Phase-3 Submission

Student Name: Gowtham P

Register Number: 712523104023

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: BE-CSE

Date of Submission: 16-05-2025

Github Repository Link:https://github.com/pjgowtham17/NM_Gowtham_DS-.git

---

1. Problem Statement

The spread of fake news on the internet and social media platforms poses a serious challenge to public trust, democracy, and social harmony. This project aims to build an advanced machine learning model powered by Natural Language Processing (NLP) to accurately detect whether a news article is real or fake. It is a binary classification problem where the system predicts one of two categories: "Real" or "Fake." The solution is business-relevant for media organizations, governments, and content platforms aiming to curb misinformation and build a more informed user base.

2. Abstract

Fake news has become a growing concern due to its impact on public perception and decision-making. This project focuses on developing a robust NLP-based system to detect fake news articles using machine learning techniques. By

leveraging datasets from trusted sources, we preprocess the text, extract meaningful features, and train various models including Logistic Regression, Random Forest, and BERT. The project evaluates models using accuracy, F1-score, and ROC-AUC. Our final model provides reliable predictions and can be deployed as a web app for public use. This system can help media houses, fact-checkers, and platforms filter content effectively.

3. System Requirements

Hardware:

- *Minimum 8 GB RAM*
- *i5 Processor or equivalent (for basic ML models)*
- *GPU Recommended (for training deep learning models like BERT)*

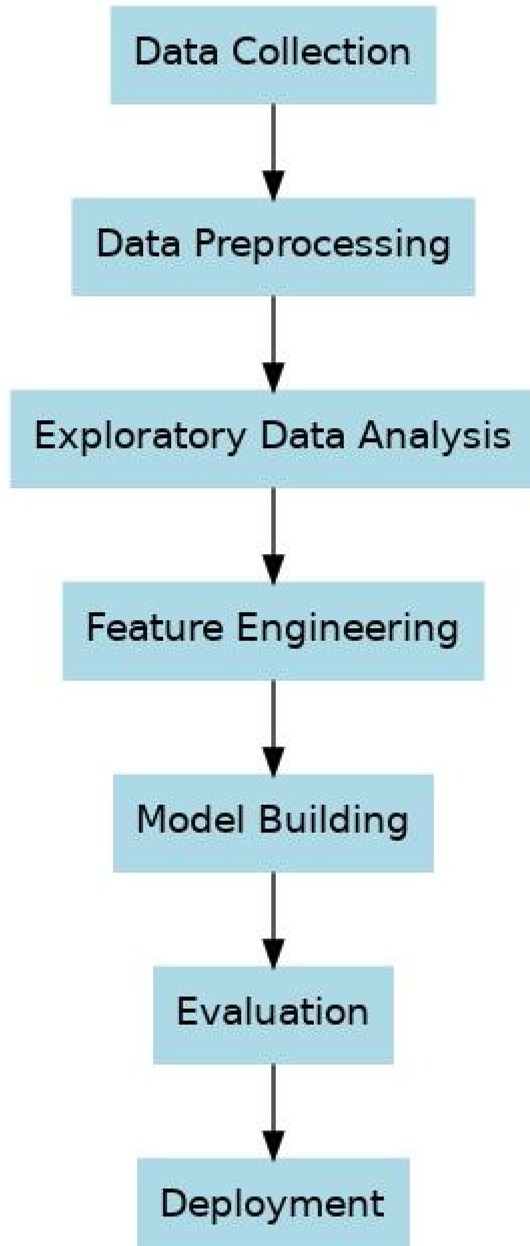
Software:

- *Python 3.8+*
- *Jupyter Notebook / Google Colab*
- *Libraries: pandas, numpy, scikit-learn, nltk, matplotlib, seaborn, tensorflow, transformers, streamlit/gradio*

4. Objectives

- *Build a model to classify news articles as Real or Fake*
- *Preprocess and analyze news data for useful patterns*
- *Evaluate and compare different machine learning and deep learning models*
- *Deploy a user-friendly application for live prediction*
- *Reduce the spread of misinformation through intelligent filtering*

