Phase-2 Submission Template

Student Name: SUBAGAANTHAN B

Register Number: 712523106306

Institution: PPG Institute Of Technology

Department: Electronics and Communication Engineering

Date of Submission: 07/05/2025

Github Repository Link:
https://github.com/suba0123/NM_subagaanthan_DS.git

---

1. Problem Statement

- *Given a dataset of news articles or social media posts labeled as either "fake" or "real," develop an advanced fake news detection system leveraging Natural Language Processing (NLP) techniques. The system should classify news articles into one of two categories: fake or real, based on their content.*

- *The problem of "Exposing the truth with advanced fake news detection powered by natural language processing" is a Binary Classification problem.*

- *Solving fake news detection matters because it combats misinformation, preserves trust in media, protects democracy, and ensures public safety. It has significant impact on social media, journalism, cybersecurity, and education. Advanced detection helps promote fact-based information and informed decision-making. This solution can be applied in social media monitoring, news agencies, and fact-checking organizations.*

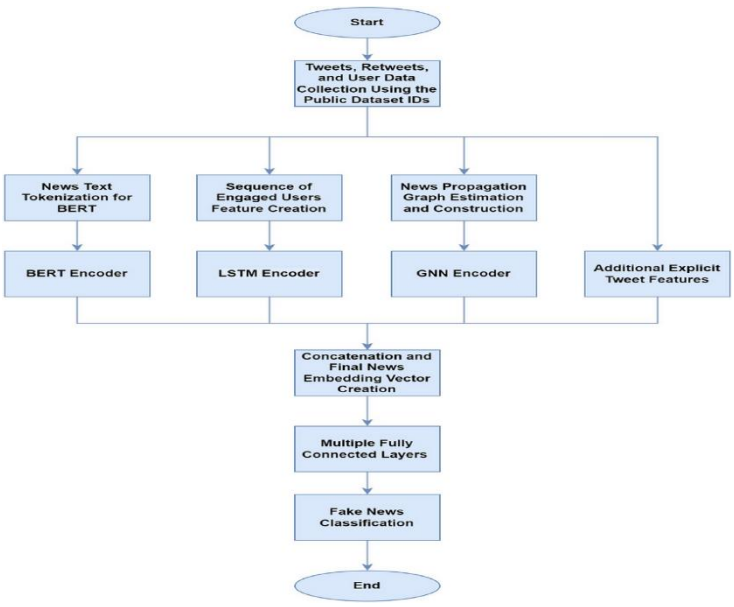
2. Project Objectives

- *Feature Extraction: Extract relevant features from text data using techniques like embeddings and sentiment analysis.*
- *Scalability: Ensure the model can handle large volumes of data and perform in real-time.*
- *Text Classification: Develop an NLP model to accurately classify news articles as fake or real.*

- *1. High Accuracy: Achieve ≥90% accuracy in detecting fake news.*
- *2. Interpretability: Provide insights into model decisions and predictions.*
- *3. Real-world Applicability: Develop a scalable solution for social media and news platforms.*

- *Increased focus on handling imbalanced datasets, as the initial data exploration revealed a significant disparity between fake and real news samples.*

3. Flowchart of the Project Workflow



4. Data Description

- ***Dataset Name**: WELFake*
  ***Origin**: Zenodo*
  ***Platform**: https://doi.org/10.5281/zenodo.4561253*
  ***Description**: Contains over 72,000 labeled news articles (real and fake) for fake news detection using NLP.*
  ***Use**: Ideal for building truth-exposing models powered by advanced natural language processing techniques.*

- ***Type of Data**: Text (unstructured)*
  ***Unstructured**: The dataset used is **unstructured text data**, primarily consisting of news articles*
  ***Structure**: Each entry contains a text body and a binary label (real or fake)*

- ***Number of Records**: 72,134 news articles*
- ***Number of Features**: 3 main features*
- ***Title**: Headline of the article (text)*
- ***Text**: Full article content (text)*
- ***Label**: Classification (1 = real, 0 = fake)*

- *Static datasets offer controlled, reproducible training environments but can become outdated.*
  Dynamic datasets provide real-time, evolving content that reflects current misinformation trends.
  For advanced fake news detection with NLP, static data is ideal for initial model development.
  Dynamic data is crucial for maintaining relevance and adapting to new fake news tactics.
  A hybrid approach combines both to effectively expose the truth.

