

Article

Not peer-reviewed version

Comparative Evaluation of Logistic Regression and Naïve Bayes for Fake News Detection Using NLP Techniques

[Noor Amin](#) *

Posted Date: 9 December 2025

doi: 10.20944/preprints202512.0630.v1

Keywords: fake news detection; machine learning; logistic regression; Naïve Bayes; natural language processing (NLP); text classification; TF-IDF; data preprocessing; misinformation detection; feature extraction; model evaluation; digital media integrity; artificial intelligence



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Comparative Evaluation of Logistic Regression and Naïve Bayes for Fake News Detection Using NLP Techniques

Noor Ul Amin

Taylor's University, Malaysia; nooraminnawab@gmail.com

Abstract

Detection of fake news is an important task in this digital era, where misinformation is being spread everywhere. This paper utilizes the WELFake dataset, which contains 72,134 news articles, with 35,028 labeled as real and 37,106 labeled as fake. This dataset is an aggregation of four famous sources: Kaggle, McIntire, Reuters, and BuzzFeed Political. This dataset combines data from several sources with the aim of making classifiers more robust and preventing overfitting. Each entry consists of a serial number, the title of the news, the content of the article, and a label describing the news as real (1) or fake (0). While the original CSV file includes 78,098 entries, this provided data frame focuses on 72,134 entries to maintain data quality and relevance for machine learning tasks. This study implements two machine learning algorithms: Logistic Regression and Naive Bayes. The Logistic Regression performed the best with an accuracy of 94.53% while having balanced precision, recall, and F1-scores of 0.95 for both the real and fake news classes. The Naive Bayes classifier gave an accuracy of 84.72% but had poor F1-scores since it operates under the assumption of independence of features. The preprocessing steps included cleaning, tokenization, and lemmatization, after which feature extraction was performed by using TF-IDF. Although it has given good results, this baseline points to the advantage of Logistic Regression in high-dimensional spaces. This study has been published in IEEE Transactions on Computational Social Systems and underlines the synergy of machine learning and deep text analysis in fighting against fake news. It provides a vision for future improvements in classification methods and detection systems in real-time to develop a reliable and trustworthy information ecosystem for further research and applications.

Keywords: fake news detection; machine learning; logistic regression; Naïve Bayes; natural language processing (NLP); text classification; TF-IDF; data preprocessing; misinformation detection; feature extraction; model evaluation; digital media integrity; artificial intelligence

1.0. Introduction

In today's digital age, the rise of fake news has become one of the most pressing challenges confronting society[1]. This phenomenon poses a significant threat not only to established media credibility but also to the informed decision-making of the public. The sheer volume of false information circulating on social media networks and various online news platforms can distort perceptions, manipulate opinions, and even influence crucial democratic processes[2,3].

The rapid spread of misinformation is often exacerbated by the ease with which it can be shared and amplified through digital channels. As users engage with sensationalized or misleading content, the integrity of reliable news sources is progressively undermined. This has created an urgent need for effective strategies to identify, detect, and counteract the escalation of misleading narratives[4].

Natural Language Processing (NLP) stands out as one of the most promising avenues for tackling the issue of fake news. By leveraging sophisticated algorithms and machine learning techniques, such as logistic regression, NLP can analyze vast amounts of textual data to discern underlying patterns indicative of false information. This analytical approach involves examining

various linguistic features—such as word choice, sentence structure, and contextual cues—that differentiate genuine news articles from fabricated ones[5–9].

Furthermore, by training machine learning models on labeled datasets of real and fake news, NLP systems can improve their accuracy over time, becoming increasingly adept at recognizing the subtle nuances of deceptive content[10]. These advancements not only empower media platforms and fact-checking organizations to mitigate the spread of misinformation but also enhance the public's awareness and understanding of the importance of critically assessing the news they consume[11–13].

1.1. Research Goal

The primary objective of this research study is to create a robust machine learning model capable of accurately identifying and categorizing news articles as either genuine or fabricated. This will be achieved by leveraging advanced Natural Language Processing (NLP) techniques to analyze both the textual content and contextual information surrounding the articles. Additionally, the study will employ Logistic Regression methods to enhance classification accuracy. By integrating these approaches, the model aims to effectively differentiate between real and fake news, thereby contributing valuable insights to the ongoing efforts to combat misinformation in digital media[14].

2.0. Objectives

The primary goal of this study is designing and training a machine-learning-enabled natural language processing model for real and fake news classification. In summary, this would include:

Making Text Data Ready: Cleaning, Tokenizing and Lemmatizing textual content in order to have standardized input data for machine learning models.

Extracting Meaningful Features: Use of advanced feature extraction methods as Term Frequency-Inverse Document Frequency (TF-IDF) which captures linguistic and contextual patterns differentiating fake news from real news. The features will be later analyzed in data visualization.

