the models on the training data and make predictions on the testing data. Finally, we evaluated the accuracy of each machine learning model using confusion matrices from the sklearn library. These matrices were used to determine the number of true positives, true negatives, false positives, and false negatives for each model. We calculated the accuracy of each model based on its prediction of real or fake news articles in the testing data.

The ROC (Receiver Operating Characteristic) curve was then used to evaluate the performance of the machine learning models. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR). To generate the ROC curve, the predict\_proba() method of the fitted model was used to obtain the predicted probabilities for each class label. Then, the roc\_curve() function from the sklearn library was used to calculate the false positive rate, true positive rate,

and threshold values for the classifier. Finally, the roc\_auc\_score() function was used to calculate the area under the ROC curve (AUC). The AUC score is a measure of the performance of the

classifier, where a score of 1 represents a perfect classifier and a score of 0.5 represents a random classifier. The ROC curve was plotted using the matplotlib library to visualize the performance

of the models. This methodology allowed us to compare the accuracy of different machine learning models in detecting fake news.

We performed feature extraction using a random forest algorithm. Feature extraction helps to identify the most important features in the dataset that are most predictive of the target variable. In this case, we used a random forest algorithm to identify the most important features in our dataset. The random forest algorithm works by creating multiple decision trees and aggregating their predictions to make the final prediction. In the feature extraction process, we trained a random forest model on our dataset and used it to rank the importance of each feature. The

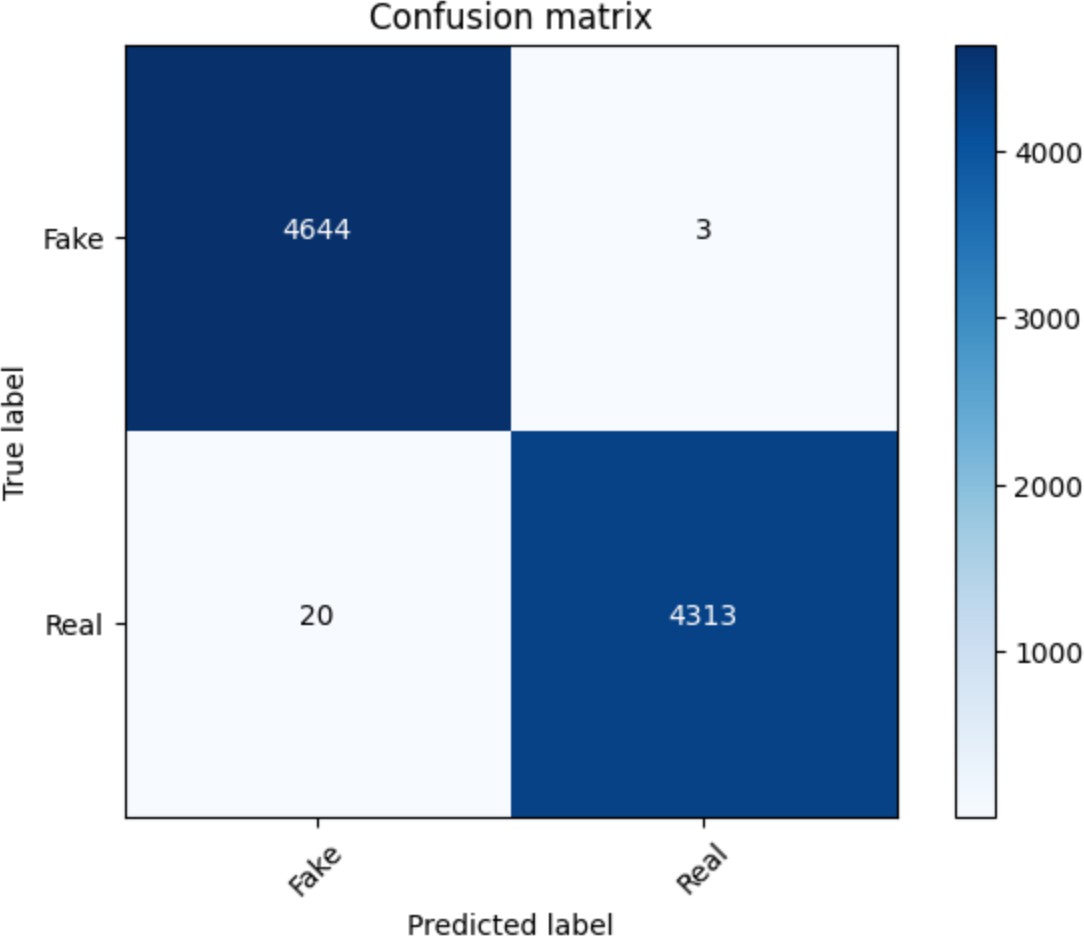
feature importance scores were then used to select the top features that were most predictive of our target variable. To perform feature extraction using random forest, we first split our dataset into training and testing sets. We then trained a random forest model on the training set and used it to rank the importance of each feature. We selected the top features based on their importance

