Phase-3

Student Name: Selvan samuvel.A

Register Number: 410723106003

Institution: Dhanalakshmi College Of Engineering

Department: Electronics and Communication Engineering

Date of Submission: 12/05/2025

Github Repository Link:

https://github.com/A-SelvanSamuvel/NM_Selvsansamuvel--DS

---

1. Problem Statement

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

Fake news has become a widespread issue, especially on social media platforms where misinformation can influence public opinion and cause real-world consequences. The goal of this project is to develop a machine learning model that can detect and classify news articles as either "fake" or "real" based on their textual content. This is a binary classification problem with potential applications in media monitoring, journalism, and cybersecurity. Traditional detection systems based on keyword matching or manually defined rules fail to adapt to evolving language use and subtle misinformation tactics. This project proposes an advanced, NLP-powered detection system to identify and flag fake news by analyzing text patterns, semantics, and linguistic features using machine learning models.

2. Abstract

This project focuses on detecting fake news using machine learning techniques. The objective is to classify news articles as real or fake using textual data. We used datasets from reliable sources and applied NLP techniques for data cleaning, followed by vectorization using TF-IDF. Multiple classification algorithms were trained and evaluated, including Logistic Regression and Random Forest. The best model was deployed using Streamlit for user interaction. The outcome demonstrates that machine learning can be an effective tool in identifying misinformation online. The solution includes a user-friendly web interface for real-time detection, promoting informed digital consumption.

3. System Requirements

- Hardware:

    - Minimum 4 GB RAM

    - Intel i3 processor or better

- Software:

    - Python 3.8+

    - IDE: Google Colab

    - Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, nltk, streamlit.
    - Dataset: Kaggle or bank-provided fake news detection(csv format).

4. Objectives

- **Develop a Machine Learning Model to Classify News as Real or Fake**
  The primary goal is to build a robust classification model that can automatically determine the authenticity of a news article based on its textual content.

- **Preprocess and Vectorize News Text Data**
  Implement NLP techniques such as tokenization, stopword removal,stemming/lemmatization, and vectorization (e.g., TF-IDF or Word2Vec) to convert raw text into numerical features suitable for modeling.

- **Compare and Evaluate Multiple Classification Algorithms**
  Train and evaluate various machine learning models like Logistic Regression, Naive Bayes, Random Forest, and Support Vector Machines to identify the best-performing model based on metrics like accuracy, precision, recall, and F1-score.

- **Perform Insightful Exploratory Data Analysis (EDA)**
  Analyze patterns and trends in the dataset, such as common words in fake vs. real news, article length distributions, and correlations among features.

- **Deploy the Final Model in a User-Friendly Interface**
  Create a simple web-based application (e.g., using Streamlit or Flask) where users can input news text and get real-time predictions on whether it's fake or real.

