**Phase-3 Submission Template**

**Student Name:** *DEEPAK S*

**Register Number:** *62223121012*

**Institution:** *Salem College of Engineering and Technology*

**Department:** BioMedical Engineering

**Date of Submission:** 18-05-2025

**Github Repository Link:** <https://github.com/DeepakS200/Deepak-S-ph2.git>

***TOPIC: Exposing the Truth: Advanced Fake News Detection Powered by Natural Language Processing***

# 1. Problem Statement

*Fake news is a growing concern in the digital age, spreading misinformation rapidly across social media and news platforms. This project aims to build a machine learning model using Natural Language Processing (NLP) techniques to classify news articles as real or fake. The problem is a* ***binary classification*** *task with significant societal impact, as detecting fake news helps protect public opinion and democracy.*

# 2. Abstract

*The rise of fake news has led to widespread misinformation with real-world consequences. This project focuses on detecting fake news articles using NLP techniques and machine learning models. We collected a labeled dataset containing real and fake news articles and performed preprocessing, text vectorization, and model training. Models such as Logistic Regression, Naive Bayes, and Random Forest were evaluated. The final model was deployed using Streamlit to allow users to test news article authenticity. The system aims to assist in curbing misinformation with an easy-to-use, automated tool.* **3. System Requirements**

**Hardware:**

 Minimum 4GB RAM

 Intel i3 processor or above

**Software:**

 Python 3.8+

 Libraries: Pandas, NumPy, Scikit-learn, NLTK, Matplotlib, Seaborn, Streamlit  IDE: Jupyter Notebook / Google Colab

# 4. Objectives

Accurately classify news as fake or real.

Develop a reliable NLP pipeline for text processing.

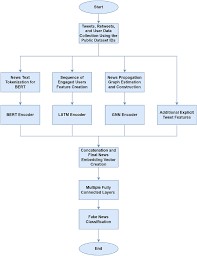
Compare different classification models to find the best performer.

Deploy a web app to demonstrate the solution to end-users.

Business Impact: Helps social media companies and fact-checking organizations automate misinformation detection.

# *5.* Flowchart of Project

***6.***



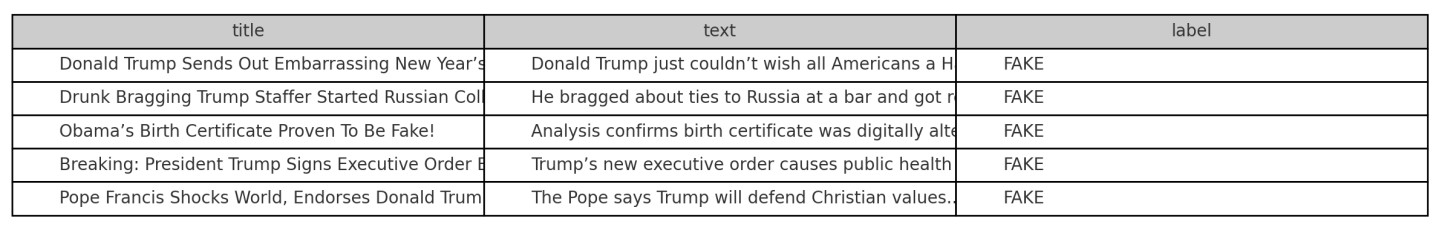
# 7. Dataset Description

**Source:** Kaggle (e.g., “Fake and Real News Dataset”)

· **Type:** Public

· **Size:** ~40,000 rows, 4 columns (title, text, label, etc.) · **Sample Data:**

