Fake News Detection with Headlines

Gared Ramirez
12/12/2023

Mentors:
Dr. Meng Li, Noah Harding Assistant Professor, Rice University Department of Statistics
Dr. Elizabeth McGuffey, Associate Teaching Professor, Rice University Department of Statistics

Acknowledgments:
We would also like to thank the support given by Arya Muralidharan, our teaching assistant.

Abstract

Fake news has become an increasing problem due to the rising use of the Internet and social media. It is important to be able to distinguish sources of fake and misleading news articles to ensure that misinformation does not sow discord, erode trust in credible sources, and negatively impact our personal and societal well-being. Moreover, in an age where many people only skim headlines without delving into the full articles, the ability to discern fake news from headlines alone becomes even more crucial. To detect and classify fake news, we implement and compare five machine learning models–naive Bayes, logistic regression, decision tree, random forest, and support vector machine–on two different datasets: a benchmark dataset and a dataset with full articles and headlines. We utilize measures such as term frequency-inverse document frequency and sentiment scores, as predictors in our models. We find that naive Bayes consistently performs best on both datasets with accuracies of 64.40% and 92.56%, respectively.

1. Introduction

Fake news is commonly defined as inaccurate information that resembles legitimate news media content (Lazer et al., 2018). It generally takes the form of two distinct categories: misinformation and disinformation. Misinformation refers to information that is unintentionally false or misleading, while disinformation involves the intentional spread of false content (Carsten Stahl, 2006). Both types are hazardous, as they have the potential to undermine public trust in news and shape individual thoughts and opinions.

Recent evidence indicates that approximately 62% of American adults utilize social media as their primary source of news consumption (Gottfried & Shearer, 2019). The high availability and speed of the Internet compound the problem by allowing easier propagation of information, including fake news. When a substantial portion of Americans rely on social media for news, the proliferation of fake news can cloud their judgment, making it challenging to distinguish between reality and misinformation and resulting in dire consequences. For example, in Texas, misinformation about vaccine risks had the potential to drop the measles vaccination coverage to below 95%. This drop in coverage could prevent herd immunity leading to more measles outbreaks (Hotez, 2016). During Hurricane Katrina, there were increased reports about lawlessness due to limited access to the impacted region. These reports, however, were often baseless rumors. As a result, communities were stigmatized based on these unfounded claims, and resources were diverted from those in need (Guarino, 2015). During the 2016 U.S. presidential election, it was discovered that a small number of social bots were spreading a large amount of disinformation including fabricated news, hoaxes, and conspiracy theories (Shao et al., 2017). Some suggest that the spread of false information shifted the results of this election, although the exact extent is difficult to quantify (Parkinson, 2018). During the devastating 2019-2020 Australian wildfires, multiple misleading images were circulated on social media. These images inaccurately portrayed the fire locations and relative risk associated and posed potential dangers to individuals (Rannard, 2020).

Due to these issues, it has become increasingly important to classify text into real or fake news. Researchers have conducted various studies to understand which text features produce the most accurate classification results. Word frequency analysis has proven to be an effective classifier in predicting fake news with up to 75.40% accuracy when utilized with a naive Bayes model and full article text (Granik & Mesyura, 2017). Although these results are fairly accurate, it is worth noting that word frequency analysis may overlook words that appear frequently in one document but rarely in others. Term frequency-inverse document frequency (tf-idf) deals precisely with this issue and produces accuracies of 89.11% and 84.97% when combined with decision tree and random forest models, respectively (Jehad & Yousif, 2020). Due to their simplicity, and high classification accuracies, these tf-idf features are often used as a baseline.

A large portion of research in fake news detection strives to identify crucial textual elements, in addition to term frequencies, to achieve the highest prediction accuracy. One example is utilizing part-of-speech (POS) attributes such as the number of nouns and verbs. Khanam et al. (2021) showed that an extreme gradient boosting model with POS attributes produced the highest accuracy of 75%. Features extracted via sentiment analysis are also useful for classification. Iwendi et al. (2022) incorporate sentiment scores that portray how positive, negative, and neutral

a given text is; they achieve an accuracy of 86.12% when combined with gated recurrent units (Iwendi et al., 2022). K-means clustering was utilized by Yazdi et al. (2020) for feature reduction. They then utilized a support vector machine (SVM) model to compare results when using all features and when using a reduced set of features. Their proposed method saw an increase in accuracy from 91.76% to 94.19%.

While prior studies have yielded reasonably accurate results, there has been limited focus on situations where the text document is short or background information about the speaker is not easily accessible, such as with news article headlines. Previously mentioned feature extractions were performed on the entire body of text or social media posts, with some spanning multiple paragraphs. However, most individuals will take no more than a glance at a headline before moving on. If individuals do not engage with the entire article, it becomes imperative to identify the optimal classification features within the headlines rather than relying on the full article text.

The purpose of this paper is to implement various machine learning models–naive Bayes, logistic regression, decision tree, random forest, and support vector machine–and compare their accuracies when classifying fake news based on article headlines. We start by replicating previous studies using all available information as predictors, including speaker history and lexicon features, such as sentiment scores and tf-idf values. Then we investigate the effect of including only semantics and lexicon features as predictors. Models are trained using both a "benchmark dataset" and "headline dataset". Testing is ultimately performed on headlines only, and models are compared using accuracy and receiver operating characteristic curve (ROC) curves.

The rest of the article is organized as follows: In Section 2, we provide details on both datasets used. In Section 3, we detail the methods utilized for feature extraction and classification. In Section 4, we present results, and in Section 5, we discuss our findings and possible future steps.

2. Data Description

Subsections 2.1–2.3 describe benchmark dataset and the headline dataset including where the data was obtained, what features are available, and preliminary data exploration. Subsection 2.4 discusses data preprocessing steps such as stop word removal and stemming.

2.1 Benchmark Dataset

This benchmark dataset, also known as LIAR, was originally presented by Wang (2017) and serves as a benchmark dataset for fake news detection. It features 12.8K statements made from 2007 to 2016 from various speakers and organizations. The average sentence length is approximately 18 words. All statements were collected and labeled with a truth meter rating by Politifact.com (Politifact, 2023**)**. Labels for a subset of 200 randomly selected statements were verified by Wang; the agreement rate measured by Cohen's kappa was 0.82, indicating the majority of statements were labeled correctly.

There are a total of six truthmeter ratings or truthfulness categories: *pants-fire*, *false*, *barely-true*, *half-true*, *mostly-true*, and *true*. *Pants-fire* are statements that are not accurate and make an

unreasonable claim. *False* are statements that are not accurate. *Barely-true* are statements that contain some truth but ignore critical facts. *Half-true* are statements that are partially accurate but take things out of context. *Mostly-true* are statements that are accurate, but need clarification. *True* are statements that are accurate and are not missing any information. The total counts are shown in Figure 1. Since our headline data only contains true and false labels, this paper focuses on those respective categories. We combine categories *pants-fire*, *false*, and *barely-true* to be *false*, and *half-true*, *mostly-true*, and *true* to be *true.*

Information about the speaker's truth history is also reported. This history includes the number of *barely-true, false, half-true, mostly-true, and pants-fire* statements they have made previously. This information is utilized in predictors to our models when replicating the results from previous papers. The speaker's location is also present within the dataset.

