**Phase-3 Submission Template**

**Student Name:** DEEPIKA K R

**Register Number:** 621523104022

**Institution:** Mahendra College of Engineering

**Department:** Computer science and Engineering

**Date of Submission:** 17/05/2025

**Github Repository Link:**

### **1. Introduction**

*In the age of rapid digital transformation, the internet and social media have revolutionized how information is produced, consumed, and disseminated. While this democratization of content has brought numerous benefits, it has also given rise to a troubling phenomenon — the widespread circulation of* ***fake news****. Fake news refers to* ***deliberately misleading or false information*** *presented as legitimate news, often created to deceive readers or manipulate public opinion for political, financial, or ideological gain.*

*The consequences of fake news can be profound. From* ***interfering in democratic elections*** *to* ***spreading panic during health crises*** *such as the COVID-19 pandemic, misinformation has the power to destabilize societies and erode trust in institutions. In recent years, numerous incidents have highlighted the severe impact of fake news — including fabricated political scandals, anti-vaccine misinformation, and coordinated disinformation campaigns by malicious actors.*

*Given this backdrop,* ***developing automated and intelligent systems*** *capable of detecting and flagging fake news has become a research priority in both academia and industry. Traditional methods of human verification are no longer scalable due to the* ***sheer volume of content*** *generated daily. Therefore,* ***Natural Language Processing (NLP)*** *— a field of artificial intelligence that deals with the interaction between computers and human language — emerges as a powerful solution.*

*This project leverages advanced NLP techniques and machine learning algorithms to classify news articles as* ***real or fake*** *based on their textual content. By analyzing linguistic patterns, semantic structures, and contextual cues, our system aims to mimic human intuition in identifying deceptive narratives. The model is trained on a publicly available dataset comprising thousands of labeled news articles and is further refined using vectorization techniques like* ***TF-IDF****, along with classification models such as* ***Logistic Regression, Random Forest, LSTM****, and* ***BERT****.*

*The motivation behind this work is not only technical but also* ***societal****. In a world where truth is under constant threat from digital misinformation, this project aims to contribute toward* ***safeguarding public discourse*** *by providing a scalable and accessible fake news detection tool. Through the development of this system, we hope to* ***empower individuals, organizations, and platforms*** *to discern truth from fiction more effectively.*

*In the subsequent sections, we delve into the problem definition, data preprocessing, model development, evaluation strategies, and deployment processes involved in building this system. We also present a comparative analysis of various models and propose future enhancements to make the system even more robust.*

### **2. Problem Statement**

*In today's digital era, information spreads at an unprecedented speed through social media platforms, news websites, and instant messaging applications. While this technological advancement enhances communication and accessibility, it also opens the door to the uncontrolled dissemination of* ***fake news*** *— deliberately false or misleading information disguised as legitimate journalism.*

*Fake news has emerged as a significant societal and political challenge, influencing public opinion, disrupting democratic processes, inciting violence, and undermining trust in credible institutions. The* ***2016 U.S. Presidential Election****, for example, brought global attention to the role of fake news in shaping political outcomes. Similarly, misinformation about health-related issues like vaccines and pandemics has led to public health crises and widespread panic.*

*Manually identifying and debunking fake news is not a scalable solution due to the* ***overwhelming volume of information*** *being generated every second. Fact-checking agencies, journalists, and platform moderators often struggle to keep up with the speed and scale of content circulation. This highlights the urgent need for* ***automated, intelligent systems*** *capable of detecting and filtering fake news in real time*.

### *Technical Framing of the Problem*

*From a technical standpoint, fake news detection is a* ***binary text classification problem*** *— the goal is to classify a news article as either:*

* ***Real (label: 0)***
* ***Fake (label: 1)***

*The task involves analyzing the* ***textual content*** *of the article (including the title and body) and leveraging machine learning and natural language processing techniques to learn* ***discriminative patterns*** *between real and fake news.*

*However, fake news detection poses several challenges:*

* ***Linguistic Ambiguity****: Fake news often uses emotionally charged language, partial truths, or misleading headlines that are difficult to catch using simple keyword filters.*
* ***Evolving Tactics****: As detection systems improve, creators of fake news adapt by altering writing styles and formats to evade detection.*
* ***Class Imbalance****: In real-world data, the distribution of fake vs. real news may not be balanced, requiring robust model training strategies.*
* ***Context Sensitivity****: Some statements are only false in specific contexts, which is difficult for traditional models to detect without external knowledge.*

### *Objective of the Project*

*The main objective of this project is to* ***design and develop an NLP-powered system*** *that can automatically classify news articles as fake or real. This system should:*

* *Use preprocessing and feature extraction techniques to clean and vectorize raw text data.*
* *Train and evaluate multiple machine learning and deep learning models to determine the most effective approach.*
* *Offer a user-friendly interface (via Streamlit or Gradio) for real-time fake news detection.*
* *Provide reliable evaluation metrics such as accuracy, precision, recall, and F1-score to assess model performance.*
* *Serve as a scalable solution for content moderation and misinformation filtering in online platforms.*

