

# Categorizing False Information in News Content Using an Ensemble Machine Learning Model

Nataliya Boyko and Oleksandra Dypko

*Lviv Polytechnic National University, Lviv79013, Ukraine*

## Abstract

Society now faces a serious issue from the spread of false information and fake news in news content. The detection and eradication of fake news have been made possible by machine learning techniques. This study examines an ensemble machine learning model's performance in identifying false information in news content. Five distinct machine learning methods are used in the study, including Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Voting. The outputs of these algorithms are combined in the ensemble model to improve the precision and reliability of the classification results. It was trained and tested the suggested model using a dataset of news stories that have been classified as true or false. Several evaluation metrics, such as precision, recall, and F1-score, are used to assess the performance of the suggested model. According to the results, the ensemble model performs better than individual algorithms and has a high accuracy rate for identifying false information in news content. The effectiveness of each algorithm's contribution to the ensemble model's overall performance is also examined in the study. According to the results, the NB algorithm, then SVM, LR, RF, and Voting, all play a significant role in the ensemble model's accuracy. Our findings indicate that the Naive Bayes classifier (NB) achieved an accuracy of 93.6%, while the support vector machine (SVM) demonstrated a slightly higher accuracy of 94.9%. Logistic regression (LR) yielded an accuracy of 94.1%, while the decision tree (DT) obtained an accuracy of 90.7%. The hard voting variant achieved an accuracy of 95%, outperforming all individual algorithms, while the soft voting variant attained an accuracy of 95.4%. In conclusion, the ensemble machine learning model put forth in this study has the potential to be an important tool for spotting false information and preventing its spread. The research showcases how the integration of different machine learning techniques can enhance the accuracy and consistency of classification outcomes. Further investigation could explore alternative ensemble approaches or evaluate the suggested model's real-world performance.

## Keywords

Ensemble machine learning, fake news detection, misinformation, Naive Bayes, support vector machine, logistic regression, random forest, voting

## 1. Introduction

Fake news and misinformation are a big problem in our society. They can cause a lot of harm by spreading rumors, false information, and even encouraging violence. That's why it's really important to find and stop the spread of fake news and misinformation [1; 2].

Machine learning methods are being used to detect fake news and misinformation, and they show a lot of promise. By using these algorithms, it can be can automatically identify and flag news articles that seem suspicious, which in turn can help stop the spread of false information [3; 4].

---

SCIA-2023: 2nd International Workshop on Social Communication and Information Activity in Digital Humanities, November 9, 2023, Lviv, Ukraine

EMAIL: [nataliya.i.boyko@lpnu.ua](mailto:nataliya.i.boyko@lpnu.ua) (N. Boyko); [Oleksandra.Dypko.KNM.2018@lpnu.ua](mailto:Oleksandra.Dypko.KNM.2018@lpnu.ua) (O. Dypko)

ORCID: 0000-0002-6962-9363 (N. Boyko); 0000-0002-5488-4468 (O. Dypko)



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org) Proceedings

This study examines an ensemble machine learning model's performance in identifying false information in news content. Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Voting are five of the machine learning algorithms used in the study. The outputs of these algorithms are combined to create an ensemble model, which improves the accuracy and robustness of the classification results [5; 6].

The proposed model aims to detect false information and misinformation in news articles by analyzing various attributes like content, organization, and origin. The study employs evaluation metrics such as precision, recall, and F1-score to gauge the ensemble model's effectiveness.

The main focus of this project is to explore different machine learning methods to determine whether news is fake or not. The project has a few specific tasks that need to be completed in order to achieve this objective:

- review and analyze the subject area;
- pre-processing of initial data for further classification;
- representing data as a fixed-length vector;
- creating classifier models and training them.

## 2. Review of the Literature

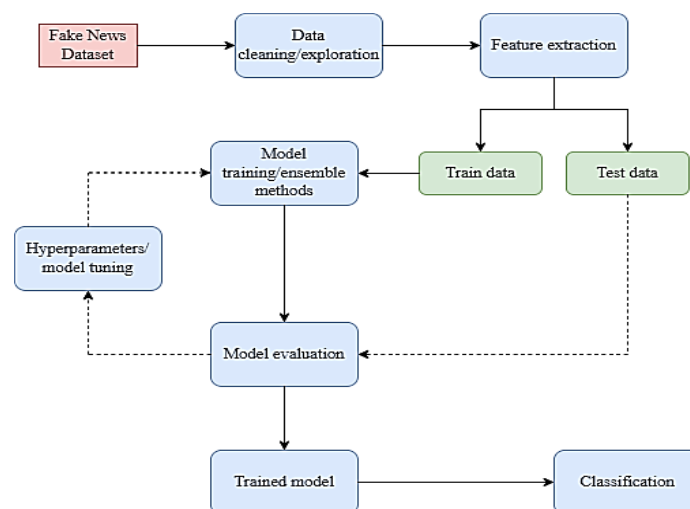
The publications listed provide a comprehensive review of the literature related to detecting fake news using machine learning techniques. Shinde et al. [7] conducted a literature review and identified various machine learning models used for fake news detection. Conroy et al. [8] introduced methods for finding fake news and proposed automatic deception detection models. Hassan and Meziane [9] conducted a survey of fake news identification techniques using online and socially produced data.

