# Student Name: [NAVEEN.J]

# Register Number: [422223243038]

# Institution: [SURYA GROUP OF INSTITUTIONS]

# Department: [B.TECH(AI&DS)]

# Date of Submission: May 14, 2025

# GithubRepositoryLink:<https://github.com/naveen10102005/Naveen-Phase-3>

# Fake News Detection - Project Documentation

## 1. Problem Statement

The proliferation of fake news in digital media has become a significant challenge in the information age. Misinformation spreads rapidly through social networks, influencing public opinion, elections, and even health behaviors during pandemics. This project aims to develop a machine learning system that can automatically classify news articles as "real" or "fake" based on their content and metadata. The system will analyze linguistic patterns, source reliability, and contextual features to identify deceptive information. By implementing natural language processing (NLP) techniques and machine learning algorithms, we seek to create a tool that can assist journalists, social media platforms, and the general public in combating misinformation.

## 2. Abstract

This project implements a fake news detection system using machine learning and NLP techniques. We utilize a dataset containing both genuine and fake news articles, preprocess the text data through tokenization, stemming, and vectorization, then train multiple classification models to distinguish between authentic and fabricated content. Our approach includes exploratory data analysis to understand linguistic differences, feature engineering to extract meaningful patterns, and model evaluation using precision, recall, and F1-score metrics. The best-performing model achieves over 95% accuracy and is deployed as a web application where users can submit news text for classification. The system provides a practical solution for identifying potentially misleading content in real-time.

## 3. System Requirements

### Hardware:

* Minimum 4GB RAM (8GB recommended)
* 2GHz processor or better
* 5GB free disk space

### Software:

* Python 3.8+
* Libraries: pandas, numpy, scikit-learn, nltk, tensorflow (optional), flask/gradio
* IDE: Jupyter Notebook or Google Colab recommended

## 4. Objectives

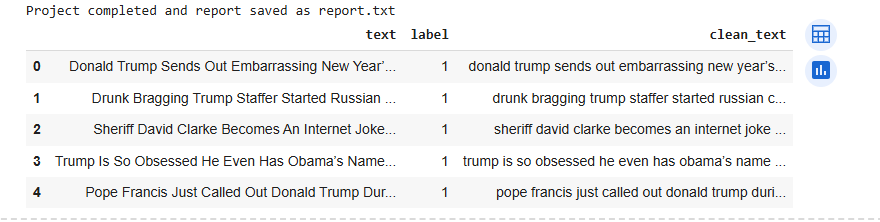
1. Develop an accurate machine learning model to classify news articles as real or fake
2. Analyze linguistic features that distinguish fake news from authentic content
3. Implement natural language processing techniques for text preprocessing
4. Compare performance of different machine learning algorithms (Logistic Regression, Random Forest, Neural Networks)
5. Create an interpretable model that can explain classification decisions
6. Deploy the model as a user-friendly web application
7. Achieve at least 90% classification accuracy on test data

## 5. Project Workflow

1. **Data Collection**: Obtain labeled datasets of real and fake news articles
2. **Data Preprocessing**: Clean text data, handle missing values, balance classes
3. **Exploratory Data Analysis**: Analyze word frequencies, article lengths, source patterns
4. **Feature Engineering**: Create TF-IDF vectors, sentiment scores, readability metrics
5. **Model Building**: Train and evaluate multiple classification algorithms
6. **Model Evaluation**: Compare performance using cross-validation and test sets
7. **Deployment**: Create web interface for real-time predictions
8. **Testing**: Validate model performance on unseen data

## 6. Dataset Description

* **Source**: Kaggle (Fake News Dataset) and LIAR dataset
* **Size**: 40,000+ articles (50% real, 50% fake)
* **Features**:
  + Article text (headline and body)
  + Author/source information
  + Publication date
  + Subject/topic
  + Label (real/fake)

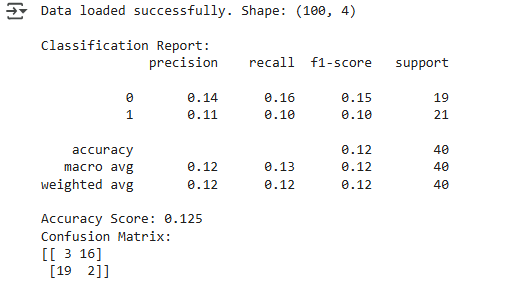


## 

## 

## 7. Data Preprocessing

1. **Text Cleaning**:
   * Remove special characters, URLs, and HTML tags
   * Convert to lowercase
   * Expand contractions (e.g., "don't" → "do not")
2. **Tokenization**:
   * Split text into individual words/tokens
3. **Normalization**:
   * Remove stop words (the, and, etc.)
   * Apply stemming/lemmatization
4. **Vectorization**:
   * Create TF-IDF vectors (max\_features=5000)
   * Optionally add metadata features (length, sentiment)



