A PROJECT REPORT

on

"FAKE NEWS DETECTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE ENGINEERING

BY

SUSHIT SEKHAR NAYAK	22052338
SUHAN MOHANTY	22052337
SWAYAMSHU S. CHOUDHURY	22052340
SAIK AFTAB	22052403

UNDER THE GUIDANCE OF
Prof. KUNAL ANAND

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

April 2025

A PROJECT REPORT

on

"FAKE NEWS DETECTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE ENGINEERING

BY

SUSHIT SEKHAR NAYAK	22052338
SUHAN MOHANTY	22052337
SWAYAMSHU S. CHOUDHURY	22052340
SAIK AFTAB	22052403

UNDER THE GUIDANCE OF
Prof. KUNAL ANAND

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"FAKE NEWS DETECTION"

submitted by

SUSHIT SEKHAR NAYAK	22052338
SUHAN MOHANTY	22052337
SWAYAMSHU S. CHOUDHURY	22052340
SAIK AFTAB	22052403

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 01/04/25

Prof. Kunal Anand
Project Guide

Acknowledgements

We are profoundly grateful to **Prof. Kunal Anand** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....…

SUSHIT SEKHAR NAYAK
SUHAN MOHANTY
SWAYAMSHU S. CHOUDHURY
SAIK AFTAB

ABSTRACT

The rapid expansion of digital media and online platforms has resulted in an increase in misleading and fabricated news. False information spreads quickly, affecting public perception, political landscapes, and social harmony. This project focuses on developing a Fake News Detection System that leverages Machine Learning to classify news articles as either real or fake based on textual data.

By incorporating Natural Language Processing (NLP) techniques, the system analyzes text patterns and applies various machine learning algorithms to predict news authenticity. Several classification models, including Logistic Regression, Decision Trees, Support Vector Machines (SVM), and Neural Networks, are trained and tested using a diverse dataset to enhance accuracy. The experimental outcomes highlight the effectiveness of AI-driven approaches in detecting misinformation and reducing the spread of deceptive content online.

Keywords:

Fake News Identification, Machine Learning, NLP, Text Analytics, Deceptive Content Detection

Contents

List of Figures

Chapter 1

Introduction

In the digital era, the rapid growth of online platforms has led to an unprecedented surge in the dissemination of news and information. While this has enabled easy access to global events, it has also paved the way for the rampant spread of **misinformation and fake news**. The ability to differentiate between authentic and deceptive news has become crucial, especially as false narratives can influence public opinion, manipulate elections, and cause social unrest. Traditional fact-checking methods are often time-consuming and inadequate in countering the rapid spread of misinformation. This highlights the pressing need for an **automated Fake News Detection System** that can analyze and classify news content efficiently.

Despite the availability of several fact-checking tools and manual verification systems, **existing solutions** suffer from major limitations, including **subjectivity, delayed verification, and scalability issues**. Many platforms rely on human fact-checkers or crowdsourced reporting, which can be inefficient and prone to bias. Additionally, most traditional filtering mechanisms struggle to adapt to evolving patterns of misinformation, making them ineffective in real-time detection. To address these shortcomings, this project leverages **Machine Learning (ML) and Natural Language Processing (NLP)** to automate the detection of fake news with improved accuracy and speed.

This report is structured to provide a comprehensive overview of the project. The following sections outline the **fundamental concepts** related to fake news detection, **problem formulation, requirement specifications, system design**, and **methodology** used for implementation. The report also includes an **evaluation of model performance, results analysis**, and a discussion on the **future scope** of the project.