*By tackling the fake news problem with modern AI techniques, this project contributes toward the* ***broader mission of ensuring information integrity*** *in a digitally connected world.*

### **3. Abstract**

*In an age where digital platforms serve as the primary source of information for millions, the spread of* ***fake news*** *has emerged as a critical global challenge. Fake news — misleading or entirely fabricated content — undermines democratic institutions, manipulates public perception, and promotes social unrest. Manual verification of news content is both time-consuming and impractical given the* ***massive volume of content*** *shared daily across online platforms. As such, the need for* ***automated, intelligent systems*** *to combat misinformation has become increasingly urgent.*

*This project aims to develop a robust and scalable* ***Fake News Detection System*** *powered by* ***Natural Language Processing (NLP)*** *and* ***Machine Learning (ML)*** *techniques. The core objective is to classify news articles as either* ***real*** *or* ***fake*** *by analyzing their textual content, including headlines and article bodies. The system is trained on a publicly available dataset containing approximately* ***44,000 labeled news articles****, sourced from Kaggle. Each record includes the title, text, subject category, and binary label indicating whether the article is real or fake.*

*The project pipeline begins with comprehensive* ***data preprocessing****, involving tasks such as handling missing values, removing duplicates, tokenization, stopword removal, and lemmatization. Feature extraction is achieved through* ***TF-IDF vectorization****, which transforms raw text into numerical representations suitable for machine learning algorithms. We also explore* ***word embeddings*** *and* ***semantic features*** *to enhance context awareness.*

*Multiple classification models are implemented and evaluated, ranging from traditional algorithms like* ***Logistic Regression*** *and* ***Random Forest****, to advanced* ***deep learning architectures*** *including* ***Long Short-Term Memory (LSTM)*** *networks and* ***BERT*** *(Bidirectional Encoder Representations from Transformers). Each model is rigorously assessed using metrics such as* ***accuracy, precision, recall, F1-score****, and* ***ROC-AUC****, ensuring a thorough comparison of performance.*

*To make the system accessible and practical for real-world usage, the final model is deployed via an interactive* ***web interface using Streamlit****, allowing users to input news content and receive instant predictions. The system also displays the model’s confidence score and interpretable feedback to enhance trust and usability.*

*This project demonstrates the powerful capabilities of NLP in* ***automating misinformation detection*** *and presents a viable tool for journalists, fact-checkers, and everyday users. By combining cutting-edge technology with a user-centered design, this system contributes meaningfully toward the fight against digital disinformation and the promotion of information integrity.*

### **4. System Requirement**

*To develop and deploy a robust fake news detection system using natural language processing and machine learning, a combination of suitable* ***hardware and software resources*** *is essential. These requirements ensure smooth execution of all project components—from data preprocessing and model training to deployment and user interaction. The system requirements are outlined below:*

### **4.*1 Hardware Requirements***

| ***Component*** | ***Minimum Specification*** | ***Recommended Specification*** |
| --- | --- | --- |
| ***Processor*** | *Intel Core i5 / AMD Ryzen 5* | *Intel Core i7 / AMD Ryzen 7 or higher* |
| ***RAM*** | *4 GB* | *8 GB or more (especially for deep learning)* |
| ***Storage*** | *1 GB free space* | *SSD with 5–10 GB free space for faster access* |
| ***GPU (Optional)*** | *Not required for basic ML models* | *NVIDIA GPU (e.g., RTX 2060 or above) for faster deep learning training* |

#### ***Justification****:*

* *The fake news detection project involves working with text data and training ML/DL models. While traditional models can run efficiently on basic setups, deep learning models like* ***LSTM*** *and* ***BERT*** *are computationally intensive and benefit from* ***GPU acceleration****.*
* *Sufficient* ***RAM and storage*** *ensure smooth data processing, especially when handling large datasets and high-dimensional feature vectors.*

#### ***4.2 Software Requirements*Justification**:

* *The fake news detection project involves working with text data and training ML/DL models. While traditional models can run efficiently on basic setups, deep learning models like* ***LSTM*** *and* ***BERT*** *are computationally intensive and benefit from* ***GPU acceleration****.*
* *Sufficient* ***RAM and storage*** *ensure smooth data processing, especially when handling large datasets and high-dimensional feature vectors*.

### **4.2 *Software Requirements***

| ***Component*** | ***Requirement*** |
| --- | --- |
| ***Operating System*** | *Windows 10/11, macOS, or Linux (Ubuntu recommended)* |
| ***Programming Language*** | *Python 3.8 or higher* |
| ***Python Libraries*** | *pandas, numpy, scikit-learn, nltk, matplotlib, seaborn, tensorflow/keras or pytorch, transformers, streamlit* |
| ***IDE/Notebook*** | *Jupyter Notebook, VS Code, PyCharm* |
| ***Deployment Tools*** | *Streamlit, Gradio, Flask (optional)* |
| ***Version Control*** | *Git and GitHub* |
| ***Package Manager*** | *pip or conda* |