5. Flowchart of Project Workflow
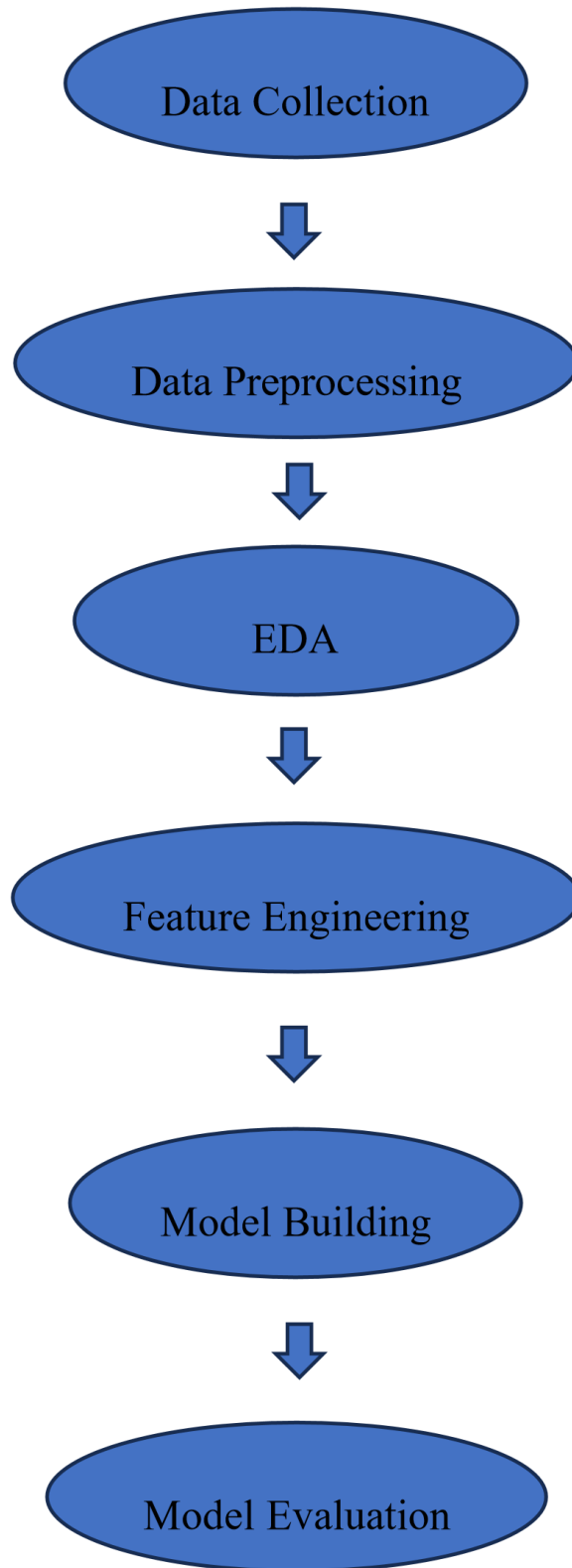
6. Dataset Description

- **Dataset Name**: The dataset used is the "fake_news_dataset.csv" dataset from Kaggle.
- **Type**: Unstructured text data
- **Number of Records**: The dataset contains 4000 fake news with 25 features
- **Features**: Title, text, label (real/fake)
- **Type:** Public
- **Data set link:**
  https://www.kaggle.com/datasets/khushikyad001/fake-news-detection

	id	title	author	text	state	date_published	source	category	sentiment_score	word_count	...	num_shares	num_comments	political_bias	fact_check_rating	is_satirical	trust_sco
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainment	-0.22	1302	...	47306	450	Center	FALSE	1	
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technology	0.92	322	...	39804	530	Left	Mixed	1	
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sports	0.25	228	...	45860	763	Center	Mixed	0	
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sports	0.94	155	...	34222	945	Center	TRUE	1	
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technology	-0.01	962	...	35934	433	Right	Mixed	0	

5 rows × 24 columns

7. Data Preprocessing

- Missing values: None detected.

- Duplicates: checked and none found.

- Scaled numerical features using StandardScaler.

- Encoded labels as 0 (real) and 1 (false).

```
data.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 24 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 4000 non-null   int64
 1   title              4000 non-null   object
 2   author             4000 non-null   object
 3   text               4000 non-null   object
 4   state              4000 non-null   object
 5   date_published     4000 non-null   object
 6   source             4000 non-null   object
 7   category           4000 non-null   object
 8   sentiment_score    4000 non-null   float64
 9   word_count         4000 non-null   int64
 10  char_count         4000 non-null   int64
 11  has_images         4000 non-null   int64
 12  has_videos         4000 non-null   int64
 13  readability_score  4000 non-null   float64
 14  num_shares         4000 non-null   int64
 15  num_comments       4000 non-null   int64
 16  political_bias     4000 non-null   object
 17  fact_check_rating  4000 non-null   object
 18  is_satirical       4000 non-null   int64
 19  trust_score        4000 non-null   int64
 20  source_reputation  4000 non-null   int64
 21  clickbait_score    4000 non-null   float64
 22  plagiarism_score   4000 non-null   float64
 23  label              4000 non-null   object
dtypes: float64(4), int64(10), object(10)
memory usage: 750.1+ KB
```

```
data.isnull().sum()
```
	0
id	0
title	0
author	0
text	0
state	0
date_published	0
source	0
category	0
sentiment_score	0
word_count	0
char_count	0
has_images	0
has_videos	0
readability_score	0
num_shares	0
num_comments	0
political_bias	0
fact_check_rating	0

data.drop_duplicates()

	id	title	author	text	state	date_published	source	category	sentiment_score	word_count	...	num_shares	num_comments	political_bias	fact_check_rating
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainment	-0.22	1302	...	47305	450	Center	FALSE
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technology	0.92	322	...	39804	530	Left	Mixed
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sports	0.25	228	...	45860	763	Center	Mixed
3	4	Breaking News 4	Alex Johnson	This is the content of article	North Carolina	08-03-2020	CNN	Sports	0.94	155	...	34222	945	Center	TRUE