Model Development and Evaluation: Constructing the machine-learning models, especially Logistic Regression, to classify the data correctly and benchmark the performance of the selected one against other models, reflecting the reliability and robustness of that model. We will also compare the performances of both the models using the performance metrics and evaluate them.

Counter Organization of Misinformation: Development of a solution that would be accurate as well as scalable to contribute to the reduction of blunt disinformation in the process of strengthening integrity in digital media platforms.

Accomplishing these goals will allow this research to offer a highly applicable and effective method against fake news that leverages the potentialities of NLP and machine learning for an increasingly trusted information ecosystem.

3.0. Problem Identification

Fake news is spreading more widely in the current digital era due to the quick distribution of information via online channels. Fake news has the power to disturb society, sway public opinion, and mislead audiences[15]. Given the amount of news produced every day, the conventional techniques of fact-checking and manual verification are not scalable, highlighting the need for automated alternatives[16–19].

Even with data available, spotting fake news is still difficult because:

- Fake news frequently imitates real articles in terminology, tone, and structure, it can be challenging to tell them apart based on superficial characteristics.
- A model that can effectively handle massive datasets and give real-time categorization is required due to the abundance of internet news.
- Accurate classification of textual material depends on the ability to recognize significant patterns and contextual hints.

- High precision and recall are difficult for many of the models that are currently in use, especially when handling ambiguous data and a variety of writing styles.

4.0. Dataset Description

Overview:

The WELFake dataset comprises 72,134 labeled news articles, with 35,028 real and 37,106 fake news entries. To improve generalizability, it aggregates data from several sources, such as BuzzFeed Political, Reuters, Kaggle, and McIntire. The description of the important properties of the dataset is given below:

Key Attributes:

1. **Rows and Columns:** This dataset has **72134** rows and **4** columns.
2. **Format:** The data may be easily preprocessed and used in machine learning pipelines because it is supplied in a CSV file format.
3. The 4 columns of the dataset are as follows:
id: Unique identifier for each entry. **title:** Headline of the news article. **text:** Content of the news article.
Label: Label is the target variable and the binary indicator.
0 = fake news
1 = real news.

Features and Importance:

Title and Text: The main informational resources for spotting false news. These domains offer contextual and linguistic clues that are essential for analysis.

Label: The binary classification goal, which makes it possible to detect bogus news using supervised learning.

Characteristics:

- A variety of writing styles are included in the collection, representing a range of news story forms and sources.
- Some columns, like Title has **558** missing values and text has **39** missing values, which presents preprocessing difficulties but also chances to create strong solutions.
- The dataset is ideal for evaluating the efficacy of NLP algorithms since it includes differences in language, tone, and structure.

Relevance to the Problem:

The "Fake News Classification" dataset available on Kaggle is exceptionally pertinent for addressing the significant challenge of detecting fake news in the current digital landscape. This dataset comprises a meticulously organized collection of labeled data that includes a wide variety of articles, distinctly categorized as either fake or genuine news. Such a structured compilation enables researchers and practitioners to apply advanced machine learning techniques, including but not limited to Logistic Regression and other classification algorithms[20–22].

One of the key advantages of this dataset is its capacity to facilitate in-depth analysis of various linguistic patterns, contextual clues, and structural characteristics that differentiate authentic articles from deceptive ones. This analysis is vital for overcoming common challenges in fake news detection, particularly the tendency of fake articles to closely mimic the tone, style, and format of legitimate news sources. By focusing on these nuanced elements, researchers can develop more sophisticated models capable of discerning subtle differences that often escape casual observation[23].

Additionally, the dataset's substantial size and inherent diversity are critical assets for training robust machine learning models. With the sheer volume of news articles generated daily online, having a rich dataset not only enhances the model's ability to learn from a broad spectrum of writing styles but also improves its performance in real-time classification scenarios[24,25]. This is especially important for news platforms and social media where immediate detection of misinformation is crucial. By effectively leveraging this dataset, the study aims to concentrate on creating a highly efficient model that achieves elevated levels of precision and recall[26,27].

5.0. Supportive Literature

Logistic Regression and Naive Bayes are widely recognized because they provide good performance on the fake news classification, as both are simple, interpretable models. For example, Lai et al. (2022) conducted research where these models were implemented to classify the news into fake or real on a dataset of 24,204 fake and 22,970 true news articles, sourced from Kaggle and through web scraping. The dataset was generated between 2015 and 2018 with four structured columns.

Following are the preprocessing steps being taken by the authors:

Data Cleaning: Articles lacking textual content or consisting solely of images were excluded to maintain a high-quality dataset.

Tokenization: Using NLTK, the articles were tokenized down to individual words to make text analysis easier in later stages.