scores and used them to train a new model. This new model was then evaluated on the testing set to determine its prediction performance.

# Results

Based on the machine learning models implemented in this study, it was found that all models were able to successfully detect fake news with high accuracy. Among the models, the decision tree performed the best with an accuracy of 99.7%. SVM also showed high accuracy with 99.53%, followed by logistic regression with 98.74% and random forest with 98.83%. Naive Bayes also showed good accuracy with 94.98%.

The results of the confusion matrices showed that all models had a high true positive rate for detecting fake news, which indicates their effectiveness in correctly classifying fake news. The false positive rate was also relatively low for all models, indicating that the models were able to avoid misclassifying real news as fake news.



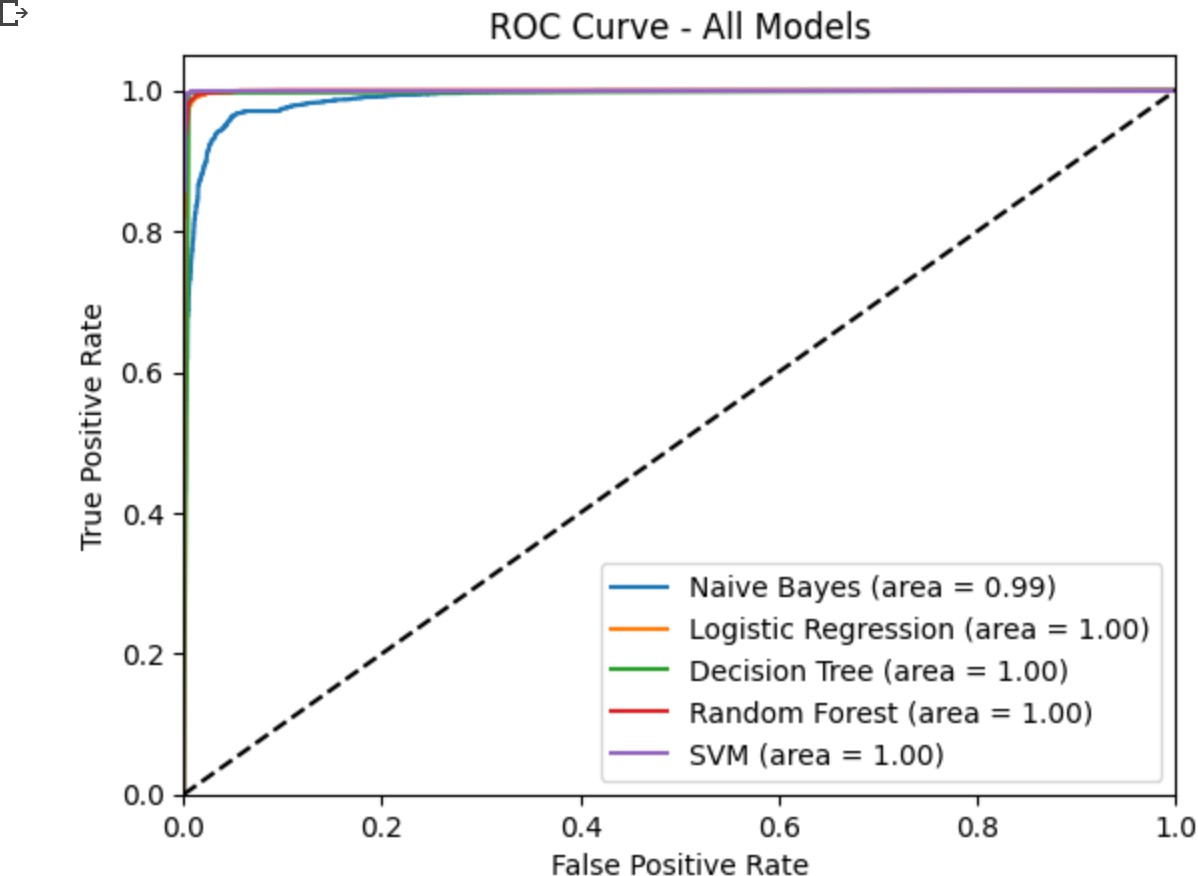
*Figure 1: Confusion matrix created from Decision Tree model without normalization*