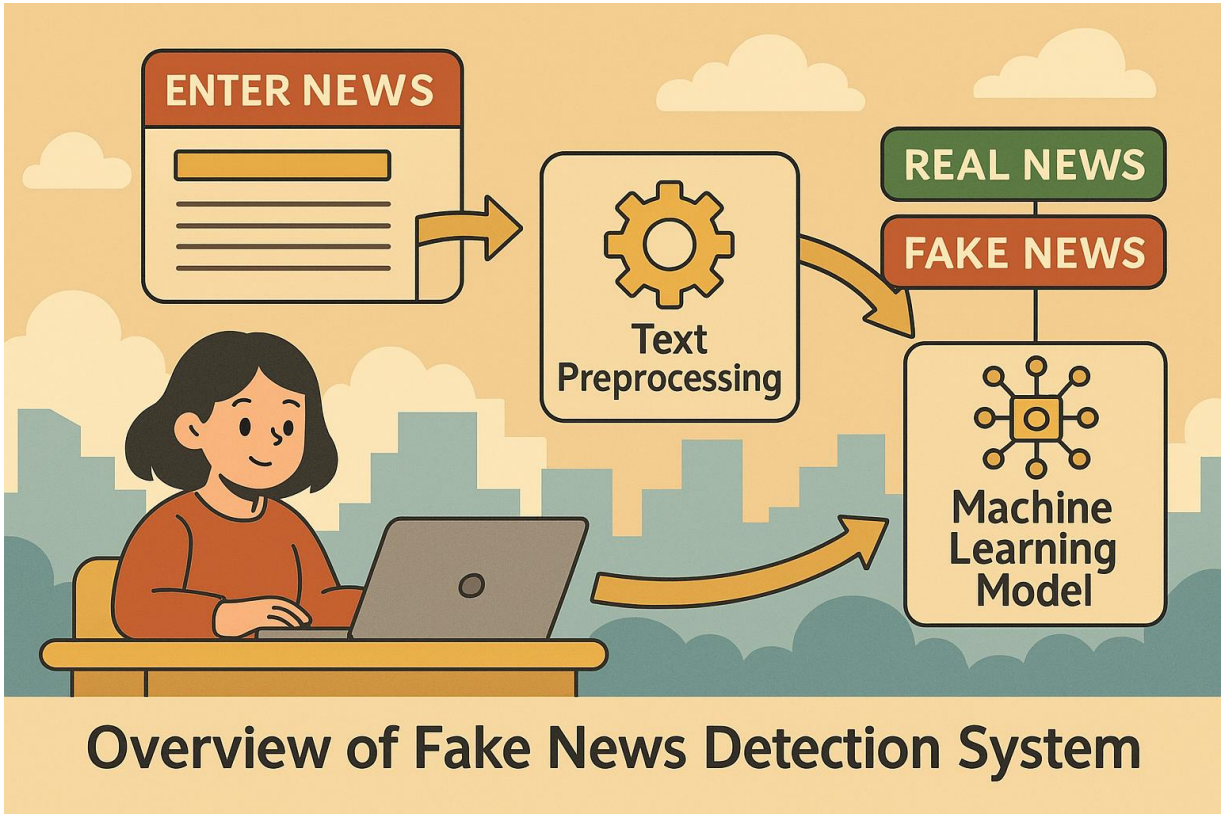


Figure 1: Overview of Fake News Detection System

Chapter 2

Basic Concepts/ Literature Review

In recent years, the rapid expansion of digital platforms has led to an explosion of information dissemination. However, this has also resulted in the rise of misinformation and fake news, which pose a threat to public trust and decision-making. Tackling this issue has become a significant concern for both the public and private sectors. To address this challenge, advanced technologies like Natural Language Processing (NLP) and Machine Learning (ML) are increasingly being employed to build automated fake news detection systems.

This chapter provides a thorough overview of the foundational technologies, algorithms, and relevant research that contribute to the development of a fake news detection system. The subsequent subsections explain the basic tools and techniques applied in this project and highlight key academic efforts in the same direction.

2.1 Natural Language Processing (NLP)

Natural Language Processing is a domain of Artificial Intelligence that allows machines to understand, interpret, and generate human language. In fake news detection, NLP plays a crucial role in analyzing and preprocessing textual content. Techniques such as tokenization, stemming, lemmatization, and stop word removal help clean and structure data for further analysis.

2.2 Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical method used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It is a widely-used text feature extraction method in classification tasks. In fake news detection, it helps in identifying significant terms that may indicate the authenticity of news content.

2.3 Machine Learning Algorithms

Several machine learning models are used for classification tasks in this project:

- **Logistic Regression:** A supervised learning algorithm used for binary classification. It predicts the probability that a given input belongs to a particular class.
- **Decision Tree Classifier:** A tree-based algorithm that splits the dataset into subsets based on feature value tests. It is interpretable and effective for categorical feature data.
- **Gradient Boosting Classifier:** An ensemble method that builds models sequentially, each correcting the errors of its predecessor. It performs well in various classification tasks and handles complex data relationships.
- **Random Forest Classifier:** An ensemble learning method that combines multiple decision trees to improve classification accuracy and reduce overfitting.

2.4 Model Evaluation Metrics

To assess the performance of classification models, the following metrics are commonly used:

- **Accuracy:** Percentage of correctly classified samples.

- **Precision & Recall:** Measure the model's ability to correctly identify fake or real news without misclassification.
- **F1 Score:** Harmonic mean of precision and recall.
- **Confusion Matrix:** A table summarizing the classification results.