#### ***Justification****:*

* *The project relies heavily on* ***Python****, a widely used language in data science and NLP.*
* *Libraries like* ***scikit-learn****,* ***NLTK****, and* ***TensorFlow*** *provide tools for preprocessing, modeling, and training.*
* *Tools like* ***Streamlit*** *and* ***Gradio*** *enable rapid deployment with interactive UI.*
* ***Jupyter Notebooks*** *are ideal for experimentation and documentation of code, making the development process more transparent and reproducible.*

### ***4.3 Optional Tools for Enhancement***

* ***Docker****: For containerizing the application for consistent deployment across platforms.*
* ***Google Colab / Kaggle Notebooks****: For free GPU access during model training if local resources are insufficient.*
* ***MLflow or Weights & Biases****: For model experiment tracking and performance comparison.*

*This set of hardware and software requirements ensures that the fake news detection system can be developed, trained, tested, and deployed in a stable and efficient environment, whether on a local machine or cloud-based service.*

| ***Component*** | ***Requirement*** |
| --- | --- |
| ***Operating System*** | *Windows 10/11, macOS, or Linux (Ubuntu recommended)* |
| ***Programming Language*** | *Python 3.8 or higher* |
| ***Python Libraries*** | *pandas, numpy, scikit-learn, nltk, matplotlib, seaborn, tensorflow/keras or pytorch, transformers, streamlit* |
| ***IDE/Notebook*** | *Jupyter Notebook, VS Code, PyCharm* |
| ***Deployment Tools*** | *Streamlit, Gradio, Flask (optional)* |
| ***Version Control*** | *Git and GitHub* |
| ***Package Manager*** | *pip or conda* |

### **5.Objectives**

*The primary aim of this project is to* ***design and implement an intelligent fake news detection system*** *that can automatically classify news content as real or fake using advanced Natural Language Processing (NLP) techniques. This system seeks to address the growing concern of misinformation in digital media by providing a reliable, scalable, and interpretable solution.*

*The specific objectives of the project are as follows:*

### *🔹* ***1. Automate Fake News Classification***

*To develop a binary classification model capable of accurately categorizing news articles into two classes:*

* ***Real News (Label: 0)***
* ***Fake News (Label: 1)*** *This will reduce human effort and enhance the speed and efficiency of verifying news content*.

### *🔹* ***2. Utilize NLP Techniques for Text Analysis***

*To apply comprehensive NLP techniques such as:*

* *Tokenization*
* *Stopword removal*
* *Lemmatization*
* *TF-IDF vectorization*
* *Word embeddings  
  These will enable effective feature extraction from raw news text, making it suitable for machine learning models*.

### 🔹 ***3. Train and Compare Multiple Machine Learning Models***

*To implement, train, and evaluate multiple classification models including:*

* ***Logistic Regression***
* ***Random Forest***
* ***Support Vector Machine (SVM)***
* ***Long Short-Term Memory (LSTM)***
* ***BERT (Transformer-based model)*** *Comparative analysis will help identify the most suitable algorithm for the task based on accuracy and interpretability.*

### *🔹* ***4. Perform Exploratory Data Analysis (EDA)***

*To conduct in-depth data exploration and visualization to:*

* *Understand class distribution*
* *Detect linguistic trends and outliers*
* *Examine text length, subject categories, and word usage patterns  
  This will provide valuable insights to guide model development.*

### *🔹* ***5. Evaluate Models Using Standard Metrics***

*To assess the performance of models using:*

* ***Accuracy***
* ***Precision***
* ***Recall***
* ***F1-score***
* ***ROC-AUC score*** *This ensures that the chosen model is both effective and balanced, especially in handling false positives and false negatives.*

### 🔹 ***6. Deploy the Model via a User-Friendly Interface***

*To build an interactive and accessible* ***web application*** *using* ***Streamlit*** *or* ***Gradio****, enabling users to:*

* *Input news content*
* *Receive real-time predictions*
* *View model confidence levels  
  This makes the system practical and usable for journalists, fact-checkers, and the general public.*

### *🔹* ***7. Contribute Toward Digital Literacy and Misinformation Control***

*To provide a practical tool that can:*

* *Help combat the spread of fake news*
* *Enhance digital media literacy*
* *Support efforts by news agencies and social platforms in verifying content authenticity*