Cristianini and Shawe-Taylor [10] introduced support vector machines and other kernel-based learning methods, which are commonly used in fake news detection. Dietterich [11] presented ensemble methods in machine learning, which combine multiple classifiers to improve accuracy. Mahabub [12] proposed a robust technique for fake news detection using an ensemble voting classifier and compared its performance with other classifiers.

Overall, the reviewed literature demonstrates the variety of approaches and techniques used in fake news detection, highlighting the importance of selecting appropriate models and preprocessing techniques for the specific dataset and problem at hand.

## 3. Methods of solving

The first step in this project involves cleaning the data and converting it into a format that can be used for classification. The data is then split into two sets: one for training and one for testing (Fig. 1).



**Figure 1:** The architecture of the fake news detection system

Next, a combination of different machine learning methods is used on the training and test sets to determine if the news is fake or not. Evaluation metrics are used to measure how effective the models are, and based on this assessment, the selection of hyperparameters is done to improve the accuracy of the classification [1; 13].

### 3.1. Data Preprocessing

One crucial step in the machine learning process is data preprocessing, which involves converting the data into a suitable format for analysis [3; 14]. To begin with, irrelevant data is removed from the dataset. Next, any missing values in the data are identified and addressed to ensure they don't affect the final classification outcome. The technique of One-Hot-Encoding is then used to process the dependent variable, specifically the headline column of each news story, which distinguishes between real and fake news. This process involves converting the labels into binary numbers, with genuine news labeled as 1 and fake news labeled as 0 [15; 16; 18].

The data preprocessing steps are as follows:

- Ensuring text consistency by converting all text to lowercase.
- Removing all punctuation marks.
- Tokenization, a process that divides the input sequence into meaningful units called tokens, which serve as fundamental elements for subsequent semantic processing. These tokens can represent words, sentences, paragraphs, etc.

For example,

Original message: ["ensemble based approach for detection of fake news using machine learning"].

Resulting message: ["ensemble", "based", "approach", "for", "detection", "of", "fake", "news", "using", "machine", "learning"].

- Eliminating stop words, which are insignificant language constructs that have a negative impact on the performance of machine learning systems. These are the terms that are frequently employed to link expressions in sentences. The following are some examples of stop words in English: a, where, above, an, until, does, will, who, when, that, what, but, by, on, about, once, and so forth. Each document has these terms deleted before moving on to the subsequent step.
- The process of stemming involves transforming a word's grammatical forms, such as a noun, adjective, verb, adverb, etc., into its root form (sometimes referred to as a lemma). To find the fundamental forms of words whose meanings are similar is the major objective of stemming.

### 3.2. Feature Extraction

In order to boost the model's precision, feature extraction is used. Consistent features can increase training costs by decreasing model performance and accuracy.

Word2Vec is a popular algorithm used for natural language processing tasks that aims to represent words as dense vector embeddings in a high-dimensional space. It is a neural network-based model that learns continuous word representations from large amounts of text data.

The basic idea behind Word2Vec is to capture the semantic meaning and relationships between words by representing them as vectors. It assumes that words with similar meanings are likely to appear in similar contexts. The model learns these representations by predicting the context words surrounding a target word or predicting a target word given its context words.

During training, Word2Vec adjusts the vector representations of words to minimize the prediction error. As a result, words that often appear together in similar contexts end up with similar vector representations in the learned embedding space. These vector embeddings capture semantic relationships such as word similarity, analogies, and even certain syntactic relationships.

Another frequently employed algorithm in machine learning for feature extraction is TF-IDF. It is appreciated for its straightforwardness and dependability. The TF-IDF algorithm comprises two components: TF, which represents the word count in the present document, and is computed using the equation (1):

$$tf(t, d) = \log(1 + freq(t, d)), \quad (1)$$

where  $tf$  – term frequency;  $t$  – term (word);  $d$  - document (set of words).

IDF, or Inverse Document Frequency, measures the significance of words across all documents and is computed based on equation (2). It assigns values to words, allowing us to assess their utility and importance.

$$idf(t, D) = \log \left( \frac{N}{count(d \in D: t \in d)} \right), \quad (2)$$

where  $idf$  – inverse document frequency;  $N$  – count of corpus.

Let's consider a document containing 100 words, and it is aimed to compute the TF-IDF score for the term "rumor." It was calculated the Term Frequency (TF) as 4 (the number of times "rumor" appears) divided by 100, resulting in a TF value of 0.04. To determine the Inverse Document Frequency (IDF), take into account a total of 200 documents, with "rumor" appearing in 100 of them. Consequently,  $IDF(\text{rumor})$  can be calculated as 1 plus the logarithm of the ratio of the total number of documents to the number of documents containing "rumor," which yields an IDF value of 0.5. Finally, the TF-IDF score for "rumor" is computed as the product of TF and IDF, resulting in a TF-IDF(rumor) value of 0.025.