The ROC curves for the different machine learning models were generated to evaluate the performance of the models. The area under the curve (AUC) was used to measure the

performance of each model, with a higher AUC indicating better performance. The AUC values for the different models were as follows: Naive Bayes (0.983), Logistic Regression (0.994),

Decision Tree (0.999), Random Forest (0.998), and SVM (0.999). The ROC curves for the

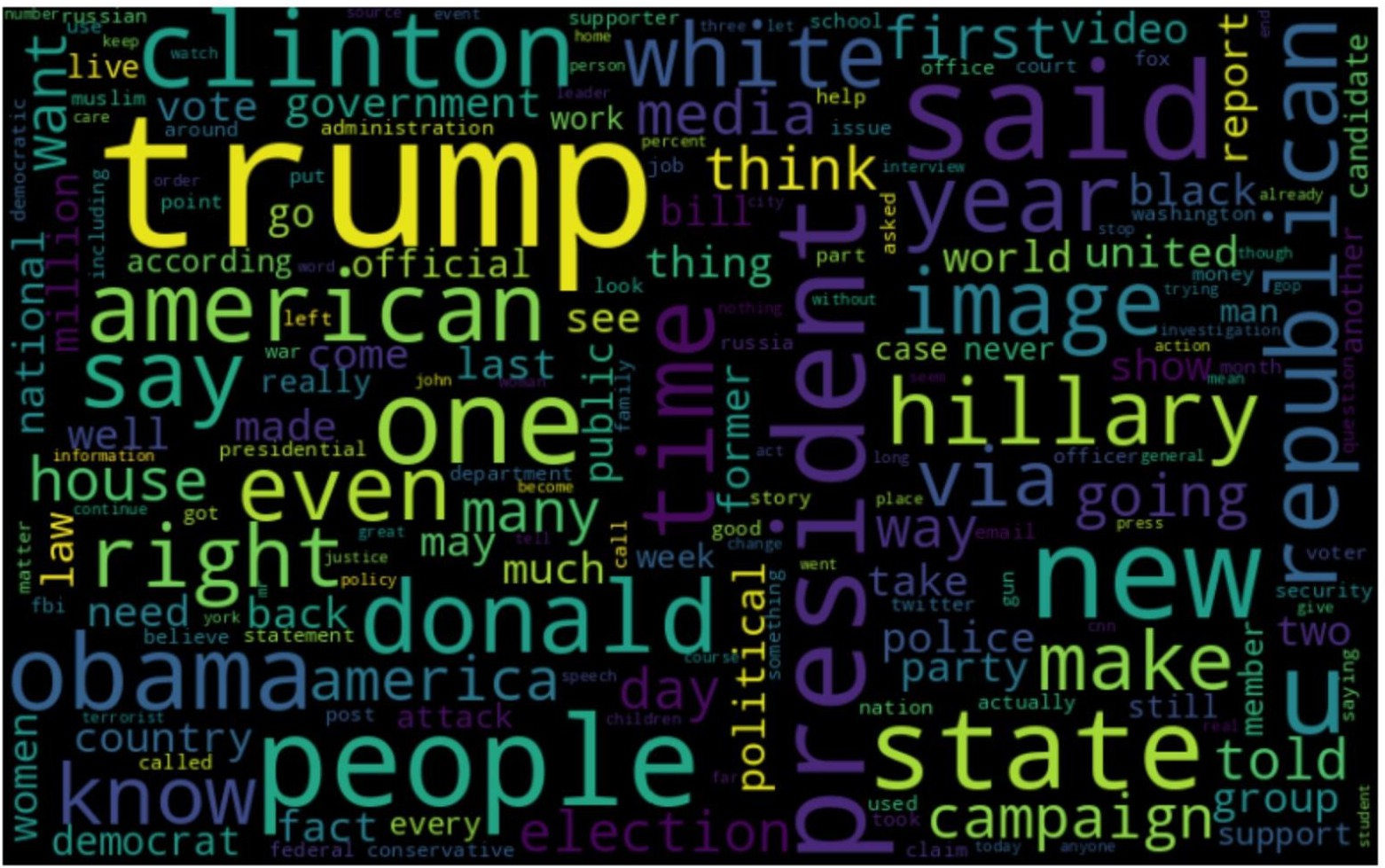
different models were plotted, and it was observed that all the models had high performance, with the curves showing a steep rise and a near-vertical line towards the top left corner of the graph. The logistic regression and decision tree models had the highest AUC values, indicating that they were the most accurate models for detecting fake news in the given dataset. These ROC curves were then plotted for each model onto a single plot, which makes it easier to compare the areas. This code was mostly generated with the help of Chat-GPT.