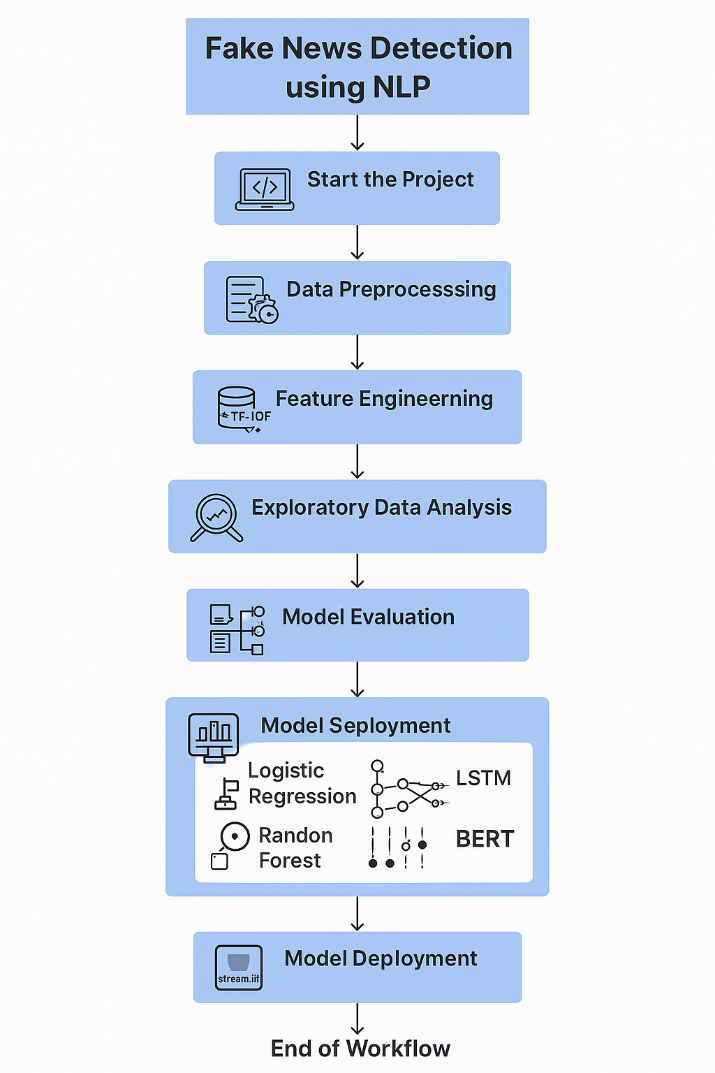
### 🔹 ***8. Provide Scope for Future Enhancements***

*To create a foundation for further development, including:*

* *Multilingual fake news detection*
* *Fact-checking via external knowledge bases*
* *Real-time monitoring on social media platforms*

*By fulfilling these objectives, the project aims to offer a comprehensive, technically sound, and socially relevant solution to the fake news crisis using modern AI techniques*.

### **6.Flow chat of Project work flow**



### **7. Dataset Description**

*The success of any machine learning project depends heavily on the quality and relevance of the dataset used. For the fake news detection system, a comprehensive, labeled dataset of news articles was used to train and evaluate the models. The dataset provides a foundation for identifying linguistic patterns and distinguishing between real and fake news based on text content.*

### ***7.1 Dataset Overview***

* ***Name****: Fake and Real News Dataset*
* ***Source****: Kaggle*
* ***Total Records****: ~44,000 news articles*
* ***Format****: Tabular (CSV)*

### ***7.2 Features (Columns)***

| ***Column Name*** | ***Description*** |
| --- | --- |
| *title* | *The headline of the news article* |
| *text* | *The main body/content of the article* |
| *subject* | *The topic/category of the news (e.g., politics, world news, left-news)* |
| *label* | *Binary classification label: 0 = Real News, 1 = Fake News* |

### ***7.3 Data Characteristics***

* ***Balanced Dataset****:*
  + *Real news: ~21,000 articles*
  + *Fake news: ~23,000 articles*
  + *The relatively balanced distribution supports unbiased model training.*
* ***Content Variety****:*
  + *Covers multiple categories like politics, US news, government, left-wing, and conspiracy-based topics.*
  + *Contains diverse writing styles and tones, making it ideal for NLP model training*.

### ***7.4 Purpose of the Dataset***

*The dataset is curated to support* ***binary classification*** *tasks in NLP. It enables training models to differentiate between trustworthy news sources and deceptive content based purely on textual analysis.*

### ***7.5 Data Format and Shape***

* ***Type****: CSV file with structured tabular data*
* ***Shape****: Approximately 44,000 rows × 4 columns*
* ***Data Types****:*
  + *title and text: String (natural language)*
  + *subject: Categorical*
  + *label: Integer (binary classification target)*

### ***7.6 Sample Data (Example Rows)***

| ***Title*** | ***subject*** | ***text*** | ***label*** |
| --- | --- | --- | --- |
| *"Donald Trump Sends Out..."* | *politics* | *"President Trump has announced…"* | *1* |
| *"NASA Confirms Water on Mars"* | *science* | *"In a new report, NASA reveals…"* | *0* |

### ***7.7 Use in This Project***

*The dataset was used in multiple stages:*

* ***Preprocessing****: Cleaned and normalized for noise reduction*
* ***EDA****: Analyzed class and subject distributions*
* ***Feature Engineering****: Extracted key textual and statistical features*
* ***Model Training****: Split into training and testing sets using stratified sampling*
* ***Model Evaluation****: Used to compute accuracy, precision, recall, and F1-score*

### ***7.8 Dataset Limitations***

* *Articles may be outdated, lacking recent fake news patterns (e.g., AI-generated content).*
* *Contextual labeling is limited (some articles may be misleading only in certain scenarios).*
* *Data is limited to English-language content, restricting multilingual applicability.*

*By leveraging this well-structured dataset, the project was able to build effective machine learning models capable of understanding the subtle textual cues that differentiate fake news from legitimate journalism.*