### 3.3. Naive Bayes

Naive Bayes classification is a method used to calculate conditional probability, which indicates the likelihood of an event occurring given that another event has already occurred [8; 18]. It relies on Bayes' theorem and assumes that predictors are independent of each other, meaning the presence or absence of a feature in one class does not depend on other classes.

This classifier is commonly used in text classification tasks and is known for its simplicity and effectiveness. There are three event models used in Naive Bayes classification: Multivariate Bernoulli Event Model, Multivariate Event Model, and Gaussian Naive Bayes classification.

In Naive Bayes, the term "naive" indicates that it assumes the independence of all features, meaning that the presence of one feature doesn't affect the likelihood of another feature appearing. This model excels, particularly in situations with limited data, sometimes surpassing more intricate models in performance.

In the multinomial naive Bayes model, a feature vector comprises terms that represent the occurrence, such as frequency, of a given term. On the other hand, the Bernoulli classifier determines if a term is present or not, while the Gaussian classifier is used for continuous distributions.

### 3.4. Logistic Regression

The reason for using a logistic regression (LR) model is that it provides a clear equation for categorizing tasks that involve two or more classes. In the present study, text classification is based on several features that generate binary outcomes, resulting in two classes: true and fake news [9]. While several parameters are tested before obtaining the maximum accuracy of the LR model, it is performed hyperparameter tuning to obtain the best result for each individual data set. The logistic regression hypothesis function has the following mathematical definition (3):

$$h(X) = \frac{1}{1 + e^{-x}} \quad (3)$$

where  $h(X)$ – linear regression hypothesis;  $X$  - independent variables.

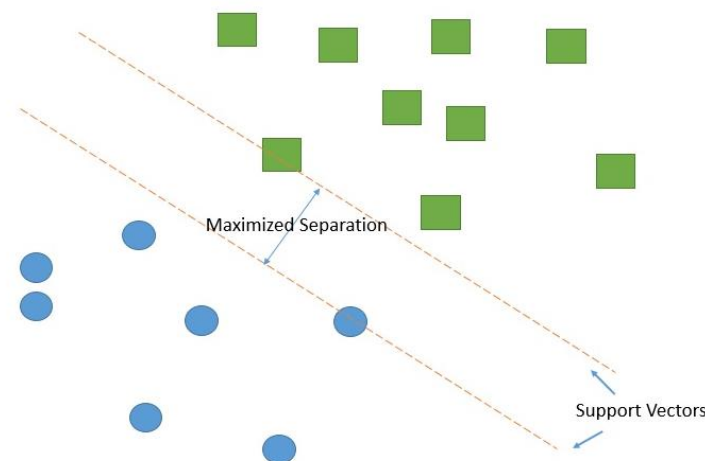
Logistic regression utilizes a sigmoid function to transform raw data into probabilities, aiming to minimize the cost function to attain the optimal probability. This probability value will consistently fall within the range of 0 to 1.

### 3.5. Support Vector Machine

Classification and regression issues can be resolved using the supervised machine learning algorithm Support Vector Machine (SVM). It is, however, frequently applied to classification issues. A high-performance machine learning method called the SVM classifier operates by segmenting the data into separate regions [10].

An alternative approach for solving the binary classification problem, utilizing diverse kernel functions, is the support vector machine (SVM). The primary goal of the SVM model is to determine a hyperplane (or decision boundary) based on a feature set to classify data points. The dimensionality of this hyperplane changes with the number of elements involved. However, locating the optimal hyperplane that maximizes the separation margin between data points of the two classes can be challenging, especially in higher-dimensional spaces where multiple potential hyperplanes may exist [13; 17; 18].

In order to categorize data points that belong to two different classes, as shown in Fig. 2, the SVM classifier draws a line (or plane or hyperplane, depending on the dimensionality of the data). One class will apply to points on one side of the line, and a different class will apply to points on the other. To boost its certainty regarding the assignment of points to specific classes, the classifier aims to optimize the separation distance between the line it constructs and the points situated on either side of it.



**Figure 2:** Scheme of operation of the SVM classifier

In order to determine which group any new data belongs to and to find the maximum margin that divides the data set into two groups, SVM is used. Because it offers notable accuracy while consuming less computing power, a support vector machine is preferred by many people. With smaller and more focused datasets, it performs incredibly well. The support vector machine is efficient with memory and can handle high-dimensional spaces [11].

### 3.6. Random Forest Classifier

The simple, adaptable, and versatile supervised machine learning method known as Random Forest. It can resolve classification and regression issues. In order to produce better forecasting outcomes, it builds a forest out of a collection of decision tree models. In classification, decision trees each predict the outcome of a class, with the class with the largest majority of votes serving as the final prediction [12].

