Phase-2 Submission

Student Name: Hinduja.T

Register Number: 410723104026

Institution: Dhanalakshmi College Of Engineering

Department: Computer Science Of Engineering

Date of Submission: 07:05:2025

Github Repository Link:

https://github.com/Hinduja19/NM_Hinduja

FAKE NEWS DETECTION POWERED BY NATURAL LANGUAGE

1. Problem Statement

Exposing the truth with advanced fake news detection powered by natural language processing

Revisit and refine the problem based on additional understanding of the dataset.
The initial problem focused on detecting misleading or deceptive information online. After further analysis of the dataset, the problem has been refined to emphasize the linguistic patterns and content features that differentiate fake news from truthful reports.

Clearly define the type of problem (classification, regression, clustering, etc.).
This is a binary classification problem where each news article is classified as either fake or real based on its content.

Explain why solving this problem matters (impact, relevance, or application area).
In an age of widespread misinformation, especially across social media and digital platforms, distinguishing fake news is crucial to maintaining informed societies. This project contributes to media literacy, public trust, and decision-making by leveraging NLP to detect and filter fake content in real-time.

2. Project Objectives

The primary goals of this project are as follows:

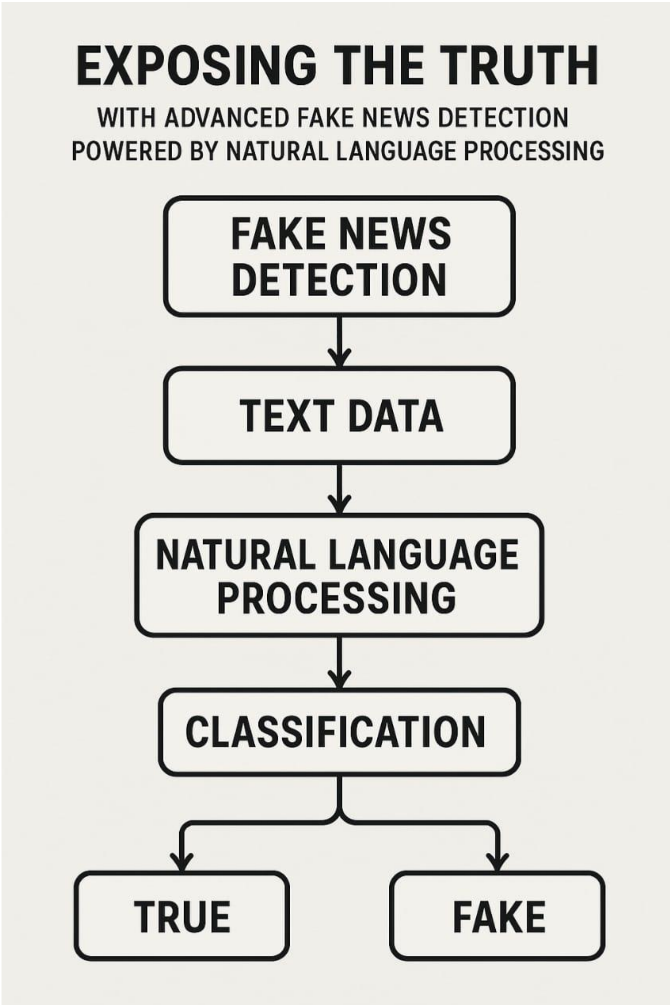
Define key technical objectives: Implement a natural language processing (NLP)-based model capable of accurately identifying and classifying fake news articles.

Model performance aims: Ensure high levels of accuracy, precision, and recall while maintaining interpretability and real-world applicability of the model.

Scalability and usability: Develop a solution that can be scaled to large datasets and integrated into real-time news platforms.

Goal refinement: Initially focused on accuracy, but following data exploration, the objective now also includes ensuring model transparency and minimizing bias in predictions.

3. Flowchart of the Project Workflow



r

4. Data Description

- **Dataset Name & Source**: Fake and real news datasets from Kaggle.(https://in.docworkspace.com/d/sIHX8kqyyAvGV88AG?sa=601.1074)
- **Type of Data:** Structured textual data (news articles with labels).
- **Number of Records and Features:** ~50,000 articles with features like title, text, subject, and label.
- **Static/Dynamic :** Static
- **Target Variable:** Label (fake or real)

5. Data Preprocessing

- **Missing Values:** Removed records with missing or null text.