2.5 Related Work and Literature Review

Numerous studies have explored automated detection of fake news using machine learning. Early research utilized simple bag-of-words models, while recent works incorporate deep learning and contextual embeddings like BERT. The common objective remains consistent—to develop robust systems capable of detecting misinformation with high accuracy.

Several benchmark datasets, such as LIAR, FakeNewsNet, and the Kaggle Fake/Real News dataset, have been extensively used in academic and industrial projects. These contributions have laid the foundation for continuous improvements in this domain.

Chapter 3

Problem Statement / Requirement Specifications

The digital era has witnessed an unprecedented rise in the dissemination of false information across online platforms. Fake news has the potential to mislead public opinion, disrupt social harmony, and manipulate political or economic landscapes. Manual fact-checking is both time-consuming and infeasible for large-scale information verification, which highlights the urgent need for an automated system capable of detecting such news articles.

The objective of this project is to design and develop a machine learning-based solution that can differentiate between authentic and fabricated news content. The system leverages multiple supervised learning algorithms to train on a large dataset of labeled news articles and accurately predict the authenticity of new input.

3.1 Project Planning

The project was developed following a structured plan to ensure clarity and effectiveness. The primary steps in project planning included:

1. Identifying the real-world problem and understanding the impact of fake news.
2. Collecting and exploring the dataset comprising real and fake news articles.
3. Preprocessing the data for cleaning and normalization.
4. Applying vectorization techniques such as TF-IDF to convert text into numerical features.
5. Training and testing multiple supervised machine learning models:
    - Logistic Regression
    - Decision Tree
    - Gradient Boosting Classifier
    - Random Forest
6. Evaluating model performance using metrics like accuracy, precision, recall, and F1-score.
7. Developing a user-friendly interface to input news content for testing.
8. Visualizing the results and comparing model outcomes.

3.2 Project Analysis (SRS)

The requirements for this system were gathered from existing challenges in content verification and insights into current misinformation trends. An initial analysis of user expectations and dataset characteristics was conducted to establish a robust system structure.

Functional Requirements:

- Users must be able to input any news text and receive a prediction (Real or Fake).
- The system should classify based on previously trained models.
- It should display prediction results and allow comparison of model performances.

Non-functional Requirements:

- The system should have high accuracy and low latency.
- It must be easy to use and should handle large input data efficiently.
- The application should be compatible with common environments like Jupyter Notebooks or Web Browsers (if deployed).

Ambiguity Analysis: Careful inspection was done to remove overlapping features, unbalanced datasets, or noisy text data. Vectorization and tokenization steps were refined to avoid word duplication or misrepresentation. The confusion matrix and performance metrics were used to ensure fair evaluation.

3.3 System Design

3.3.1 Design Constraints

The project was developed using the following tools and environment:

1. **Hardware Requirements:**

   - Processor: Intel i5 or above
   - RAM: Minimum 8 GB
   - Storage: At least 2 GB free for dataset and dependencies