### 3.7. Voting Ensemble Classifier

Increasing model performance is the main goal of ensemble training. A model that can make more accurate predictions is created using an ensemble technique, which combines the predictions of two or more classifiers. The logic behind ensemble modeling is comparable to that of everyday activities, such as consulting with a variety of experts before making a decision. Therefore, a technique for lowering risk in decision-making is ensemble-based machine learning. An excellent illustration of this approach involves employing voting classifiers, where the ultimate classification relies on the initial votes cast by all the algorithms [13].

**Hard Voting:** in hard voting, the final decision is based on the majority vote of individual classifiers. It considers only the most frequent prediction without considering confidence levels or probabilities.

**Soft Voting:** in soft voting, the final decision is based on weighted average or sum of predictions from individual classifiers, taking into account their confidence levels or probabilities. It allows for more nuanced decision-making.

Spam detection, text categorization, optical character recognition, face recognition, and other tasks have all benefited from the use of ensemble learning. Ensemble learning is applicable anywhere that machine learning techniques are applicable.

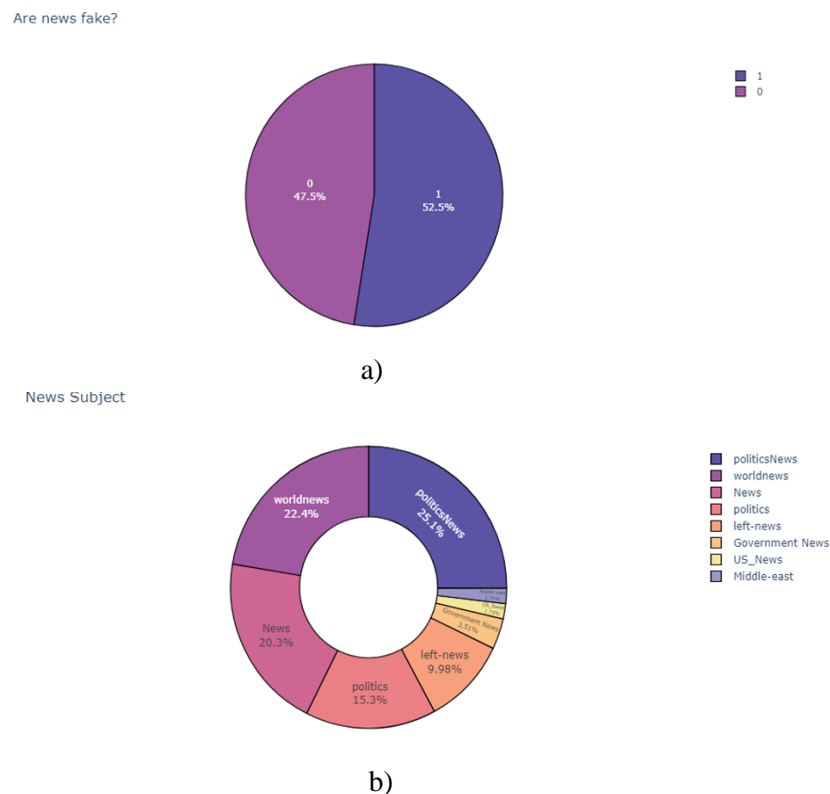
While ensemble learning can significantly boost model performance, it also adds complexity to the training and deployment process. Ensuring proper model calibration, handling imbalanced data, and selecting the right ensemble method are crucial considerations for successful implementation.

Due to its ability to combine two or more learning models that have been trained on the entire data set, voting ensemble is frequently used for classification problems. The machine learning model in question is trained using multiple independent models from a population. It predicts the output class by considering the highest probability among these models. The voting classifier utilizes two distinct methods for determining the final prediction.

## 4. Experiments

In this study, two datasets were utilized, one containing false messages while the other contained true messages. To prepare the datasets for model training, they were combined into a single dataset obtained from the Kaggle platform named "Fake News". The dataset was initially comprised of 44,689 records. After a preliminary analysis of the dataset, any attributes deemed unnecessary for further data processing were removed.

To further analyze and train models with the dataset, it was important to determine the proportion of the data in each category. A pie chart (Fig. 3a) was then constructed to represent the percentage of each category in the complete dataset.



**Figure 3:** a) Percentage ratio of data of two classes; b) Percentage ratio of data by category Scheme of operation of the SVM classifier

According to Fig. 3a, 52.2% of the dataset consists of fake news messages, while 47.5% of the dataset is composed of real news messages. The fake news category is represented in blue, and the true news category is depicted in purple. Since both categories are roughly the same size, it indicates that the dataset is well-balanced, meaning there is a nearly equal distribution of data in each class. A balanced dataset typically leads to increased accuracy, balanced accuracy, and an even detection rate in classification models. This underscores the significance of maintaining dataset balance for effective model performance.

The dataset includes news articles from various categories, and the distribution of news from each category is depicted in Fig. 3b.