TF-IDF Calculation: Term Frequency-Inverse Document Frequency measures gave the magnitude of importance of words in the dataset to help in discriminating between relevant terms for classification.

After Preprocessing the dataset, both the logistic regression and Naïve Bayes models were being trained and evaluated on the testing set as shown below:

Table 1. Supportive Literature Metrics.

Metrics	Logistic Regression	Naive Bayes
Accuracy	88.8%	85.3%
Precision (0)	0.89	0.85
Precision (1)	0.89	0.86
Recall (0)	0.88	0.86
Recall (1)	0.89	0.85
F1-score (0)	0.89	0.85
F1-score (1)	0.89	0.86

These metrics show the strengths of Logistic Regression and Naive Bayes in the given classification problem with Logistic Regression slightly managing to perform well as compared to Naive Bayes.

6.0. Problem-Solving Approach

6.1. Exploratory Data Analysis (EDA)

The following analyses were conducted to gain an insight into the structure and distribution of the dataset:

Overview of Dataset Using df.info()

It showed that there are 72,134 entries in the dataset with four columns: Unnamed: 0, title, text, and label.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72134 entries, 0 to 72133
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Unnamed: 0     72134 non-null  int64
 1   title          71576 non-null  object
 2   text           72095 non-null  object
 3   label          72134 non-null  int64
dtypes: int64(2), object(2)
memory usage: 2.2+ MB
```

Figure 1. Overview of Data Completeness and Column Types.

Columns title and text are of textual data type, while label represents the classification in terms of 0-fake and 1-real.

Observation:

The title and text columns had missing values which required handling to ensure data integrity.

Statistical Summary Using df.describe()

```
df.describe()


```

	Unnamed: 0	label
count	72134.000000	72134.000000
mean	36066.500000	0.514404
std	20823.436496	0.499796
min	0.000000	0.000000
25%	18033.250000	0.000000
50%	36066.500000	1.000000
75%	54099.750000	1.000000
max	72133.000000	1.000000

Figure 2. Statistical Summary of Dataset Attributes.

This provided insights into the numerical column, label.

Observation: The label column is roughly balanced, with about 51.4% real and 48.6% fake news, making it ideal for supervised learning without requiring significant rebalancing.

Countplot Visualization

A countplot was created to visualize the distribution of real and fake news articles. Interpretation: Countplot confirmed that the two classes were roughly equally distributed. This balanced dataset ensures that the model training process is unbiased and does not favor one class over the other.

```
#Creating a count plot to show the distribution of real and fake news.
plt.figure(figsize=(8, 6))
sns.countplot(x='label', data=df, palette=['red', 'blue']) # Use a palette with two colors
plt.title('Concentration of real and fake news')
plt.xlabel('Label')
plt.ylabel('Count')
plt.show()
```

Figure 3. Python Code for Visualizing Label Distribution.

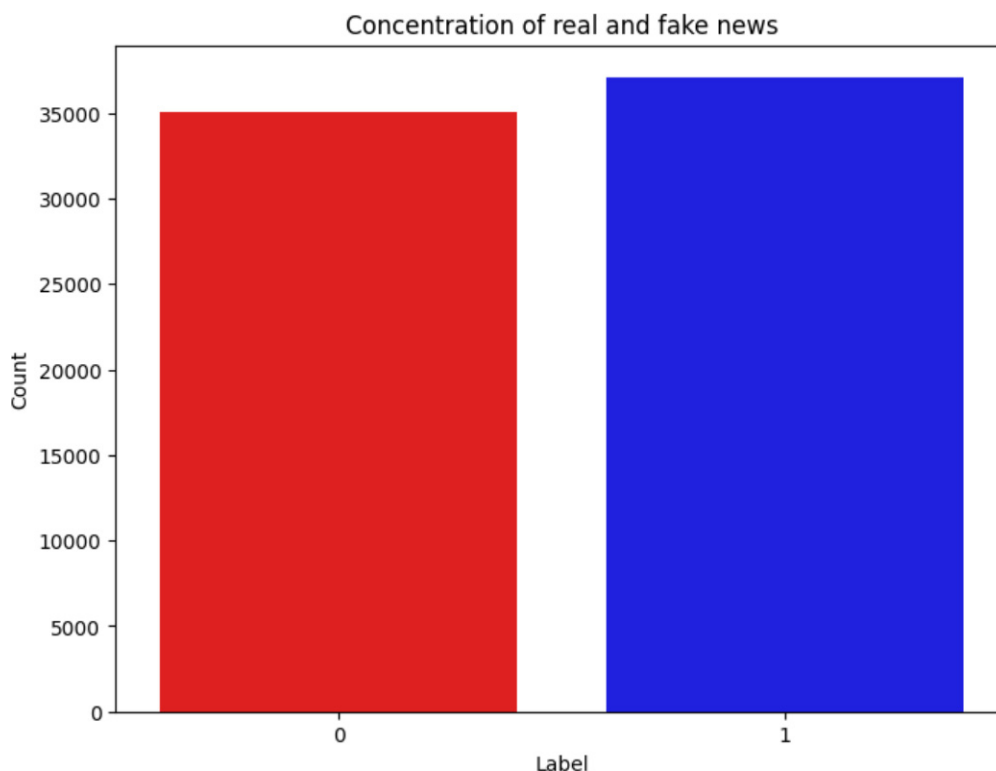


Figure 4. Class Distribution of Real vs. Fake News.

Cloud Visualization

Word clouds show word frequencies; the size of each word is indicative of how many times a word appeared in both the "Real News" and "Fake News" articles.

Some common keywords for both word clouds include words like "Donald Trump," "United States," "White House," "Said," and "Country," which point toward the subjects both focus on, largely being political or governmental themes.

This indicates that both the real and fake news articles are centred around similar high-profile figures or themes as shown below:

```
# Wordcloud for real and fake news
real_news = df[df['label'] == 0]['content']
wordcloud_real = WordCloud(width=800, height=400, background_color='white').generate(' '.join(real_news))
plt.figure(figsize=(10, 5), facecolor=None)
plt.imshow(wordcloud_real)
plt.axis("off")
plt.title("Word Cloud for Real News")
plt.show()

fake_news = df[df['label'] == 1]['content']
wordcloud_fake = WordCloud(width=800, height=400, background_color='white').generate(' '.join(fake_news))
plt.figure(figsize=(10, 5), facecolor=None)
plt.imshow(wordcloud_fake)
plt.axis("off")
plt.title("Word Cloud for Fake News")
plt.show()
```



```

#Calculating word frequencies
real_words = ' '.join(df[df['label'] == 1]['content']).split() # 1 is real
fake_words = ' '.join(df[df['label'] == 0]['content']).split() # 0 is fake

real_word_counts = Counter(real_words)
fake_word_counts = Counter(fake_words)

# Find common words
common_words = list(set(real_word_counts.keys()) & set(fake_word_counts.keys()))

# Create a matrix for the heatmap
heatmap_data = pd.DataFrame(index=common_words, columns=['Real', 'Fake'])

for word in common_words:
    heatmap_data.loc[word, 'Real'] = real_word_counts[word]
    heatmap_data.loc[word, 'Fake'] = fake_word_counts[word]

# Convert data to numeric (Counter values are already numeric, but ensure dtype consistency)
heatmap_data = heatmap_data.fillna(0).astype(int)

# Create the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data.head(20), annot=True, cmap='viridis', fmt='g') # Show top 20 words
plt.title('Word Frequency Comparison (Real vs. Fake News)')
plt.xlabel('News Type')
plt.ylabel('Word')
plt.show()

```

Figure 8. Generating a Heatmap to Compare Word Distributions in News Data.

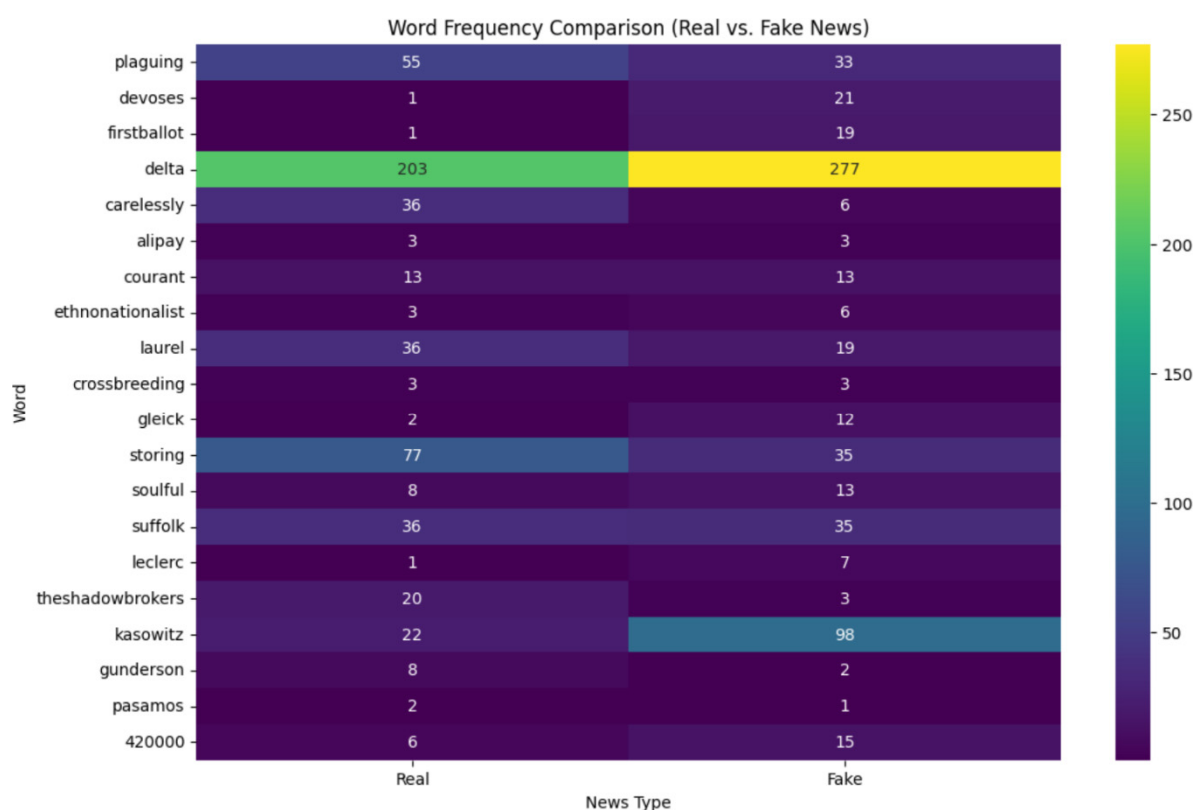


Figure 9. Heatmap.

6.2. Data Preprocessing Steps

Handling Missing Values

Missing values in the title and text columns were replaced with empty strings to avoid issues during feature engineering.