Each statement is also associated with a topic. The top five topics are *health care*, *taxes*, *elections*, *immigration*, and *education*. Table 1, shows the proportion of truthfulness categories within each topic. *Health Care* has a slightly larger proportion of false statements compared to the rest of the topics. Similarly, *election* has a slightly larger proportion of true statements compared to the rest of the topics.

2.2 Headline Dataset

This dataset is composed from two different sources. Ahmed et al. (2017) collected 21.4K legitimate news articles from Reuters.com (Reuters Editorial, 2023**)** . The articles were collected from January 2016 to December 2017. Each entry contains the title, the original article text, the subject, and the date published. All articles contain text that is greater than 200 words. Each subject is labeled by topic, although this will not be utilized in our analysis as they only contain 2 different labels. Since these articles are from a reputable source, they are labeled as containing true information.

In addition, Ahmed et al. (2017) collected 23.5K fake news articles from various non-reputable sources from January 2016 to December 2017. Journalists from Politifact verified the validity of each article. Similar to the Reuters dataset, each entry contains the title, the original text, the subject, and the date published.

2.3 Data Exploration and Comparison

We created word cloud diagrams to visually present the prevalent words found within the benchmark dataset, the titles from the headline dataset, and the full article text from the headline dataset. Figure 2 displays these word clouds. As expected, there is overlap in common words from the headline text and headline titles such as "Trump", "Republican", and "Democrat". These common words reflect the time period during which the data was collected, coinciding with the election of Donald Trump as the President of the United States. In contrast, the word cloud generated from the benchmark dataset exhibits analogous political terms, albeit with a focus on the years from 2007 to 2017. Noteworthy terms within this timeframe include "Obama", "Democrat", and "health care". Single letter words such as "s" indicate an apostrophe was present such as "Trump's".

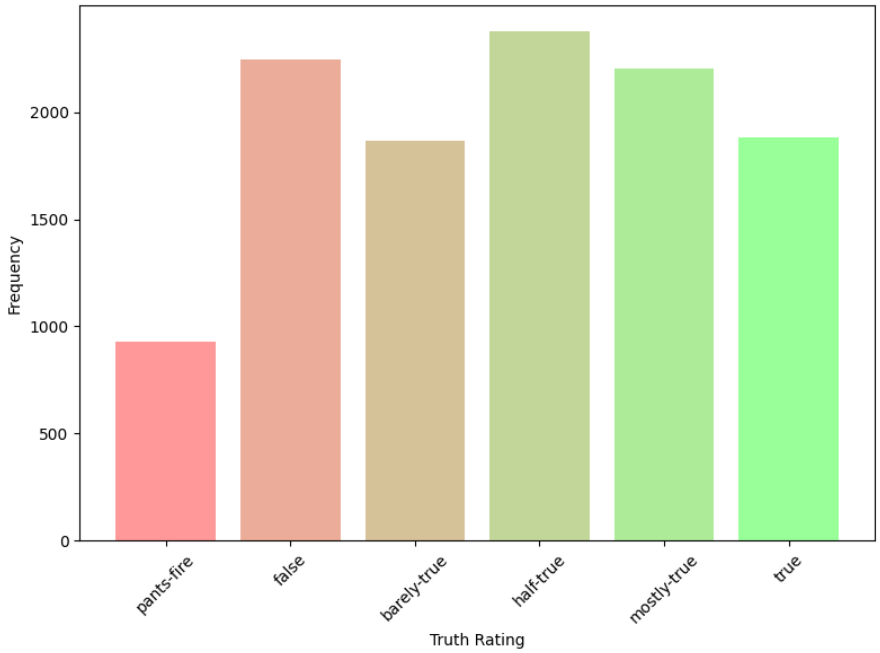

Figure 1: Truth rating frequency from the benchmark dataset. Ordered from most false to most true.

Table 1: Proportion of truthfulness within each of top five prevalent topics from the benchmark (LIAR) dataset. Bold values indicate the highest truth proportion category for each topic.

Topic	Pants-Fire	False	Barely-True	Half-True	Mostly-True	True	Size
Health Care	0.090	**0.243**	0.181	0.202	0.135	0.150	421
Taxes	0.067	0.174	0.168	**0.211**	0.205	0.174	327
Election	0.113	0.205	0.120	0.152	0.177	**0.233**	283
Immigration	0.103	0.210	0.166	**0.225**	0.203	0.092	271
Education	0.041	0.139	0.150	0.225	**0.255**	0.191	267

2.4 Data Preprocessing

Before analyzing text data, certain preprocessing steps are necessary to speed up runtime by removing noise and ensuring only relevant features are utilized. Such modifications include stop-word removal, punctuation removal, and stemming. Natural Language Toolkit (NLTK) was used for all these modifications (NLTK Project, 2023**)**.

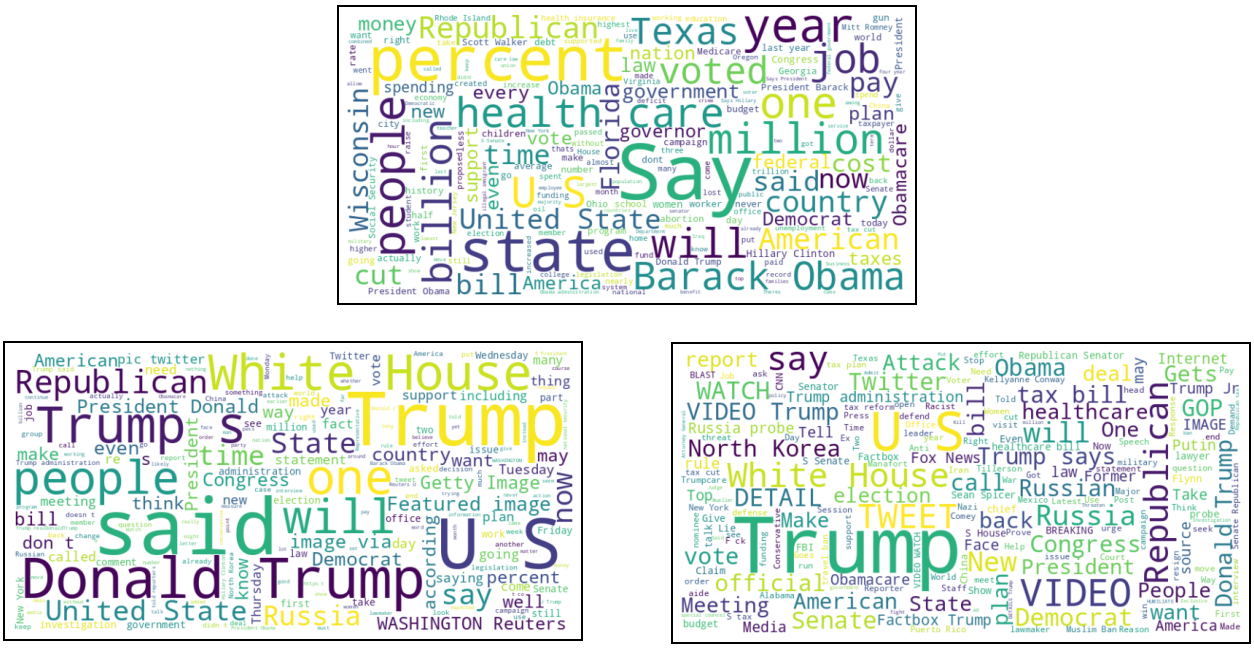

Figure 2: Top word cloud displays the most common words from the benchmark dataset text. Bottom left word cloud displays the most common words from the full text from the headline dataset. Bottom right word cloud displays the most common words from the titles from the headline dataset.