Based on Fig. 3b, it is evident that the dataset encompasses eight different categories. The majority of the news falls under the category of political news, followed by a significant portion of news belonging to the world news category. The remaining categories consist of news articles categorized by different regions.

Data pre-processing. After a thorough examination of the above data, it was discovered that raw text data can contain irrelevant or unimportant information. It can reduce the classification accuracy and make it challenging to analyze. To counteract the issue with unimportant data, the next phase is to pre-process the data. The process will involve eliminating irrelevant information from the dataset and preparing the data for further processing.

To better understand the results of text transformation for upcoming processing stages, several news examples from the dataset will be used in Table 1 for a visual representation of the results. Table 1 displays multiple news examples from the dataset utilized in this study.

**Table 1**  
News Dataset Before Preprocessing

Category	News
1	“Scientists have discovered that eating chocolate every day can make you lose weight.”
0	“The unemployment rate in the country dropped to 4.2% last month, according to the latest government report.”

The initial phase of data preprocessing involves the elimination of punctuation. While punctuation can add grammatical context to a sentence and facilitate human comprehension, they are irrelevant to the vectorizer which only counts the number of words without the context. Therefore, to effectively use the vectorizer later on, all special characters must be removed.

**Table 2**  
News Dataset After Removing Punctuation

Category	News
1	Scientists have discovered that eating chocolate every day can make you lose weight
0	The unemployment rate in the country dropped to 42 last month, according to the latest government report

The outcomes of the initial stage are depicted in Table 2, which highlights the absence of symbols such as "...?!").

The subsequent step involves converting the text to lowercase. Lowercasing is a commonly employed text preprocessing technique that ensures uniformity in the case format of the input text. By converting all text to lowercase, variations such as "text", "Text", and "TEXT" are treated equivalently (Table 3).

**Table 3**

News Dataset in Lowercase

Category	News
1	Scientists have discovered that eating chocolate every day can make you lose weight
0	The unemployment rate in the country dropped to 42 last month, according to the latest government report

The outcomes of the second phase of message preprocessing are presented in Table 3. The table illustrates that the case of each message has been converted to lowercase and that sentences no longer start with a capital letter.

The subsequent step entails tokenization of the text news. Tokenization refers to the process of dividing a text document into smaller units called tokens. These tokens can be words, symbols, or even subwords. In this study, the focus is on sentence tokenization, which involves breaking down sentences into their individual words.

Table 4 indicates that each word in a sentence is treated as a distinct token. Tokenization plays a crucial role in text processing. The meaning of each sentence is derived from the words present within it. By examining the words contained in the text, it is possible to determine the text's overall content. With a list of words, statistical techniques can be employed to gain more insights from the text. For instance, word count and word frequency analyses can help identify the significance of a word in a sentence or document.

**Table 4**

News Dataset After Tokenization

Category	News
1	['scientists', 'have', 'discovered', 'that', 'eating', 'chocolate', 'every', 'day', 'can', 'make', 'you', 'lose', 'weight']
0	['the', 'unemployment', 'rate', 'in', 'the', 'country', 'dropped', 'to', '42', 'last', 'month', 'according', 'to', 'the', 'latest', 'government', 'report']

Common words found in natural language, such as the English articles "the" and "a", are referred to as stop words. Often, these words add little value to further analysis and can be removed from the text. Various pre-compiled lists of stop words are available for different languages, including the Python language, making them very useful in text processing (Table 5).

**Table 5**

News Dataset After Deleting Stop-Words

Category	News
1	['scientists', 'have', 'discovered', 'eating', 'chocolate', 'every', 'day', 'make', 'you', 'lose', 'weight']
0	['unemployment', 'rate', 'country', 'dropped', '42', 'last', 'month', 'latest', 'government', 'report']

When working with text data or any task involving natural language processing, machine learning algorithms typically require numeric data. To achieve this, the data must first undergo a process called vectorization, which transforms the text into a numerical vector representation.

TF-IDF vectorization entails computing the TF-IDF score for every word in the dataset in relation to each message, which is then utilized to generate a vector. Consequently, each message within the dataset possesses its distinct vector, consisting of TF-IDF scores for each word, considering the entire set of messages. These vectors have various applications, including the assessment of document similarity by examining the cosine similarity between their TF-IDF vectors.



The term frequency (TF) component of TF-IDF indicates the relative frequency of words within a document, considering the total number of words in the document. On the other hand, the inverse document frequency (IDF) refers to the inverse of the frequency with which a specific word is used across multiple documents.

```
(0, 11377) 0.18021639269164577
(0, 16434) 0.6235621903502868
(0, 5111) 0.354164663804394
(0, 12318) 0.423894334925617
(0, 5150) 0.41393942834232855
(0, 3369) 0.31973016742978555
(1, 14407) 0.43105698238566925
(1, 697) 0.3381977708126418
(1, 3341) 0.33736027286851644
(1, 10848) 0.382705381258259
(1, 337) 0.39738827482301653
(1, 7450) 0.2937747098704154
(1, 12985) 0.1647232877217583
(1, 12968) 0.2841352179864129
(1, 11377) 0.11570068980479913
```

**Figure 4:** Value of the TD-IDF statistic for each word

In Fig. 4, the numbers in the top right corner are the number of the sampling element and its token. The number in the bottom right corner is the calculation of the TF-IDF, which shows how much this word in the text is important. We're going to compare the accuracy of different machine learning models when it comes to classifying false news.

