Phase-3 Submission

Student Name: Priyadharshan P

Register Number: 712523205044 **Institution:** PPG Institute of Technology

Department: Information Technology **Date of Submission:** 16-05-2025

Github Repository Link:

https://github.com/dharshu0623/NM_Priyadharshan_DS

1. Problem Statement

The rise of social media and digital platforms has accelerated the spread of fake news, which can distort public opinion, incite conflict, and erode trust in media. This issue is particularly dangerous in sensitive areas like politics, health, and finance. Manual fact-checking is insufficient due to the volume and speed of content generation, highlighting the need for automated detection systems.

This project aims to build a Fake News Detection System using Natural Language Processing (NLP) and machine learning to classify news articles as real or fake based on their textual content.

- ***Problem Type**: Supervised Learning – Binary Classification*
- ***Objective**: Classify articles as FAKE (1) or REAL (0)*
- ***Approach**: Apply NLP for feature extraction and use models like Logistic Regression, SVM, and Random Forest*
- ***Relevance**: Supports media literacy, helps platforms flag deceptive content, and combats misinformation at scale*

1. Abstract

In today's information-driven society, the spread of fake news has become a serious concern, impacting public perception and decision-making. This project addresses the problem by developing a system capable of detecting fake news articles using Natural Language Processing (NLP) and machine learning techniques. The primary objective is to classify news content as either real or fake based on textual features, thereby helping to curb the spread of misinformation.

The project involves a comprehensive pipeline including data preprocessing, feature extraction using methods like TF-IDF, exploratory data analysis, and the implementation of various classification models such as Logistic Regression, Support Vector Machine (SVM), and Random Forest. Model performance is evaluated using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. Visualizations are used to interpret results and highlight key patterns in fake versus real news content. The final system is deployed using a simple web interface, enabling real-time prediction and promoting user awareness and digital integrity.

1. System Requirements

Hardware Requirements

- ***RAM**: Minimum 4 GB (8 GB recommended for faster training and visualization)*
- ***Processor**: Dual-Core CPU (Intel i3 or equivalent; i5 or higher recommended for smoother execution of large models)*
- ***Storage**: Minimum 2 GB free space (for datasets, libraries, and model files)*
- ***Internet**: Required for downloading datasets and using cloud-based notebooks (if applicable)*

Software Requirements

- ***Programming Language**: Python 3.7 or higher*
- ***IDE/Environment**:*
  - *Google Colab* (preferred for students without high-end hardware)
  - *Jupyter Notebook* (for local development)

- ***Required Python Libraries**:*
  - *pandas – for data handling*
  - *numpy – for numerical operations*
  - *nltk, spacy – for natural language processing*
  - *scikit-learn – for machine learning models and evaluation*
  - *matplotlib, seaborn, wordcloud, plotly – for visualization*

- *streamlit or flask (optional, for deployment)*

1. Objectives

The goal of this project is to ***develop a machine learning-based system*** *that classifies news articles as real or fake using Natural Language Processing (NLP). This addresses the increasing spread of misinformation by providing an automated, scalable solution.*