Stop words are common words found within a language that are considered to add little or no substantive meaning. Stop words are often removed to reduce noise and improve the efficiency of machine learning models. Types of stop words that we removed are articles such as “a” and “the”, conjunctions such as “but” and “or”, and prepositions such as “on” and “at”. In addition, the benchmark dataset contains words similar to “say” and “states” at the beginning of each statement. These words were also removed.

Stemming is the process of simplifying words to their root form by removing prefixes, suffixes, and affixes. Similar to stop word removal, stemming decreases noise by allowing words with similar meanings to be grouped rather than being interpreted differently. For example, “jumps”, “jumped”, and “jumping” are all reduced to “jump”. The Porter Stemming algorithm implemented from NLTK (NLTK Project, 2023**)** was utilized for this step.

3. Methods

Subsection 3.1 discusses feature extraction techniques such as term frequency-inverse document frequency and sentiment analysis. Subsection 3.2 discusses the machine learning models utilized. Figure 3 illustrates the workflow that we followed including preprocessing steps, models used, and model effectiveness metrics.

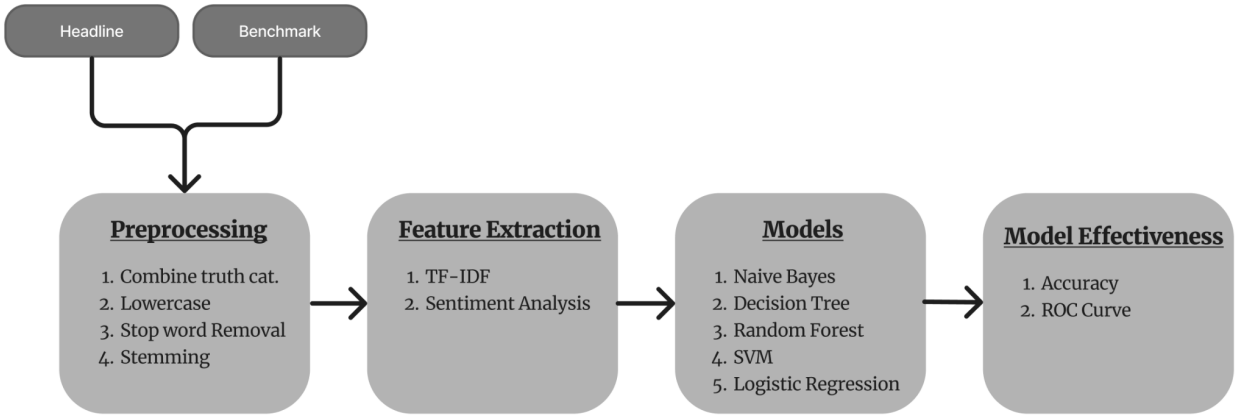

Figure 3: Workflow for working with the data. Step 1 includes preprocessing on both the benchmark and headline datasets. Step 2 includes extracting features. Step 3 includes training our models. Step 4 includes testing our models using accuracy and ROC curves.

3.1 Feature Extraction

Feature extraction is the process in which relevant information is obtained from the original text data. This step lays the groundwork for subsequent analyses, enabling a more focused exploration of underlying patterns and insights within our datasets. We extracted two different features from our data. Subsection 3.1.1 discusses the term frequency - inverse document frequency, while Subsection 3.1.2 examines sentiment analysis.

3.1.1 Term Frequency - Inverse Document Frequency (tf-idf)

Term frequency-inverse document frequency (tf-idf) is a numerical statistic used to evaluate the importance of a term. In our context, a document is the text from an article or headline, and a term is an individual word within that text. The importance of each term within a document is weighted relative to the other documents. The tf-idf is the product of term frequency and the inverse document frequency.

The term frequency (tf) measures how frequently a term occurs in a specific document. More specifically the tf for a given term w in document d is as follows.

$$TF_{w,\,d} = \frac{|w|}{|d|},$$

where $|w|$ represents the number of times w appears in document d, and $|d|$ is the total number of terms found in d.

The inverse document frequency (idf) measures how unique a term is across the entire collection of documents. More specifically the idf for a term w is

$$IDF_{w} = log(\frac{|D|}{|D_{w}|}),$$

where $|D|$ represents the total number of documents and $|D_w|$ represents the number of documents containing term w.

The resulting tf-idf for a term *w* and document d is calculated by

$$TF\_IDF_{w,d} = TF_{w,d} \times IDF_w.$$

Higher tf-idf scores indicate that a term is distinctive in a particular document. Lower scores indicate that the term is either common across documents or not important in the document.

3.1.2 Sentiment Analysis

Sentiment analysis is a technique used to classify the emotional tone expressed in a piece of text into positive, negative, neutral, or compound. When utilizing the VADER sentiment library (vaderSentiment, 2018), each term within a document is given a valence score, x. This score ranges from negative four to positive four. Scores closer to negative four indicate an extremely negative emotional score, while scores closer to positive four indicate an extremely positive emotional score. Scores that are close to 0 indicate a neutral tone (Hutto & Gilbert, 2014). A compound score is generated by summing all the valence scores from a document, then normalizing to be between -1 and 1. The formula is as follows:

$$y = \sum_{x \in X} x;$$
$$C = \frac{y}{\sqrt{y^2 + \alpha}},$$

where X is all the valence scores in the document and α is a normalization constant. For our paper, the default value of 15 was utilized for α.

In addition to the compound score, each document receives a positive, neutral, and negative score. Since positive and negative scores are negatively correlated, we opted to only include the compound score to ensure that our features were independent from each other.

3.2 Classification Methods

The subsequent subsections discuss the models that were trained and tested. These include naive Bayes, logistic regression, decision tree, random forest, and support vector machine.

3.2.1 Naive Bayes

Naive Bayes is a statistical model based on conditional probabilities, which calculates the likelihood of K potential outcomes, denoted as C_k, given a set of n independent features

represented as the vector $x = (x_1, \dots, x_n)$. These probabilities can be calculated through the application of Bayes' theorem in the following formula:

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}.$$

Here, $p(C_k)$ is the prior probability for class C_k, $p(x|C_k)$ is the likelihood of the vector of n features given the class, and $p(x)$ is the joint probability of the features occurring. Since $p(x)$ does not depend on the classes it is effectively a constant; therefore only the numerator is of interest.

3.2.2 Logistic Regression

Logistic regression is a classification method used to predict the likelihood of an event based on a set of independent predictor variables. In logistic regression, a logit transformation is applied to the odds, which is the ratio of the probability of success to the probability of failure. This transformation is frequently referred to as the log odds or the natural logarithm of odds, and the full logistic model is expressed using the following mathematical expressions:

$$logit(p(x)) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + B_j x_j)}};$$
$$\ln\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x_1 + \dots + B_j x_j.$$

When applied to our context, we define a success as an article being fake,and the set of features, x, is the tf-idf values and the sentiment score obtained from the text or headline. Therefore, $p(x)$ is the probability that a certain article is fake. The value β_0 is an intercept term, and the values $\beta_1, \dots, \beta_j$ are the regression coefficients for each of the j features.

3.2.3 Decision Tree

Decision tree is a hierarchical model that is used to represent the most probable outcome given a set of conditional statements and feature inputs. Each internal node of the decision tree represents a condition on a feature within the model. For example, within our model, a node could be the condition “Is the compound sentiment score less than 0?”. Each leaf node represents a specific class or outcome, such as fake or legitimate, based on the path taken from the root node.