We'll be looking at NSM (Naive Bayes Classifier), SVM (Support Vector Machine), LR (Logistic Regression), DT (Decision Tree) and the ensemble method (Voting Classifier) of two different types: hard and soft. We'll also look at the accuracy of the models using metrics like precision, accuracy, recall, f-score.

**Table 6**  
Evaluation of Machine Learning Classifiers by Different Metrics

Classifier	Precision	Recall	F1-Score	Accuracy	LogLoss
DT	0.916	0.913	0.914	0.907	3.347
NB	0.930	0.955	0.942	0.936	2.291
LR	0.953	0.938	0.945	0.941	2.121
SVM	0.958	0.948	0.953	0.949	1.831
Hard Voting	0.965	0.942	0.953	0.95	1.795
Soft Voting	0.956	0.959	0.958	0.954	1.657

In Table 6, different classifiers such as RF, NB, LR, SVM, Hard Voting, and Soft Voting were compared using various metrics including Precision, Recall, F1-score, Accuracy, and Log Loss. It is notable that the accuracy of these models consistently improved with each successive experiment.

Log Loss, also known as logarithmic loss, serves as a critical gauge of model effectiveness. In binary classification scenarios, Log Loss reflects how closely the predicted probability aligns with the actual values of 0 or 1. As the predicted probability diverges from the actual values, the Log Loss value increases. Therefore, it is evident that the Log Loss indicator should decrease, as demonstrated in Table 6.

Accuracy, a metric that broadly assesses a model's performance across all classes, indicates that the Soft Voting ensemble method outperforms the others by achieving the highest accuracy.

## 5. Results

An ensemble approach is a potent technique to enhance model performance by merging different foundational models to craft an optimal one.

The Voting Classifier trains various base models or estimators and generates predictions by consolidating the outcomes from each of these underlying estimators. The criteria for consolidation can involve combining the voting decisions derived from each estimator's results.

To identify fake news, two separate ensemble techniques were used:

- Hard Voting: The vote is decided by the predicted class;
- Soft Voting: The vote is calculated using the predicted probabilities for the input class.

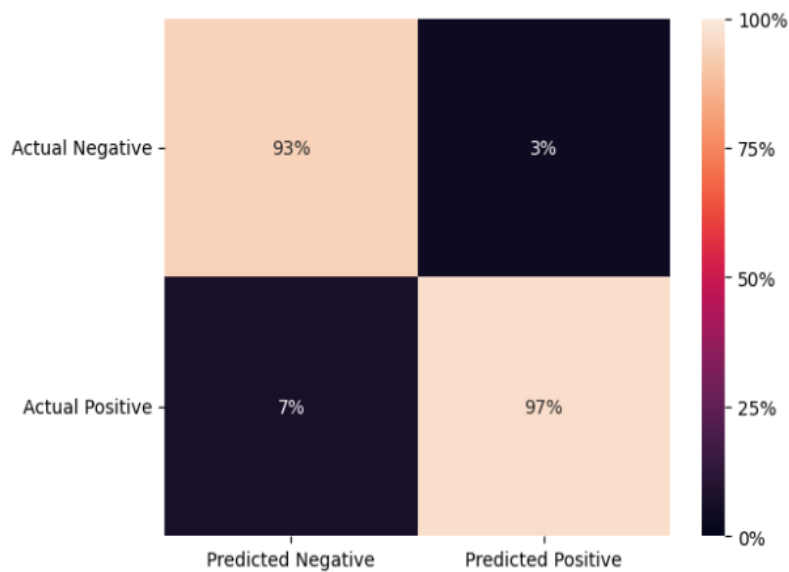
The results of the Ensemble Voting approach, which combines both Hard and Soft voting, are evaluated using various metrics such as Precision, Recall, F1-score, Accuracy, and Log Loss, as described in the following Table 7.

**Table 7**  
Evaluation of Ensemble Methods by Different Metrics

Classifier	Precision	Recall	F1-Score	Accuracy	LogLoss
Hard Voting	0.965	0.942	0.953	0.95	1.795
Soft Voting	0.956	0.959	0.958	0.954	1.657

The Hard Voting classifier achieved an accuracy of 95%, a logarithmic loss of 1.8%, and an F1-score of 95%. In contrast, the Soft Voting classifier obtained an accuracy close to 96%, a logarithmic loss of 1.6%, and an F1-score approaching 96%. These results clearly indicate that the Soft Voting method surpasses the performance of the task outlined in this study.