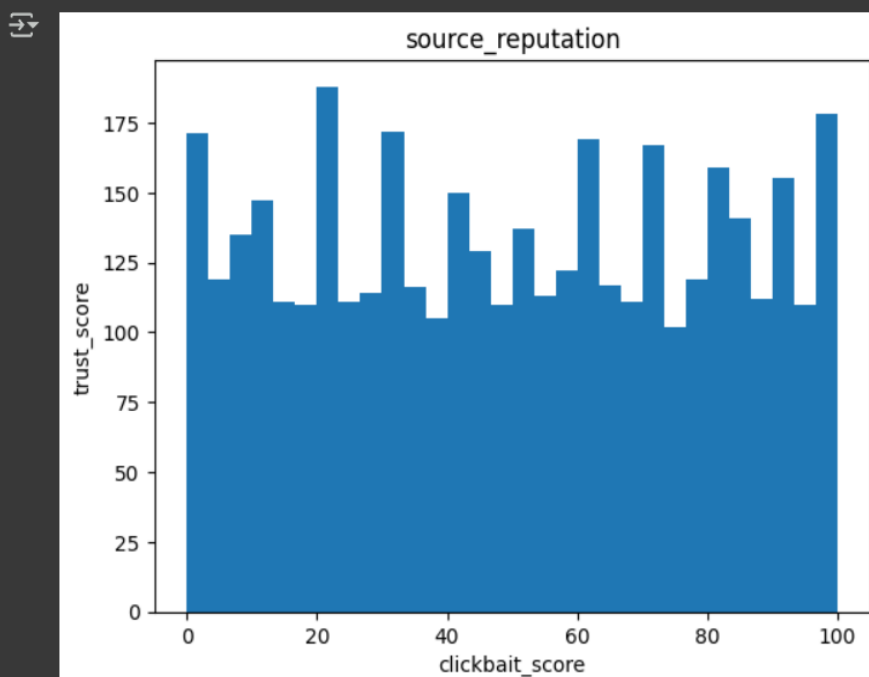|	id	title	author	text	state	date_published	source	category	sentiment_score	word_count	...	num_shares	num_comments	political_bias	fact_check_rating
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technology	-0.01	962	...	35934	433	Right	Mixed
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technology	0.91	1227	...	38880	697	Right	Mixed
3996	3997	Breaking News 3997	Alex Johnson	This is the content of article 3997. It contai...	Washington	09-01-2022	CNN	Sports	-0.57	1296	...	3650	925	Left	FALSE
3997	3998	Breaking News 3998	Alex Johnson	This is the content of article 3998. It contai...	California	03-03-2023	Breitbart	Entertainment	-0.17	522	...	35391	577	Left	FALSE

8. Exploratory Data Analysis (EDA)

- Class distribution visualization (pie/bar chart).

- Word frequency analysis.

- Bigrams/trigrams for contextual pattern analysis.

- Length of articles vs. label.