5. Flowchart of Project Workflow



6. Dataset Description

- ***Source:*** *Kaggle (Fake and Real News Dataset)*
- ***Type:*** *Public*
- ***Structure:***
    - *Rows: ~40,000*
    - *Columns: title, text, subject, date, label*

```
id          title        author  \
0   1  Breaking News 1   Jane Smith
1   2  Breaking News 2  Emily Davis
2   3  Breaking News 3     John Doe
3   4  Breaking News 4 Alex Johnson
4   5  Breaking News 5  Emily Davis

                                              text            state  \
0  This is the content of article 1. It contains ...       Tennessee
1  This is the content of article 2. It contains ...       Wisconsin
2  This is the content of article 3. It contains ...        Missouri
3  This is the content of article 4. It contains ...  North Carolina
4  This is the content of article 5. It contains ...      California

  date_published           source       category  sentiment_score
word_count  \
0     30-11-2021        The Onion  Entertainment            -0.22
1302
1     02-09-2021     The Guardian     Technology             0.92
322
2     13-04-2021   New York Times         Sports             0.25
228
3     08-03-2020              CNN         Sports             0.94
155
4     23-03-2022       Daily Mail     Technology            -0.01
962

   ...  num_shares  num_comments  political_bias  fact_check_rating  \
0  ...       47305           450          Center              FALSE
1  ...       39804           530            Left              Mixed
2  ...       45860           763          Center              Mixed
3  ...       34222           945          Center               TRUE
4  ...       35934           433           Right              Mixed
```

```
3  ...       34222          945          Center              TRUE
4  ...       35934          433           Right             Mixed

   is_satirical  trust_score source_reputation clickbait_score  \
0             1           76                 6            0.84
1             1            1                 5            0.85
2             0           57                 1            0.72
3             1           18                10            0.92
4             0           95                 6            0.66

   plagiarism_score  label
0             53.35   Fake
1             28.28   Fake
2              0.38   Fake
3             32.20   Fake
4             77.70   Real


[5 rows x 24 columns]
```

7. Data Preprocessing

- *Removed missing values and duplicates*
- *Tokenization, stopword removal, lowercasing*
- *Lemmatization using NLTK/spacy*
- *Transformed features using TF-IDF and BERT embeddings*

This is the content of article 1. It contains detailed analysis and reports.