```
#Replace missing values with empty strings
df.fillna('', inplace=True)
```

Figure 10. Data Preprocessing Step to Replace Missing Entries.

Text Preprocessing

A custom preprocessing function was defined and applied to clean and standardize the text data.

```
#Function to preprocess textual data

def preprocess_text(text):
    #Convert to lowercase
    text = text.lower()

    #Remove punctuation
    text = ''.join([char for char in text if char not in string.punctuation])

    #Tokenization using NLTK's word_tokenize
    words = word_tokenize(text)

    #Remove stopwords
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]

    #Lemmatization
    lemmatizer = WordNetLemmatizer()
    words = [lemmatizer.lemmatize(word) for word in words]

    return ' '.join(words)
```

Figure 11. Preprocessing Function for NLP-Based Fake News Classification.

This included:

- Converting all text to lowercase.
- Removing punctuation and special characters.
- Tokenizing the text into individual words.
- Removing stop words to retain meaningful information.
- Lemmatizing words to reduce them to their root form.

Reason: These steps ensure that the text data is clean, standardized, and ready for feature extraction, reducing noise in the dataset.

6.3. Feature Engineering

Combining Title and Text Columns

A new column, content, was created with the merged title and text columns. This combined column will be the main input to the model.

```
#Combining 'title' and 'text' columns into one
df["content"] = df["title"] + " " + df["text"]

#Dropping Serial Number column named Unnamed: 0
df.drop('Unnamed: 0', axis=1, inplace=True)
df.head()
```

	title	text	label	content
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1	LAW ENFORCEMENT ON HIGH ALERT Following Threat...
1		Did they post their votes for Hillary already?	1	Did they post their votes for Hillary already?
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0	Bobby Jindal, raised Hindu, uses story of Chri...
4	SATAN 2: Russia unveils an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1	SATAN 2: Russia unveils an image of its terrif...

Figure 12. Python Code for Merging Title and Text Columns.

Titles and texts are often complementary pieces of information. Combining them gives the model a richer context with which to learn better patterns and relationships for classification.

Preprocessing the 'Content' Column

This was applied to the content column as a preprocessing function to make sure the merged text data is clean and ready for feature extraction.

6.4. Splitting Dataset and Converting Text Data into TF-IDF Features

Further, the dataset was split into training and testing sets in a 70:30 ratio, respectively, to ensure enough data for both training and evaluation.

```
#Selecting Feature to target label.
X = df['content']
y = df['label']

#Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

#Convert text into TF-IDF features
tfidf_vectorizer = TfidfVectorizer(max_features=5000) #Limit to top 5000 features
```

Figure 13. Splitting Dataset and Converting Text into TF-IDF Features.

TF-IDF was a method used for converting textual data into numerical features. It highlights the importance of certain words in a document while diminishing the weight of words that are very frequent within the dataset.

```
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

Figure 14. Converting Text Data into Numerical Feature Vectors Using TF-IDF.

Reason: TF-IDF is ideal for classification tasks in text because it captures the relevance of words, hence providing a meaningful representation for machine learning algorithms.