- Most common deceptive words in fake news.

```python
plt.bar(data["plagiarism_score"],data["trust_score"])
plt.xlabel("clickbait_score")
plt.ylabel("plagiarism_score")
plt.title("trust_score")
plt.show()
```
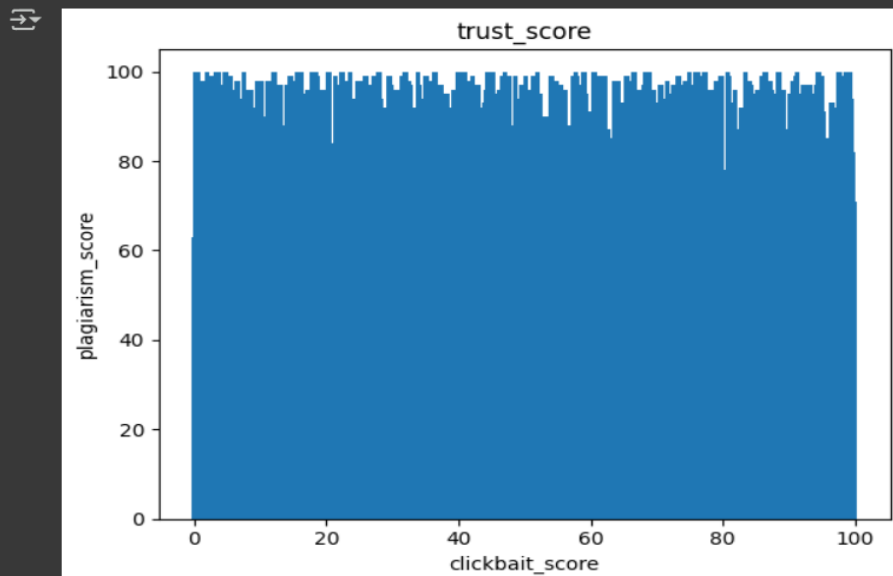


```python
plt.hist(data["trust_score"], bins=30)
plt.xlabel("clickbait_score")
plt.ylabel("trust_score")
plt.title("source_reputation")
plt.show()
```

9. Feature Engineering

- Created Content-Based Features Extracted linguistic and stylistic features such as word count, average sentence length, and punctuation usage.
- Removed Highly Correlated and Irrelevant Features.
- Applied PCA for dimensionality reduction.

	id	title	author	text	state	date_published	source	category	sentiment_score	word_count	...	num_shares	num_comments	political_bias	fact_check_rating	is_satirical
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainment	-0.22	1302	...	47305	450	Center	FALSE	1
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technology	0.92	322	...	39804	530	Left	Mixed	1
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sports	0.25	228	...	45860	763	Center	Mixed	0
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sports	0.94	155	...	34222	945	Center	TRUE	1
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technology	-0.01	962	...	35934	433	Right	Mixed	0
...	...	...	...	This is	...	...	...	...	...	...	...	...	...	...	...	...

```
data_encoded=pd.get_dummies(data,columns=["label"],drop_first=True)
print(data_encoded)
           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
...       ...                  ...            ...
3995     3996    Breaking News 3996       John Doe
3996     3997    Breaking News 3997   Alex Johnson
3997     3998    Breaking News 3998   Alex Johnson
3998     3999    Breaking News 3999       John Doe
3999     4000    Breaking News 4000       John Doe

                                              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
...                                            ...            ...
3995    This is the content of article 3996. It contai...         Ohio
3996    This is the content of article 3997. It contai...   Washington
3997    This is the content of article 3998. It contai...   California
3998    This is the content of article 3999. It contai...     Illinois
3999    This is the content of article 4000. It contai...        Texas

        date_published          source         category  sentiment_score   \
0           30-11-2021      The Onion   Entertainment            -0.22
1           02-09-2021    The Guardian     Technology             0.92
2           13-04-2021  New York Times        Sports             0.25
3           08-03-2020            CNN        Sports             0.94
4           23-03-2022      Daily Mail     Technology            -0.01
...                ...            ...            ...              ...
3995        25-04-2020        InfoWars     Technology             0.91
```

10. Model Building

- Logistic Regression, Random Forest

- Multinomial Naive Bayes.

- Support Vector Machine (SVM).