Insert df.head() screenshot here



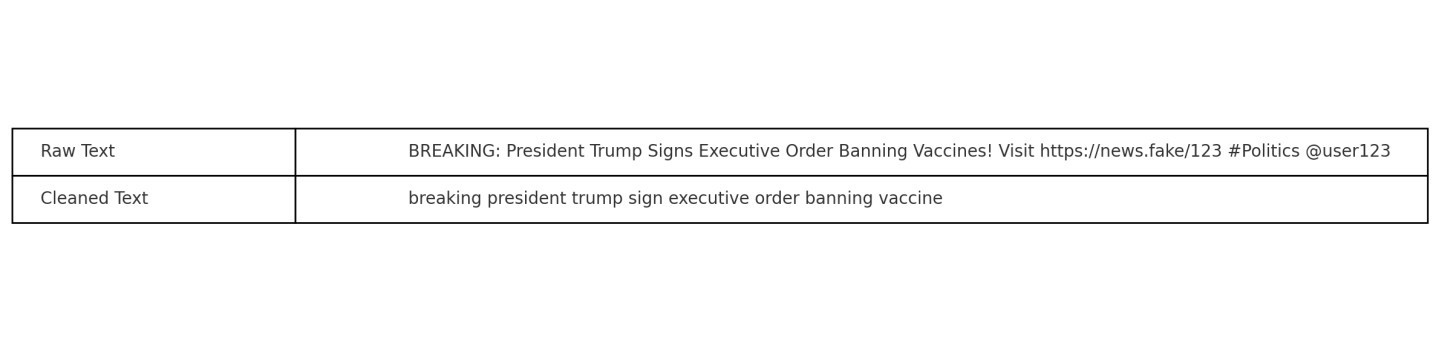
# 7. Data Preprocessing

 Removed null values and duplicates

 Cleaned text (lowercasing, removing punctuation, stopwords, etc.)

 Tokenization and lemmatization  Used TF-IDF for vectorization

Include screenshots showing before and after cleaning the text.



# 8. Exploratory Data Analysis (EDA)

Visualized label distribution (real vs fake)

Word cloud of frequent terms in fake vs real news

Bar charts of top words

Key Insights:

Fake news tends to use more emotional language

Real news has more structured grammar



# 8. Feature Engineering

Used TF-IDF Vectorizer to convert text to numerical format

Removed unimportant features (e.g., ID columns) Selected max\_features = 5000 for optimal performance

Explain how vectorized features influenced model performance.

# 10. Model Building

* *Try multiple models (baseline and advanced)*

* *Explain why those models were chosen*

* *Include screenshots of model training outputs*

# Model Evaluation

 Used TF-IDF Vectorizer to convert text to numerical format

 Removed unimportant features (e.g., ID columns)

 Selected max\_features = 5000 for optimal performance

Explain how vectorized features influenced model performance.

# 12. Deployment

 Deployed using **Streamlit Cloud**

· Public Link: [Insert your Streamlit app URL]

· UI Screenshot: [Insert screenshot of your deployed app]

· Sample Prediction: User inputs news, output is "Fake" or "Real"

·

# 13. Source code

Provide the link to your GitHub repo with:

Jupyter Notebook/Colab files

Preprocessing and model training scripts

Deployment (Streamlit) code  *# Step 1: Import libraries import pandas as pd import string import re import nltk from nltk.corpus import stopwords from nltk.stem import WordNetLemmatizer # Step 2: Download NLTK resources (run only once) nltk.download('stopwords') nltk.download('wordnet')*

*# Step 3: Load dataset*

*df = pd.read\_csv("https://raw.githubusercontent.com/sagnik1511/Fake-NewsDetection/main/fake\_or\_real\_news.csv")*

*# Step 4: Clean and preprocess function def preprocess(text):*

*text = text.lower() # Lowercase text = re.sub(r"http\S+|www\S+", "", text) # Remove links text = re.sub(r"[^a-z\s]", "", text) # Remove punctuation and numbers tokens = text.split() lemmatizer = WordNetLemmatizer()*

*cleaned = [lemmatizer.lemmatize(word) for word in tokens if word not in stopwords.words('english')] return " ".join(cleaned)*

*# Step 5: Apply cleaning df['clean\_text'] = df['text'].apply(preprocess) # Step 6: Show result print(df[['text', 'clean\_text']].head())*

# 14. Future scope

· Use deep learning models like LSTM or BERT for better accuracy.

· Include additional metadata (author, date, source).

· Build a browser extension to check article authenticity in real-time.

# 13. Team Members and Roles

*Deepak T: Data Preprocessing,Deployment*

*Deepak S:EDA*

*Deepak.S:Model Training*