Entropy, the measure of the impurity in the observations, is utilized to develop the decision tree model. Mathematically entropy is expressed as the following:

$$E(S) = \sum_{i=1}^{K} - p_i log_2 p_i,$$

where S is the current state (node), p_i is the probability of outcome i at state S, and K is the total number of possible outcomes (in our case, two). From the current state S, entropy after basing a split on attribute (feature) X is the weighted average of entropy at each child node, and is denoted $E(S, X)$. The resulting information gain can be defined as the following:

$$IG(S, X) = E(S) - E(S, X).$$

The tree is formed by applying a greedy algorithm that selects the split with the largest information gain at each step. Figure 4 portrays an example of a decision tree.

3.2.4 Random Forest

Random forest models are an ensemble of decision tree models. They utilize a technique called bagging, which creates multiple different training subsets from the training data with replacement and then the classification output that has the majority of counts is the overall output. The process of bagging helps mitigate the issue of overfitting. The following is a general algorithm for generating random forests.

Random Forest Algorithm

---

1. Randomly select n data entries and m features from the original data. For our model, each data entry is the text for an article and the features are the tf-idf and compound sentimental analysis score.
2. Construct x individual decision trees from each sample, and generate an output c, the classification output. For our model, the outputs will be "True" or "False".
3. Final output is the class that has the most counts from all decision tree models.

3.2.5 Support Vector Machine

Support vector machines (SVMs) are supervised machine learning models that create decision boundaries or hyperplanes between classes. In the case that the data is linearly separable, two parallel hyperplanes can be selected such that the distance between them is as large as possible. The region between them is called the margin. A hyperplane that maximizes this margin is selected, and can be described by the following:

$$\min ||w||_2^2 \text{ such that } y_i(w^T x_i - b) \geq 1 \; \forall i \in \{1, \ldots, N\},$$

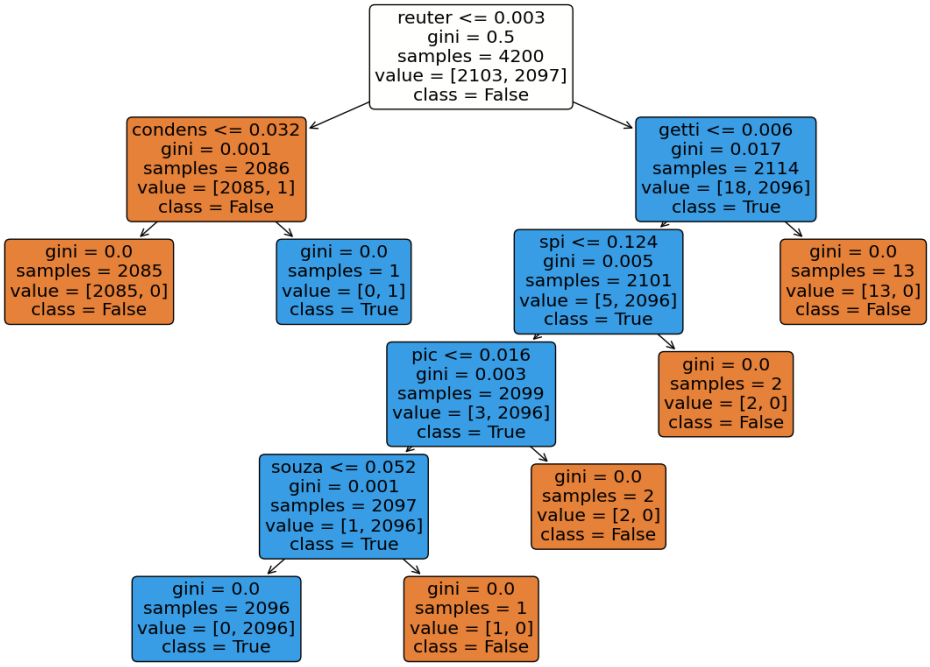

Figure 4: Example decision tree that predicts if a document is legitimate (True) or fake (False). Each condition is the first line of each node, and is based on the term frequency-inverse document frequency value for the given word.

where w is the normal vector to the hyperplane, x_i is the set of N features for observation i, y_i is the class of observation i, and $\frac{b}{||w||}$ is the offset of the hyperplane from the origin along the normal vector w.

In the case where the data is not linearly separable, the hinge loss function is utilized. This puts a penalty on data that is on the incorrect side of the hyperplane. More specifically, the following equation is minimized:

$$\lambda||w||^2 + [\frac{1}{N}\sum_{i=1}^{N} max(0,\ 1\ -\ y_i(w^T x_i\ -\ b))].$$

The only new parameter introduced is λ, which determines the trade-off between increasing the margin size and ensuring each x_i lies on the correct side of the margin.

3.3 Training and Testing Sets

We trained five models–naive Bayes, logistic regression, decision tree, random forest, and support vector machine–using four different approaches. Approach one included training and testing on the benchmark dataset using all available predictors, such as speaker, speaker history, and location. This approach served as a baseline to compare our subsequent approaches. Approach two included removing all features that were not common to both the benchmark

dataset and headline dataset; only the tf-idf and sentiment scores were used in this approach. We performed training on the benchmark dataset and testing on both the benchmark data and headlines from the headline dataset. Approach three consisted of training on the full article text from the headline dataset, while approach four consisted of training on the headlines from the headline dataset. Both approach three and four were tested on the full article text and the headlines. These approaches are displayed in Figure 5. In each approach, the respective dataset was split into 70% training and 30% testing.

4. Results

The accuracy results for approaches one and two are presented in Table 2. More specifically, this contains the testing accuracies when all predictors are included from the benchmark dataset, and when only the common predictors between the two datasets are present. Random forest performed the best on the full benchmark test with an accuracy of 72.45%. On the other hand, naive Bayes performed the best on the benchmark subset and headline tests with accuracies of 61.56% and 64.40%, respectively. In general, there was a decrease in accuracy as we removed features from the benchmark dataset and another decrease when we generalized our models to the headline dataset.

The testing accuracies for approaches three and four are presented in Table 3. More specifically, these approaches were trained on the full article text and headline text, respectively. The testing accuracies reflect when each approach was tested on both the full article text and headline text. In approach three, all models achieved above 95% accuracy, with random forest obtaining the highest accuracy of 99.67% when tested on full articles. For this approach, there was a general drop in accuracies across all models when tested across the headlines. Naive Bayes outperformed all models with an accuracy of 90.33%. It is also important to note that although random forest and decision tree have high accuracies when tested on the full article text; both models drop to 50.17% and 51.17%, respectively when tested on the headlines. In approach four, logistic regression achieved the highest accuracy of 91.33% with naive Bayes following with an accuracy of 91.06% when testing on the full article. When testing on the headline dataset, naive Bayes had the highest accuracy of 92.56%.

The corresponding ROC curves are displayed in Figure 6 and Figure 7 for all approaches. The AUC score is the area under the curve and is also shown in these figures. An AUC that is closer to 1 is better, whereas an AUC that is closer to 0.5 indicates the model predictions are as accurate as a coin flip. The dashed lines in these graphs indicate the curve for random guessing. We find that naive Bayes has the highest AUC in most cases or close to it.