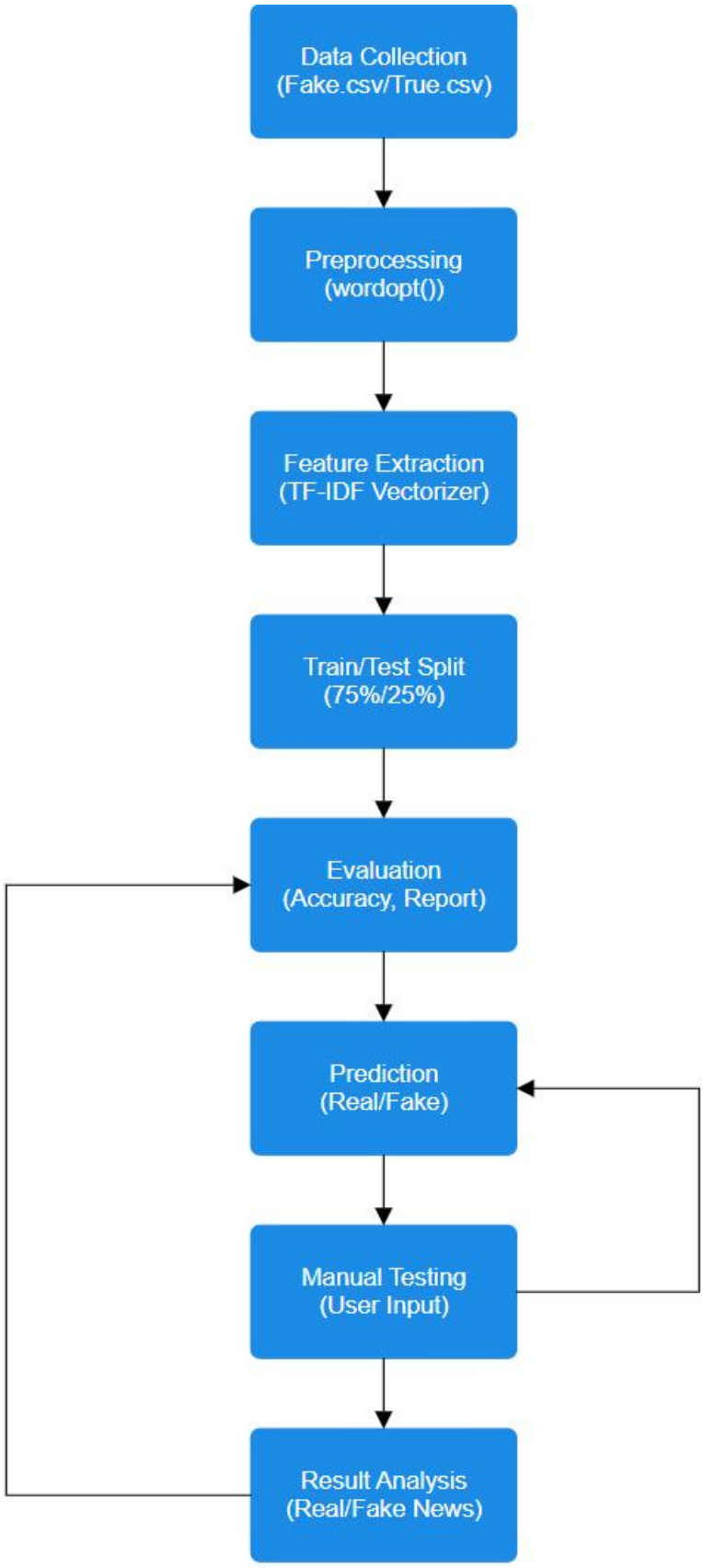
2. **Software Requirements:**

   - Python 3.8+
   - Jupyter Notebook / Google Colab, Smart Draw
   - Libraries: pandas, numpy, sklearn, matplotlib, seaborn, nltk, xgboost
   - Dataset: [Fake.csv, True.csv] containing over 40,000 news articles

3. **Setup Constraints:**

   - Requires internet access to install Python packages and dependencies
   - Needs preprocessing steps like tokenization, stopword removal, and TF-IDF vectorization before model training.

3.3.2 Block Diagram



The block diagram illustrates the entire workflow of the Fake News Detection system, beginning from data acquisition to the final prediction output.

1. Data Collection:

The system uses two labeled datasets—Fake.csv and True.csv—which contain fake and real news articles respectively. These serve as the foundation for model training.

2. **Preprocessing:**

   Raw text data is cleaned using a custom wordopt function. This step removes punctuation, converts text to lowercase, eliminates stopwords, and prepares the data for analysis.

3. **Feature Extraction:**

   The cleaned text is converted into numerical format using TF-IDF Vectorizer, which transforms text into feature vectors based on word importance.

4. **Train/Test Split:**

   The dataset is split into 75% for training and 25% for testing, ensuring the model is trained on a major portion and evaluated on unseen data.

5. **Model Training:**

   Four machine learning models are trained using the training data:

   - Logistic Regression (LR)
   - Decision Tree (DT)
   - Gradient Boosting Classifier (GB)
   - Random Forest (RF)

6. **Evaluation:**

   Each trained model is evaluated using metrics like accuracy and classification reports to determine their performance.

7. **Prediction:**

   The best-performing model is used for making predictions on new user inputs.

8. **Manual Testing:**

   A user interface allows manual input of news text. The system then outputs whether the news is Fake or Real based on the trained model.

Each block represents a modular component of the system, and arrows indicate the logical flow of data and processing.

Chapter 4

Implementation

This chapter outlines the complete implementation approach taken during the development of the Fake News Detection system. The project was designed and executed using Python and machine learning techniques, with a focus on natural language processing (NLP) for effective classification of real and fake news.

