White Paper
By Achraf Safsafi
Bellevue University, DSC 680, Prof. Catie Williams

Fake News Detection Using Machine Learning Methods

Business Problem:

Fake news is a real and dangerous issue at all times. However, with the presence and abundance of various social media platforms, this phenomenon has become more widespread and worse than ever before, as it can have grave financial and social consequences, whether for both individuals or entities. In this project, I will address this issue from the technical side by applying different machine learning algorithms to choose the best to build a model that can classify news as real or fake. So the research question I will answer is as following " Which best machine learning classifier model can be applied for detecting fake news?".

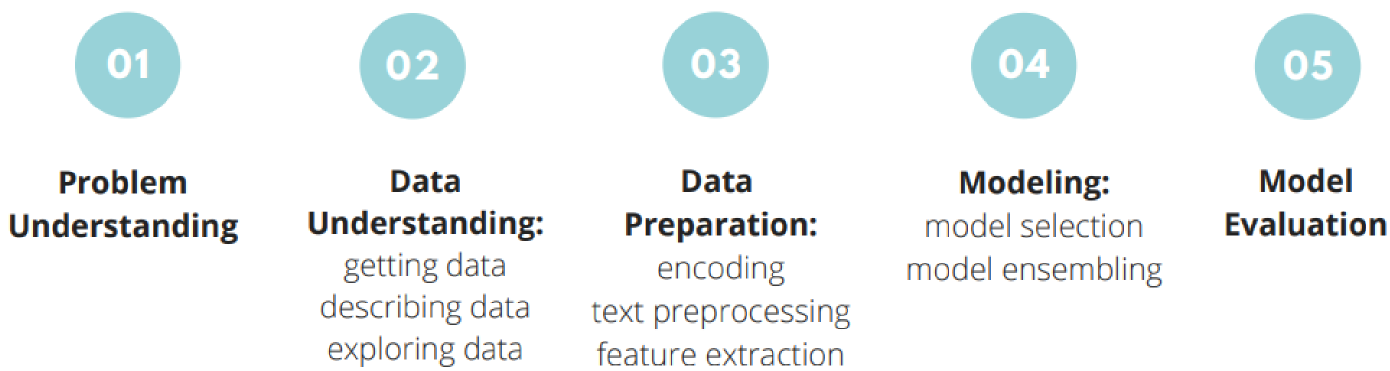
Background/History:

The spread of fake news is an old and new topic that negatively affects individuals and societies alike. It existed from the Pre–Printing Press Era, when Procopius of Caesarea (500–ca. 554 AD), the principal historian of Byzantium, used fake news to discredit Emperor Justinian and his wife. And the phenomenon still continues until our present time, the Internet Era, during the 2016 US election campaign when many different websites posted fake and biased stories. But with the spread and ease of access to social media, the phenomenon has worsened more and more. The worry is that misleading articles will become the basis on which society and individuals depend as the primary source of news. In contrast, the real information will become lost between lies and rumors. For this reason, it has become necessary to search for mechanisms and methods to distinguish fake news from real ones. Here, the role of machine learning comes through its Natural Language Processing (NLP) and classification techniques to create a detector that could solve this issue.

Assumptions:

- Each news article is a mix of topics.
- There is slight multicollinearity among text features.

Implementation Plan:



Data Explanation:

The dataset for the project is found on Kaggle. It is a CSV file that occupies up to about 30 MB of space. The dataset has four columns and 6335 rows representing the number of articles. The columns include:

- id: the unique id for a news article
- title: the title of a news article
- text: the text of the article
- label: label denotes whether the news is Real or Fake

The dataset has no missing values and is perfectly balanced. The number of news records contains an equal number of samples from the real and fake classes, as shown in Figure 1.

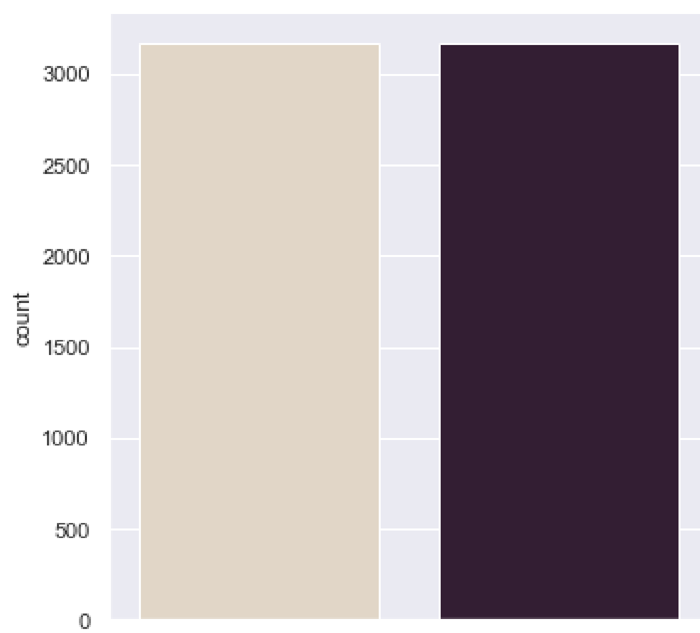


Figure 1:Class distribution

The label values were encoded as '1' in case of fake news and '0' in case of real news. Then, the id and title columns were removed as there is no need for them in the analysis. Several NLP preprocessing methods were applied to the dataset, including removing numeric characters, special characters, and stopwords. Bigram extraction, trigram extraction, and lemmatizing also were made. As a result, after text cleaning, the word number of the text corpus decreases by 46%, from 29826765 words to 13635565 words. Figure 2 & Figure 3 represent the 100 most frequent words before and after preprocessing, respectively.

Figure 3: Word cloud after text preprocessing

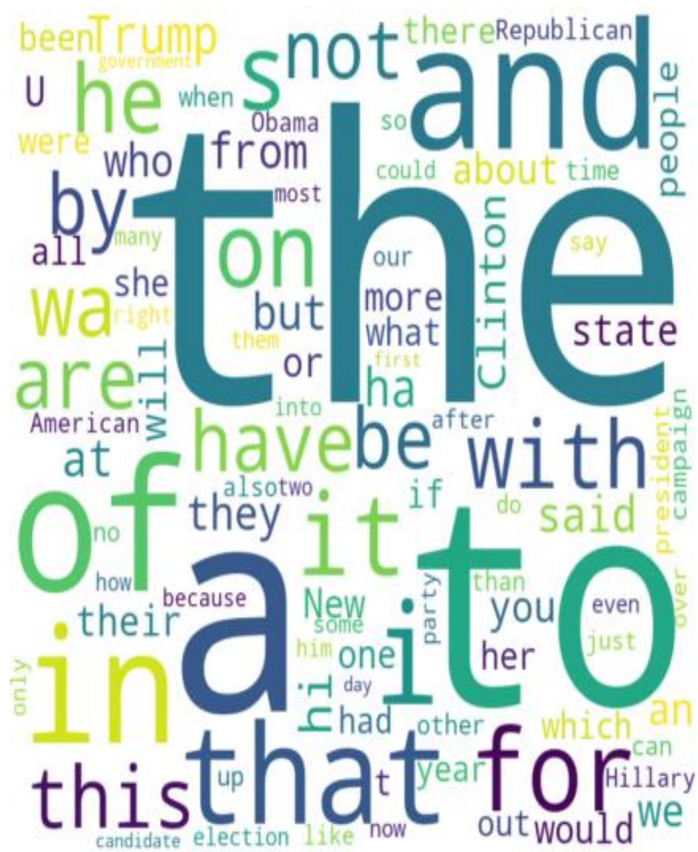


Figure 2: Word cloud before text preprocessing

The next step was feature extraction. The latent Dirichlet allocation (LDA), a topic model algorithm, was applied to perform this task. LDA is an unsupervised classification method that tries to divide textual data into abstract topics.

Evaluating the topic model was needed to choose the optimal number of topics for LDA. The evaluation was done using coherence scores, and the best LDA model was a model with 32 topics, as shown in Figure 4. Unneeded variable ('text') was dropped, and the 32 extracted topics were used as features. Finally, the dataset was ready for modeling.

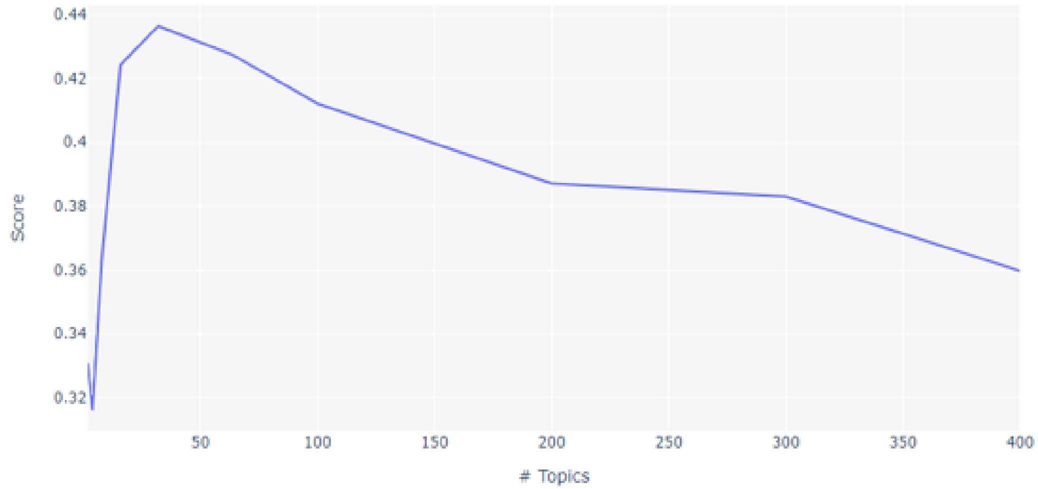


Figure 4:Determining optimal number of topics using coherence scores

Methods:

A large part of the project was performed by PyCaret, an auto-machine learning library in Python that allows the user to perform many different machine learning tasks with fewer lines of code. The library lets execute many tasks that take much time if traditional Python libraries perform them.

After dataset preparation, 18 traditional Machine learning classification algorithms were trained using 10-fold cross-validation. The classifiers include SVM - Linear Kernel, SVM - Radial Kernel, Gaussian Process Classifier, MLP Classifier, Ridge Classifier, Random Forest Classifier, Quadratic Discriminant Analysis, Ada Boost Classifier, Gradient Boosting Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine, and CatBoost Classifier. Next, the performance of the classifiers was compared based on their Accuracy. There are different classification metrics, but as the dataset is perfectly balanced, the Accuracy score was selected as the best metric to evaluate the models.

Next, ensemble modeling approaches include blending and stacking, were applied using the three best models. In the end, all previous models, including individuals and ensemble models, their performance were evaluated to determine the final classifier that offers the best Accuracy.

Analysis:

Table 1 summarizes the average Accuracy reached by each individual algorithms. The Light Gradient Boosting Machine algorithm got a maximum Accuracy score of 84.91%, followed by the Extreme Gradient Boosting algorithm with 84.46%, then the Gradient Boosting Classifier algorithm with 84.37%. The Random Forest Classifier algorithm achieved an average accuracy of 84.17%, followed by the Extra Trees Classifier algorithm with 83.85%. It is evident that the average Accuracy scores between the top five algorithms are very close. The difference between the first model and the fifth one is 1.06%.

Table 1: The average performance results for each algorithm (sorted using Accuracy from highest to lowest)

Model	Accuracy	AUC	Recall	Prec.
Light Gradient Boosting Machine	0.8491	0.9253	0.8606	0.8414
Extreme Gradient Boosting	0.8446	0.9216	0.8520	0.8394
Gradient Boosting Classifier	0.8437	0.9200	0.8489	0.8404
Random Forest Classifier	0.8417	0.9171	0.8489	0.8369
Extra Trees Classifier	0.8385	0.9171	0.8543	0.8282
Ada Boost Classifier	0.8166	0.9013	0.8114	0.8200
K Neighbors Classifier	0.7912	0.8622	0.7600	0.8107
Ridge Classifier	0.7763	0.0000	0.7731	0.7783
Linear Discriminant Analysis	0.7763	0.8611	0.7731	0.7783
Logistic Regression	0.7742	0.8613	0.7690	0.7775
Decision Tree Classifier	0.7711	0.7711	0.7740	0.7697
Quadratic Discriminant Analysis	0.7494	0.8351	0.6748	0.7934
Naive Bayes	0.7429	0.8224	0.7420	0.7439
SVM - Linear Kernel	0.7325	0.0000	0.7672	0.7188
Dummy Classifier	0.4993	0.5000	0.3000	0.1497

Table 2 shows the performance results obtained for each ensemble method using the three best individual models, including Light Gradient Boosting Machine, Extreme Gradient Boosting, and Gradient Boosting Classifier. Based on these results, the blending ensemble and the stacking ensemble methods achieved average Accuracy of 85.29% and 85.61%, respectively, which are the highest accuracies compared to the accuracy of the base models.

Table 2:Performance results for ensemble models

	Accuracy	AUC	Recall	Prec.
Blending:	0.8529	0.9273	0.8656	0.8442
Stacking :	0.8561	0.9245	0.8629	0.8514

Conclusion:

After different NLP text preprocessing techniques were applied to the news dataset, several machine learning models, including traditional classification algorithms and ensemble methods, were built to classify the news into fake or real. Based on the results, the Light Gradient Boosting Machine model obtained the best performance compared to the other individual models. However, the stacking ensemble model achieved the highest performance in this study.

limitations:

The limitation of this study is that it deals only with textual data like the body text. But other essential components the news content could have, such as images and videos, need to be considered.

Recommendations:

Based on this study's results, machine learning methods could reach an acceptable accuracy in detecting fake news. In the case of news text-based, the stacking ensemble model is recommended as a good fake news detector.

Future Uses/Additional Applications:

Besides detecting fake news, the classifier also can be applied to detect spam emails, or it can be used to classify the sentiment of a given text document as positive or negative.

Challenges/Issues:

I do not think I faced some severe challenges. At first, when I thought about this project, the first thing I considered was the unbalanced classification problem. But fortunately, when I found a dataset with an equal number of samples for each class, I realized then that the class distribution would not be an issue.

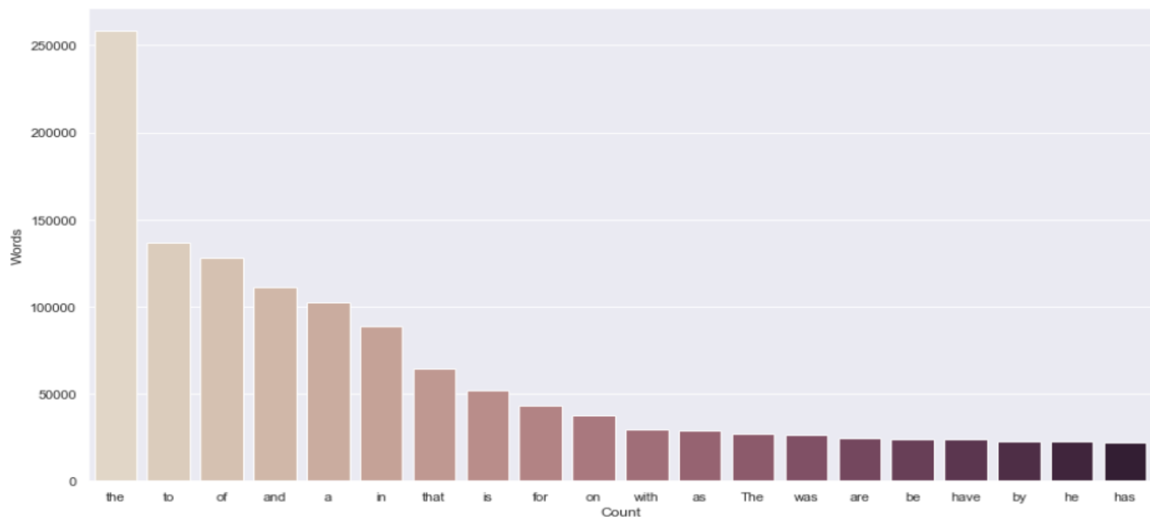
Ethical Considerations:

Although the project aims to contribute to the fight against fake news, some ethical concerns need to be taken into account. Suppose the model is used in some platforms as publishing censorship. So the model would highlight articles classified as fake for deleting them. In this case, there could be a human rights problem related to the right to publish, especially since there is no consensus on what is considered news fake. In all cases, there is no way to stop the right of people to publish their articles based on models that are most likely to have some errors. In this case, how will misclassified articles be handled?

Appendix

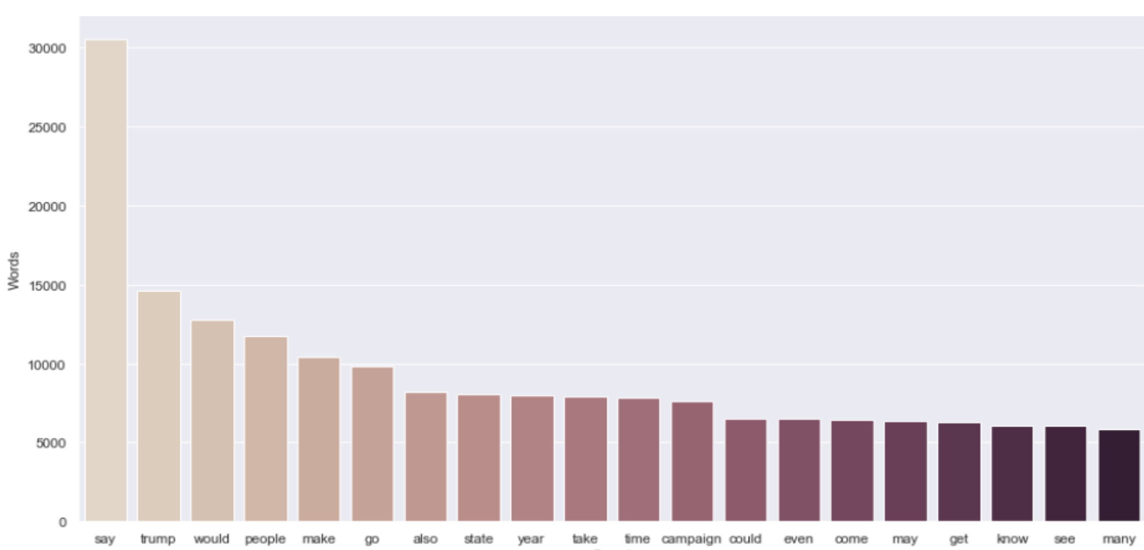
Words Frequency Before and After Text Preprocessing

Top words before cleaning



.

Top words after cleaning



References:

Burkhardt, J. M. (2016). Chapter 1. History of fake news | Burkhardt | Library technology reports. Retrieved from https://journals.ala.org/index.php/ltr/article/view/6497

Home. (2021, July 04). Retrieved from https://pycaret.org/

Figueira, Á., & Oliveira, L. (2017, December 14). *The current state of fake news: Challenges and opportunities*. Procedia Computer Science. Retrieved December 5, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050917323086.

Jain, A., Khatter, H., & Shakya, A. (2019, September). A smart System for Fake News Detection Using Machine Learning. Retrieved December 5, 2021, from https://ieeexplore.ieee.org/document/8977659/

Albahr, A., & Albahr, M. (2020). An empirical comparison of fake news detection using different machine learning algorithms. Retrieved from https://thesai.org/Publications/ViewPaper?Volume=11&Issue=9&Code=IJACSA&SerialNo=17

Fake news dataset. (2019). Retrieved from https://www.kaggle.com/kamleshsolanki/fake-news-dataset