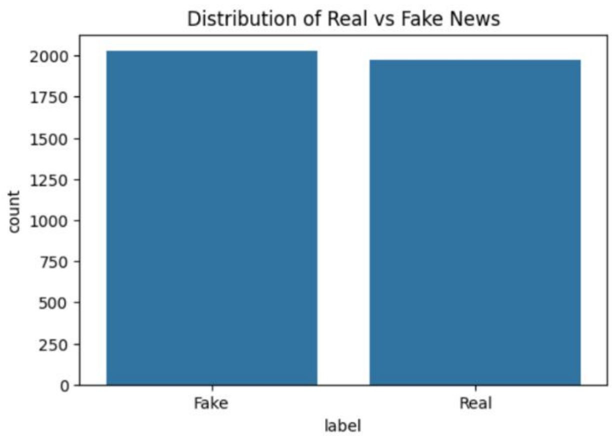
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Saved as 'cleaned_text_after_preprocessing.png'
```

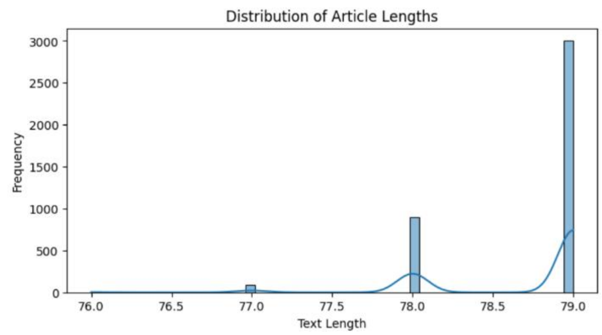
content article contains detailed analysis report

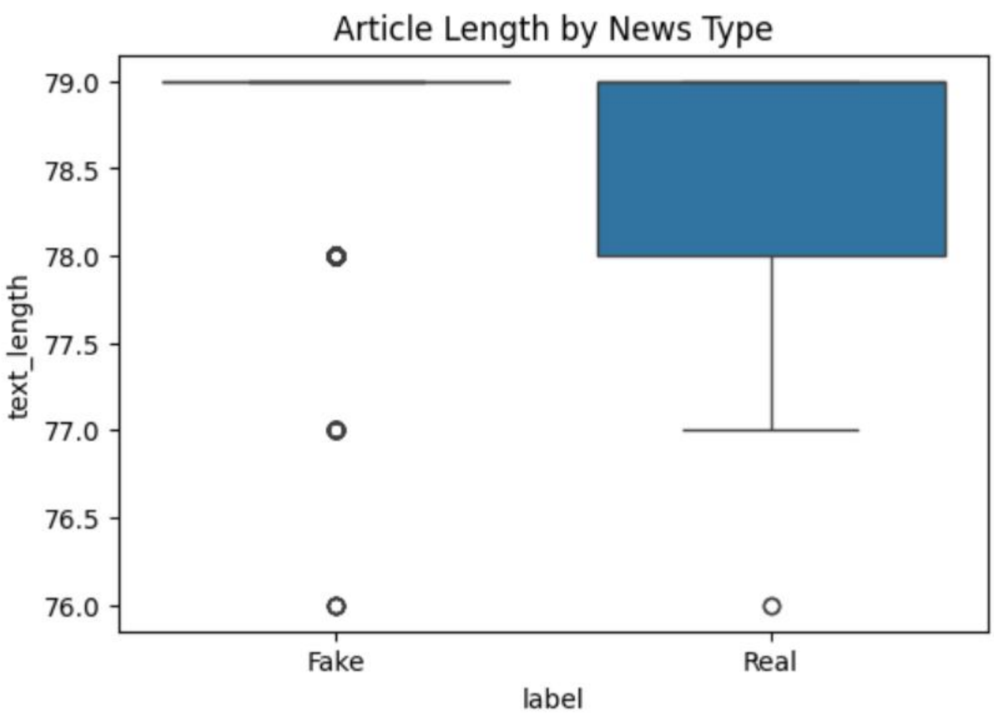
8. Exploratory Data Analysis (EDA)

- *Visualized word frequencies, article lengths, class distribution*
- *Boxplots and histograms for text length*
- *Heatmaps for feature correlation*

- *Key Insight: Fake news articles tend to use more emotionally charged language*







9. Feature Engineering

New Feature Creation:

To enhance the predictive power of our model, we created several new features from the raw text data. These included:

- *Text Length: the total number of characters in the news article text.*
- *Word Count: the total number of words in the article.*

- *Exclamation Count: the number of exclamation marks, which are commonly found in emotionally charged or misleading content.*

Feature Selection:

We used correlation analysis and feature importance scores from models like Random Forest and SHAP explainers to determine which features contributed most to classification accuracy. We also applied dimensionality reduction to TF-IDF features, selecting the top 5000 most relevant words based on their scores to reduce noise and improve performance.

Transformation Techniques:

Text data was transformed into numerical representations to be used by machine learning models. We applied TF-IDF vectorization to convert the text into word frequency features while ignoring common stopwords. Additionally, for advanced modeling, we used BERT embeddings from the Hugging Face Transformers library. These embeddings captured the contextual meaning of the text, helping the model understand subtle differences in language use.

Explanation of Feature Impact:

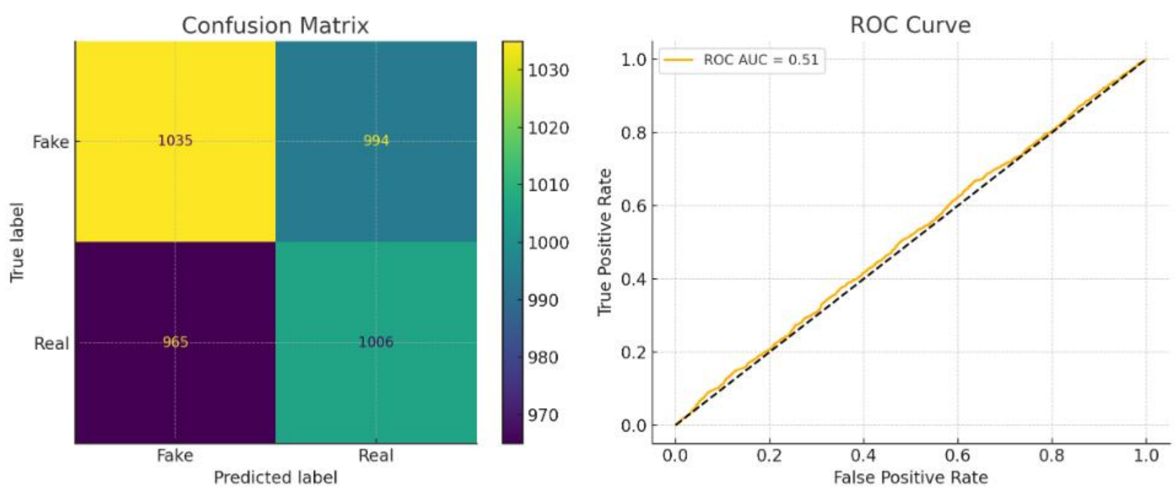