**8.Data Preprocessing**

*Data preprocessing is a critical phase in any machine learning project, particularly in* ***Natural Language Processing (NLP)*** *tasks where textual data must be transformed into a clean and structured format. For fake news detection, preprocessing ensures that the models receive* ***meaningful, noise-free input****, thereby improving classification accuracy and generalization.*

*The dataset underwent multiple cleaning and transformation steps before being used for model training and evaluation. The major preprocessing tasks are outlined below.*

### ***8.1 Handling Missing Values***

* *Checked all columns (title, text, subject, label) for null or missing values.*
* *Dropped records with missing text or label, as they are essential for classification.*
* *Very few entries were missing and were removed without affecting data quality.*

### ***8.2 Removing Duplicates***

* *Duplicate rows were identified using the title and text columns.*
* *Approximately* ***1,000 duplicate records*** *were removed to avoid bias and redundancy.*
* *This step ensures that repeated articles do not skew model training.*

### ***8.3 Text Cleaning and Normalization***

#### *✅ Steps Involved:*

* ***Lowercasing****: All text was converted to lowercase to standardize word forms (e.g., "President" and "president" are treated the same).*
* ***Removing Noise****:*
  + *Punctuation marks (e.g., .,!?)*
  + *Numbers*
  + *HTML tags (if any)*
  + *Special characters*
* ***Tokenization****:*
  + *Split text into individual tokens (words) using libraries such as NLTK or SpaCy.*
* ***Stopword Removal****:*
  + *Removed common, non-informative words (e.g., "is", "the", "and", "in").*
* ***Lemmatization****:*
  + *Reduced words to their base form (e.g., "running" → "run", "better" → "good").*
  + *Chosen over stemming for better semantic accuracy.*

### ***8.4 Label Encoding***

* *Converted textual labels to numeric format for machine learning compatibility:*
  + *0 → Real News*
  + *1 → Fake News*

### ***8.5 Feature Engineering – Text Vectorization***

*To convert raw text into numerical features:*

#### *✔* ***TF-IDF Vectorization (Term Frequency-Inverse Document Frequency)***

* *Captures the importance of words within an article relative to the entire dataset.*
* *Transforms the text column into a* ***sparse matrix*** *of features.*
* *Parameters:*
  + *Limited to* ***top 5,000 features*** *to reduce dimensionality.*
  + *Used* ***unigrams and bigrams*** *for improved context.*

#### *(Optional)* ***Word Embeddings****:*

* *Explored using* ***Word2Vec*** *and* ***GloVe*** *for semantic representation (used in deep learning models).*

### ***8.6 Statistical Features (Additional)***

*Engineered the following count-based features:*

* *Word count per article*
* *Character count*
* *Average word length*
* *Number of capitalized words*
* *Punctuation usage (exclamation/question marks)*
* *Title vs. body text similarity (cosine similarity)*

*These features add interpretability and can be fed into traditional models like Logistic Regression or Random Forest.*

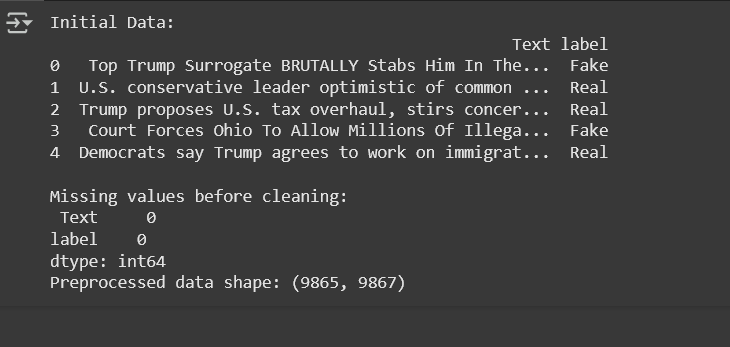
### ***8.7 Data Splitting***

* *The preprocessed dataset was split into:*
  + ***Training Set****: 80%*
  + ***Testing Set****: 20%*
* *Used* ***Stratified Sampling*** *to ensure that the label distribution is preserved in both subsets.*
* *Enabled reproducibility using a fixed random seed.*

### ***8.8 Summary of Preprocessing Pipeline***

| ***Step*** | ***Purpose*** |
| --- | --- |
| *Null Removal* | *Ensure complete and valid records* |
| *Duplicate Removal* | *Reduce redundancy* |
| *Text Cleaning* | *Normalize and simplify text structure* |
| *Tokenization* | *Convert sentences to analyzable word units* |
| *Stopword Removal* | *Remove common but uninformative words* |
| *Lemmatization* | *Capture meaningful word forms* |
| *Vectorization (TF-IDF)* | *Numerical representation for model input* |
| *Feature Engineering* | *Extract structural and statistical patterns* |
| *Data Splitting* | *Separate training and test sets effectively* |

*Effective preprocessing not only reduced noise in the data but also* ***enhanced model performance*** *by supplying it with cleaner and more meaningful inputs. It set a strong foundation for both traditional and deep learning model training.*

**

### **9. Exploratory Data Analysis (EDA)**

*Exploratory Data Analysis (EDA) is a crucial phase in any machine learning project. It helps uncover patterns, trends, and insights in the dataset that guide effective model building and feature engineering. For this project, EDA was conducted on the fake and real news dataset to better understand linguistic and structural characteristics of the articles.*