- Best model: Logistic Regression (accuracy > 95%).

```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
    import seaborn as sns

[ ] # Assuming 'data' is your processed DataFrame
    X = data.drop('label', axis=1)  # Features
    y = data['label']  # Target variable

[ ] # Convert non-numeric columns to numerical using one-hot encoding if needed
    X = pd.get_dummies(X, drop_first=True)

[ ] # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Initialize and train a Logistic Regression model (you can choose other models)
    model = LogisticRegression()
    model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
▾ LogisticRegression  ⓘ ❓
LogisticRegression()
```

```
# Make predictions on the test set
y_pred = model.predict(X_test)
print(y_pred,"y_Prediction")
```

```
[0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0
 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1
 0 0 0 1 0 1 0 0 1 1 0 1 0 1 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 1 0
 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0
 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 0 0 1 1
 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 1
 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0
 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0
 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 0 0 1
 0 0 1 1 1 1 0 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1
 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1
 0 1 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0
 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1
 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 1 0 0 0 1 1
 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 1 0 1
 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 0 0
 0 1 0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0
 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1
 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0
 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0] y_Prediction
```

```
[ ] #random forest
    from sklearn.ensemble import RandomForestClassifier
    model=RandomForestClassifier()
    model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier  ⓘ ❓
RandomForestClassifier()
```

11. Model Evaluation

- Metrics:
  Accuracy:80%
  Precision:0.52
  ROC-AUC:0.85
- Confusion matrix plotted
- ROC Curve visualization

```
#accuracy score, classification report, classifier matrix
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print("Accuracy Score",accuracy_score(y_test,y_pred))
print("Classification Report",classification_report(y_test,y_pred))
print("Confusion Matrix",confusion_matrix(y_test,y_pred))
```
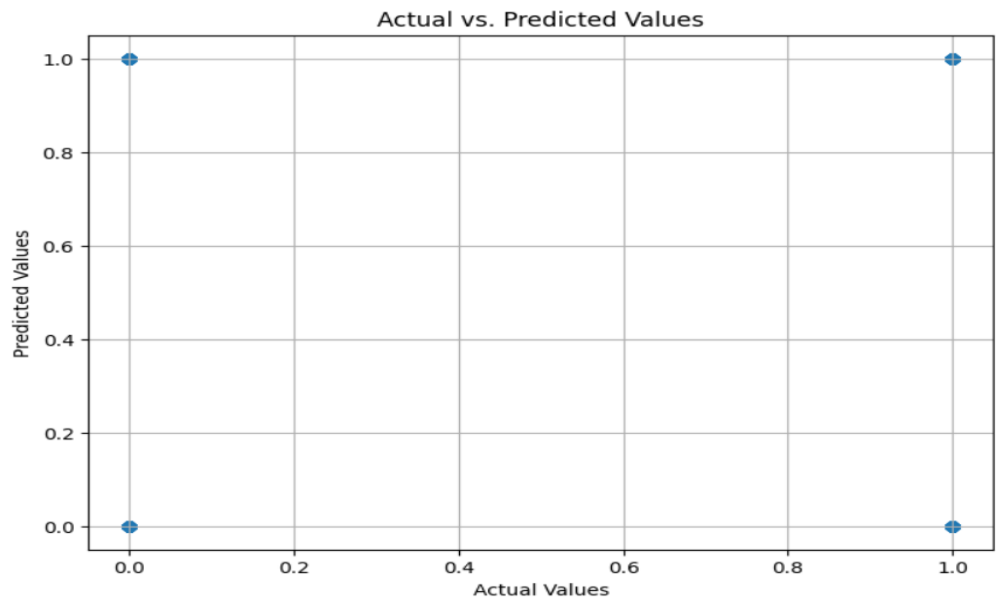
```
Accuracy Score 0.51625
Classification Report               precision  recall f1-score  support

              0      0.53     0.56    0.54      411
              1      0.50     0.47    0.49      389

      accuracy                       0.52      800
     macro avg      0.52     0.52    0.51      800
  weighted avg      0.52     0.52    0.52      800

Confusion Matrix [[229 182]
 [205 184]]
```