5. Discussion

In this work, we explored feature extraction from article headlines and text in an effort to classify fake and real news. Feature extraction methods included term frequency-inverse document frequency (td-idf) and sentiment analysis. Five statistical models were utilized, including naive Bayes, logistic regression, decision tree, random forest, and support vector machine. Four

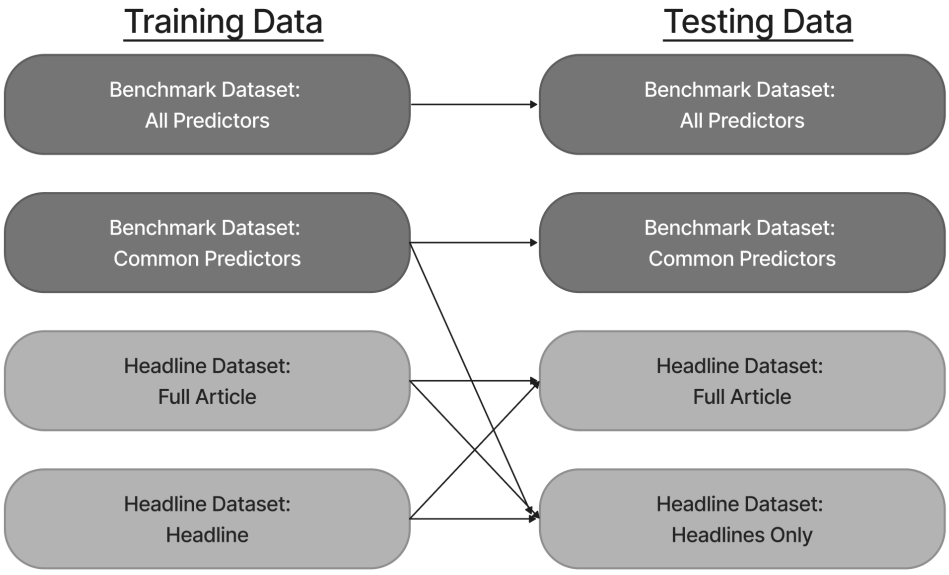

Figure 5: Approaches used to train and test the effectiveness of each model. The left column contains the training datasets and the right column contains the testing datasets. Arrows from the left to right columns indicate which portions of the datasets were tested using the corresponding training dataset.

Table 2: Accuracies for testing data utilizing the benchmark dataset as training data. Full Benchmark data uses the speaker's truthfulness history, Benchmark subset removes the truthfulness history, Headline data is tested on only the headlines. Bold values indicate the highest accuracy for each training-testing pair.

Training Set	Testing Set	Naive Bayes	Logistic Regression	Decision Tree	Random Forest	SVM
Benchmark: All Predictors	**Benchmark: All Predictors**	0.7040	0.6338	0.6851	**0.7245**	0.6109
Benchmark: Common Predictors	**Benchmark: Common Predictors**	**0.6156**	0.6093	0.5493	0.6125	0.5232
Benchmark: Common Predictors	**Headline**	**0.6440**	0.6025	0.5150	0.5480	0.4500

approaches were created to train and test both our benchmark and headline datasets. Approach one included training and testing on the benchmark dataset using all available predictors. Approach two included all common features from both datasets, and was tested on both the benchmark dataset and headlines from the headline dataset. Approach three consisted of training on the full article text from the headline dataset; while approach four consisted of training on the

Table 3: Accuracies for testing data utilizing the article text from the headline dataset as training data. Article Text indicates the testing subset from the full article, Headline data is tested on only the headlines. Bold values indicate the highest accuracy for each training-testing pair.

Training Set	Testing Set	Naive Bayes	Logistic Regression	Decision Tree	Random Forest	SVM
Full Article	**Full Article**	0.9683	0.9850	0.9917	**0.9967**	0.9737
Full Article	**Headline**	**0.9033**	0.8017	0.5017	0.5117	0.8683
Headline	**Full Article**	0.9106	**0.9133**	0.6094	0.6950	0.7342
Headline	**Headline**	**0.9256**	0.9133	0.8333	0.8306	0.8661

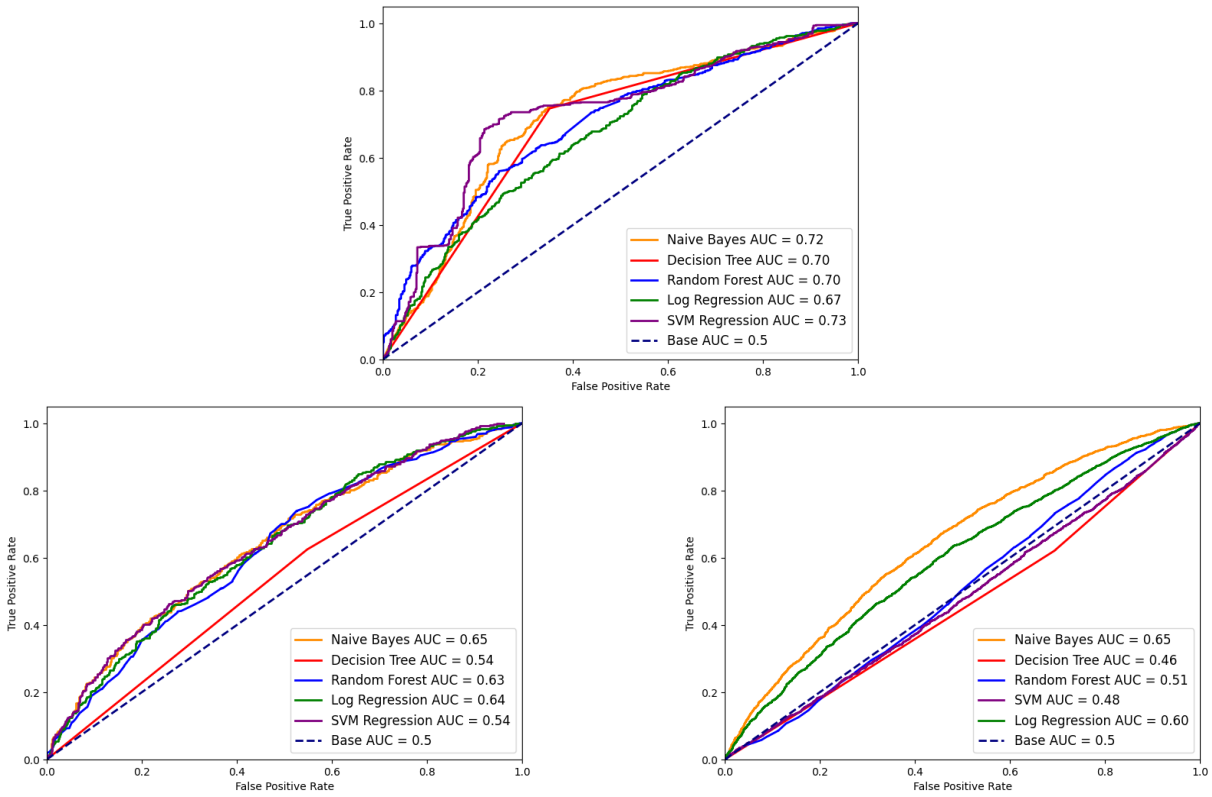

Figure 6: The top row contains the ROC curve for approach one–training and testing on the full benchmark dataset. The bottom row contains ROC curves for approach two–training on the common benchmark predictors. The bottom left is when tested on the benchmark subset, and the bottom right is when tested on the headlines from the headline dataset.