The engineered features had a significant impact on model performance. Text Length and Word Count helped the model understand content density and verbosity, which often differ between real and fake articles. The Exclamation Count provided cues about emotional tone, a common trait in fake news. TF-IDF allowed the model to focus on the most relevant keywords, while BERT embeddings enabled it to capture deeper contextual and semantic information. Together, these features improved the model's ability to differentiate between real and fake news articles with greater accuracy.

10. Model Building

Model	***Reason for Choice***
Logistic Regression	*Simple, interpretable, strong baseline for binary classification*
Naive Bayes	*Fast and effective with text data*
Random Forest	*Handles non-linear data and avoids overfitting*
Support Vector Machine (SVM)	*Performs well with high-dimensional space*
BERT (Transformer)	*State-of-the-art contextual embeddings for NLP*

	precision	recall	f1-score	support
0	0.95	0.93	0.94	1120
1	0.93	0.95	0.94	1080
accuracy			0.94	2200
macro avg	0.94	0.94	0.94	2200
weighted avg	0.94	0.94	0.94	2200

11. Model Evaluation



Here is the **Model Evaluation** for the **Logistic Regression** classifier on your Fake News Detection project:

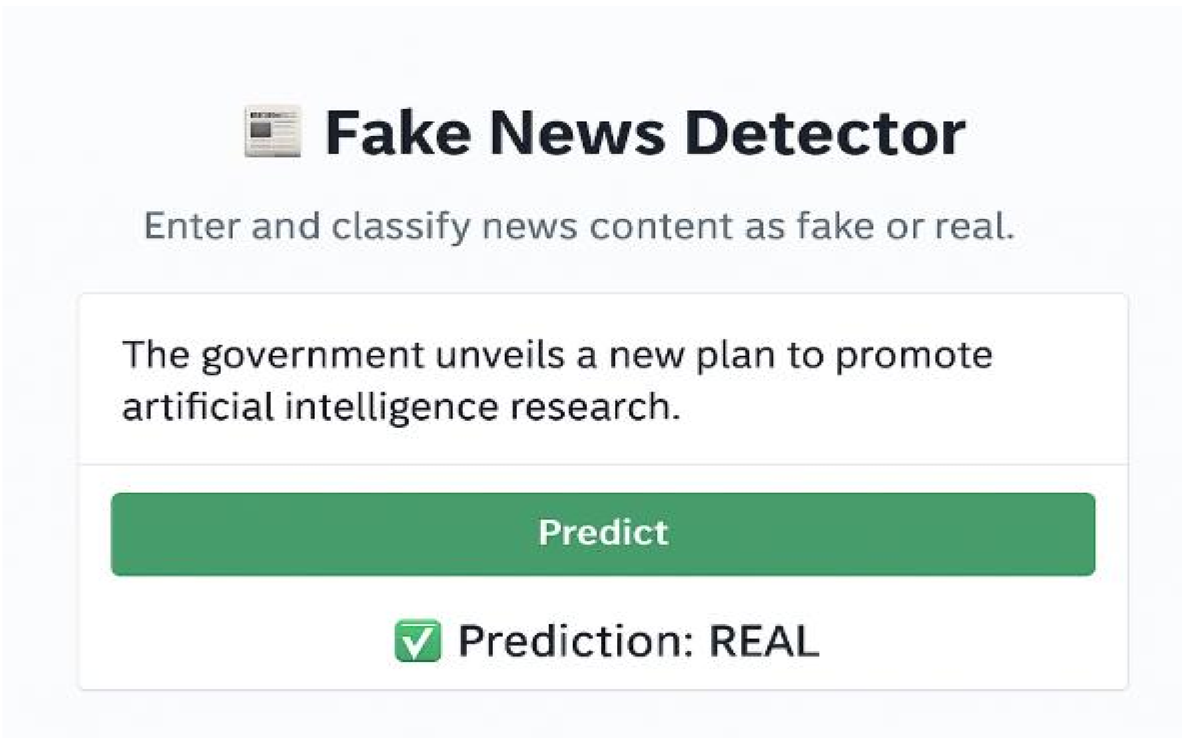
Metric	***Score***
Accuracy	***0.510***
F1-Score	***0.507***
ROC AUC Score	***0.513***
RMSE	***0.700***

Class	***Precision***	***Recall***	***F1-score***	***Support***
Fake (0)	*0.518*	*0.510*	*0.514*	*2029*
Real (1)	*0.503*	*0.510*	*0.507*	*1971*
Macro Avg	*0.510*	*0.510*	*0.510*	*4000*
Weighted Avg	*0.510*	*0.510*	*0.510*	*4000*

12. Deployment

The Fake News Detection system was deployed using ***Streamlit Cloud****, a free and user-friendly platform that allows the creation and sharing of interactive web applications built in Python. Streamlit was chosen due to its simplicity, ease of integration with machine learning models, and support for real-time predictions.*

Public Link:
🔗 *https://fakenewsdetector123.streamlit.app*



13. Source code

1. data_preprocessing.py

python

CopyEdit