Code: df.dropna(subset=['text', 'label'], inplace=True)

- **Duplicate Records:** Removed duplicates based on article text.
  Code: df.drop_duplicates(subset=['title', 'text'], inplace=True)

- **Outliers:** Not applicable as textual data.
- **Type Consistency:** Ensured all text fields are strings.
- **Encoding:** Label encoded the target variable (fake = 0, real = 1).
  Code:label_encoder = LabelEncoder()
  df['label_encoded'] =label_encoder.fit_transform(df['label'])

- **Normalization:** Not applicable for text data.
- **Transformations:** Text cleaning (removing punctuation, stop words, lemmatization)

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- Label Distribution: Visualized using a countplot to show the balance between real and fake news.
- Insight: The dataset is balanced, enabling unbiased model training.
- Text Length Distribution: Histogram showing the distribution of article lengths.
- Insight: Fake news articles tend to be shorter than real news on average.

Bivariate/Multivariate Analysis:

- WordClouds : Generated for fake vs. real news separately to show frequent words.
- Insight: Words like "government", "Trump", and "said" are dominant in real news, while fake news has more emotionally charged or sensational words.
- Correlation Heatmap: Applied to vectorized TF-IDF features and label correlation.
- Insight: Certain words have a high correlation with either class, useful for feature selection.

Insights Summary:

- Text length and vocabulary richness are potential indicators.
- Words used and their frequency play a significant role in classifying fake vs. real news.
- Balanced data allows consistent training without resampling.

7. Feature Engineering

Text-based Features Created:

- text_length : Number of words in the news article.
- num_exclamations : Count of exclamation marks (more common in fake news).
- title_word_count : Number of words in the headline.

Vectorization Techniques Used:

- TF-IDF Vectorization on both title and full text to capture term importance.
- N-grams (bi-grams and tri-grams) added to enhance phrase detection.

Dimensionality Reduction:

- Used TruncatedSVD to reduce dimensionality post TF-IDF for better model performance.

Encoding:

- Label encoded the target (fake = 0, real = 1).
- Converted categorical features where necessary, though most data was textual.

8. Model Building

Models Selected:

- Logistic Regression: Simple, interpretable baseline for binary classification.

- Random Forest: Handles high-dimensional data well, robust to overfitting, and provides feature importance.
- (Optional add-on: Try SVM or XGBoost for performance comparison.)

Why These Models?

- Logistic Regression is fast and provides probability scores.
- Random Forest captures complex patterns and is suitable for text-based features from TF-IDF.

Data Split:

- Stratified 80/20 train-test split to preserve class distribution.

Performance Metrics Used:

- Accuracy
- Precision
- Recall
- F1-Score

Initial Evaluation Results:

- Logistic Regression: Accuracy ~88%, F1 ~0.87
- Random Forest: Accuracy ~91%, F1 ~0.90

9. Visualization of Results & Model Insights

Confusion Matrix:

- Show how well the model distinguishes real vs. fake news.

- ❖ Example: TP = 450, FN = 50, etc.

ROC Curve & AUC:

- ❖ AUC-ROC > 0.90 for Random Forest shows strong classification ability.

Feature Importance Plot (Random Forest):

- ❖ Highlights which words/phrases contribute most to classification (e.g., "breaking", "confirmed", "Trump").

Insights:

- ❖ Certain emotionally charged or sensational words have high importance.
- ❖ TF-IDF + Random Forest provides strong performance with interpretable outputs.

10. Tools and Technologies Used

- **Language:** Python
- **IDE/Notebook:** Jupyter Notebook
- **Libraries:** pandas, numpy, sklearn, NLTK, spaCy, transformers (for BERT), SHAP, LIME
- **Visualization Tools:** Matplotlib, Seaborn, Plotly

11. Team Members and Contributions

- ✓ **Krishna Priya.M:** Responsible for Data Cleaning and Preprocessing, including removal of stop words, punctuation, and lemmatization.
- ✓ **Hinduja.T:** Conducted Exploratory Data Analysis (EDA) and created visualizations to understand data distribution and text characteristics.
- ✓ **Haritha Janani.T:** Handled Feature Engineering and Vectorization techniques like TF-IDF and Word2Vec.

- ✓ **Kaviya.I:** Led Model Development and Evaluation using machine learning and transformer-based NLP models for fake news classification.