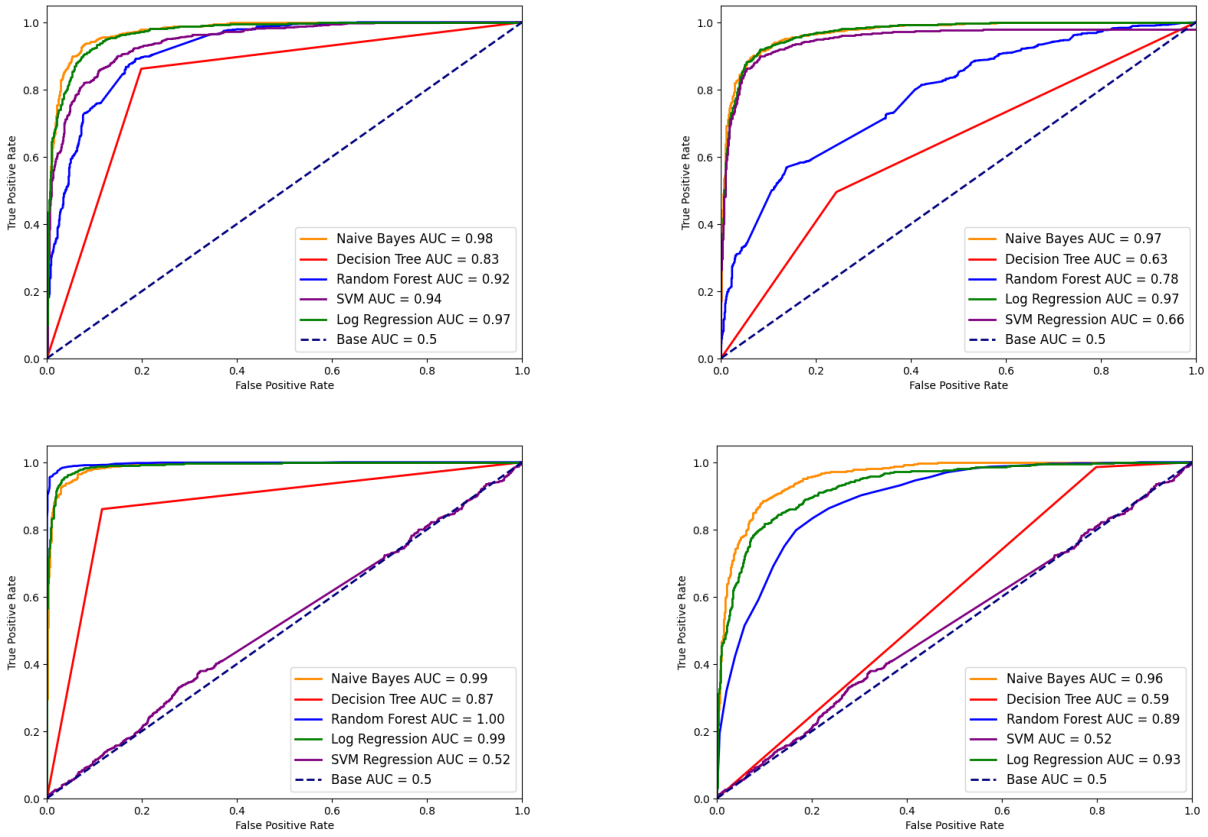

Figure 7: The top row is the ROC curves for approach 3–training on the full article text within the headline dataset. Top left is when tested on the full article text; top right is when tested on the headlines. The bottom row is the ROC curves for approach 4–training on the headlines within the headline dataset. Bottom left is when tested on full article text; bottom right is when tested on the headlines.

headlines from the headline dataset. Both approach three and four were tested on the full article text and the headlines.

Looking at previous studies of the same benchmark dataset, we can compare how well our models perform. Jehad & Yousif (2020) incorporated tf-idf into their features and produced 89.11% and 84.97% accuracies when combined with decision tree and random forest models. Yazdi et al. (2020)'s SVM approach to feature reduction showed an accuracy increase from 91.76% to 94.19% on the benchmark dataset. A partial explanation to our lower results in this scenario could be that we only utilized two categories while other literature used all six truth-categories. This was done to analyze how well our models generalize when tested on headlines, which only contained the two categories. However, the outcomes in Table 2 are not unexpected. As we removed features, especially those as important as speaker truthfulness history, we saw a large dip within all our model accuracies. The same results were seen when we decreased the number of document words by testing on the headline text.

On the other hand, approach three from Table 3 produced both high and low accuracies. When testing on the full article text all the models produced above 95% accuracy. However, when tested on the article headline, accuracies above 90% but as low as 50.17% were produced. When looking more closely at how the tree based models classified data we can see a reason for these drops. The decision tree model classified text from the headline data as false the majority of the time, while the random forest model would classify the same text as being legitimate the majority of the time. The issue for the low accuracy arose from the headline data itself. All the true news articles–which were collected from Reuters.com (Reuters Editorial, 2023)–contained the word "Reuters" within the full article text. Similarly the majority of the fake articles would contain the words "Getty Images" at the end of each article. The models were picking up on these specific words and creating short trees that would use these single words as indicators for classifying new data. Therefore, when testing on other full articles that follow similar patterns, the models would perform extremely well; however, when generalized to the headlines–which do not follow this pattern–the models would only take into account these specific words and not perform as well. To combat these issues, potential further work would include removing common words within each dataset that provide unintentional information about the validity of a new article.

In contrast, approach four did not have the same issues as approach three. The headlines did not have specific, misleading words that were present in all real or all fake news articles. Therefore, while the accuracies were slightly lower than approach three, the models generalized well and produced high accuracies when tested on both the full articles and the headlines. Although headlines are typically shorter than the full article text, the extracted features within them have the potential to be stronger indicators of whether a news article is legitimate or fake when given either the full text or only the headline.

We do acknowledge that these models and results have limitations. Both datasets require their fake news articles to be manually fact checked and labeled. Additionally, the legitimate news articles from the headline dataset are assumed to be real based on their news station. This introduces the possibility of error in cases where documents were labeled incorrectly, potentially leading to uncertainty within our accuracies. Moreover, tf-idf only factors single words and their relative frequencies within a document, and does not take into consideration the meaning of multiple words grouped together. Finally, our models only accounted for two categories: true and false. This required us to collapse the six original truthfulness categories, leading to some ambiguity regarding articles that contained both fake and true information.

Our research indicates that information extracted from article headlines can serve as valid predictors for classifying fake and legitimate news. Considering all four approaches and each models' accuracy, naive Bayes consistently achieved the highest performance or close to it. The reliability and capacity to generalize across various datasets position naive Bayes as a robust candidate for future endeavors, such as mobile applications or browser extensions. Another potential approach is to aggregate each models' result into a majority vote. This technique can provide a more holistic perspective by leveraging the collective insights from each model, potentially enhancing the overall robustness of the system. These outcomes can help keep individuals well informed and up to date on current information, even if they do not have access to the entire news article, such as in paid subscriptions. Potential further research in this area

include: testing and training the models on articles from different news stations, exploring models that account for multiple categories or provide results on a numerical scale, and examining models that output results as probabilities rather than category labels.

In summary, the ongoing evolution and refinement of predictive models for distinguishing between fake and legitimate news have the potential to significantly contribute to the dissemination of accurate and timely information. These advancements, coupled with the exploration of diverse datasets and the incorporation of probabilistic outputs, hold promise for enhancing the accessibility and reliability of news delivery through applications and browser extensions.