4.1 Methodology

The following steps were systematically followed to build the Fake News Detection system:

1. **Data Collection:** Two publicly available datasets were used — Fake.csv for false news and True.csv for legitimate news articles.
2. **Preprocessing:** The text data was cleaned using custom preprocessing functions to remove unwanted symbols, punctuation, stop words, and apply lowercasing.
3. **Feature Extraction:** The preprocessed text was transformed into numerical format using the TF-IDF vectorizer which gives importance to rare words while minimizing the weight of frequently occurring ones.
4. **Train-Test Split:** The data was split into 75% training and 25% testing to validate the performance of the model.
5. **Model Building:** Four popular classification models were implemented and trained:

    - Logistic Regression
    - Decision Tree Classifier
    - Gradient Boosting Classifier
    - Random Forest Classifier

6. **Prediction & Evaluation:** After training, the models were evaluated using metrics such as accuracy, classification reports, and confusion matrices.

7. **Manual Input Testing:** An interactive text box was used in the front-end interface for user input to test real-time classification.

4.2 Testing OR Verification Plan

The system was verified through model evaluation and manual testing. Below is a sample test plan:

Test ID	Test Case Title	Test Condition	System Behaviour	Expected Result
T01	Real News Classification	Input a known true news article	System classifies it correctly	Output: Real
T02	Fake News Classification	Input a fabricated news article	System classifies it correctly	Output: Fake
T03	Empty Input	Leave the input box empty	System shows validation or error	Prompt: "Please enter news text"
T04	Irrelevant Input	Enter nonsensical text like "asdjkh123"	System processes input	Output: Fake (likely due to low match)
T05	Mixed Real and Fake Phrases	Input a mix of real and fake lines	System evaluates context	Output based on dominant features

4.3 Result Analysis OR Screenshots

The output of the model training and testing was visually represented using accuracy comparison graphs. Screenshots of the front-end predictions were also captured.

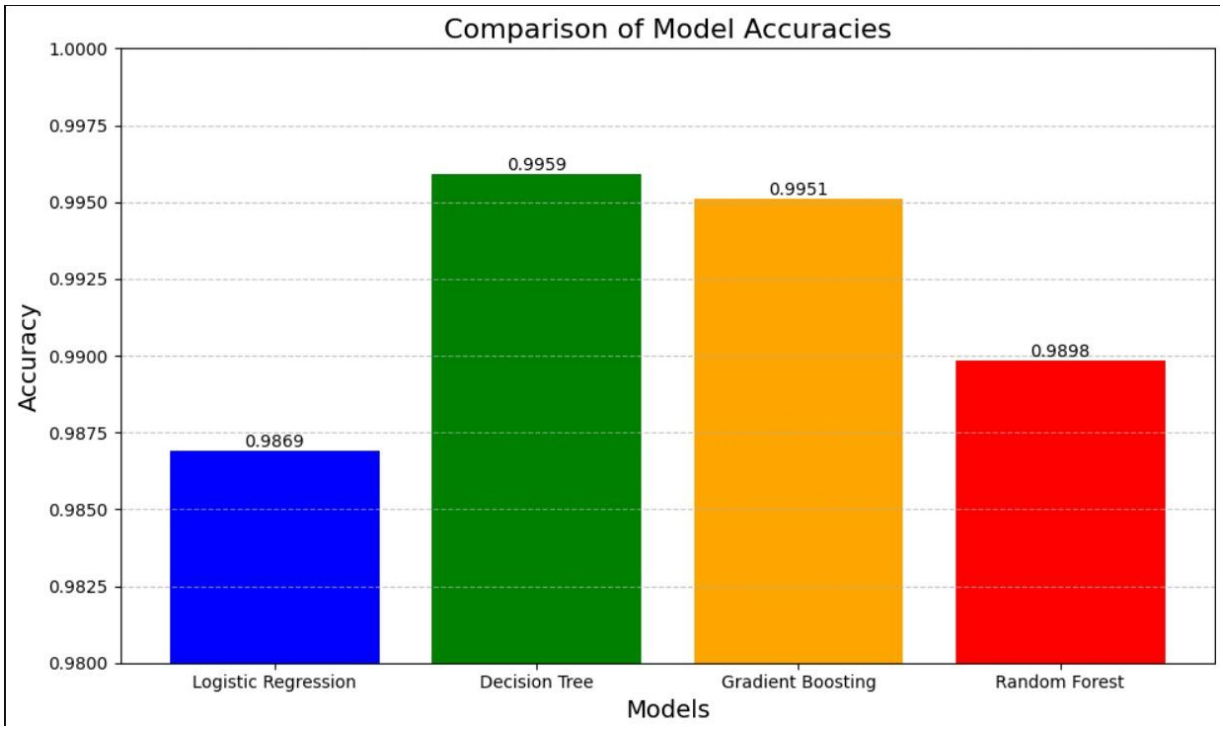
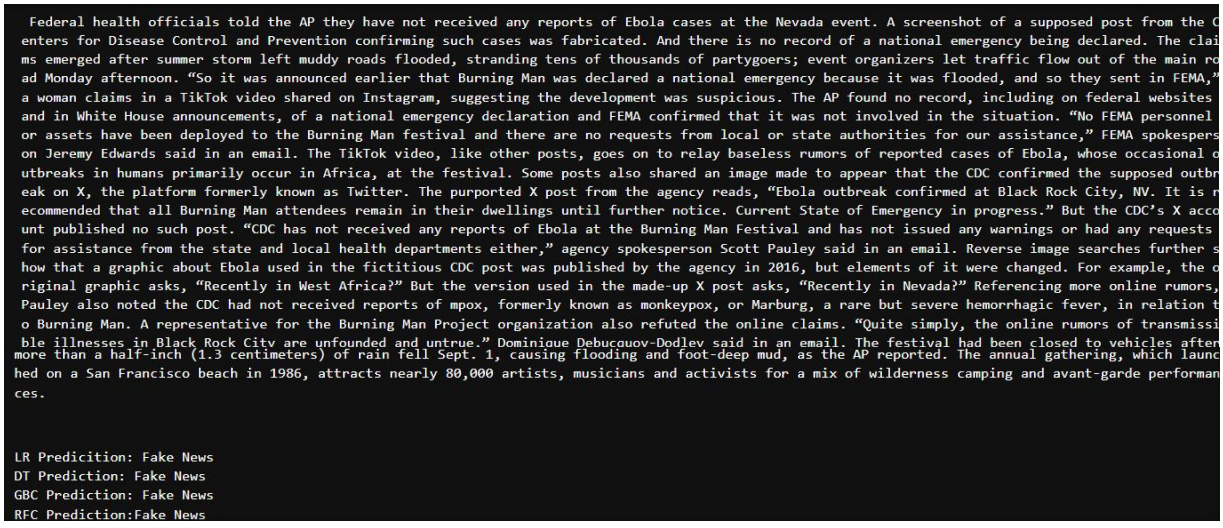