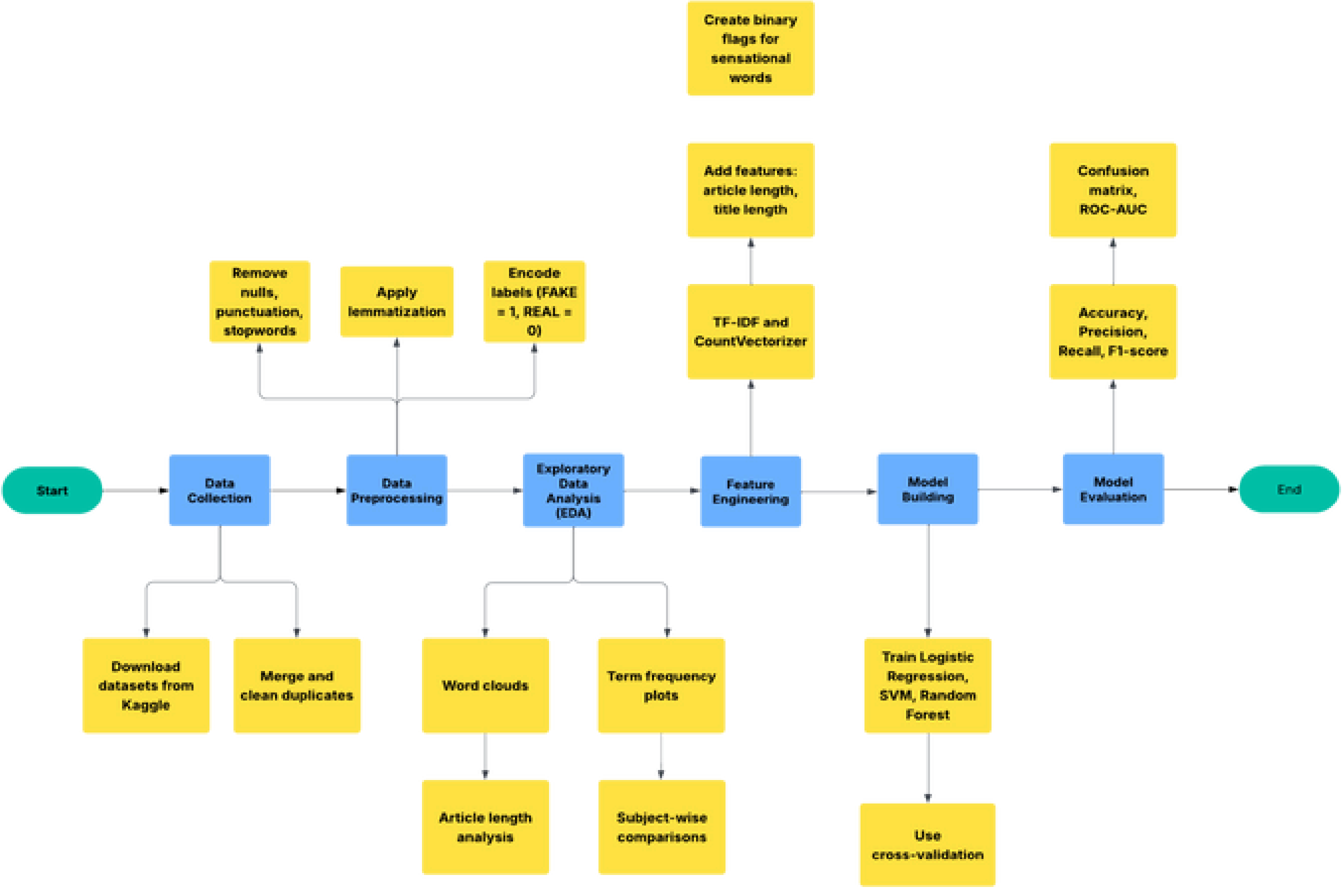
Key Objectives:

- *Build a supervised ML model to classify news content as* FAKE *or* REAL*.*
- *Preprocess and clean text data for accurate model input.*
- *Extract features using techniques like TF-IDF and n-grams.*
- *Evaluate models (Logistic Regression, SVM, Random Forest) using metrics such as Accuracy, F1-score, and ROC-AUC.*
- *Visualize insights through EDA and model interpretation tools.*
- *(Optional) Deploy a simple web interface for real-time prediction.*

Expected Impact:

- *Supports fact-checking and media verification.*
- *Helps users and platforms flag misleading content.*
- *Promotes digital literacy and trust in information.*

1. Flowchart of Project Workflow


Create binary flags for sensational words
Add features: article length, title length
Confusion matrix, ROC-AUC
Remove nulls, punctuation, stopwords
Apply lemmatization
Encode labels (FAKE = 1, REAL = 0)
TF-IDF and CountVectorizer
Accuracy, Precision, Recall, F1-score
Start
Data Collection
Data Preprocessing
Exploratory Data Analysis (EDA)
Feature Engineering
Model Building
Model Evaluation
End
Download datasets from Kaggle
Merge and clean duplicates
Word clouds
Term frequency plots
Train Logistic Regression, SVM, Random Forest
Article length analysis
Subject-wise comparisons
Use cross-validation

1. Dataset Description

Source:

*The dataset was obtained from **Kaggle**. Specific sources include:*

- *Fake and Real News Dataset by Clément Bisaillon*
- *Other supplementary datasets from HuggingFace, Zenodo, and Kaggle.*

Type:

Public – The dataset is freely accessible and available for download from Kaggle and other open platforms.