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. *Lecture Notes in Computer Science*, 10618, 127–138. https://doi.org/10.1007/978-3-319-69155-8_9

Carsten Stahl, B. (2006). On the difference or equality of information, misinformation, and disinformation: a critical research perspective. *Informing Science: The International Journal of an Emerging Transdiscipline, 9,* 083–096. https://doi.org/10.28945/473

Gottfried, J., & Shearer, E. (2019, December 10). News use across social media platforms 2016. *Apo.org.au*. https://apo.org.au/node/64483

Granik, M., & Mesyura, V. (2017). Fake news detection using naive Bayes classifier. *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON).* https://doi.org/10.1109/ukrcon.2017.8100379

Guarino, M. (2015, August 16). Misleading reports of lawlessness after Katrina worsened crisis, officials say. *The Guardian.* https://www.theguardian.com/us-news/2015/aug/16/hurricane-katrina-new-orleans-looting-violence-misleading-reports

Hotez, P. J. (2016). Texas and its measles epidemics. *PLOS Medicine, 13*(10), e1002153. https://doi.org/10.1371/journal.pmed.1002153

Hutto, C., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. https://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Iwendi, C., Mohan, S., khan, S., Ibeke, E., Ahmadian, A., & Ciano, T. (2022). Covid-19 fake news sentiment analysis. *Computers and Electrical Engineering, 101*, 107967. https://doi.org/10.1016/j.compeleceng.2022.107967

Jehad, R., & A.Yousif, S. (2020). Fake news classification using random forest and decision tree (J48). *Al-Nahrain Journal of Science, 23*(4), 49–55. https://doi.org/10.22401/anjs.23.4.09

Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021). Fake news detection using machine learning approaches. *IOP Conference Series: Materials Science and Engineering, 1099*(1), 012040. https://doi.org/10.1088/1757-899x/1099/1/012040

Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news. *Science*, *359*(6380), 1094–1096. https://doi.org/10.1126/science.aao2998

NLTK Project. (2009). Natural language toolkit — NLTK 3.4.4 documentation. *Nltk.org.* Retrieved September 7, 2023, from https://www.nltk.org/

Parkinson, H. J. (2016, November 14). Click and elect: how fake news helped Donald Trump win a real election | Hannah Jane Parkinson. *The Guardian.* https://www.theguardian.com/commentisfree/2016/nov/14/fake-news-donald-trump-election-alt-right-social-media-tech-companies

PolitiFact. (2019). Retrieved September 7, 2023, https://www.politifact.com/

Python. (2019, May 29). Python. *Python.org; Python.org*. Retrieved September 7, 2023, from https://www.python.org/

Rannard, G. (2020, January 7). Misleading maps of Australia fires go viral. *BBC News; BBC News*. https://www.bbc.com/news/blogs-trending-51020564

Reuters Editorial. (2019). Business & financial news, U.S & international breaking news | Reuters. *Reuters*. https://www.reuters.com/

Shao, C., Ciampaglia, G., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. https://www.andyblackassociates.co.uk/wp-content/uploads/2015/06/fakenewsbots.pdf

vaderSentiment. (2018, April 23). PyPI. Retrieved September 7, 2023, from https://pypi.org/project/vaderSentiment/

Wang, W. Y. (2017, July 1). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. *ACLWeb; Association for Computational Linguistics.* https://doi.org/10.18653/v1/P17-2067

Yazdi, K. M., Yazdi, A. M., Khodayi, S., Hou, J., Zhou, W., & Saedy, S. (2020). Improving fake news detection using k-means and support vector machine approaches. *International Journal of Electronics and Communication Engineering, 14*(2), 38–42. https://publications.waset.org/10011058/improving-fake-news-detection-using-k-means-and-support-vector-machine-approaches

Appendix

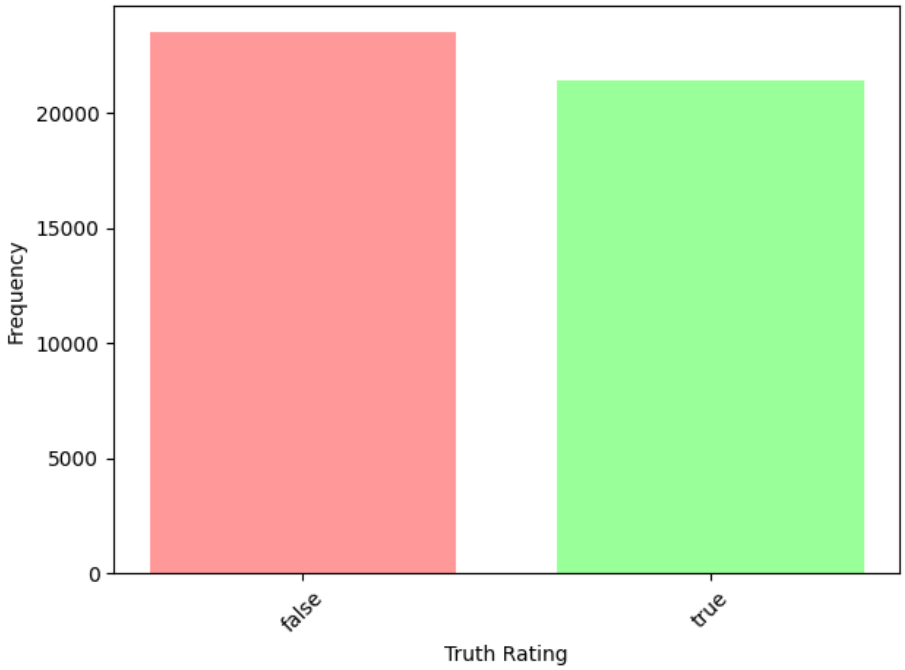

Figure 8: Truth rating frequency from the headline dataset.

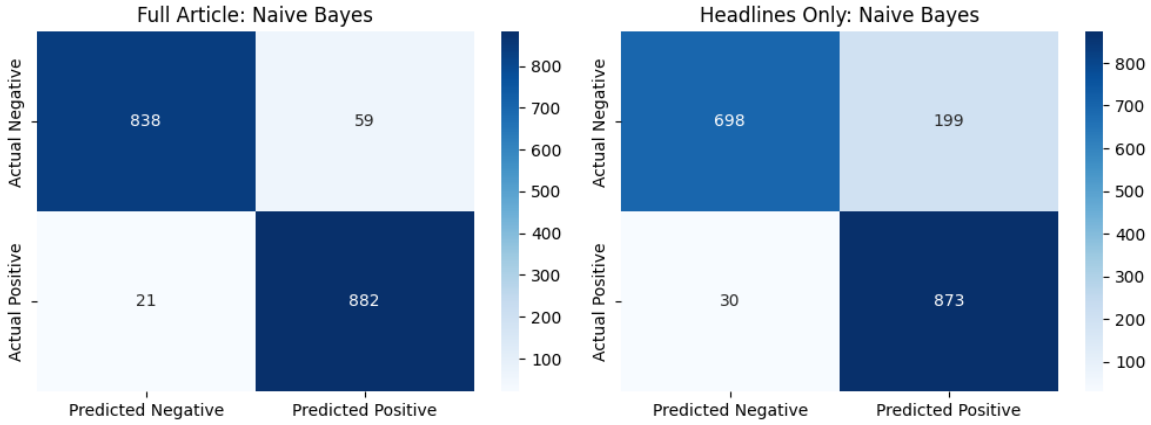

Figure 9: Confusion matrices for the naive Bayes model when trained on the full article text. The left was tested on the full article and the right was tested on the headlines.