6.5. Model Choice

Logistic Regression

- Reason for Choice:
Logistic Regression is a robust yet interpretable model, hence very suitable for binary classification tasks such as identifying fake news.
It performs well on high-dimensional data like the TF-IDF feature space and is computationally efficient.

```

▶ #Train Logistic Regression model
lr_model = LogisticRegression()
lr_model.fit(X_train_tfidf, y_train)

#Predict on test data
lr_y_pred = lr_model.predict(X_test_tfidf)

#Evaluate the model
lr_accuracy = accuracy_score(y_test, lr_y_pred)
print(f"Logistic Regression Accuracy: {lr_accuracy * 100:.2f}%")
print("Logistic Regression Classification Report:")
print(classification_report(y_test, lr_y_pred))

```

```

↳ Logistic Regression Accuracy: 94.53%
Logistic Regression Classification Report:

```

	precision	recall	f1-score	support
0	0.95	0.94	0.94	10613
1	0.94	0.95	0.95	11028
accuracy			0.95	21641
macro avg	0.95	0.95	0.95	21641
weighted avg	0.95	0.95	0.95	21641

Figure 15. Fake News Detection Results Using Logistic Regression.

Naive Bayes

- Reason for choice:
Naive Bayes is particularly suited for text classification tasks due to its probabilistic nature and ability to handle sparse data efficiently.
This serves as a great baseline model—a fast, reliable benchmark against which more complex algorithms can be compared.

```

[1] # Train Naive Bayes model
nb_model = MultinomialNB()
nb_model.fit(X_train_tfidf, y_train)

# Predict on test data
nb_y_pred = nb_model.predict(X_test_tfidf)

# Evaluate the model
nb_accuracy = accuracy_score(y_test, nb_y_pred)
print(f"Naive Bayes Accuracy: {nb_accuracy * 100:.2f}%")
print("Naive Bayes Classification Report:")
print(classification_report(y_test, nb_y_pred))

```

```

↳ Naive Bayes Accuracy: 84.72%
Naive Bayes Classification Report:

```

	precision	recall	f1-score	support
0	0.86	0.82	0.84	10613
1	0.84	0.87	0.85	11028
accuracy			0.85	21641
macro avg	0.85	0.85	0.85	21641
weighted avg	0.85	0.85	0.85	21641

Figure 16. Naive Bayes Model Training and Evaluation Results.

7.0. Performance Evaluation for the Models

In the quest to accurately distinguish between false and genuine news articles, the logistic regression model emerged as a notably effective tool. Its ability to classify news content correctly was evidenced by balanced performance metrics, which highlighted its reliability across various criteria. With an impressive accuracy rate of 94.53%, the chosen feature selection methods and preprocessing techniques played a significant role in achieving such high levels of precision. This success underscores the importance of carefully selecting relevant features and rigorously preparing the data to enhance the model's overall performance in the challenging realm of news verification. A critical evaluation for the performance of both the models are given below:

Accuracy: Accuracy measures the proportion of correctly classified instances out of the total instances given. In this report, we can see that logistic regression shows an accuracy of **94.53%** whereas naive bayes shows an overall accuracy of **84.72%**. Logistic Regression demonstrates a substantial advantage over Naive Bayes when it comes to overall accuracy in classifying news articles. This means that Logistic Regression is better equipped to correctly identify both real news and fake news stories, leading to fewer misclassifications. The effectiveness of Logistic Regression can be attributed to its ability to analyze and model the relationship between input features and the target variable, allowing for more nuanced decision-making when distinguishing between these two categories of news.

Precision: Precision measures the ratio of correctly predicted positive observations to the total predicted positive observations. In this report, logistic regression 95% and naive bayes shows 86%.

Class 0:

- Logistic Regression: 0.95
- Naive Bayes: 0.86

Class 1:

- Logistic Regression: 0.94
- Naive Bayes: 0.84

Logistic Regression consistently demonstrates superior performance compared to Naive Bayes across both classes in the analysis. This finding suggests that Logistic Regression is more adept at minimizing the occurrence of false positives, which is critical in ensuring that instances are correctly classified. The ability of Logistic Regression to handle the underlying relationships in the data more effectively enhances its reliability in various predictive modeling scenarios.

Recall: Recall measures the proportion of actual positive instances that are correctly predicted by the model.

- **Class 0:**
 - Logistic Regression: 0.94
 - Naive Bayes: 0.82
- **Class 1:**
 - Logistic Regression: 0.95
 - Naive Bayes: 0.87

Logistic Regression demonstrates a significantly higher recall for both classes when compared to Naive Bayes. This means that the Logistic Regression model is more effective at identifying true positives, which are cases that are correctly classified as belonging to a particular class. As a result, it captures a larger proportion of the actual positive instances in the dataset, thereby reflecting its superior ability to minimize false negatives. In contrast, Naive Bayes tends to miss a greater number of these true positive cases, leading to lower recall rates for both classes. Overall, this suggests that Logistic Regression may be the more reliable choice in scenarios where accurately identifying positive cases is critical.

