Github Link: https://github.com/madhivanan007/Madhi

Project title:Exposing the Truth - Advanced Fake News Detection Using NLP

PHASE-2:

1. Problem Statement

Misinformation and fake news pose significant challenges in today's digital landscape, influencing public opinion and distorting facts. The goal of this project is to develop a *Natural Language Processing (NLP)-based model* capable of distinguishing between genuine and fake news articles. The detection process leverages linguistic patterns, semantic analysis, and contextual clues to assess authenticity.

By applying *machine learning techniques* to real-world news datasets, this project aims to improve early misinformation identification, aiding journalists, policymakers, and users in verifying the credibility of online content.

2. Project Objectives

- Develop an NLP-powered machine learning model to classify news articles as *fake or legitimate*.

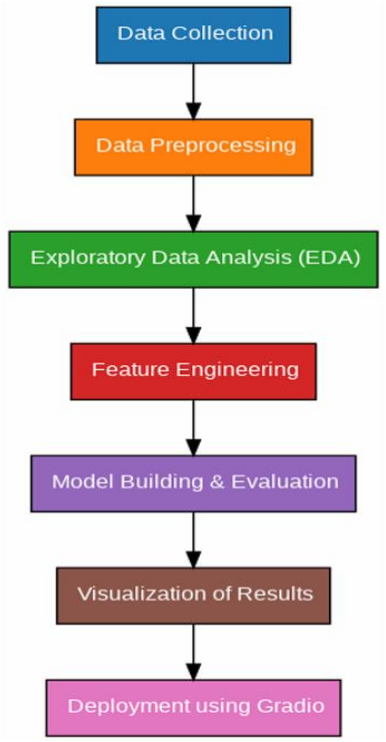
- Identify key linguistic and contextual features contributing to misinformation.

- Evaluate model interpretability and robustness against adversarial attacks.

- Deploy an interactive user-friendly interface using *Gradio* for real-time news verification.

- Enhance detection accuracy using deep learning methods like LSTMs or Transformers.

3. Flowchart of the Project Workflow



4. Data Description

-Dataset Name: Fake News Dataset (e.g., LIAR dataset, Kaggle Fake News dataset)

Source: Public repositories (Kaggle, UCI ML Repository, OpenNews datasets)

Records & Features: Thousands of labeled news articles with attributes like headline, article text, source, and publication date

Target Variable: **Binary label* (Real/Fake)

Dataset Type: Structured text data

5. Data Preprocessing

Text Cleaning: Removing stopwords, special characters, and non-text elements.

Tokenization & Lemmatization: Breaking down sentences and reducing words to their root forms.

Vectorization: Converting text data into numerical format using **TF-IDF* and *Word Embeddings (Word2Vec, BERT embeddings)*.

Handling Imbalanced Data: Using **SMOTE* or class weighting techniques to balance fake and real news distribution.

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- Word frequency distribution
- Sentiment analysis of fake vs real news

Bivariate & Multivariate Analysis:

- *Word cloud visualization* highlighting common terms in fake news vs legitimate news
- *Correlation matrix* of linguistic features (e.g., exaggeration, sensational words)

Key Insights:

- Fake news often uses *strong emotional language* and lacks credible sources.
- Legitimate news exhibits *neutral sentiment* with fact-based reporting.

7. Feature Engineering

- Extracted *NLP-driven features*:
  - Sentence complexity, readability scores, and emotional tone
  - Source credibility indicators (e.g., verified domains)
  - Named Entity Recognition (NER) for identifying real vs fabricated entities
- Created *syntactic & semantic patterns* using dependency parsing.

8. Model Building

Algorithms Used:

Logistic Regression (Baseline model)

Random Forest (Captures non-linear relationships)

Transformer-based models (BERT, RoBERTa)* for advanced contextual learning

Train-Test Split:

- 80% training, 20% testing

- Used *Stratified Sampling* to preserve class distribution

Evaluation Metrics:

- Accuracy, Precision, Recall, *F1-score* (crucial for imbalanced data)

9. Visualization of Results & Model Insights

Feature Importance:

- Visualized using SHAP plots for interpretability.

- *High importance* given to sentiment, linguistic complexity, and publication source.

Model Comparison:

- Compared model performance using *ROC-AUC curves* and *Confusion Matrices*.

User Testing:

- *Gradio Interface* enables users to input article text and receive a credibility assessment.

10. Tools and Technologies Used

Programming Language: Python 3

Notebook Environment: Google Colab

Key Libraries:

- nltk, spaCy for NLP preprocessing

- pandas, numpy for data handling

- matplotlib, seaborn, plotly for visualizations

- scikit-learn, transformers for model training

- Gradio for deployment

11. Team Members and Contributions

M.Kumaran: data preprocessing, EDA.

N.Madhivanan:feature engineering, model development.

M.Pooja: evaluation, documentation.