Fake News Detection and Evaluation with Confusion Matrix

Aditya Chakraborty
Bachelor of Technology
(Ocean Engineering and Naval Architecture)
2024-28
Indian Institute of Technology, Kharagpur

Period of Internship:
25th August 2025 - 19th September 2025

Report submitted to:
IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

Abstract

This project focuses on detecting fake news and evaluating the results with the help of a confusion matrix. The task was treated as a supervised learning problem since the datasets contained true and fake labels. The initial steps included preprocessing, exploratory data analysis, merging, and shuffling the data using Pandas. Word2Vec was used to vectorize the news content by capturing semantic meanings, trained first on a BBC news dataset. Machine learning models such as Logistic Regression and Random Forest were then applied, and their performance was measured using accuracy, precision, recall, and f1-score. Finally, a confusion matrix was created to visualize the predictions, and the models were saved using the pickle library for future use.

Introduction

The rise of social media and digital platforms has made the spread of information faster than ever before. While this has many benefits, it has also opened the door for fake news to spread widely, often misleading people and creating confusion. Detecting fake news has therefore become an important problem, and this project aims to address it by applying machine learning techniques to automatically identify whether a news article is genuine or fake.
To work on this problem, we used a dataset compiled from real-world sources. The genuine articles were collected from Reuters.com, a globally trusted news website, while the fake news articles were gathered from various unreliable sources identified by Politifact (a U.S.-based fact-checking organization) and Wikipedia. The dataset contains articles on a wide range of topics, with a major focus on politics and world news. For wider accessibility, the dataset is also available on Kaggle at: www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets.

In terms of technology, this project makes use of Python as the main programming language, along with popular libraries such as Pandas for data preprocessing and cleaning, Scikit-learn for model building and evaluation, and Word2Vec for text vectorization. Word2Vec helps capture the semantic meaning of words and was trained on a BBC news dataset before applying it to our main dataset. For classification, we used machine learning algorithms like Logistic Regression and Random Forest, both of which are well-suited for supervised learning tasks.

The procedure began with basic data preprocessing and exploratory data analysis (EDA), followed by merging the datasets and shuffling them to remove any bias. Once the data was cleaned and prepared, Word2Vec was used to transform the textual data into numerical vectors that could be fed into the models. The dataset was then split into training and testing sets to properly evaluate model performance. Accuracy, precision, recall, and F1-score were calculated to understand how well the models were performing, and a confusion matrix was used to visualize the predictions. Finally, the trained models were serialized using the pickle library so that they could be reused later without retraining.

The purpose of this project is to explore how machine learning can be used to tackle the pre-

-ssing issue of misinformation. By building a system that can automatically classify news articles as real or fake, we aim to take a small step toward reducing the impact of fake news in society. While this project is limited to the dataset used, it provides a foundation for more advanced models that can be applied to larger, more diverse datasets in the future.

Several topics were covered as part of this internship in the first two weeks such as Python basics, Numpy and Pandas, Machine Learning Overview and Model Testing, LLM Fundamentals and in the last and the most important Communication Skills.

Objective

- To build a machine learning model that can automatically classify news articles as real or fake.
- To illustrate the importance of data preprocessing, EDA, and feature representation in handling text-based datasets.
- To compare the performance of different algorithms (Logistic Regression and Random Forest) using standard metrics like accuracy, precision, recall, and F1-score.
- To demonstrate the use of Word2Vec for capturing semantic meaning in textual data.
- To show how trained models can be saved and reused through serialization using the pickle library.

Methodology

The project on fake news detection was carried out step by step, beginning with data collection and ending with model building, evaluation, and saving. Each stage was carefully designed to ensure the dataset was handled properly and the models were trained effectively. Below is a detailed explanation of the entire workflow.

1. Data Collection

The dataset used in this project was compiled from real-world sources. The genuine news articles were scraped from Reuters.com, a trusted and reputed news website. On the other hand, fake news articles were gathered from unreliable platforms identified by Politifact and Wikipedia. These articles cover a range of topics, with a strong focus on politics and world news.

2. Tools and Libraries Used

- Python – Main programming language.
- Pandas, NumPy – For data cleaning, manipulation, and preprocessing.
- Matplotlib/Seaborn – For visualization during Exploratory Data Analysis (EDA).
- Scikit-learn – For dataset splitting, model training, and evaluation.
- Word2Vec (gensim) – For text vectorization and embedding.
- Pickle – For saving and reusing trained models.

3. Data Preprocessing and Cleaning

- Merged the two datasets (real and fake news).

- Removed unwanted symbols, punctuation, and irrelevant characters from the news content.
- Lowercased all text for uniformity.
- Shuffled the dataset to avoid any bias in training and testing.
- Checked for missing values and duplicates, and handled them accordingly.