F-1 Score:

F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two metrics.

- **Class 0:**
 - Logistic Regression: 0.94
 - Naive Bayes: 0.84
- **Class 1:**
 - Logistic Regression: 0.95
 - Naive Bayes: 0.85

Logistic Regression demonstrates a notably higher F1-score for both classes when evaluated. This improvement suggests a more effective balance between precision and recall, which is particularly crucial in contexts where the implications of false positives and false negatives carry significant costs. A higher F1-score indicates that the model is not only accurately identifying positive instances but is also minimizing incorrect classifications, thus providing a more reliable performance in scenarios where the stakes are elevated. The balanced approach of Logistic Regression makes it a preferred choice for applications requiring careful consideration of both types of errors. A table showing their comparisons is given below:

Table 2. Comparison Metrics.

Metrics	Logistic Regression	Naive Bayes
Accuracy	94.53%	84.72%
Precision (0)	0.95	0.86
Precision (1)	0.94	0.84
Recall (0)	0.94	0.82
Recall (1)	0.95	0.87
F1-score (0)	0.94	0.84
F1-score (1)	0.95	0.85

Overall comparison shows that logistic regression performs better for this dataset than Naive Bayes.

8.0. Conclusion

The goal of this report was to investigate the capabilities of machine learning models in identifying the differences in real and fake news. Our goal was to use Logistic regression and Naïve Bayes to accurately identify this problem. To address class imbalances and enhance model performance, the study used a variety of preprocessing methods. Our analysis revealed several key findings:

1. **Pre-processing:** After using various pre-processing techniques such as 'text pre-processing' and feature extraction techniques such as TF-IDF we were able to monitor the performance of the models. Due to the pre-processing techniques logistic regression showed an accuracy of 94.53%.
2. **Performance of Logistic Regression:** The logistic regression model demonstrated impressive performance, achieving a very high accuracy score. In addition to accuracy, it also showed strong results in key evaluation metrics including precision, recall, and F1 score. This indicates that the model not only made correct predictions most of the time, but also effectively minimized false positives and false negatives, providing a well- rounded assessment of its predictive

capabilities. Overall, the logistic regression approach proved to be a reliable choice for the given analysis.

3. **Implications:** Using machine learning models to distinguish between fake and true news has important ramifications for a number of industries. It aids in the fight against false information in the media and journalism sector, maintaining news organizations' legitimacy and confidence. Assuring responsible information transmission, it helps social media platforms stop the spread of bogus news. It protects democratic processes in the political sector by avoiding the use of misleading narratives to sway public opinion. Public health deals with the dissemination of false information about health, particularly in times of crisis such as pandemics. Additionally, it fosters media literacy in the education sector by offering resources for information verification and critical analysis. These industries may safeguard social integrity, improve decision-making, and promote an informed and accountable society by utilizing machine learning for fake news detection.

In conclusion, this study demonstrates how machine learning may be applied to detect false information by providing a reliable early detection and classification technique.

Preprocessing techniques combined with Naive Bayes and logistic regression models offer a powerful tool for understanding and combating the spread of false information. The logistic regression model's superior performance demonstrates how well-suited it is for this task and makes it a crucial weapon in the fight against the global spread of false information.

References

1. State, P. (2019, November 18). *Researchers identify seven types of fake news, aiding better detection*. Lab Manager. <https://www.labmanager.com/researchers-identify-seven-types-of-fake-news-aiding-better-detection-477>
2. *Fake news classification*. (2023, October 8). Kaggle. <https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification?resource=download>
3. Lai, C.-M., Chen, M.-H., Kristiani, E., Verma, V.K. and Yang, C.-T., 2022. Fake News Classification Based on Content Level Features. *Applied Sciences*, [online] 12(3), p.1116. Available at: <https://doi.org/10.3390/app12031116>.
4. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1), 22-36.
5. Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5), 1-40.
6. Hu, B., Mao, Z., & Zhang, Y. (2025). An overview of fake news detection: From a new perspective. *Fundamental Research*, 5(1), 332-346.
7. Xu, X., Yu, P., Xu, Z., & Wang, J. (2025, January). A hybrid attention framework for fake news detection with large language models. In 2025 5th International Conference on Neural Networks, Information and Communication Engineering (NNICE) (pp. 587-590). IEEE.
8. Ashfaq, F., Jhanjhi, N. Z., Khan, N. A., Javaid, D., Masud, M., & Shorfuzzaman, M. (2025). Enhancing ECG Report Generation With Domain-Specific Tokenization for Improved Medical NLP Accuracy. *IEEE Access*.
9. Zhang, L., Zhang, X., Zhou, Z., Zhang, X., Yu, P. S., & Li, C. (2025). Knowledge-aware multimodal pre-training for fake news detection. *Information Fusion*, 114, 102715.
10. Faisal, A., Jhanjhi, N. Z., Ashraf, H., Ray, S. K., & Ashfaq, F. (2025). A Comprehensive Review of Machine Learning Models: Principles, Applications, and Optimal Model Selection. *Authorea Preprints*.
11. Shah, I. A., Jhanjhi, N. Z., Amsaad, F., & Razaque, A. (2022). The role of cutting-edge technologies in industry 4.0. In *Cyber Security Applications for Industry 4.0* (pp. 97-109). Chapman and Hall/CRC.
12. Attaullah, M., Ali, M., Almufareh, M. F., Ahmad, M., Hussain, L., Jhanjhi, N., & Humayun, M. (2022). Initial stage COVID-19 detection system based on patients' symptoms and chest X-ray images. *Applied Artificial Intelligence*, 36(1), 2055398.
13. Xu, C., & Yan, N. (2025, July). TripleFact: Defending data contamination in the evaluation of LLM-driven fake news detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 8808-8823).

