**Phase-3 Submission Template**

**Student Name:** M.Saravanan

**Register Number:** 621523104143

**Institution:** Mahendra College Of Engineering

**Department:** Computer Science and Engineering

**Date of Submission:** 09/05/2025

**Github RepositoryLink:**https://github.com/Selvabharathy13



### **1. Problem Statement**

*With the exponential growth of online content, fake news has become a significant threat to public opinion, political stability, and public health. Misinformation spreads faster than ever, misleading readers and creating confusion. There is an urgent need for intelligent systems that can accurately detect and flag fake news using computational approaches like Natural Language Processing (NLP) and Machine Learning.*

### **2. Abstract**

*The rise of social media and digital platforms has accelerated the spread of fake news, misleading the public and undermining trust in media. This project aims to develop a robust fake news detection system powered by Natural Language Processing (NLP) and Machine Learning (ML).*

### **3. System Requirements**

### Hardware Requirements:

### Processor: Intel i5/i7 or equivalent

### RAM: 8GB minimum (16GB recommended)

### Storage: At least 2GB for datasets and model files

### GPU (for deep learning models): NVIDIA CUDA-enabled GPU (optional but recommended)

### Software Requirements:

### Operating System: Windows 10 / Ubuntu / macOS

### Python 3.8+

### Jupyter Notebook / Google Colab

### Required Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, nltk, spacy, tensorflow/keras, flask (for deployment)

### **4. Objectives**

To develop a system that can accurately classify news as fake or real using NLP and ML.

To analyze linguistic features and patterns in textual data.

To compare the performance of multiple classification models.

**5. Flowchart of Project Workflow**

*┌────────────────────┐*

*│ Data Collection │*

*└────────┬───────────┘*

*↓*

*┌────────────────────┐*

*│ Data Preprocessing│*

*└────────┬───────────┘*

*↓*

*┌───────────────▼────────────────┐*

*│ Exploratory Data Analysis (EDA)│*

*└────────┬───────────────────────┘*

*↓*

*┌────────▼────────┐*

*│Feature Engineering│*

*└────────┬────────┘*

*↓*

*┌──────▼──────┐*

*│Model Building│*

*└──────┬──────┘*

*↓*

*┌───────▼────────┐*

*│Model Evaluation│*

*└───────┬────────┘*

*↓*

*┌───────▼────────────┐*

*│ Visualization & Insights│*

*└────────────────────┘*

### **6. Dataset Description**

* *Dataset Name: Fake and real news dataset*
* *Source: Kaggle or other news corpus datasets*
* *Attributes:*
* *title: Headline of the article*
* *text: Full article content*
* *subject: Topic/category of the news*
* *date: Publication date*
* *label: 0 for real, 1 for fake*

### 

### **7. Data Preprocessing**

* *Removing null values*
* *Dropping duplicates*
* *Cleaning text (lowercasing, removing punctuation, stopwords, and numbers)*
* *Tokenization and lemmatization*
* *Balancing the dataset (if imbalanced)*

### 

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

* *Word frequency analysis*
* *Word clouds for fake vs real news*
* *Article length distribution*
* *Frequency of top keywords*
* *Distribution of topics (subject)*
* *Temporal patterns if using date*

### 

### **9. Feature Engineering**

### Bag of Words (BoW)

### TF-IDF Vectorization

### N-grams (bigrams/trigrams)

### Sentiment scores

### POS tag distribution

### Named Entity Recognition (NER) counts

### **10. Model Building**

### Preprocessing: Lowercasing, stopword removal, lemmatization, punctuation cleaning.

### Feature Extraction: TF-IDF Vectorization, n-gram models.

### Algorithms Used:

### Logistic Regression

### Naïve Bayes

### Random Forest

### Support Vector Machine (SVM)

### XGBoost

### LSTM/BERT for deep learning

### **11. Model Evaluation**

* *Metrics Used:*
* *Accuracy*
* *Precision*
* *Recall*
* *F1-Score*
* *ROC-AUC*
* *Evaluation Tools:*
* *Confusion Matrix*
* *ROC Curve*
* *Precision-Recall Curve*

### **12. Deployment**

*The final model is integrated into a Flask web application.*

*Users can paste or upload news articles and receive instant feedback on authenticity.*

*Hosted locally or on platforms like Heroku, Render, or Streamlit.*

**13. Source code**

*All code is organized in the following modules:*

*data\_preprocessing.py – Cleaning and tokenizing data*

*eda\_visualizations.ipynb – Graphs and statistics*

*feature\_engineering.py – TF-IDF and NLP pipeline*

*model\_training.ipynb – Model training and evaluation*

*app.py – Flask application for deployment*

*templates/ – HTML interface for the app*

*(You can upload the actual code to GitHub or zip it for submission.)*

**14. Future scope**

*Incorporate multilingual fake news detection for regional news sources.*

*Integrate fact-checking APIs like Google Fact Check or Snopes.*

*Expand to audio/video content analysis using multimodal ML models*

*Use real-time social media monitoring to flag suspicious content.*

*Build a browser extension for live news authenticity checks.*

**13. Team Members and Roles**

*1. Selvabharathy K – NLP & Feature Engineering Lead*

*Handled text preprocessing using NLTK and spaCy.*

*Developed TF-IDF and n-gram models for feature extraction.*

*Contributed to model training and optimization.*

*2. Samuel Paul V – Data Analysis & Visualization Expert*

*Performed Exploratory Data Analysis (EDA) to uncover patterns in fake vs real news.*

*Created visual dashboards including word clouds, bar plots, and distribution charts.*

*Helped document findings and insights.*

*3. Saravanan M – Machine Learning Engineer*

*Built and evaluated multiple ML models including Logistic Regression, SVM, and XGBoost.*

*Tuned hyperparameters and managed model comparison metrics.*

*Supported backend integration for deployment.*

*4. Sanjeevi V – Deep Learning & Model Deployment Specialist*

*Designed and trained LSTM and BERT-based models for enhanced text classification.*

*Implemented the final model in a Flask web application.*

*Managed model testing and deployment environment.*

*5. Sakthivel E – Project Manager & System Designer*

*Oversaw project progress and coordinated team activities.*

*Created the system design flowchart and documented the architecture.*

*Ensured quality control and compiled the final report and presentation.*