Phase-3

Exposing the Truth with Advanced Fake News Detection Powered by Natural Language Processing

Student Name: AKALYA. S

Register number: 620123106003

Institution: AVS ENGINEERING COLLEGE

Department: ECE

Date of submission: 17/05/2025

Github repository link:http://github.com/akalya1611-Akalya-S-naanmudhalvan-project-

1. Problem Statement

The rapid spread of fake news on digital platforms undermines public trust, influences elections, and spreads misinformation. Detecting and mitigating fake news is essential for preserving the integrity of information.

2. Abstract

This project leverages Natural Language Processing (NLP) to build an automated fake news detection system. By analyzing text-based features from news articles and applying machine learning models, the system aims to accurately classify newsas real or fake, aiding users and organizations in identifying credible information.

3. System Requirements

- Hardware: 8GB RAM, Intel i5 or higher processor
- Software: Python 3.8+, Jupyter Notebook, Scikit-learn, Pandas, NLTK, Flask (for deployment)
- OS: Windows/Linux/MacOS

4. Objectives

- Collect and preprocess news data.
- Perform exploratory data analysis to uncover patterns.
- Engineer features suitable for fake news classification.
- Train and evaluate machine learning models.
- Deploy the best-performing model as a web service.

5. Flowchart of Project Workflow

Data Collection -> Data Preprocessing -> Exploratory Data Analysis -> Feature Engineering -> Model Building -> Model Evaluation -> Deployment

6. Dataset Description

The dataset contains labeled news articles with fields such as title, text, subject, and label (real/fake). Common sources include Kaggle datasets like "Fake and Real News Dataset."

7. Data Preprocessing

- Removing null values
- Text normalization (lowercasing, removing punctuation)
- Stop word removal

- Tokenization
- Lemmatization

8. Exploratory Data Analysis

- Word frequency analysis
- Distribution of real vs. fake labels
- Subject-wise comparison
- Word clouds

9. Feature Engineering

- TF-IDF vectorization
- CountVectorizer
- N-grams
- Sentiment scores

10. Model Building

- Algorithms: Logistic Regression, Naive Bayes, Random Forest, Support Vector Machines
- Training with cross-validation

11. Model Evaluation

- Accuracy, Precision, Recall, F1-score
- Confusion Matrix
- ROC-AUC curve analysis

12. Deployment

The model is deployed as a RESTful API using Flask, enabling web integration. A simple UI allows users to input news text and receive classification output.

13. Source Code

All code is written in Python and structured into modules:
- data preprocessing.py
- model_training.py
- evaluate_model.py
- app.py (Flask deployment)

14. Feature Scope

- Multi-language support
- Real-time detection from URLs
- User feedback integration for model refinement

15. Team Members and Roles

Data Engineer :KOVARTHANA. M. S

Role: Manages data collection and pipeline.

Collect news data from APIs/websites.

Clean and store data in databases.

Build ETL pipeline

NLP Specialist: KOKILA. V

Role: Processes and analyzes text data.

Preprocess text (tokenization, stemming).

Apply NLP models (TF-IDF, BERT, etc.)

Extract features for model training.

Documentation Specialist : AKALYA. S

Role: Prepares technical and user documentation.

Write project reports and manuals.

Document model design and API usage.

Maintain project logs and references.