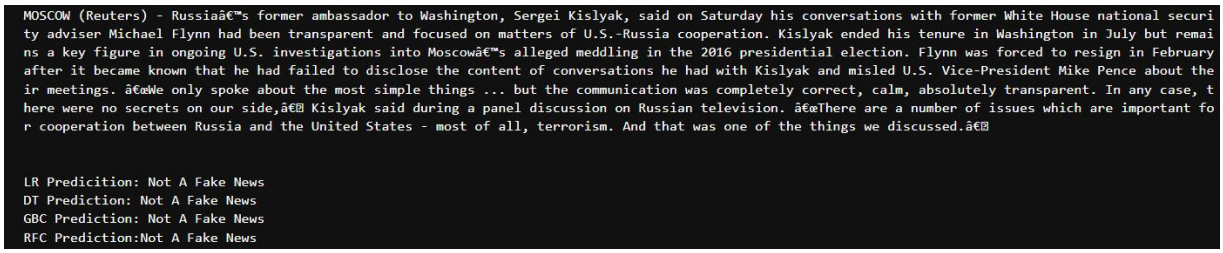
Figure 2: Accuracy Comparision of each model

Prediction Screenshots:

(Fake News Check):



(Real News Check):



4.4 Quality Assurance

To ensure the system was built and tested effectively:

- Code was written following clean coding practices and reviewed in stages.
- The ML models were tested on a large dataset with over 40,000 combined entries.
- Multiple iterations of training and validation ensured robustness.
- Cross-validation techniques were used during development to prevent overfitting.
- TF-IDF vectorization and text normalization techniques were applied consistently to maintain input quality.

Chapter 5

Standards Adopted

This chapter outlines the various standards that were considered and followed during the design, coding, and testing phases of the Fake News Detection system. Following established standards ensures that the project is maintainable, readable, and of high quality.

5.1 Design Standards

During the design phase of the project, standard practices were followed to ensure consistency and clarity. The following design standards and guidelines were adopted:

- UML (Unified Modeling Language): Widely accepted modeling language was used to represent the system components such as use case diagrams, class diagrams, sequence diagrams, and activity diagrams.

- IEEE 1016-2009: Recommended guidelines for system and software design descriptions were considered while preparing design documentation.
- Database Design Standards: Although this project is primarily machine learning-based, if any storage structure is used, normalization rules and entity-relationship modeling principles were applied.
- Modular Design: The system was structured in a modular way to promote separation of concerns and easier debugging or maintenance.