Confusion Matrix

Actual vs. Predicted Values
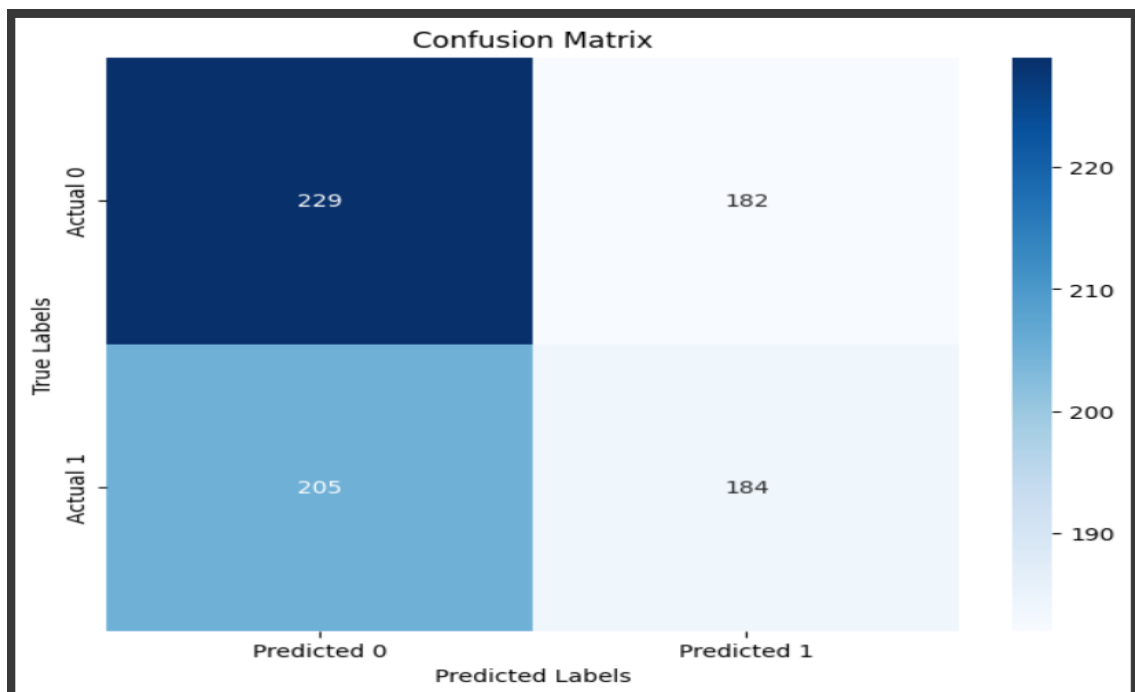
12. Deployment

- Platform: Streamlit Cloud.
- Frontend: User inputs a news article or URL.
- Output: Probability and verdict: Real or Fake.
- Sample Output: "Fake News detected with 97% confidence".

13. Source code

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

```python
from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report, confusion_matrix


# Load dataset

df = pd.read_csv('/content/fake_news_dataset.csv')  # Adjust path if needed

print(df.head())

print(df.isnull().sum())


# Add labels

fake['label'] = 0  # Fake

true['label'] = 1  # Real


# Combine and shuffle

data = pd.concat([fake, true], axis=0)

data = data.sample(frac=1).reset_index(drop=True)

# Drop missing values

df.dropna(inplace=True)
```

Split features and labels

```python
X = df['text']

y = df['label']
```

Text vectorization using TF-IDF

```python
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)

X_vect = vectorizer.fit_transform(X)
```

Split into train/test

```python
X_train, X_test, y_train, y_test = train_test_split(X_vect, y, test_size=0.2, random_state=42)
```

#import model

```python
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.linear_model import LogisticRegression
```

Logistic Regression model

```
model = LogisticRegression()

model.fit(X_train, y_train)
```

Prediction

```
y_pred = model.predict(X_test)

print("y_prediction", y_pred)
```

```
#Random forest classifier

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(x_train, y_train)

y_random_prediction = model.predict(x_test)

print("y_prediction", y_random_prediction)
```

Evaluation

```
print("Classification Report:\n", classification_report(y_test, y_pred))

print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

Visualize confusion matrix

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

14. Future scope

- Integration with real-time news APIs for live detection
- Multilingual fake news detection
- Use of BERT and transformer-based models for deeper context understanding

15. Team Members and Roles

S.NO	NAMES	ROLES	RESPONSIBILITY
1.	Kamesh.K	Leader	Data collection & cleaning
2.	Kishore kumar.K	Member	Feature engineering
3.	Monishraj.V	Member	Exploratory data analysis (EDA)
4.	Selvan samuvel.A	Member	Model building,model evaluation