FAKE NEWS DETECTION USING MACHINE LEARNING

Deepika. R[1], Sinchana. A. J[2], Balaram M[3]

[1,2]Student, Department Of BCA, BMS College Of Commerce And Management, Bengaluru, Karnataka, India.

[3]Assistant Professor, Department Of BCA, BMS College Of Commerce And Management, Bengaluru, Karanataka, India.

DOI: https://www.doi.org/10.58257/IJPREMS35173

ABSTRACT

The main goal of the Fake News Detection project is to address the spread of false information online. This will be done by creating an automated system that can accurately distinguish between real and fake news articles. The project will include a detailed text classification process, covering data collection, preprocessing, feature extraction, model training, and evaluation. To improve accuracy in identifying fake news, various machine learning algorithms like Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, and Support Vector Machine (SVM) will be used. Additionally, the project will feature a user-friendly interface for uploading data and interacting with the model, as well as visualization tools for presenting results. Future plans for the project include real-time data processing, support for multiple languages, and ongoing learning through user feedback to ensure the system remains effective against changing misinformation tactics. In summary, this project lays a strong foundation for combating fake news, encouraging informed decision-making, and maintaining public trust in the media.

Keywords: Fakenews detection, logistic regression, feature extraction, visualization, user interface, decision tree.

1. INTRODUCTION

In today's technology-driven society, the rapid spread of false information is a major concern. It is crucial to detect and prevent the dissemination of fake news in order to maintain the credibility of data and support informed decision-making. This project aims to address this issue by creating an advanced text classification system specifically designed to identify fake news. By utilizing cutting-edge machine learning techniques and powerful libraries such as pandas, scikit-learn, and nltk, the system is built to be strong and scalable. The workflow includes important stages such as data preprocessing, feature extraction, model training, and evaluation. Data preprocessing involves cleaning the data, handling missing values, removing punctuation, and filtering out irrelevant words. Tokenization and stemming are then applied to standardize the text data for machine learning algorithms. Feature extraction is performed using TF-IDF vectorization, which converts the text into a numerical format that captures the importance of each word. Various classification algorithms, including Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, and Support Vector Machine (SVM), are utilized. Each model is trained on preprocessed data and evaluated based on accuracy, precision, recall, and F1-score metrics to ensure reliable performance. Visualization tools such as word clouds and bar charts provide insights into common words found in real and fake news articles. The system also includes a user-friendly frontend interface for uploading datasets, initiating model training, and viewing classification results. Thorough testing is conducted to ensure that the system meets performance, scalability, reliability, and security standards for deployment. In summary, this project introduces a highly accurate and reliable text classification system for detecting fake news, offering an effective solution to combat misinformation in the digital age.

2. METHODOLOGY

2.1 Data Ingestion

To begin, the initial stage entails gathering and importing the dataset into the system. The dataset usually consists of news articles that have been categorized as either real or fake. This information is extracted from a CSV file and imported into a Pandas DataFrame to facilitate further analysis. The load_data() function is employed to read the CSV file, while clean_columns() is utilized to eliminate extraneous columns and retain only pertinent data for examination.

2.2 Data Preprocessing

Preprocessing data is essential for improving the precision and importance of features used in training machine learning models. This process includes handling missing values, randomizing data, and refining text. The remove_missing_values() function replaces empty strings or placeholders for missing data. The shuffle_data() function mixes up the dataset to prevent bias. The preprocess_text() function refines the text by removing punctuation, stopwords, and applying tokenization and stemming to standardize the data for feature extraction.

2.3 Feature Extraction

After preprocessing the text data, the next step involves converting it into a numerical format that is compatible with machine learning algorithms. This can be done using TF-IDF Vectorization, where the tfidf_vectorizer() function transforms the cleaned text data into TF-IDF vectors, highlighting the significance of words in the dataset. Alternatively, Count Vectorization can be used with the count_vectorizer() function to represent the text as a bag-of-words model, showing the frequency distribution of words.