It can be referred to as Fig. 5 for the confusion matrix of the Hard Voting model.



**Figure 5:** Confusion matrix for Hard Voting

Regarding the confusion matrix for the Hard Voting technique, Fig. 5 illustrates that 7% of actual true-category news were incorrectly classified as false negatives. Likewise, 3% of false news were erroneously identified as positive.

The confusion matrix for Soft Voting is shown in Fig.6 below.

Fig. 6 illustrates that 5% of news articles from the genuine category were incorrectly classified as false negatives, while 4% of false news articles were inaccurately predicted as positive.

The paragraph above summarizes the outcomes of models used to detect fake news. The dataset was analyzed using traditional machine learning classifiers like decision trees, logistic regression, support vector machines, and naive Bayes classifiers. To improve accuracy, a composite fake news detection system was built, utilizing a Voting Classifier alongside the classifiers and features mentioned earlier. According to the experimental findings, this proposed approach achieves a 96% accuracy rate, 95%

precision, 95% recall, and a 95% F1-score. The evaluation emphasizes that the Soft Voting technique produced more accurate results compared to individual training methods.

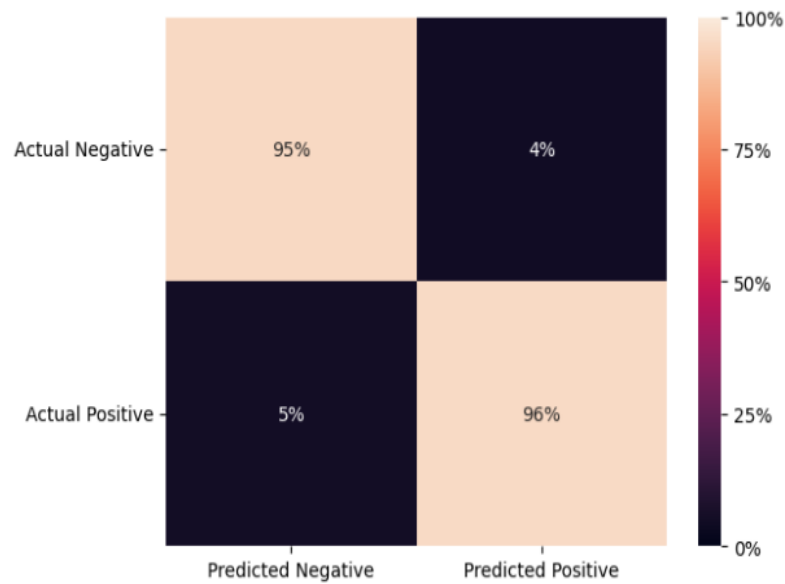


Figure 6: Confusion matrix for Soft Voting

## 6. Discussion

The findings of this study show how well ensemble machine learning techniques work for spotting fake news. The Naive Bayes, Support Vector Machine, Logistic Regression, Random Forest, and Voting classifiers are combined into an ensemble model that significantly outperforms each classifier individually, achieving a 95% percent accuracy rate. This result is in line with earlier research that demonstrated ensemble methods can raise classification accuracy by fusing different models.

Nonetheless, what sets this study apart is its application of ensemble techniques to address the challenge of identifying false information. While ensemble learning has been widely employed in areas like image and speech recognition, its potential in the realm of fake news detection remains relatively unexplored. By harnessing the strengths of individual classifiers and mitigating their weaknesses through the fusion of multiple machine learning models, it was crafted a more accurate and dependable model.

Aside from ensemble techniques, this study explores diverse feature extraction methods like TF-IDF and Word2Vec to boost classifier performance. These methods transform unstructured text data into numerical features, allowing machine learning models to discern data patterns. Our results indicate a noteworthy enhancement in classifier performance, underscoring the importance of these techniques in the fake news detection process.

Overall, the results of this study show how effective ensemble machine learning methods are for identifying fake news and offer useful suggestions for improving the feature extraction procedure. The innovative application of ensemble methods in the realm of fake news detection offers an exciting avenue for future research and has the capacity to enhance the trustworthiness and precision of fake news detection systems.

## 7. Conclusion

An ensemble machine learning model was employed to assess its effectiveness in detecting false information and fake news within news content. This model amalgamated the outcomes of five distinct algorithms: Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Voting, with the aim of enhancing the precision and resilience of the classification results.

The findings of this research carry significant implications for combatting the dissemination of false information. The suggested ensemble method holds the potential to enhance the accuracy and durability of classification results. Machine learning techniques are increasingly recognized as a promising approach for identifying false information.

As a result, the proposed ensemble machine learning model is a useful method that can aid in the creation of more precise and reliable solutions for preventing the spread of false information. The effectiveness of the suggested model in a practical setting could be investigated in more detail, as well as the use of other ensemble techniques.