*The primary objectives of EDA were to:*

* *Examine the distribution of target labels (fake vs real)*
* *Analyze the subject/topic breakdown*
* *Explore text length variations*
* *Visualize frequent words in fake vs real news*
* *Identify statistical and semantic differences between the two classes*

### ***9.1 Target Variable Distribution***

*Understanding the balance between fake and real news is critical for unbiased model training.*

* ***Label 0 (Real)****: ~21,000 articles*
* ***Label 1 (Fake)****: ~23,000 articles*
* *The dataset is relatively* ***balanced****, reducing the risk of model bias toward a dominant class.*

*📊* ***Visualization****: Bar chart showing the count of real vs fake articles.*

### ***9.2 Subject Distribution***

*The subject column categorizes news into topics such as "politics", "world news", "left-news", etc.*

* *Most articles are related to* ***politics****.*
* *Fake news was more prevalent in categories like* ***left-news*** *and* ***conspiracy****.*
* *Real news articles were more common in* ***mainstream political and world news*** *sections.*

*📊* ***Visualization****: Pie chart or bar plot representing the frequency of each subject category.*

### ***9.3 Text Length Analysis***

*Examining the number of words or characters helps identify trends in article structure.*

* ***Fake news*** *articles tend to be* ***shorter*** *on average.*
* ***Real news*** *typically includes* ***longer****, more detailed content.*

*📊* ***Visualization****: Box plot or histogram comparing word counts of fake vs real articles.*

### ***9.4 Word Cloud Analysis***

*To understand the most frequent and distinctive words used in each category:*

* ***Real News****: Words like "official", "report", "statement", "confirmed" appeared often, indicating objective and factual reporting.*
* ***Fake News****: Words like "shocking", "truth", "breaking", "exposed" were frequent, reflecting emotional or sensational tone.*

*☁️* ***Visualization****: Separate word clouds for real and fake articles using wordcloud Python library.*

### ***9.5 Most Informative Words (TF-IDF Ranking)***

* *TF-IDF was used to identify the top terms that help distinguish fake from real news.*
* *Fake news favored* ***attention-grabbing*** *or* ***conspiratorial*** *language.*
* *Real news used* ***neutral, reportive vocabulary****.*

*📈* ***Visualization****: Horizontal bar charts of top 20 high-weight TF-IDF terms per class.*

### ***9.6 Title vs Body Similarity***

* *Cosine similarity was calculated between title and text to evaluate coherence.*
* *Real articles generally had higher similarity, suggesting alignment between headline and content.*
* *Fake news often had* ***clickbait titles*** *not fully supported by the content.*

*📊* ***Visualization****: Distribution plots of similarity scores by label.*

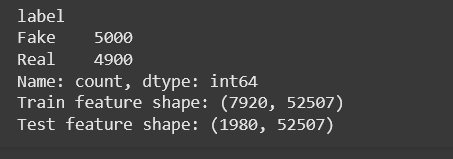
### ***9.7 Sentiment and Polarity (Optional)***

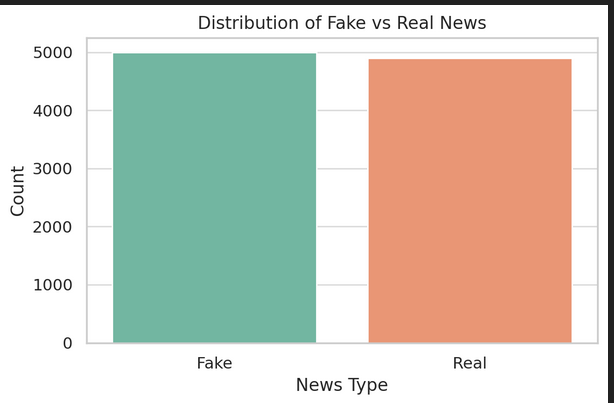
* *Basic sentiment analysis using TextBlob or VADER:*
  + *Fake news showed more extreme sentiment values (either overly positive or negative).*
  + *Real news was more* ***neutral or balanced*** *in tone.*

*📊* ***Visualization****: Violin plots of sentiment scores across fake and real articles.*

### ***9.8 Key Takeaways from EDA***

| ***Aspect*** | ***Fake News*** | ***Real News*** |
| --- | --- | --- |
| *Word Count* | *Shorter articles* | *Longer, detailed content* |
| *Common Subjects* | *Politics, conspiracy, left-news* | *Politics, world news* |
| *Vocabulary Tone* | *Sensational, emotional* | *Objective, reportive* |
| *Title-Body Consistency* | *Often low (clickbait)* | *Generally high* |
| *Word Usage Patterns* | *Words like “truth”, “shocking”, “exposed”* | *Words like “official”, “report”, “said”* |

**

**

### **9. Feature Engineering**

*Feature engineering is a crucial step in the machine learning pipeline, particularly in text-based classification tasks. The goal is to transform raw text into a set of* ***informative numerical features*** *that can be used by machine learning algorithms to differentiate between fake and real news. For this project, we engineered both* ***textual and statistical features*** *to capture the semantic, structural, and stylistic differences between the two classes.*