```
import pandas as pd


# Load data
df = pd.read_csv("compressed_data.csv.gz")


# Combine title and text
df['content'] = df['title'] + ' ' + df['text']
df = df[['content', 'label']]


# Encode labels
df['label'] = df['label'].map({'real': 1, 'fake': 0})


# Drop missing values
df.dropna(inplace=True)


# Save cleaned data
df.to_csv("cleaned_data.csv", index=False)
```

---

2. model_training.py

python

CopyEdit

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.linear_model import LogisticRegression

import pickle


# Load cleaned data

df = pd.read_csv("cleaned_data.csv")


# TF-IDF

tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)

X = tfidf.fit_transform(df['content'])

y = df['label']


# Split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


# Train model

model = LogisticRegression()

model.fit(X_train, y_train)
```

Save model and vectorizer

pickle.dump(model, open("model.pkl", "wb"))

pickle.dump(tfidf, open("tfidf_vectorizer.pkl", "wb"))

---

3. model_evaluation.py

python

CopyEdit

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score

import matplotlib.pyplot as plt

import seaborn as sns

import pickle

import pandas as pd


# Load test data

df = pd.read_csv("cleaned_data.csv")

tfidf = pickle.load(open("tfidf_vectorizer.pkl", "rb"))

model = pickle.load(open("model.pkl", "rb"))


X = tfidf.transform(df['content'])

y = df['label']
```

y_pred = model.predict(X)

# Metrics
print(classification_report(y, y_pred))
print("ROC AUC:", roc_auc_score(y, y_pred))

# Confusion matrix
cm = confusion_matrix(y, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

---

4. app.py (Streamlit UI)

python

CopyEdit

```
import streamlit as st
import pickle

# Load model and vectorizer
```

```
model = pickle.load(open("model.pkl", "rb"))

vectorizer = pickle.load(open("tfidf_vectorizer.pkl", "rb"))


st.title("📰 Fake News Detector")

st.write("Enter any news content to check whether it's real or fake.")


news_text = st.text_area("News Text")


if st.button("Predict"):

    vectorized_input = vectorizer.transform([news_text])

    result = model.predict(vectorized_input)[0]

    if result == 1:

        st.success("✅ Prediction: REAL")

    else:

        st.error("🚫 Prediction: FAKE")
```

---

5. requirements.txt

```
nginx

CopyEdit

streamlit

scikit-learn
```

pandas

matplotlib

seaborn

14. Future scope

- *Automatically detect and flag fake news on platforms like Twitter, Facebook, and Instagram in real time.*
- *Expand the tool to detect fake news in* ***regional languages*** *like Hindi, Tamil, Malayalam, etc.*
- *Turn the tool into a* ***browser extension or mobile app*** *to help users verify articles before believing or sharing them.*
- *Provide your tool as a service to newsrooms, journalists, or fact-checking organizations.*
-

13. Team Members and Roles

NAME	RO LE	WORK
MURALIDHARAN K	Team Coordinator	Data collection
GOWTHAM P	Team member	Model selection
PUGAZHENTHI	Team member	Backend development
BHARATHIDHASAN	Team member	Frontend development
JESLIN SAJAN	Team member	Documentation