5.2 Coding Standards

To maintain clean, efficient, and understandable code, several best practices and coding standards were followed throughout the development:

- Proper naming conventions were used for variables, functions, and classes (e.g., camelCase for variables and PascalCase for class names).
- Functions were kept short and modular, focusing on a single responsibility per function.
- Consistent indentation and spacing was maintained to improve readability.
- Comments and docstrings were included to explain the purpose of code blocks and individual functions.
- PEP 8 (Python Enhancement Proposal 8), the official coding standard for Python, was used to guide the structure and formatting of Python code.
- Code was segmented logically, and try-except blocks were used for proper error handling.

…...

5.3 Testing Standards

To ensure the quality and reliability of the system, standard testing methodologies and guidelines were followed:

- IEEE 829 (Standard for Software Test Documentation) was referred to for preparing test plans and test cases, including Test ID, conditions, expected outcomes, and behavior.
- Testing was performed using unit testing and manual validation.

- Precision, recall, accuracy, and F1-score metrics were used to evaluate the performance of each machine learning model.
- A significant portion of the data was kept aside as a test set to ensure the models were evaluated on unseen inputs.
- Cross-validation was used to ensure the model's generalizability and avoid overfitting.

Real-time testing was carried out through the web interface for user input validation.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

The project titled Fake News Detection using Machine Learning successfully demonstrates the ability of supervised learning models to classify news articles as either real or fake. Throughout the development process, various stages such as data preprocessing, feature extraction using TF-IDF, model training with multiple algorithms (Logistic Regression, Decision Tree, Gradient Boosting, and Random Forest), and evaluation were carried out to ensure the reliability of the system.

The implemented system helps in automating the detection of misinformation by providing a straightforward interface where users can input any news content and receive an immediate classification. Among the tested models, ensemble methods showed higher accuracy, validating the importance of combining multiple decision mechanisms in classification tasks.

This project proves to be a practical approach in addressing the growing concern of fake news circulation, particularly in digital and social media platforms.

6.2 Future Scope

While the current system is efficient for text-based binary classification, there are several enhancements that can be explored in future iterations of this work:

- Deep Learning Models: Implementation of models like LSTM, BERT, or transformers for better contextual understanding and improved accuracy.
- Multilingual Support: Expanding the system to support news articles written in multiple regional and global languages.
- Fake News Source Detection: Incorporating source verification mechanisms to identify and flag unreliable publishers or websites.
- Real-time Integration: Deploying the model into web browsers, news apps, or social media platforms to provide real-time news verification for end-users.
- Visual and Multimedia Content Analysis: Including image and video content verification, which is a growing concern in modern misinformation.

Overall, the project lays a strong foundation and opens several avenues for future enhancements that can contribute toward building a more trustworthy digital information ecosystem.

References

[1] Scikit-learn: Machine Learning in Python, [Online]. Available: https://scikit-learn.org/

[2] GeeksforGeeks, “Fake News Detection using Machine Learning,” [Online]. Available: https://www.geeksforgeeks.org/fake-news-detection-using-machine-learning/?ref=gcse_ind

[3] GitHub, “Fake News Detection Project,” [Online]. Available: https://github.com/

[4] Kaggle, “Fake and Real News Dataset,” [Online]. Available: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[5] Python Software Foundation, “Python Language Reference,” [Online]. Available: https://www.python.org/

[6] IEEE Standards Association, “IEEE Standard for Software Verification and Validation,” [Online]. Available: https://standards.ieee.org/

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

FAKE NEWS DETECTION

SAIK AFTAB
22052403

Abstract: The aim of this project is to detect fake news from textual content using machine learning algorithms. By leveraging various classification models and natural language processing techniques, the system classifies news as real or fake. The dataset was sourced from open platforms and preprocessed before model training.

Individual contribution and findings: I was responsible for the Data Collection and Preprocessing phase of the project. This included exploring and acquiring the Fake.csv and True.csv datasets from Kaggle and GitHub repositories. My tasks involved inspecting dataset structure, verifying data quality, and writing preprocessing logic using the wordopt() function to clean text data—removing punctuation, converting to lowercase, removing stop words, etc.

I planned this phase by researching standard NLP preprocessing steps and comparing various libraries. I implemented the cleaning logic using Python and Pandas, and ensured compatibility for further feature extraction. Through this, I learned about data wrangling challenges and how data quality directly impacts model accuracy. The preprocessing pipeline was finalized after several iterations and testing on a sample subset.