4. Exploratory Data Analysis (EDA)

- Visualized the distribution of fake vs. real news.
- Checked word frequencies and common terms in both categories.
- Observed article lengths and token counts to understand dataset characteristics.

5. Text Vectorization

To feed text into machine learning models, we needed a numerical representation. We used Word2Vec, which captures semantic meaning by embedding words into vector space. Before applying it to the dataset, it was first trained on a BBC news dataset to improve accuracy.

6. Model Building and Selection

We tested two different supervised learning algorithms:

- Logistic Regression – A simple yet effective baseline model for classification tasks.
- Random Forest Classifier – A more robust ensemble model for better performance on complex data.

The dataset was split into training (75%) and testing (25%) sets using Scikit-learn. This ensured that the models were trained on one portion of the data and evaluated on unseen data to check generalization.

7. Model Evaluation

The models were evaluated using the following metrics:

- Accuracy Score – To check overall correctness.
- Precision, Recall, F1-score – To balance false positives and false negatives.
- Confusion Matrix – To visualize correct vs. incorrect classifications.

8. Model Serialization

After training and evaluation, the models were serialized using the pickle library. This allows them to be saved and reused later without retraining.

9. GitHub Repository

All the Python code, preprocessing scripts, and model files have been uploaded to GitHub for reference.

https://github.com/Adiborty-Code/IDEAS-TIH-ISI-Kolkata-Project-Repo

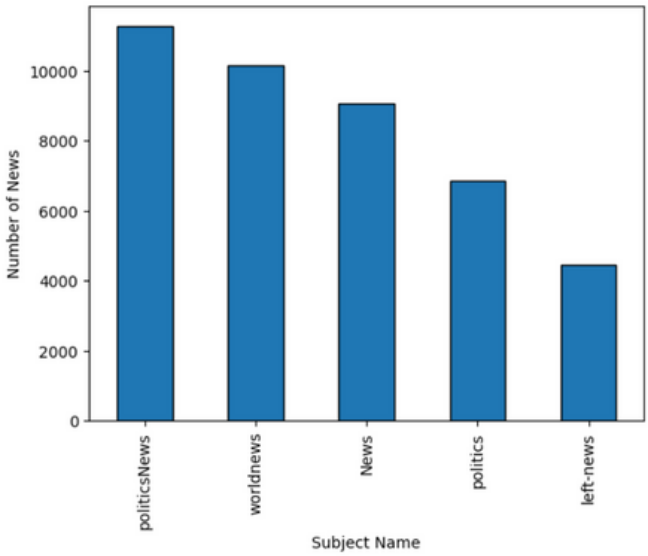
Data Analysis and Results

This section presents the outcomes of the project in the form of descriptive analysis, inferential analysis, and machine learning model comparisons. The results are supported by tables and visualizations to give a clear understanding of the dataset characteristics and model performance.

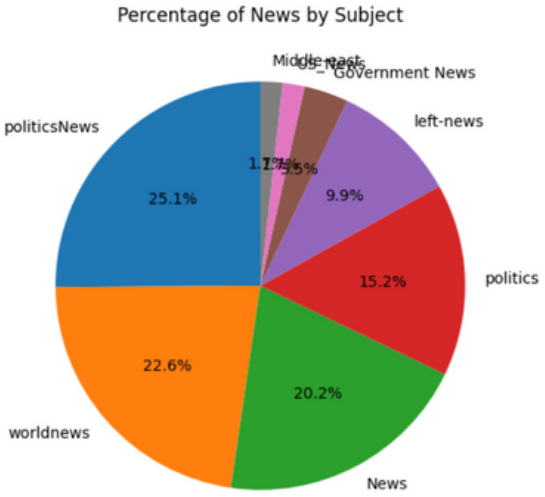
1. Descriptive Analysis

To understand the dataset better, an exploratory analysis was carried out. The dataset contained multiple subjects, and their distribution was studied.

- The top 5 most common subjects in the dataset were identified and plotted in a bar chart. This visualization clearly shows which topics dominate the dataset and helps understand the data composition.



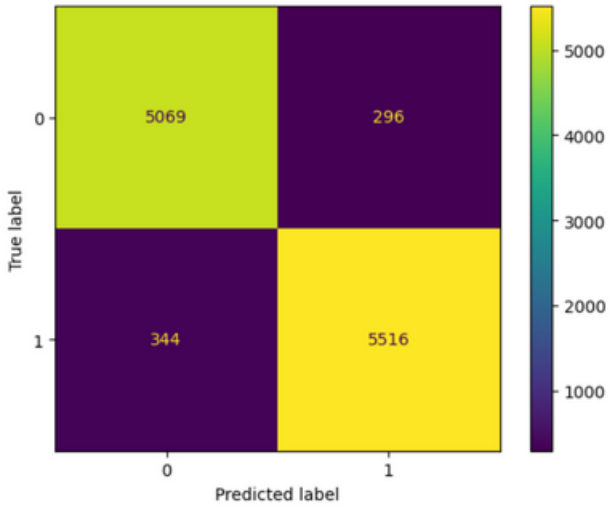
- A pie chart was generated to show the percentage distribution of news articles by subject. This gives a holistic view of how different categories contribute to the dataset.