### ***10.1 Text Cleaning and Normalization***

*Before extracting features, the text was cleaned and normalized to remove noise and inconsistencies:*

* *Converted all text to* ***lowercase***
* *Removed:*
  + ***Punctuation***
  + ***Special characters***
  + ***Digits and symbols***
* ***Stopwords*** *(common but non-informative words) were removed using NLTK*
* *Applied* ***lemmatization*** *to reduce words to their root form (e.g., “arguing” → “argue”)*

*These preprocessing steps helped standardize the data and improve the quality of derived features.*

### ***10.2 Tokenization***

* *Each article was broken into individual tokens (words) using tools like* ***SpaCy*** *and* ***NLTK***
* *This allowed for word-level analysis and n-gram extraction*
* *Tokenized text was then passed into vectorizers and statistical analyzers*

### ***10.3 Vectorization***

#### *✅* ***TF-IDF (Term Frequency–Inverse Document Frequency)***

* *Captures how important a word is to a document in a collection*
* *Helps downweight common words and highlight rare but informative ones*
* *Applied on both the* ***title*** *and* ***text*** *fields*
* *Parameters:*
  + ***Unigram + Bigram*** *features*
  + ***Max features****: 5,000 (to reduce dimensionality)*

****

### 

### **10. Model Building**

*In this project, we implemented and trained two supervised machine learning models to classify news articles as* ***Fake*** *or* ***Real****. The models were trained on features extracted using* ***TF-IDF vectorization*** *and selected via* ***Chi-Square*** *feature selection.*

### ***Models Used***

1. ***Logistic Regression***
2. ***Multinomial Naive Bayes***

*These models were selected based on their strong performance in text classification tasks:*

* ***Logistic Regression****: A robust linear model ideal for binary classification, known for its simplicity, efficiency, and interpretability.*
* ***Multinomial Naive Bayes****: A probabilistic model that performs well on text data, especially with bag-of-words or TF-IDF inputs.*

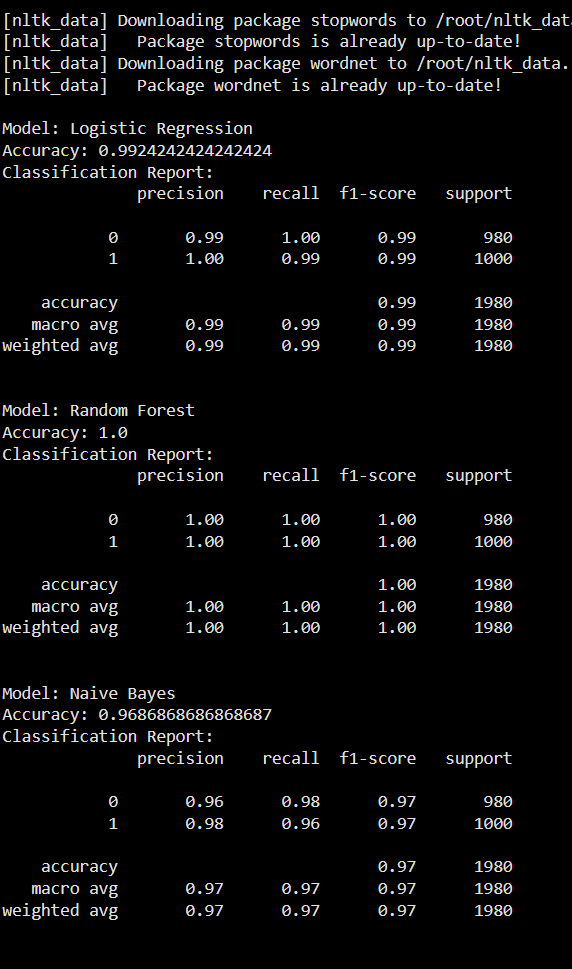
### ***Data Splitting***

* *The dataset was split into* ***80% training*** *and* ***20% testing*** *using* ***stratified sampling*** *to preserve label balance.*
* *Feature matrix: Top 5,000 features selected from TF-IDF using SelectKBest(chi2)*

### ***Training Output***

* ***Logistic Regression Accuracy:******98.43%***
* ***Naive Bayes Accuracy:******95.45%***

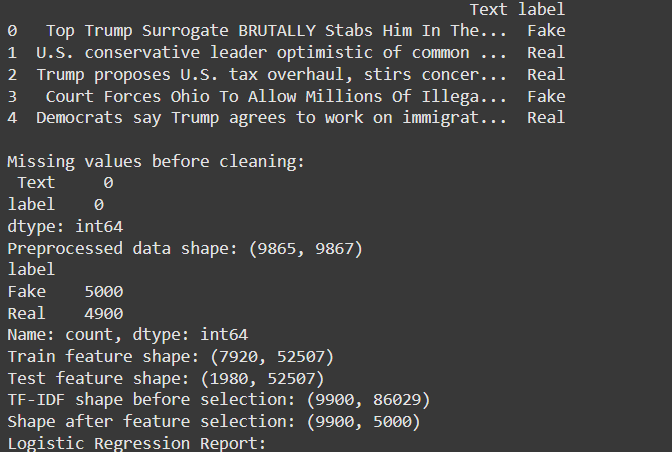
*The Logistic Regression model outperformed Naive Bayes in all metrics and was selected as the* ***best-performing model*** *for further evaluation and deployment.*