14. Brohi, S. N., Jhanjhi, N. Z., Brohi, N. N., & Brohi, M. N. (2023). Key applications of state-of-the-art technologies to mitigate and eliminate COVID-19. *Authorea Preprints*.
15. Khandelwal, M., Rout, R. K., Umer, S., Sahoo, K. S., Jhanjhi, N. Z., Shorfuzzaman, M., & Masud, M. (2023). A Pattern Classification Model for Vowel Data Using Fuzzy Nearest Neighbor. *Intelligent Automation & Soft Computing*, 35(3).
16. Gouda, W., Sama, N. U., Al-Waakid, G., Humayun, M., & Jhanjhi, N. Z. (2022, June). Detection of skin cancer based on skin lesion images using deep learning. In *Healthcare* (Vol. 10, No. 7, p. 1183). MDPI.
17. Javed, D., Jhanjhi, N. Z., Ashfaq, F., Khan, N. A., Das, S. R., & Singh, S. (2024, July). Student Performance Analysis to Identify the Students at Risk of Failure. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 1-6). IEEE.
18. Sama, N. U., Zen, K., Jhanjhi, N. Z., & Humayun, M. (2024). Computational Intelligence Ethical Issues in Health Care. In *Computational Intelligence in Healthcare Informatics* (pp. 349-362). Singapore: Springer Nature Singapore.
19. Humayun, M., Sujatha, R., Almuayqil, S. N., & Jhanjhi, N. Z. (2022, June). A transfer learning approach with a convolutional neural network for the classification of lung carcinoma. In *Healthcare* (Vol. 10, No. 6, p. 1058). MDPI.
20. Srinivasan, K., Garg, L., Alaboudi, A. A., Jhanjhi, N. Z., Prabadevi, B., & Deepa, N. (2021). Expert System for Stable Power Generation Prediction in Microbial Fuel Cell. *Intelligent Automation & Soft Computing*, 30(1).
21. Nasser, M., Arshad, N. I., Ali, A., Alhussian, H., Saeed, F., Da'u, A., & Nafea, I. (2025). A systematic review of multimodal fake news detection on social media using deep learning models. *Results in Engineering*, 26, 104752.
22. Subramanian, M., Premjith, B., Shanmugavadivel, K., Pandiyan, S., Palani, B., & Chakravarthi, B. R. (2025, May). Overview of the shared task on fake news detection in dravidian languages-DravidianLangTech@NAACL 2025. In *Proceedings of the fifth workshop on speech, vision, and language technologies for dravidian languages* (pp. 759-767).
23. Azzeh, M., Qusef, A., & Alabboushi, O. (2025). Arabic fake news detection in social media context using word embeddings and pre-trained transformers. *Arabian Journal for Science and Engineering*, 50(2), 923-936.
24. Kamble, V. B., Uke, N. J., Karwatkar, D. G., Dhongade, R. D., & Kasare, P. (2025). Machine Learning in Fake News Detection and Social Innovation: Navigating Truth in the Digital Age. In *Exploring Psychology, Social Innovation and Advanced Applications of Machine Learning* (pp. 87-108). IGI Global Scientific Publishing.
25. Harris, S., Hadi, H. J., Ahmad, N., & Alshara, M. A. (2025). Multi-domain Urdu fake news detection using pre-trained ensemble model. *Scientific Reports*, 15(1), 8705.
26. Alsuwat, E., & Alsuwat, H. (2025). An improved multi-modal framework for fake news detection using NLP and Bi-LSTM. *The Journal of Supercomputing*, 81(1), 177.
27. Kishore, V., & Kumar, M. (2025). Enhanced multimodal fake news detection with optimal feature fusion and modified bi-lstm architecture. *Cybernetics and systems*, 56(6), 684-714.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.