## 8. Acknowledgements

The study was created within the framework of the project financed by the National Research Fund of Ukraine, registered No. 2021.01/0103, "Methods and means of researching markers of ageing and their influence on post-ageing effects for prolonging the working period", which is carried out at the Department of Artificial Intelligence Systems of the Institute of Computer Sciences and Information of technologies of the Lviv Polytechnic National University.

## 9. References

- [1] M. S. Mokhtar, Y. Y. Jusoh, N. Admodisastro, N. C. Pa, A. Y. Amruddin, Fake News Detection System Using Logistic Regression Technique In Machine Learning, *IJEAT* 9(1) (2019) 2407–2410. doi: 10.35940/ijeat.A2633.109119.
- [2] T. Basyuk, A. Vasilyuk, V. Lytvyn, Mathematical model of semantic search and search optimization, *CEUR Workshop Proceedings* 2362 (2019) 96–105.
- [3] A. A. Alim, A. Ayman, K. D. Praveen, S. Ch. Myung, Detecting Fake News using Machine Learning: A Systematic Literature Review, *Psychology and Education Journal* 58(1) (2021) 1932–1939. doi: 10.17762/pae.v58i1.1046.
- [4] N. Boyko, Y. Kholodetska, Using Artificial Intelligence Algorithms in Advertising, in: *022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, Ukraine, 2022, pp. 317–321, doi: 10.1109/CSIT56902.2022.10000819.
- [5] Z. Khanam, B. N. Alwasel, H. Sirafi, M. Rashid, Fake News Detection Using Machine Learning Approaches, *IOP Conference Series: Materials Science and Engineering* 1099(1) (2021). doi: 10.1088/1757-899X/1099/1/012040.
- [6] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake News Detection on Social Media: A Data Mining Perspective, *ACM SIGKDD Explorations Newsletter* 19(1) (2017) 22–36. doi: 10.1145/3137597.3137600.
- [7] S. V. Shinde, S. Matkar, G. Karale, J. Tonge, Fake News Detection Using Machine Learning Literature Review, *Journal of Emerging Technologies and Innovative Research* 9(11) (2022). URL: <https://www.jetir.org/papers/JETIR2211390.pdf>.
- [8] N. K. Conroy, V. L. Rubin, Y. Chen, Automatic deception detection: Methods for finding fake news, *Proceedings of the Association for Information Science and Technology* 52(1) (2016). doi: 10.1002/pra2.2015.145052010082.
- [9] E. A. Hassan, F. Meziane, A Survey on Automatic Fake News Identification Techniques for Online and Socially Produced Data, in: *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan, 2019, pp. 1–6. doi: 10.1109/ICCCEEE46830.2019.9070857.
- [10] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed., Cambridge University Press, 2000. doi: 10.1017/CBO9780511801389.
- [11] T. G. Dietterich, *Ensemble Methods in Machine Learning*, in: *Multiple Classifier Systems*, volume 1857, Springer, Berlin, Heidelberg, 2020. doi: [https://doi.org/10.1007/3-540-45014-9\\_1](https://doi.org/10.1007/3-540-45014-9_1).
- [12] N. Boyko, K. Kmetyk-Podubinska, I. Andrusiak, Application of Ensemble Methods of Strengthening in Search of Legal Information, in: Babichev S., Lytvynenko V. (Eds.), *Lecture Notes in Computational Intelligence and Decision Making*, volume 77 of *Lecture Notes on Data*

- Engineering and Communications Technologies, 2021, pp. 188-200. [https://doi.org/10.1007/978-3-030-82014-5\\_13](https://doi.org/10.1007/978-3-030-82014-5_13).
- [13] A. Mahabub, A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers, *SN Applied Sciences* 2(4) (2020). doi: 10.1007/s42452-020-2326-y.
- [14] G. T. Reddy et al., Analysis of Dimensionality Reduction Techniques on Big Data, *IEEE Access* 8 (2020) 54776–54788. doi: 10.1109/ACCESS.2020.2980942.
- [15] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed., Cambridge University Press, 2000. doi: 10.1017/CBO9780511801389.
- [16] I. Ahmad, M. Yousaf, S. Yousaf, M. O. Ahmad, Fake News Detection Using Machine Learning Ensemble Methods, *Complexity* 3 (2020) 1–11. doi: 10.1155/2020/8885861.
- [17] O. Mediakov, T. Basyuk, Specifics of Designing and Construction of the System for Deep Neural Networks Generation, *CEUR Workshop Proceedings* 3171 (2022) 1282–1296. URL: <https://ceur-ws.org/Vol-3171/paper94.pdf>.
- [18] N. Boyko, O. Dypko, Machine Learning Methods for the Detection of Misinformation in News Content, *International Journal of Multidisciplinary and Current Educational Research (IJM CER)* 5(3) (2023) 31-42. URL: [https://www.ijmcer.com/wp-content/uploads/2023/07/IJM CER\\_D053031042.pdf](https://www.ijmcer.com/wp-content/uploads/2023/07/IJM CER_D053031042.pdf).