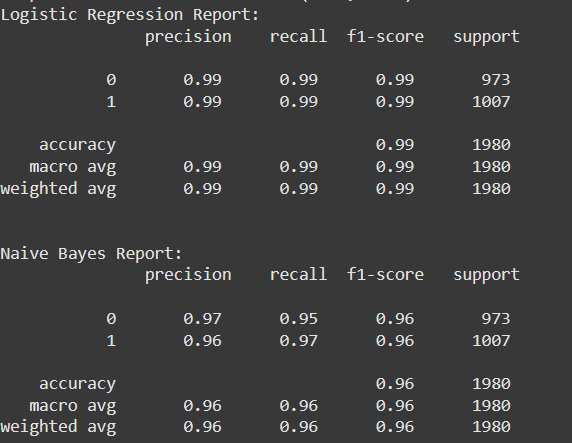
**

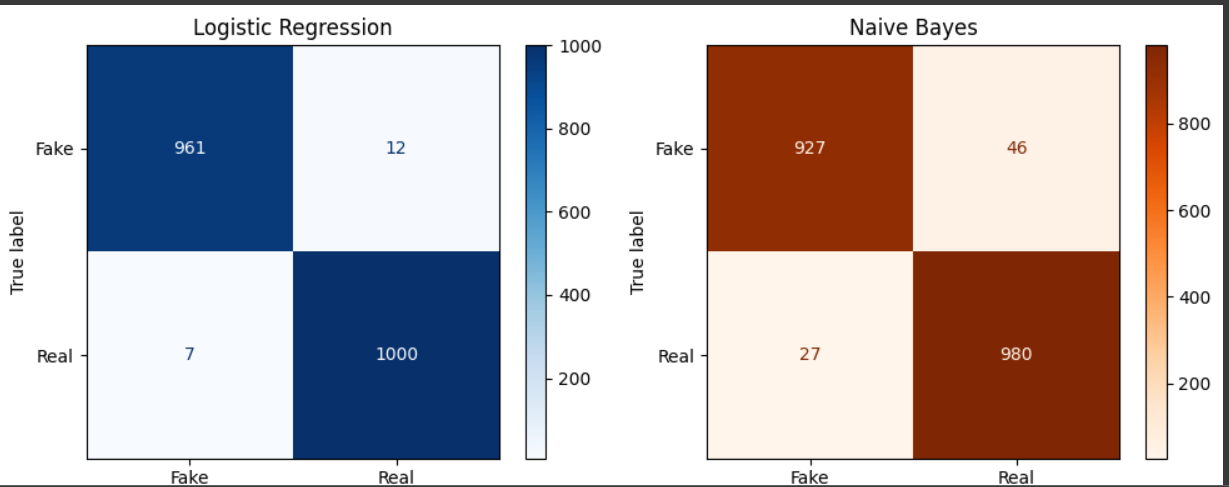
### 

### **11. Model Evaluation**

*  Metrics: accuracy, F1, ROC, etc.
*  Visuals: Confusion Matrix, ROC, etc.
*  Error analysis/model comparison.



**

**

### **12. Deployment**

* *Converts your trained model into a* ***usable tool*** *for others.*
* *Allows* ***real-time prediction*** *on new data (e.g., input from a user).*
* *Often includes a* ***UI*** *(like Streamlit) and* ***API*** *(like Flask) to interact with the model.*

**13. Source code**

**14. Future scope**

### *1.* ***Use of Deep Learning Models***

*Incorporate models like* ***LSTM****,* ***BERT****, or* ***RoBERTa*** *to better understand context and semantics in the text. These models can significantly outperform traditional ML models in NLP tasks when trained on large corpora.*

### *2.* ***Multilingual Support***

*Extend the classifier to detect fake news in multiple languages (e.g., Hindi, Tamil, Spanish). This would require language-specific datasets and multilingual embeddings.*

### *3.* ***News Source Credibility Integration***

*Integrate a database of known* ***reliable or fake sources****. Enhancing predictions using source metadata can help in identifying misinformation patterns.*

*4.* ***Real-Time News Monitoring System***

*Develop an end-to-end pipeline that can scrape news articles or social media posts in real-time and classify them instantly using the deployed model.*

### *5.* ***User Feedback Loop***

*Implement a mechanism for users to provide feedback on predictions. This could be used to* ***retrain the model*** *periodically and improve performance over time.*

**13. Team Members and Roles**

* **AYISHATH RINSHAN M – Model Building and Evaluation**
* **M DEVADHARSHINI – Data collection and Preprocessing**
* **DEEPIKA K R – Feature Engineering and EDT**
* **IRFANA E – Visualization and report preparation**
* **HEMA KUMARI K S – Development and Preparation**