*Figure 2: Fitted ROC Curve plotted for the all models*

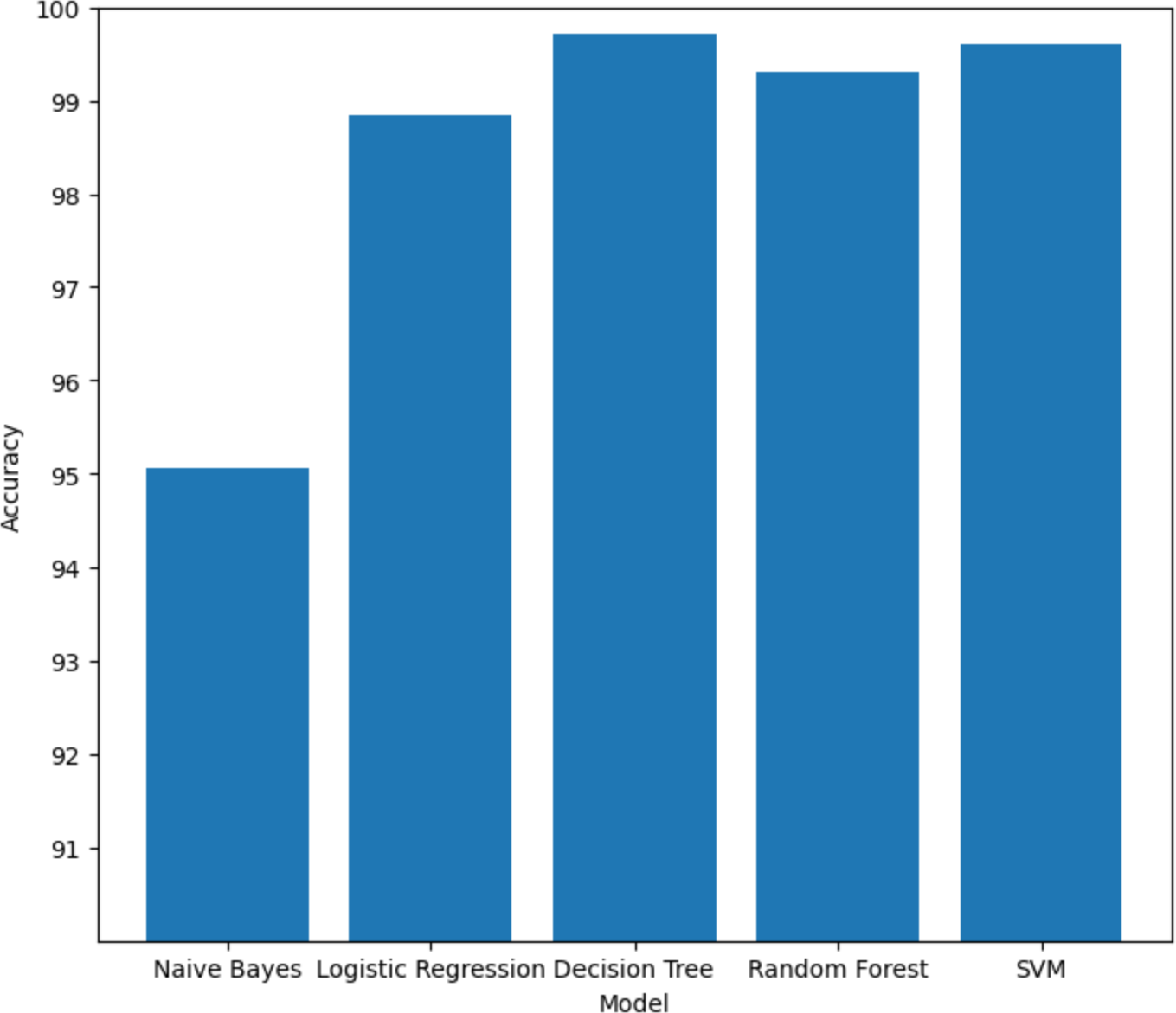
The word clouds generated for the real and fake news datasets showed clear differences in the most commonly used words. The real news dataset had words such as "trump", "president", "government", and "house" as some of the most frequently used words. In contrast, the fake news dataset had words such as "trump", "people", "president", "state", and "obama" as some of the most frequently used words. This difference in the most commonly used words suggests that the models were able to distinguish between real and fake news based on the language used in

