Phase-2 Submission Template

Student Name: ARJUN.A

Register Number: 720323104012

Institution: AKSHAYA COLLEGE OF ENGINEERING AND TECHNOLOGY

Department: B.E.COMPUTER SCIENCE AND ENGINEERING

Date of Submission: 12.05.25

Github Repository Link: [Update the project source code to your Github Repository]

---

1. Problem Statement

In today's digital age, the spread of misinformation and fake news poses a significant threat to society. This project aims to develop a system that detects fake news articles using Natural Language Processing (NLP) techniques. The problem is framed as a **binary classification task** where each news article is labeled as *real* or *fake*.

This problem is crucial for enhancing media credibility, promoting public awareness, and assisting social media platforms in mitigating the effects of misinformation.

2. Project Objectives

- Build a machine learning pipeline for fake news classification using NLP techniques.
- Preprocess and clean the text data for better feature extraction.
- Use multiple models such as Logistic Regression and Random Forest for performance comparison.
- Achieve high accuracy, precision, and recall in identifying fake news.
- Provide interpretable model insights and key influential features

3. Flowchart of the Project Workflow

(Data Collection) → (Data Cleaning & Preprocessing) → (EDA) → (Feature Engineering) → (Model Training) → (Evaluation) → (Insights & Deployment)

4. Data Description

- **Dataset Name**: Fake and Real News Dataset
- **Source**: Kaggle
- **Type**: Text (unstructured data)

- **Number of Records**: ~44,000 articles
- **Features**: Title, Text, Subject, Date
- **Target Variable**: Label (Real/Fake)
- **Static Dataset**

5. Data Preprocessing

- Removed null or missing entries
- Dropped duplicate records
- Removed punctuation, stopwords, and HTML tags
- Applied lowercasing and tokenization
- Performed lemmatization
- Encoded labels (0: Fake, 1: Real)
- Converted text into numerical form using TF-IDF vectorization

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis**: Distribution of article lengths, word counts
- **Bivariate Analysis**: Correlation of article subject with label
- **Insights**: Fake news tends to have exaggerated or sensational titles; real news uses more factual language.

7. Feature Engineering

- Created features such as:
  - Article length
  - Presence of clickbait words
- Vectorized text using:
  - TF-IDF
  - N-grams (bi-grams)
- Optional: Dimensionality reduction using Truncated SVD

8. Model Building

- **Models Used**:
  - Logistic Regression
  - Random Forest
- **Justification**: Logistic Regression offers interpretability; Random Forest provides better performance for complex patterns.
- **Split**: 80/20 training/testing with stratification
- **Metrics**: Accuracy, Precision, Recall, F1-score

9. Visualization of Results & Model Insights

- Confusion Matrix for both models
- ROC-AUC Curve comparison

- Feature Importance from Random Forest
- TF-IDF top terms analysis
- Logistic Regression coefficient plots

10. Tools and Technologies Used

- **Language**: Python
- **IDE**: Google Colab
- **Libraries**: pandas, numpy, sklearn, nltk, matplotlib, seaborn, xgboost
- **Visualization**: matplotlib, seaborn, plotly

11. Team Members and Contributions

- **DINESH.K** – Data Collection and Cleaning

Responsible for sourcing the dataset, handling missing values, removing duplicates, and preparing raw data for analysis.

- **KARTHIK.M** – Exploratory Data Analysis (EDA)

Conducted univariate and bivariate analyses, visualized trends and patterns, and summarized key data insights.

- **GOKUL.M**– Feature Engineering & Text Preprocessing

Performed text preprocessing (tokenization, lemmatization, vectorization) and created new features using NLP techniques.

- **ARJUN.A** – Model Building and Evaluation

Implemented machine learning models (Logistic Regression, Random Forest), optimized performance, and evaluated with appropriate metrics.

- **HARIPANDIAN.P** – Documentation & Visualization

Compiled project documentation, prepared plots (confusion matrix, ROC, feature importance), and managed the GitHub repository