2.4 Model Training and Evaluation

After extracting the features, the next step is to train different machine learning models and assess their performance. This involves:

Training Models: The train_model() function is employed to train various machine learning models, such as Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, and SVM, using the training data.

Evaluating Models: The evaluate_model() function is used to evaluate the trained models using metrics like accuracy, precision, recall, F1-score, and confusion matrix. The classification_report() function generates comprehensive classification reports, offering valuable insights into the performance of each model.

2.5 Visualization

Visualization is essential for comprehending data and model outcomes. The module comprises various functions, including:

Word Cloud Generation: The plot_word_cloud() function produces word clouds to visually represent the most commonly used words in real and fake news articles.

Bar Charts for Key Words: The plot_top_words() function creates bar charts that illustrate the frequency of the most important words.

Confusion Matrix: The plot_confusion_matrix() function exhibits the confusion matrix for classification models, aiding in the visualization of the model's performance.

2.6 User Interface

The project incorporates a user interface to enhance user engagement, offering the following functionalities:

Dataset Upload: Users have the ability to upload their datasets for analysis by utilizing the upload_dataset() function.

Training Initiation: The initiate_training() function enables users to commence the process of training the model.

Result Presentation: The display_results() function showcases the classification results, encompassing visualizations and performance metrics.

3. MODELING AND ANALYSIS

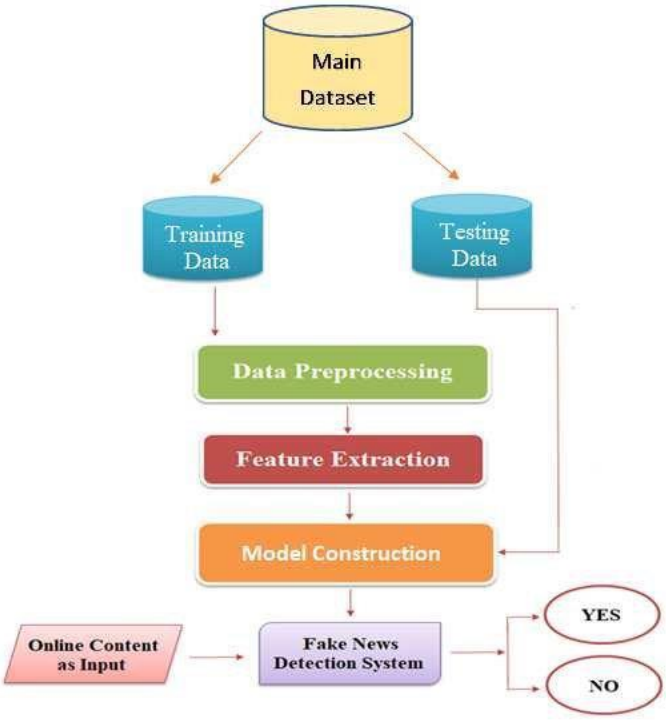
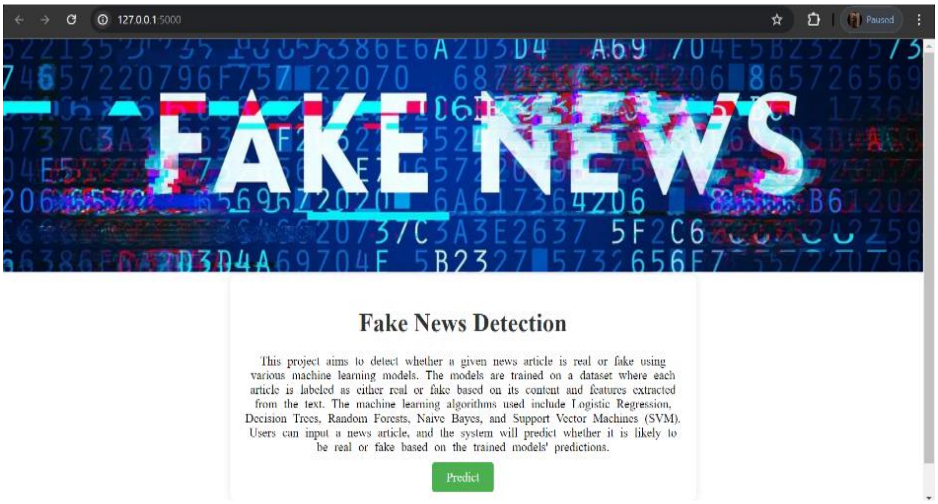