Size and Structure:

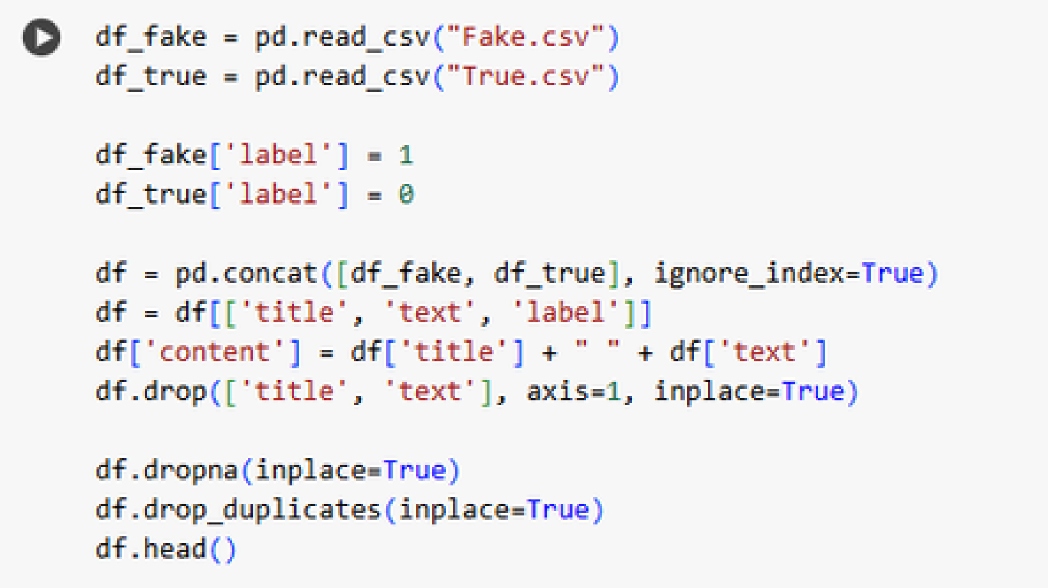
- ***Number of rows:** Approximately **44,919 news articles***

- *Fake: ~23,502*
- *Real: ~21,417*

- ***Number of columns**: Typically **4 to 5** depending on the dataset*

- ***Common features**: title, text, subject, date*

- ***Target column**: label (FAKE = 1, REAL = 0)*

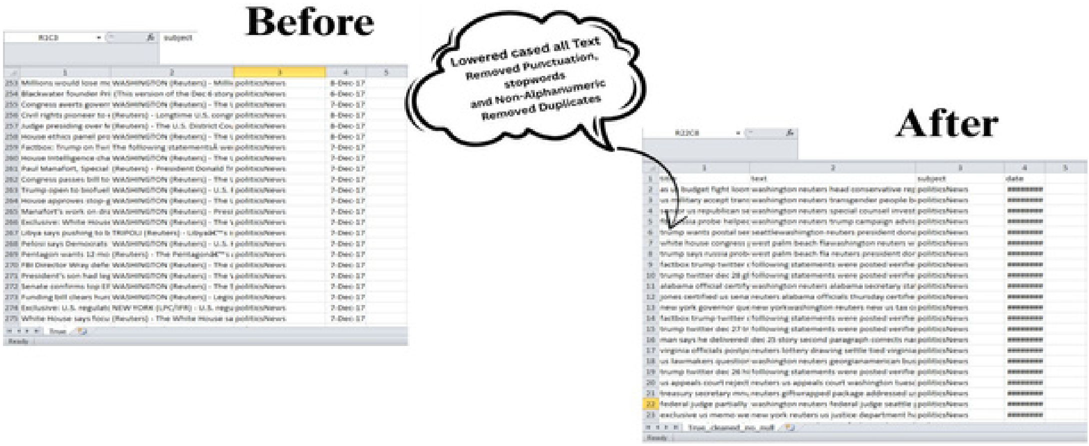
- 

```
df_fake = pd.read_csv("Fake.csv")
df_true = pd.read_csv("True.csv")

df_fake['label'] = 1
df_true['label'] = 0

df = pd.concat([df_fake, df_true], ignore_index=True)
df = df[['title', 'text', 'label']]
df['content'] = df['title'] + " " + df['text']
df.drop(['title', 'text'], axis=1, inplace=True)

df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df.head()
```