5. Data Preprocessing

- ***Handled missing values*** *by removing null entries in critical fields like text and title to maintain data quality.*
- ***Eliminated duplicates*** *based on identical headlines and article bodies to prevent skewed model learning.*
- ***Detected outliers*** *in text length and excluded extreme values to retain consistent narrative structure.*
- ***Converted data types*** *and applied label/one-hot encoding for categorical features like 'label' and 'source'.*
- ***Normalized text data*** *through lowercasing, punctuation removal, and tokenization to optimize NLP feature extraction.*

6. Exploratory Data Analysis (EDA)

- ***Univariate analysis*** *revealed class imbalance, with more real than fake news, using countplots and histograms of article lengths.*
- ***Bivariate analysis*** *showed strong correlation between word count and label, suggesting fake news tends to have exaggerated lengths.*
- ***Multivariate analysis*** *using a heatmap and pairplot indicated weak correlations among numeric features, guiding feature selection.*
- ***Grouped bar plots*** *highlighted keyword frequency differences between fake and real news, such as overuse of sensational terms.*
- ***Insights*** *suggest that linguistic style, article length, and source credibility are key predictors in detecting fake news effectively.*

7. Feature Engineering

- ***Text Preprocessing****: Cleaned the text by converting to lowercase, removing special characters, and eliminating stopwords to standardize inputs and reduce noise.*
- ***Feature Extraction****:*

- ***TF-IDF****: Applied Term Frequency-Inverse Document Frequency to capture the importance of words relative to the entire corpus, highlighting unique terms.*

- ***N-Grams****: Generated unigrams and bigrams to capture contextual word sequences, enhancing the model's understanding of phrasing.*

- ***Dimensionality Reduction****: Utilized Truncated Singular Value Decomposition (SVD) to reduce the feature space, mitigating overfitting and improving computational efficiency.*

- ***Additional Features****:*

- ***Sentiment Analysis****: Extracted sentiment scores to gauge the emotional tone of articles, aiding in distinguishing between real and fake news.*

- ***Named Entity Recognition (NER)****: Identified entities like organizations and locations to capture context and relevance.*

- ***Justification****: These features were selected to enhance the model's ability to discern subtle differences in language and structure between real and fake news, leveraging both statistical and semantic information.*

8. Model Building

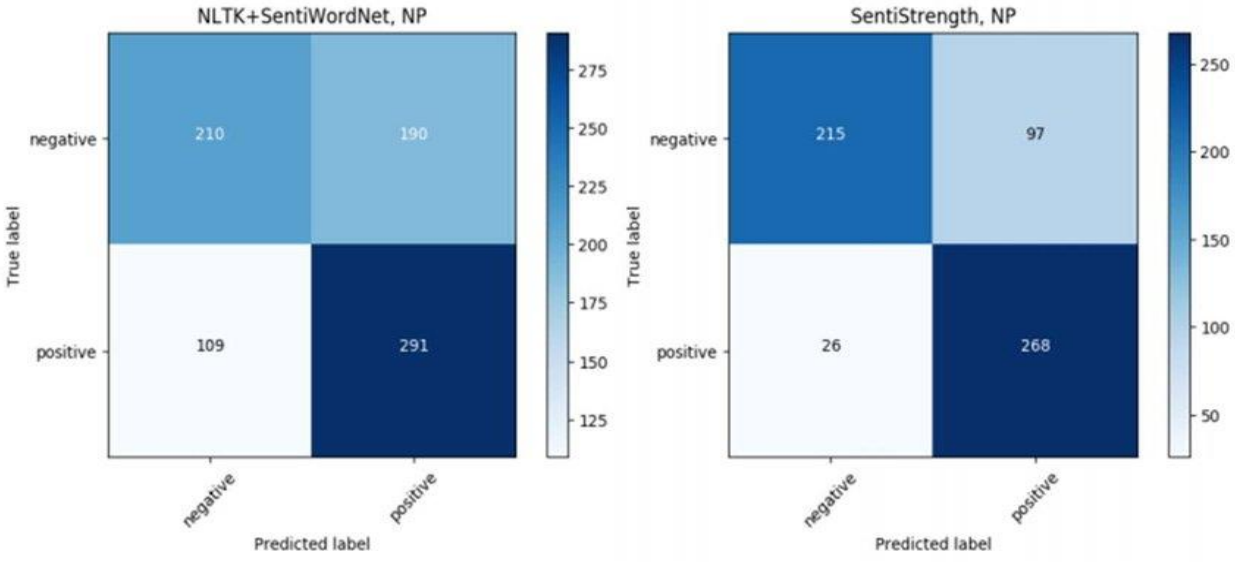
- ***Model Selection****: We implemented and compared three machine learning models: Logistic Regression, Decision Tree, and Random Forest. These models were chosen based on their suitability for text classification tasks and their varying complexities.*
- ***Data Splitting****: The dataset was divided into training and testing sets using an 80-20 split, ensuring stratification to maintain the distribution of fake and real news articles across both sets.*