Figure 1: System Architecture

4. RESULTS AND DISCUSSION

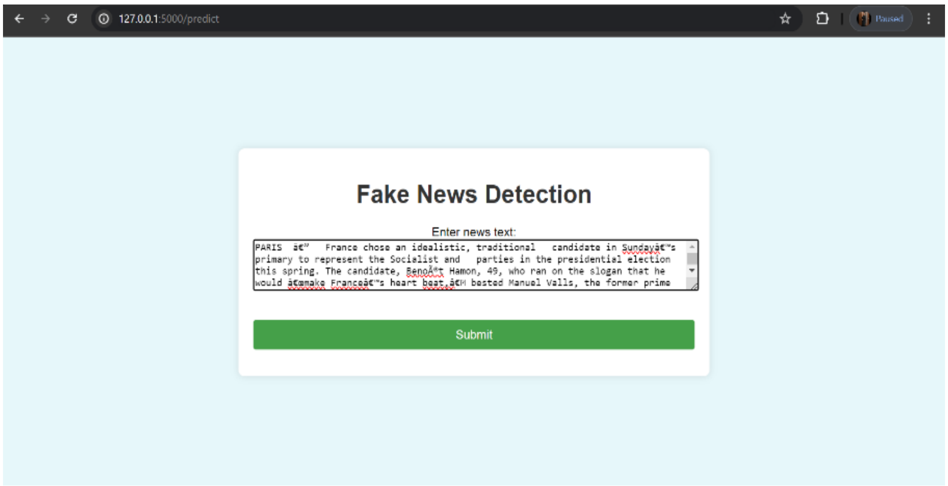
The primary goal of this project is to examine the Kaggle dataset, a widely used dataset for identifying false news and categorizing them as reliable or fake. We thoroughly analyzed the Kaggle dataset and utilized a confusion matrix to present our findings. To detect false news, we employed five distinct algorithms: Random Forests, Naïve Bayes, Logistic Regression, Decision Tree, and SVM. By executing the algorithm code on the Anaconda platform, the python code automatically generates the confusion matrix using the cognitive learning library.



Fake News Detection

This project aims to detect whether a given news article is real or fake using various machine learning models. The models are trained on a dataset where each article is labeled as either real or fake based on its content and features extracted from the text. The machine learning algorithms used include Logistic Regression, Decision Trees, Random Forests, Naive Bayes, and Support Vector Machines (SVM). Users can input a news article, and the system will predict whether it is likely to be real or fake based on the trained models' predictions.