the articles.



*Figure 3: Word Cloud created for fake news dataset*

Overall, the results of this study suggest that machine learning algorithms can be effective in detecting fake news. The decision tree model was found to be the most accurate, followed closely by SVM, logistic regression, and random forest. This study provides a starting point for further research in the field of fake news detection, and highlights the potential of machine learning for addressing this important issue.



*Figure 4: Comparative graph amongst models using accuracy score*

*Chat-GPT*

Chat-GPT was utilized in this study to develop code on pre-processing the data by concatenating the data frames and shuffling the data using the nltk library. Concatenating the dataframes ensures that all the relevant data is included and can be analyzed together and is an important step in data integration. These preprocessing tasks can help to ensure that the data is as clean and meaningful as possible, which can improve the accuracy of the machine learning models used in the study. Chat-GPT was also used in the Jupyter notebook to develop the final comparative graph between the accuracy of the models used, through the matplotlib library. The comparative model histogram shows the distribution of accuracy scores for each algorithm, allowing for a quick and easy comparison of their overall performance. Another crucial part of the visualization of this study generated by Chat-GPT, was the ROC Curve. The ROC curve provides a more detailed analysis of the model’s ability to distinguish between true and fake news, taking into account the true positive rate and false positive rate. The ROC curve was plotted for each model in a single plot, allowing for easy visual comparison between the area under the curve. Chat-GPT was lastly used for feature extraction using the Random Forest algorithm. A random forest model was trained on our dataset and used it to rank the importance of each feature by creating multiple decision trees then aggregating their predictions to make the final prediction. By selecting only

the top features, we were able to reduce the dimensionality of our dataset.

# Conclusion

After researching and demonstrating machine learning models in the study of fake news detection, the results were encouraging. The use of word clouds, TF-IDF vectorization, and a variety of machine learning models including Naive Bayes, Logistic Regression, Decision Tree,

Random Forest, and SVM have provided high accuracy in detecting fake news. These results indicate that these methods can be used to identify fake news articles with a high degree of confidence. However, there may be limitations in the dataset used for this study, and the

effectiveness of these methods may depend on the characteristics of the dataset.. There may be potential biases in the selection of features or the training and testing of the models that need to be addressed in future research. The results of this study provide a baseline for the development of more effective methods for detecting fake news in the future. With the increasing use of fake

news in today's society, the ability to accurately detect and prevent its spread is very important in maintaining a well-informed public and making sure our social system is balanced and honest.

# Future Work

In this study, we used a relatively simple approach to text preprocessing and vectorization.

However, there are many more advanced techniques that could potentially improve the accuracy of fake news detection, such as deep learning models and sentiment analysis. Exploring these techniques could be an interesting avenue for future research.

In this study, we also treated all fake news as a single category. However, there are many different types of fake news, ranging from outright lies to misleading statistics to biased

reporting. It would be interesting to explore whether different types of fake news are more or less difficult to detect using machine learning algorithms, and whether certain models are better suited for certain types of fake news. This also includes using more updated data sets as well as using data from more sources and social media API. We can use a multitude of different sources in the future, which will give us a larger sample size as well as a larger population representation.

# Works Cited

OpenAI. (n.d.). Chat-GPT. Retrieved April 21, 2023, from https://openai.com/chat-gpt/

Bisaillon, C. (2020, March 26). *Fake and real news dataset*. Kaggle. Retrieved April 26, 2023, from [https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset](http://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset)