DECODING DISINFORMATION: A MACHINE LEARNING FRAMEWORK FOR NEWS VERIFICATION AND THEMATIC CLASSIFICATION

Zoey Chappell, Luke McEwen, Daniel Wolosiuk
Rochester Institute of Technology
Rochester
zac9557@rit.edu, ltm4331@rit.edu, dw4102@rit.edu

ABSTRACT

Untrustworthy news spreads rapidly across social media and can have catastrophic effects. This research presents a customizable framework for detecting and classifying fake news using both supervised (Naive Bayes, K-NN) and unsupervised (K-Means, DBSCAN) machine learning techniques. Using a dataset of over 9800 raw labeled news articles, a custom cleaning algorithm is created and applied to Naive Bayes, K-NN, K-Means, and DBSCAN.The custom cleaning algorithm removes stop words, improperly formatted tokens, numerical values, and other tokens with little to no value. Interestingly, we found that stop words improved the performance of DBSCAN and K-Means. Common clusters identified by the unsupervised learning algorithms are U.S. Foreign Policy Making, Oil Drilling in the Midwest, and Federal Tax Law. We found that Naive Bayes had the best overall performance, with accuracy, precision, recall, and F1 Score of 97% while K-NN followed closely with 94%. DBSCAN underperformed on this dataset. This work contributes a flexible, transparent, and tailored approach for fake news detection with applications in information integrity efforts.

***K*eywords** Fake News Detection · Machine Learning · Supervised Learning · Unsupervised Learning

1 Introduction

Fake news is defined as a form of entirely fabricated news that is presented as truth and can refer to political satire or propaganda [1]. The affects of fake news range from damaging personal reputations to influencing political decisions, prompting many governments to investigate and enact legislation against fake news [1]. The challenge of fake news is multifaceted, making it challenging to determine where to focus efforts. As discussed in of[2], public concern about fake news arose in the United Statues during the 2016 presidential election and this theme is represented in the chosen dataset.

The News Detection (Fake or Real) Dataset, created by Nitish Jolly and found on Kaggle, contains 9865 raw news articles that are labeled to be real or fake[3]. A sample of the dataset is shown in Figure 1. A brief description of stop words and their effectiveness is discussed and applied to our data cleaning strategies [4].

This project leverages the News Detection (Fake or Real) Dataset to train both unsupervised and supervised machine learning models. A key component of our approach is a custom data processing function, in which a mix-and-match approach can be taken to the data cleaning process. This is essential for preparing the dataset for the machine learning algorithms.

For unsupervised learning, we use K-Means and DBSCAN to cluster the raw text articles based on context similarity. Then, we will manually analyze the clusters to determine a cluster label that describes the content of the articles. For example, political or satire articles. The supervised learning algorithms used are K-Nearest Neighbors (K-NN) and Naive Bayes, and the goal will be identifying false news from real. Again, the two models will be compared to determine the best performance.

The ultimate goal of this project is to contribute to the broader effort of identifying and understanding fake news by identifying context patterns. The techniques we used can be applied to social media or news platforms to support media literacy and empower Internet users with tools to verify online information.

Text	label
9865 unique values	Fake 51% Real 49%
Top Trump Surrogate BRUTALLY Stabs Him In The Back: 'He's Pathetic' (VIDEO) It s looking as though ...	Fake
U.S. conservative leader optimistic of common ground on healthcare WASHINGTON (Reuters) - Republican...	Real

Figure 1: Example entry in the News Detection (Fake or Real) Dataset by Nitish Jolly

2 Literature Review

Edson Tandoc writes this article to discuss the phenomenon of fake news, and provides definitions, examples, and potential solutions [1]. The foundation provided by Tandoc is built upon by Greifeneder et. al., where the effects of fake news are explored more deeply. For example, the psychology and pervasiveness, as well as where to focus efforts on, are discussed in this article [2].

Nitish Jolly provided the dataset used in this paper [3]. Stop words and their usage are explained in [4].

3 Methodology

The primary objective of this project is to develop a customization library of resources for detecting and classifying fake news. Supervised machine learning algorithms will use the real/fake labels provided to determine whether an untested news article is real or fake. Unsupervised machine learning will be utilized to classify the data into families that were not originally derived from the provided dataset. Examples of families might include political bias, satire, ideological framing, or sensationalism. This project can be used to provide insight into the most common types of disinformation that are spread and will apply to any text-based media.

3.1 Data Selection and Preprocessing

We used the News Detection (Fake or Real) Dataset by Nitish Jolly, which consists of 9865 news articles. Each entry in the database is a full-text news article and a label, which was used during the supervised learning algorithms (Jolly, N). 51% of the articles are labeled real and 49% are labeled fake. The nearly equivalent distribution mitigates concerns related to class imbalance that introduce bias to the algorithms. However, the dataset was initially unprocessed and required extensive cleaning before being suitable for use in machine learning algorithms. The first step involved reading the data from a CSV file using the pandas library. The text of each article was then tokenized using the RegexpTokenizer from the Natural Language Toolkit (NLTK), which split the text into individual word tokens. The labels were extracted into a separate list.

3.1.1 Customizable Cleaning Function

The tokenized dataset was processed through a customizable cleaning function designed to remove irrelevant tokens. An essential component of this process is the bag_of_words() function, which is a helper function for cleaning and a standalone application. It generates a unique set of all words present in the dataset, which were then manually analyzed to identify and compile a custom list of unwanted tokens. The words in the custom unwanted list are words that we deemed do not have a use in identifying fake or real news, and include:

- Months and weekdays, which may not have semantic weight.
- Media identifiers and outlet names, unless needed for targeted analysis.
- Arbitrary strings, abbreviations, or malformed tokens.

The tokens are categorized to allow customization, for instance if the algorithm was looking for election month fake news, the months would be a useful feature. This is shown in Figure 2.

```
[# months
 'january', 'february', 'march', 'april', 'may', 'june', 'july', 'august', 'september', 'october', 'november', 'december',
  # week days
 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday',
# stopwords not removed
'said', 'could', 'soon', 'told', 'says', 'also', 'since', 'much', 'like', 'every', 'went', 'made', 'might', 'would', 'puts'
# note, this line is like news anchors - could keep if wanted to do something with
'cspan', 'fox', 'cn', 'snl'
# random
'http', 'https', 'getty', 'ohm', 'lulu', 'lrso', 'sulu', 'oompa', 'jaly', 'abney', 'jpg', 'sdf', 'ubs', 'ch', 'ygd', 'ttp'
'kwame', 'shiya', 'boas', 'kiche', 'assa', 'omni', 'jaly', 'pamby', 'acdc', 'gopac', 'tenga',  'ym', 'noy', 'oz', 'da', 'ge'
'kassy', 'w1', 'erika', 'cotti', 'df1', 'bpfh', 'fku', 'lir', 'gaier', 'syed', 'dje', 'edva', 'abedi', 'hk', 'zoe', 'eog', 'kok'
'zakka', 'karim', 'madi', 'svcs', 'oag', 'ramaj', 'eroc', 'eau' 'haass', 'kteg', 'dubke', 'sergi', 'kirt',
'ddhq', 'kptv', 'gwich','s1ppi', 'josie'
]
```

Figure 2: Custom unwanted list.

The cleaning process itself proceeds as follows:

1. All tokens are converted to lowercase to mitigate case-sensitivity issues.
2. Common stop words are removed using NLTK's built-in list of English stop words. Note, stop words are widely used words such as "the" or "in" (Web Communications, 2025).
3. Filter the tokens, based on:
   (a) Exact matches to words in the custom unwanted list.
   (b) Tokens that are numeric or contain letters and numbers (e.g., 2017 Election).
   (c) Tokens that include underscores.
   (d) Tokens that match random alphanumeric strings (e.g. u1rd4b6cz2) identified using regular expressions.

3.1.2 Data Transformation

Once the cleaning step is completed, the tokenized data is converted into various forms. Firstly, it is converted into feature vectors using the Term Frequency - Inverse Document Frequency (TF-IDF) method. TfidfVectorizer from scikit-learn was configured to extract both unigrams and bigrams, with an adjustable max_features parameter to control vocabulary size. Principal Component Analysis (PCA) was applied to reduce the dimensionality of the sparse TF-IDF matrix, resulting in an improved interpretability of the data while retaining the most significant parts. Finally, the data was split into training and testing sets using the train_test_split function from scikit-learn. A fixed random seed was used to ensure a reducible split.

3.2 Supervised Learning Algorithms

The objective of supervised learning is accurate classification of untested fake news articles. Such a technology can serve numerous purposes, including the detection (and blockage) of misinformation and disinformation in a content-filtering application. Two models were used for supervised learning: k-NN and Naive Bayes.

K-NN was considered for its non-linear decision boundaries and and interpretability. Non-linear decision boundaries are valuable because complex patterns can be observed when combined with well structured feature space. Interpretability is considered because "nearest neighbors" is very useful for explaining why a prediction decision was made.

Naive Bayes was primarily considered for its resistance to overfitting data. Independence assumptions are present in this model, which keeps the model "simple". After training, Naive Bayes can very quickly predict labels on untested data. This is very useful for application-level implementations, such as content filtering systems. In other words, Naive Bayes is very practical for a real-world situation.

Both models feature a configurable "vocabulary size" parameter. This parameter selects the "top x most important words" of a given text and uses only those for consideration. For Naive Bayes, this parameter is not meaningful since model performance remains nearly the same. For K-NN, however, it is found that model performance is higher when the vocabulary size "max_features" parameter is smaller.

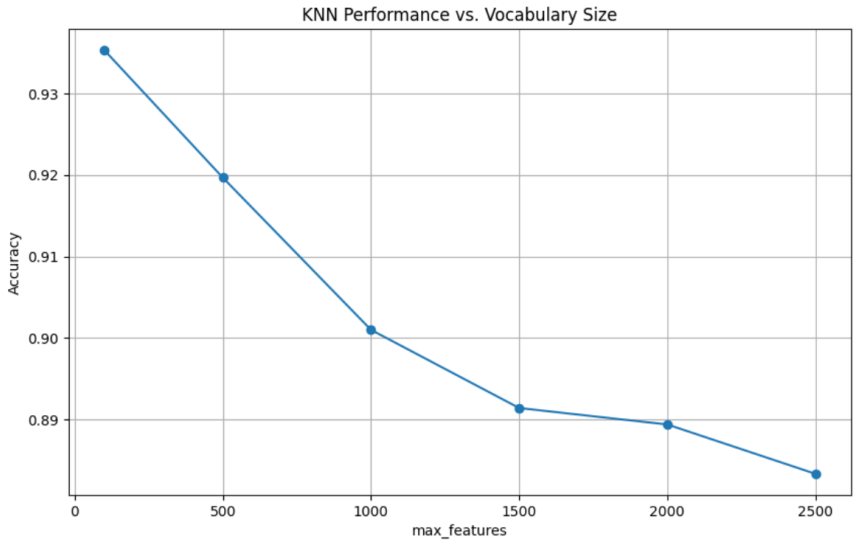

Figure 3: Graph of vocabulary size versus Accuracy for K-NN.

Experimentation was conducted with an 80/20 training and testing set split. Although both models exhibited good performance, Naive Bayes consistently observed a slightly higher performance than shown in K-NN in every tested metric.

To generate these metrics, both models had configured parameters which maximized performance. For K-NN, the "max vocabulary" size was set to 100, and for Naive Bayes the same parameter was configured to 1000.

3.3 Unsupervised Learning Algorithms

The goal of the unsupervised learning portion is to attempt to discover groupings of news articles without the need to label. The two algorithms used for this included K-Means and DBSCAN. K-Means was picked for its simplicity, speed, and interpretability. K-Means is really efficient at partitioning data into distinct clusters making a good baseline. DBSCAN was chosen for its ability find outliers offering a complementary perspective to K-Means. Both used a set of vectorized articles using TF-IDF, followed by dimensionality reduction with PCA. K-Means was implemented with a variable number of clusters which looped from k=2 to k=20. For each value the clustering output was compared against the true labels using metrics. These scores were plotted to visualize performance variance between the range of numbers. To attempt to overcome a few of the shortcomings of K-Means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was also tested. DBSCAN does not require a specific number of clusters but instead relies on two

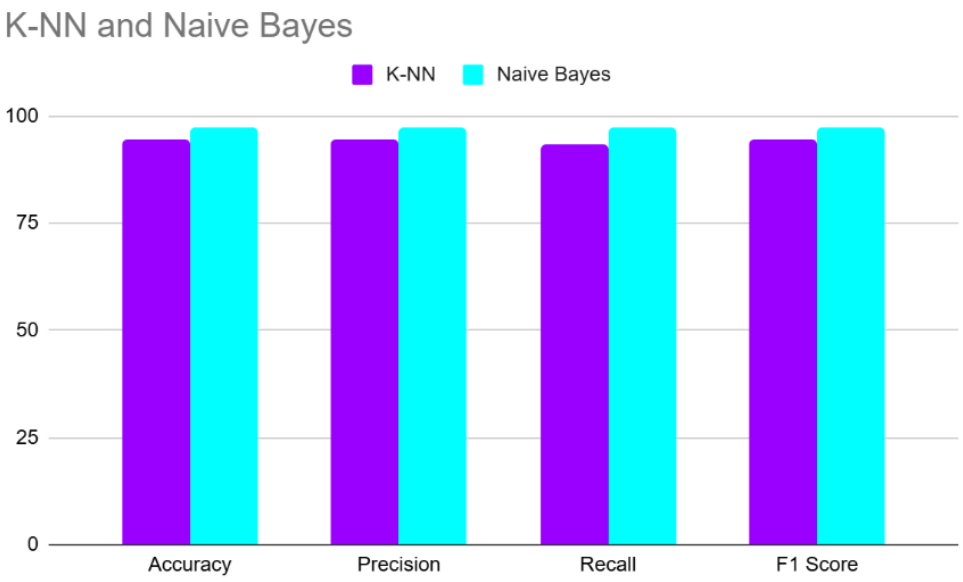

Figure 4: Performance metrics of K-NN versus Naive Bayes.

key parameters, neighborhood radius (eps), and minimum points to form a cluster (min_samples). The eps parameter was again varied from 0.3 to 1.2 and then scores of the clustering were plotted and compared to see which value of epsilon has the best success.

Figure 5: Graph of clusters.

4 Results

When using unsupervised machine learning algorithms, stop words can improve the performance, as shown in Figure 7. This contrasts with our initial hypothesis, as we expected removing stop words to lead to more accurate clustering results.

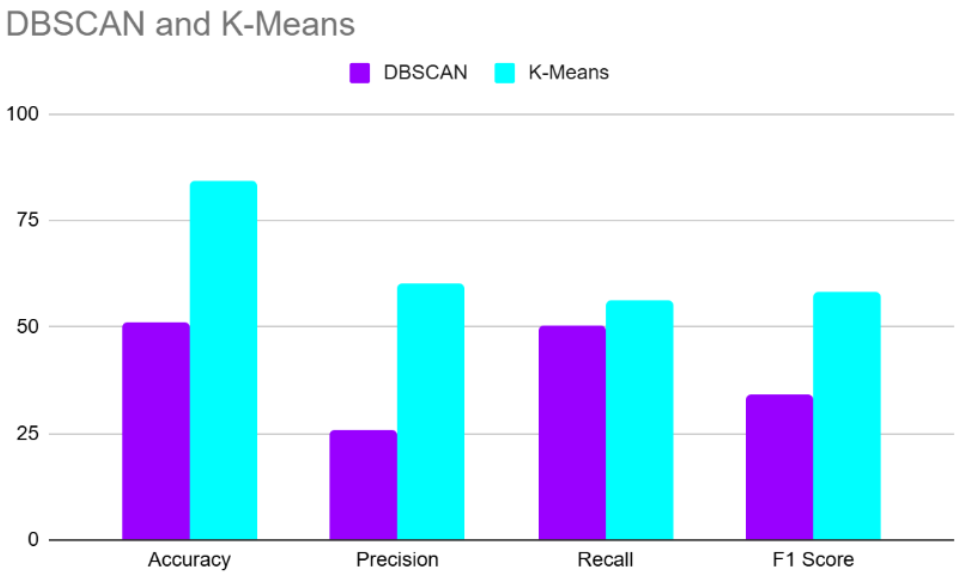

Figure 6: Performance metrics of DBSCAN versus K-Means.

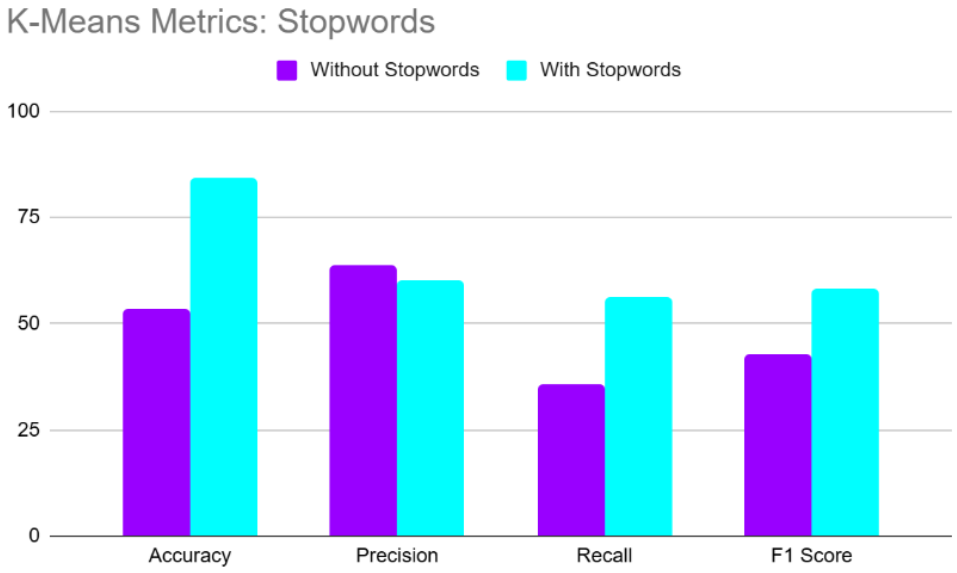

Figure 7: K-Means performance metrics.

Among the unsupervised algorithms, DBSCAN had the weakest performance with an accuracy of 50.6%, precision at 25.3%, recall at 50.0%, and a F1 Score of 33.6%.

In contrast, supervised learning algorithms consistently outperformed the unsupervised algorithms accross all performance metrics. This is demonstrated in Figure 8. Naive Bayes had the best performance, with 97% across all metrics categories. K-NN followed closely behind with 94%.