Individual contribution to project report preparation: I contributed to Chapter 1 (Introduction) and Chapter 2 (Basic Concepts) of the group project report, including details about the problem domain, the importance of fake news detection, and descriptions of preprocessing steps and TF-IDF.

Individual contribution for project presentation and demonstration: I handled the explanation of the data collection and cleaning phase during the presentation. I demonstrated how raw text data is preprocessed before feature extraction and explained the logic behind the wordopt() function with examples.

Full Signature of Supervisor: ……

Full signature of the student: ……..

SUSHIT SEKHAR NAYAK
22052338

Abstract: The aim of this project is to detect fake news from textual content using machine learning algorithms. By leveraging various classification models and natural language processing techniques, the system classifies news as real or fake. The dataset was sourced from open platforms and preprocessed before model training.

Individual contribution and findings: I was responsible for the Feature Extraction and Vectorization phase of the project. After the text data was cleaned, I applied the TF-IDF Vectorizer to convert the processed textual content into numerical features. This transformation was crucial to enable the machine learning models to interpret the data.

My planning involved comparing TF-IDF with other techniques like Bag of Words and choosing the most effective one for this context. I implemented the TF-IDF logic using Scikit-learn's TfidfVectorizer, set appropriate parameters such as max_features, and verified the shape and type of resulting feature matrices.

I learned how essential vectorization is in NLP pipelines and understood how feature extraction impacts model accuracy. I also helped prepare the dataset for training and testing using train_test_split().

Individual contribution to project report preparation: I contributed to Chapter 3 (System Design) and Chapter 4.1 (Methodology) of the group report. I detailed the pipeline design, preprocessing modules, and explanation of the vectorization logic using TF-IDF.

Individual contribution for project presentation and demonstration: I explained the TF-IDF feature extraction step during the presentation and demonstrated how raw text transforms into numerical features. I highlighted its importance for improving ML model performance.

Full Signature of Supervisor: …….

Full signature of the student: ……..

SUHAN MOHANTY
22052337

Abstract: The aim of this project is to detect fake news from textual content using machine learning algorithms. By leveraging various classification models and natural language processing techniques, the system classifies news as real or fake. The dataset was sourced from open platforms and preprocessed before model training.

Individual contribution and findings: I was responsible for the Model Training and Evaluation phase. I implemented four different classification models: Logistic Regression, Decision Tree, Gradient Boosting, and Random Forest using Scikit-learn. Each model was trained using the TF-IDF features from the previous step.

I planned and tested each model with default and tuned parameters, then compared their performance using accuracy, confusion matrix, and classification reports. This helped us decide which models performed best for the task.

I learned how to interpret various evaluation metrics, handle overfitting, and make comparisons between different ML algorithms. Among all, Logistic Regression performed the best, which was selected for integration with the frontend.

Individual contribution to project report preparation: I contributed to Chapter 4.2 (Testing) and Chapter 4.3 (Result Analysis) of the project report. I created the test case table and presented graphs of accuracy and confusion matrix.

Individual contribution for project presentation and demonstration: I presented the model evaluation results and explained which model we finalized and why. I also walked through accuracy comparison and classification report output.

Full Signature of Supervisor: ……

Full signature of the student: ……..

SWAYAMSHU S. CHOUDHURY
22052340

Abstract: The aim of this project is to detect fake news from textual content using machine learning algorithms. By leveraging various classification models and natural language processing techniques, the system classifies news as real or fake. The dataset was sourced from open platforms and preprocessed before model training.

Individual contribution and findings: I was responsible for the Frontend Testing and Integration of the project. I created the user interface where users can input custom news text, which is passed through the preprocessing and prediction pipeline. I ensured that the model integrates smoothly and returns results clearly as “Real” or “Fake”.

My planning involved testing the pipeline with multiple sample inputs, handling text encoding issues, and ensuring results were accurate and understandable. I also helped design the final output display and corrected UI elements.

I gained experience in connecting ML models with user interfaces, performing end-to-end validation, and testing edge cases to ensure the system behaves correctly for different types of input text.

Individual contribution to project report preparation: I contributed to Chapter 4.4 (Quality Assurance) and Chapter 5 (Standards Adopted) of the group report, including frontend quality testing and interface output screenshots.

Individual contribution for project presentation and demonstration: I demonstrated the frontend interface by inputting real-time news text and showing how the system identifies it as real or fake. I also answered questions about output handling and frontend logic.

Full Signature of Supervisor: …….

Full signature of the student: ……..

TURNITIN PLAGIARISM REPORT