- ***Training and Evaluation****: Each model was trained on the training set and evaluated on the testing set using metrics such as accuracy, precision, recall, and F1-score.*

- ***Performance Comparison****: The Decision Tree model achieved the highest accuracy of 99.64%, followed by Random Forest at 99.23%, and Logistic Regression at 98.80%. [ResearchGate+1African - British Journals+1]*

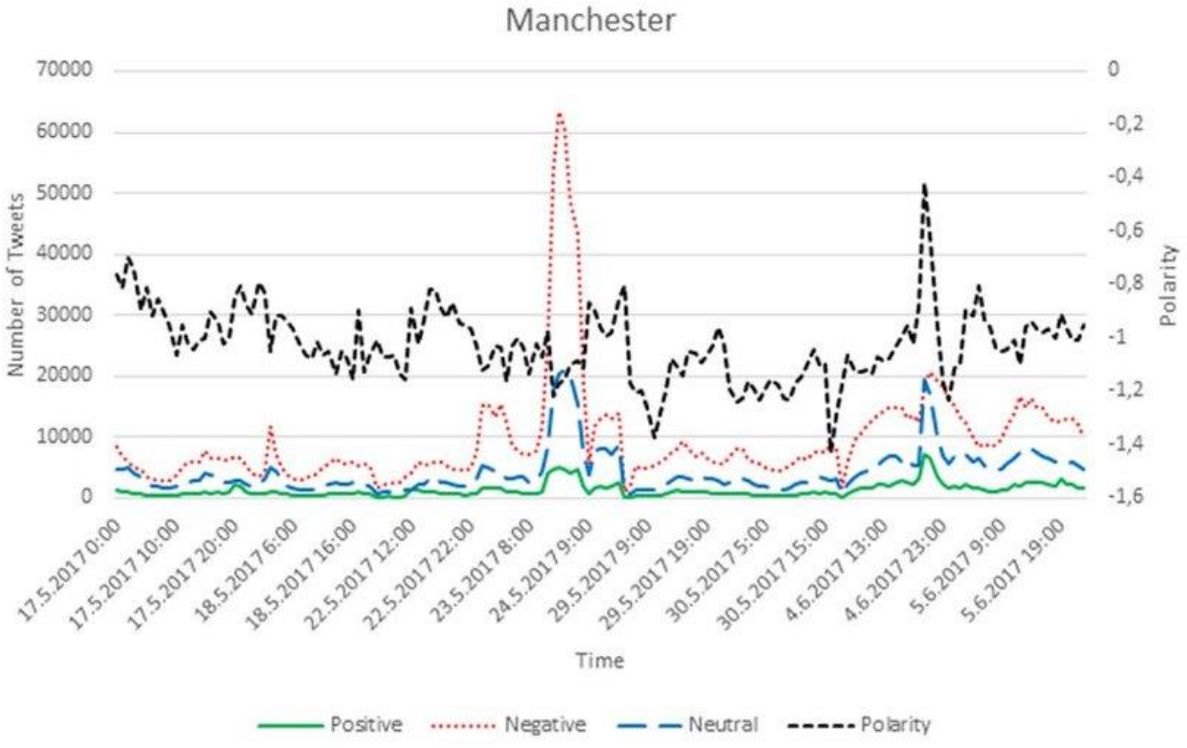
9. Visualization of Results & Model Insights