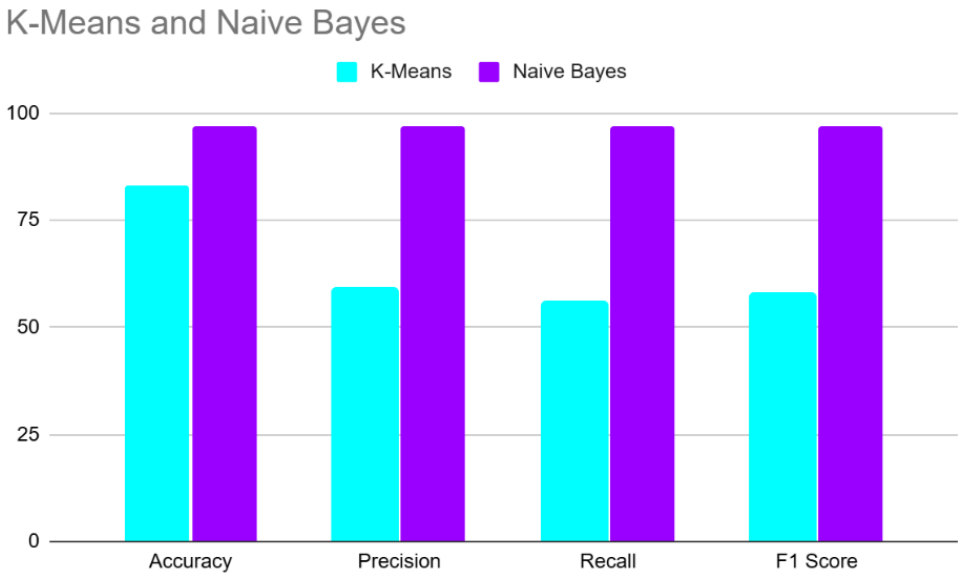

Figure 8: Performance metrics of K-Means (unsupervised) versus Naive Bayes (supervised)

5 Discussion

This study takes a look at both supervised and unsupervised learning techniques. It compares two supervised algorithms on predicting fake news articles. Along with two unsupervised algorithms on grouping the news articles together without any predefined labels. The outstanding performance of Naive Bayes and K-NN highlight the effectiveness of predicting with well labeled data. The K-NN model was found to highly sensitive to high vocabulary size with a smaller sized list improved performance. Naive Bayes remained strong regardless of the size of the vocabulary making it more robust for general use.

The results from the unsupervised model did not share this same black and white approach as before. This time K-Means and DBSCAN were tasked with grouping the data together, and then we as a team labeled the data as we saw it. K-Means struggled with a higher number of clusters reaching peaks at 3 and 8. However we wanted to see how the data looked at a low and high cluster count. For 3 clusters we created the labels, U.S. Foreign Policy Making, Oil Drilling in the Midwest, Federal Tax Law. However looking at the output from 20 clusters where the metrics didn't do very well. We were able to create much more niche labels based on the output which told a much larger story about the news articles we were looking at. Some of the labels we created for 20 clusters include Anti-Vaccine and Public Health Skepticism, FBI and Intelligence Communities, Iran - Iraq Conflict and Oil Geopolitics, Fake News and Clickbait Patterns. Unfortunately DBSCAN did not have the same effect on the dataset and would only ever make up to 3 clusters since number of clusters can't be specified.

The customizable cleaning and preprocessing framework proved to be essential allowing for targeted analysis. This flexibility enhances the framework's potential for real world applications such as social media monitoring and news aggregation.

6 Conclusion

Supervised and unsupervised models training on fake/real news data sets is an impactful technology that can be implemented in numerous applications. This can be used in news aggregation platforms, social media platforms, and digital literacy sources to increase service quality. The categorization of real/fake news via machine learning is useful to spot trends in the spread of misinformation of current events, such as what is common in social media platforms.

Acknowledgments

We would like to thank our professor Saniat Sohrawardi for his guidance, feedback, and support throughout this project. We also appreciate the resources and facilities provided by the ESL Global Cybersecurity Institute at Rochester Institute of Technology, which made this research possible.

Tools Used:

ChatGPT

Usage: Artificial intelligence tools were used to support this project in the following ways: brainstorming appropriate libraries, detecting coding errors, formatting regular expressions, generating code for data visualization, and refining grammar and clarity in written content.

Verification: Cross-checked with library manual page and manual testing.

Prohibited Use Compliance: I confirm this work adheres to course AI policies, with no unauthorized use in this presentation. All AI-assisted components meet required substantial modification standards.

References

[1] Edson C. Tandoc Jr. The facts of fake news: A research review. *Sociology Compass*, 13(9):1–9, 2019.

[2] Rainer Greifeneder, M. Jaffe, E. Newman, and N. Schwarz. The psychology of fake news. In *The Psychology of Fake News*, pages 11–25. Routledge, 2020. Original work published 2025.

[3] Nitish Jolly. News detection (fake or real) dataset. `https://www.kaggle.com/datasets/nitishjolly/news-detection-fake-or-real-dataset/code`, 2024. Kaggle.com.

[4] Web Communications. Stopwords. `https://sraf.nd.edu/textual-analysis/stopwords/`, 2025. Software Repository for Accounting and Finance.