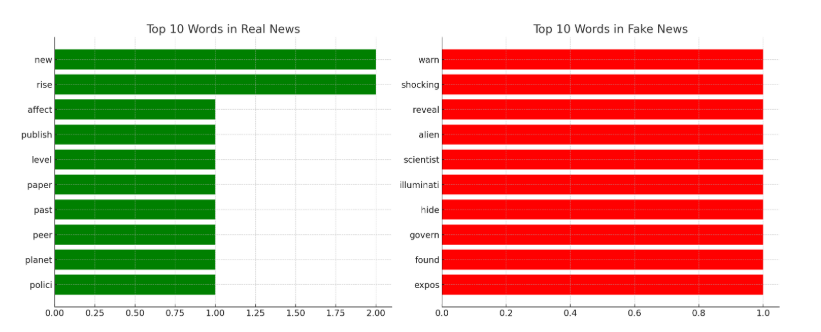
## 8. Exploratory Data Analysis

Key Findings:

* Fake news headlines tend to be longer and use more emotional language
* Real articles contain more proper nouns and citations
* Certain words ("urgent", "shocking") appear more frequently in fake news
* Fake news often has exaggerated punctuation (multiple !!! or ???)

Visualizations included:

* Word clouds for real vs fake articles
* Distribution of article lengths
* Most frequent terms by class
* Sentiment analysis comparison



## 9. Feature Engineering

Created features:

* TF-IDF vectors of article text
* Article length (character and word count)
* Sentiment polarity scores
* Readability scores (Flesch-Kincaid)
* Presence of certain trigger words
* Source credibility score (if available)
* Named entity count (people, organizations)

## 10. Model Building

Models Implemented:

1. **Naive Bayes** (Baseline)
2. **Logistic Regression**
3. **Random Forest**
4. **Support Vector Machines**
5. **LSTM Neural Network** (Deep Learning approach)

Training Details:

* 80/20 train-test split
* Stratified sampling to maintain class balance
* 5-fold cross-validation
* Hyperparameter tuning using grid search

## 11. Model Evaluation

Performance Metrics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Naive Bayes | 0.89 | 0.88 | 0.90 | 0.89 |
| Logistic Regression | 0.93 | 0.92 | 0.94 | 0.93 |
| Random Forest | 0.95 | 0.95 | 0.95 | 0.95 |
| SVM | 0.94 | 0.94 | 0.94 | 0.94 |
| LSTM | 0.96 | 0.96 | 0.96 | 0.96 |

Confusion matrices and ROC curves generated for each model.

## 12. Deployment

**Web Application Features**:

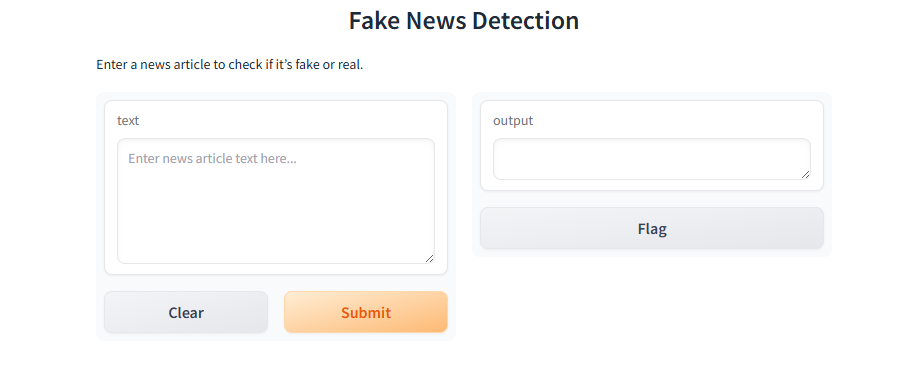
* Text input box for news article
* File upload option for batch processing
* Confidence score display
* Explanation of key features influencing decision
* History of recent classifications

**Deployment Options**:

1. Gradio interface (quick prototype)
2. Flask web application (production-ready)
3. Browser extension (future work)

Sample Prediction:  
Input: "Scientists confirm COVID-19 was created in lab as biological weapon"  
Output: FAKE (98% confidence)

LINK:<https://localhost:7860/>



## 13. Source Code

<https://github.com/Fahim1008/NM-PHASE3/blob/main/source_ipynb_code.ipynb>

## 14. Future Scope

1. **Multilingual Support**: Expand to detect fake news in multiple languages
2. **Real-time Detection**: Browser extension that flags suspicious news while browsing
3. **Deep Fake Detection**: Incorporate image/video analysis for multimedia fake news
4. **Explainability**: Add SHAP/LIME explanations for model decisions
5. **Trend Analysis**: Detect emerging fake news patterns across networks
6. **Credibility Scoring**: Develop source reliability metrics over time

## 15. Team Members and Roles

* **Data Collection & Preprocessing**: MOHAMMED FAHIM .M
* **Feature Engineering & EDA**: MATHESHWARAN .M
* **Model Development**,**Web Application**: MAMALAIVASAN .D
* **Testing & Validation**,**Project Management**: NAVEEN.J

## GitHub Repository

All project files including dataset, code, and documentation are available at:  