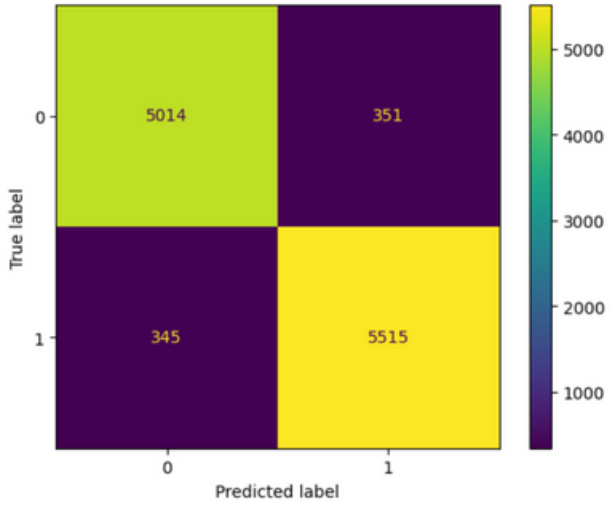


From these visualizations, it was observed that most of the news articles were concentrated in the areas of politics and world news, while other topics had comparatively fewer articles. This indicates that the dataset has a bias toward political content, which aligns with the common sources of fake news.

2. Inferential Analysis

To evaluate the models, we used classification metrics such as accuracy, precision, recall, and F1-score. A confusion matrix was also generated to compare the predicted labels with the actual labels.





From the above confusion matrices (Logistic Regression - Above, Random Forest - Below), it was found that the models performed fairly well in detecting both real and fake news, though a few misclassifications occurred. This is expected in text classification problems due to the complexity of language and overlapping word usage between real and fake articles.

3. Comparative Analysis of Machine Learning Models

The project tested two supervised machine learning algorithms: Logistic Regression and Random Forest. Their performance was compared using standard evaluation metrics.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.943	0.949	0.941	0.945
Random Forest	0.938	0.940	0.941	0.941

From the comparative analysis, it was observed that:

- Logistic Regression achieved an accuracy of 0.943, making it the better-performing model in this case.
- Random Forest achieved an accuracy of 0.938, performing slightly lower than Logistic Regression despite being an ensemble method.

This shows that while ensemble models like Random Forest often handle non-linearity and complex data patterns effectively, in this dataset Logistic Regression outperformed it. This highlights that simpler models can sometimes work better for text classification problems, especially when the features are well-preprocessed and vectorized.

Conclusion

This project explored the problem of fake news detection using machine learning techniques. The dataset was compiled from reliable and unreliable sources, giving a balanced set of genuine and fake articles for analysis. The data went through a proper preprocessing pipeline including cleaning, exploratory analysis, merging, shuffling, and vectorization using Word2Vec.

Two machine learning models—Logistic Regression and Random Forest—were trained and tested. Based on the results, Logistic Regression achieved the highest accuracy of 0.943, while Random Forest closely followed with 0.938. The confusion matrices and evaluation metrics confirmed that both models were effective in distinguishing between real and fake news. Interestingly, Logistic Regression, despite being a simpler model, slightly outperformed Random Forest. This shows that well-preprocessed text data can often be handled effectively with simpler algorithms.

The findings highlight that machine learning can indeed provide a strong foundation for building systems that automatically classify news articles. However, challenges remain, especially since fake news evolves continuously, and models trained on one dataset may not always generalize well to newer content.

Recommendations for Future Work

- Try to enhance the model's accuracy by using boosting methods such as AdaBoost, Gradient Boosting, or XGBoost.
- Use TF-IDF or other vectorizers instead of Word2Vec and study how much it impacts model performance.
- Expand the dataset to include more diverse sources and subjects beyond politics and world news.
- Explore deep learning approaches (e.g., LSTMs, GRUs, or Transformer-based models like BERT) for improved classification.

Appendix

1. https://matplotlib.org/stable/api/index.html (Matplotlib API Reference)
2. https://scikit-learn.org/stable/api/index.html (Scikit-Learn API Reference)
3. Python Data Analysis: Data Wrangling with Pandas, NumPy, & Jupyter by Wes McKinney — Third

Github Repository of this project :- https://github.com/Adiborty-Code/IDEAS-TIH-ISI-Kolkata-Project-Repo