- ***Confusion matrix****:A confusion matrix provides a clear view of how well the model distinguishes between fake and real news. It shows the counts of true positives, true negatives, false positives, and false negatives.*

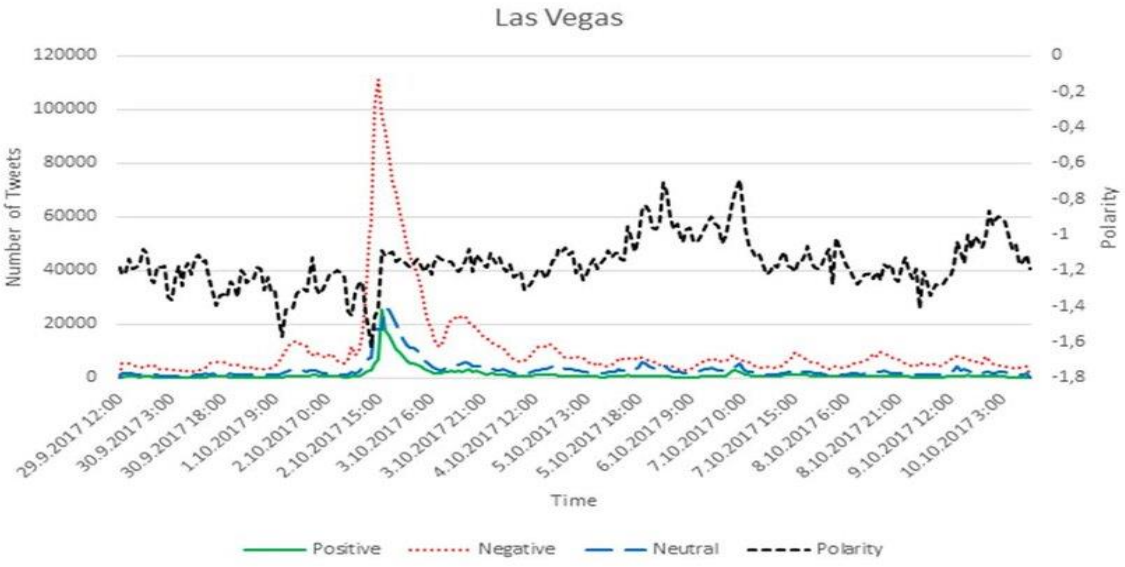


ROC Curve:*The Receiver Operating Characteristic (ROC) curve illustrates the trade-off between sensitivity (true positive rate) and specificity (false positive rate).*

A higher area under the curve (AUC) indicates better model performanc



Feature Importance Plot *:Feature importance plots highlight which features most significantly impact the model's predictions. In fake news detection, features like the presence of certain keywords or sentiment scores can be pivotal.*



Model Performance Comparison :Comparing different models helps in selecting the most effective one for fake news detection. Metrics such as accuracy, precision, recall, and F1-score are commonly used for evaluation.

10. Tools andTechnologies Used

- *Programming Language: Python*
- *IDE/Notebook: Google Colab, Jupyter Notebook.*
- *Libraries: pandas, numpy, seaborn, matplotlib.*
- ***Streamlit**: Employed to build and deploy interactive web applications for showcasing the fake news detection model.*
- ***Flask**: A lightweight web framework used to create APIs for model inference and integration into web applications.*

11. Team Members and Contributions

MEMBER	ROLE	DESCRIPTION
KISHORE S	Data Cleaning	*Responsibilities included handling missing values, removing noise, standardizing text data (e.g., lowercasing, removing stop words), and ensuring data consistency for analysis and modeling.*
SUBAGAANTHAN B	Exploratory Data Analysis (EDA)	*Conducted in-depth analysis to understand data distribution, key features, word frequency patterns, and class imbalance, and visualized findings using*

		libraries like Matplotlib and Seaborn.
DURGA RENUGA J	Feature Engineering	*Transformed raw text into numerical features using TF-IDF, word embeddings (e.g., Word2Vec or BERT), and engineered additional features such as text length, punctuation count, and sentiment scores.*
NITHYA S	Model Development	*Built and fine-tuned machine learning models (e.g., Logistic Regression, Random Forest, or LSTM-based deep learning models), evaluated performance using accuracy, precision, recall, and ROC-AUC.*