Figure 2: Front Page

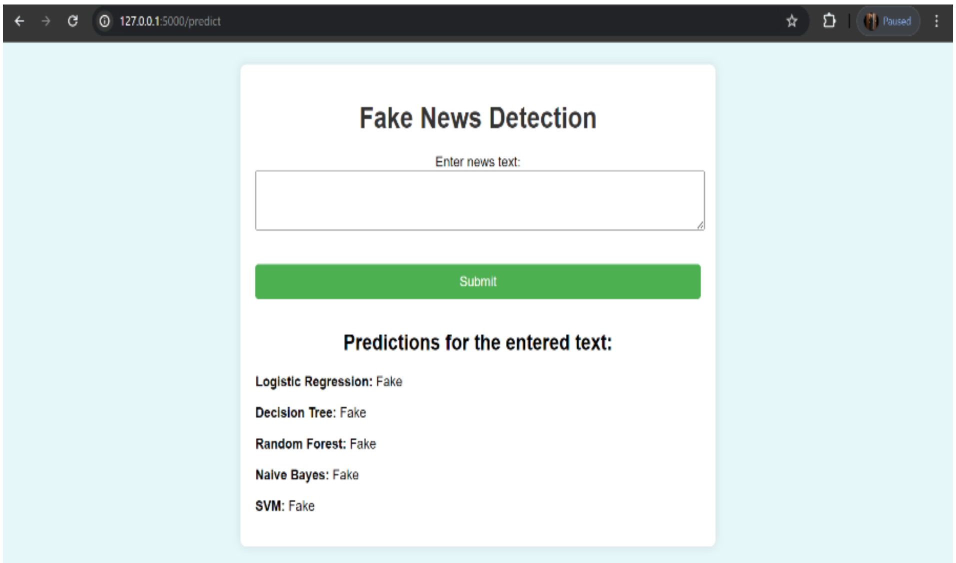


Fake News Detection

Enter news text:

PARIS â€" France chose an idealistic, traditional candidate in Sundayâ€™s primary to represent the Socialist and parties in the presidential election this spring. The candidate, BenoÃ®t Hamon, 49, who ran on the slogan that he would â€œmake Franceâ€™s heart beat,â€ bested Manuel Valls, the former prime

Figure 3: Submission Page



Fake News Detection

Enter news text:

Predictions for the entered text:

Logistic Regression: Fake

Decision Tree: Fake

Random Forest: Fake

Naive Bayes: Fake

SVM: Fake

Figure 4: Result Page

The findings suggest that ensemble methods such as Random Forest performed better than individual models in terms of accuracy and generalization. Logistic Regression and SVM also showed impressive performance, indicating their reliability in text classification tasks. Although the Decision Tree model showed some signs of overfitting, it still performed well. Multinomial Naive Bayes, known for its simplicity, delivered competitive performance and fast computation, making it a practical option for large-scale applications.

5. CONCLUSION

The Fake News Detection project has successfully developed a comprehensive system using machine learning to identify and categorize fake news. The project followed a structured process that included data ingestion, preprocessing, feature extraction, model training, evaluation, and visualization to achieve accurate results with various models. The study revealed that ensemble methods such as Random Forest were effective in distinguishing between real and fake news, with Logistic Regression and SVM also showing strong performance. Data preprocessing, which involved handling missing values and cleaning text, was essential for precise classification. Visualization techniques provided valuable insights into the data and model outcomes. The project also featured a user-friendly interface and model persistence for easy interaction and reuse of trained models. Moving forward, the project will focus on real-time data processing, multilingual support, and advanced NLP techniques to enhance accuracy. Incorporating user feedback and ethical standards will be crucial for real-world implementation. This project showcases the potential of machine learning in combating misinformation and promoting informed decision-making in society.

6. REFERENCES

[1] Abdullah-All-Tanvir, Mahir, E. M., Akhter S., & Huq, M. R. (2019). Detecting Fake News using Machine Learning and Deep Learning Algorithms. 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, Malaysia, 2019, pp.1-5, https://doi.org/10.1109/ICSCC.2019.8843612

[2] S. B. Parikh, V. Patil, and P. K. Atrey, “On the Origin, Proliferation and Tone of Fake News,” Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019, pp. 135–140, 2019.

[3] Khanam, Z., et al. "Fake news detection using machine learning approaches." IOP conference series: materials science and engineering. Vol. 1099. No. 1. IOP Publishing, 2021.

[4] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.(2019).

[5] Zhang, Jiawei, Bowen Dong, and S. Yu Philip. "Fakedetector: Effective fake news detection with deep diffusive neural network." 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020.