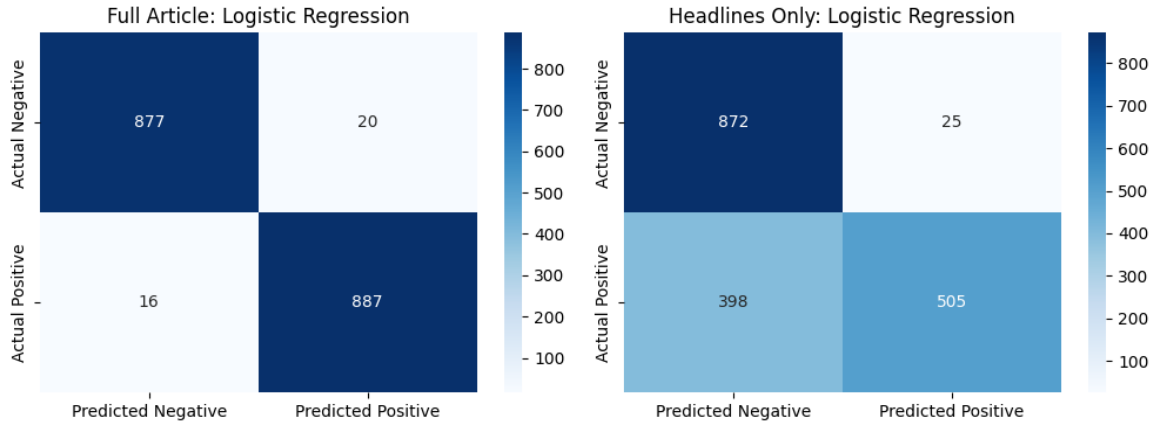

Figure 10: Confusion matrices for the logistic regression model when trained on the full article text. The left was tested on the full article and the right was tested on the headlines.

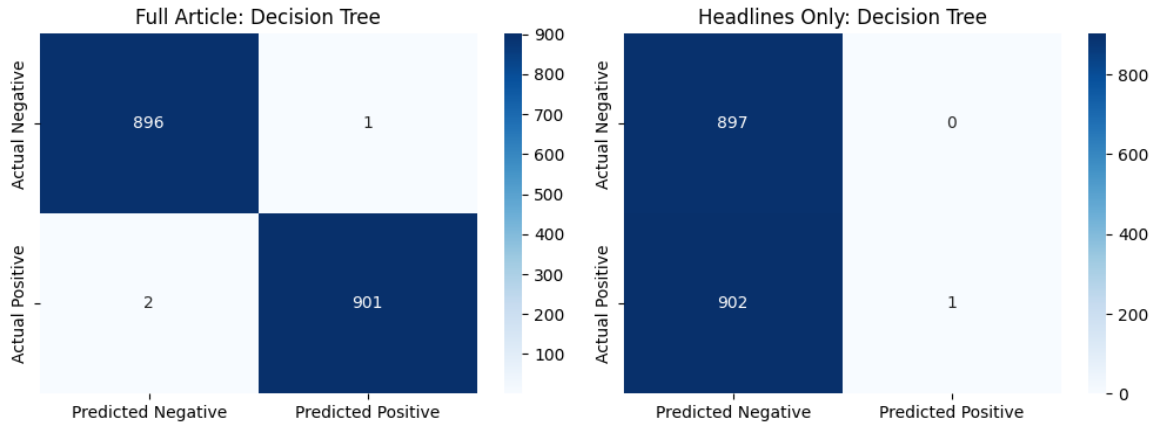

Figure 11: Confusion matrices for the decision tree model when trained on the full article text. The left was tested on the full article and the right was tested on the headlines.

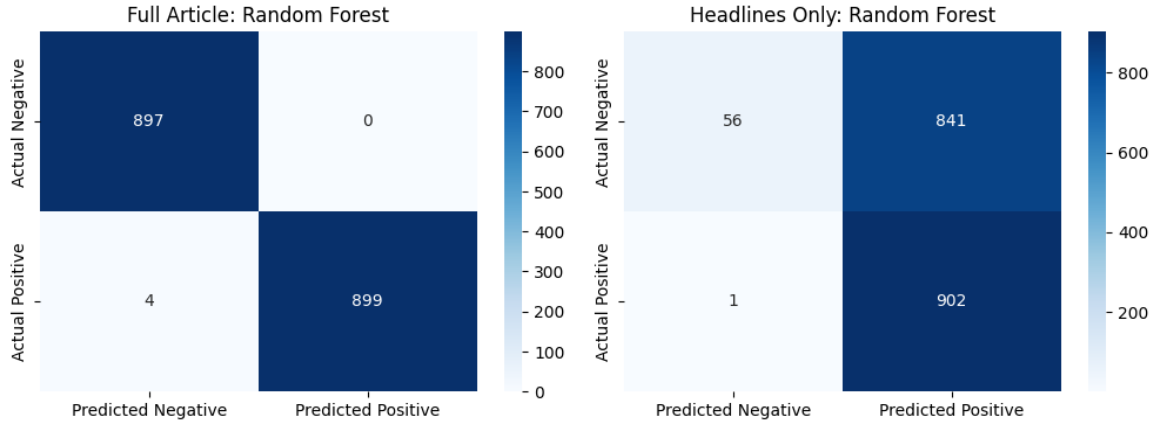

Figure 12: Confusion matrices for the random forest model when trained on the full article text. The left was tested on the full article and the right was tested on the headlines.

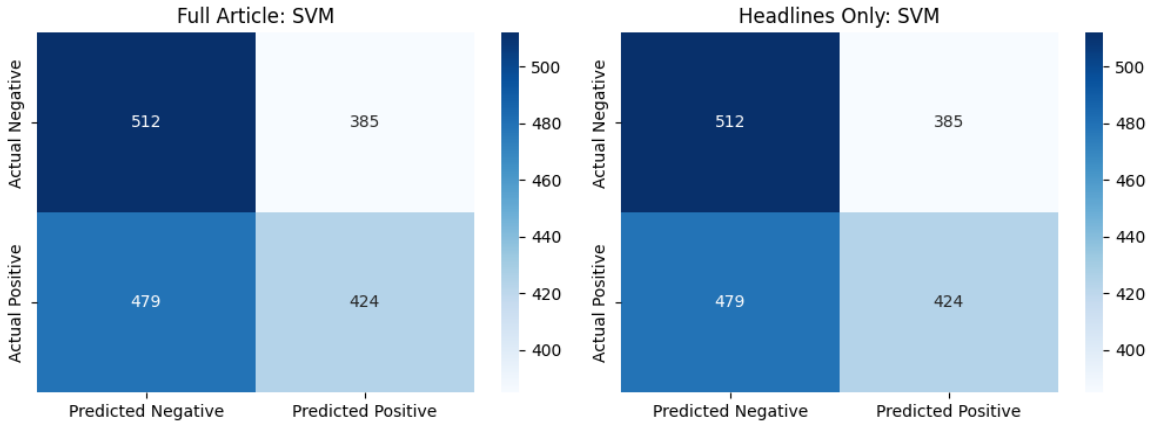

Figure 13: Confusion matrices for the support vector machine model when trained on the full article text. The left was tested on the full article and the right was tested on the headlines.