- *df.head( )*

1. Data Preprocessing

- *Removed null values, duplicates*
- *Standardized date format*
- *Text cleaning: lowercasing, punctuation/stopword removal, lemmatization*
- *Encoded labels as binary*
- *Feature vectorization using TF-IDF*


Before
Lowered cased all Text Removed Punctuation, stopwords and Non-Alphanumeric Removed Duplicates
After

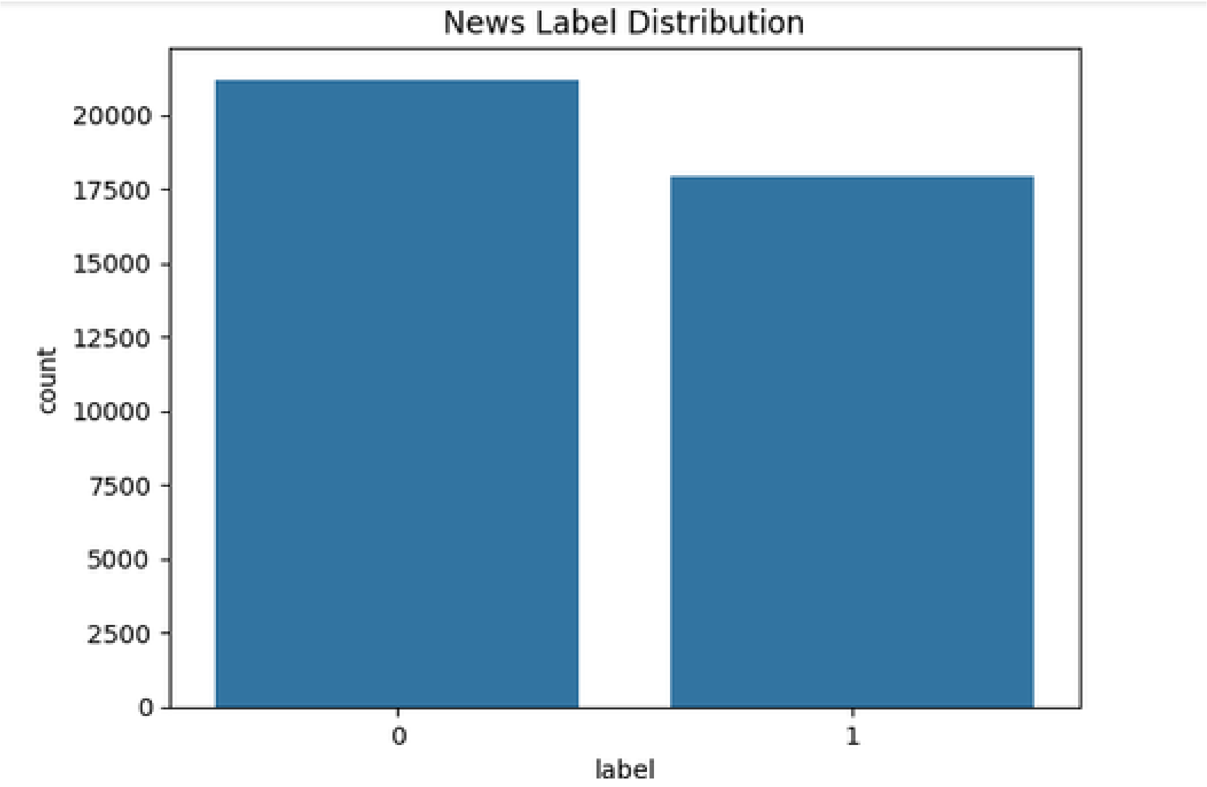
1. Exploratory Data Analysis (EDA)

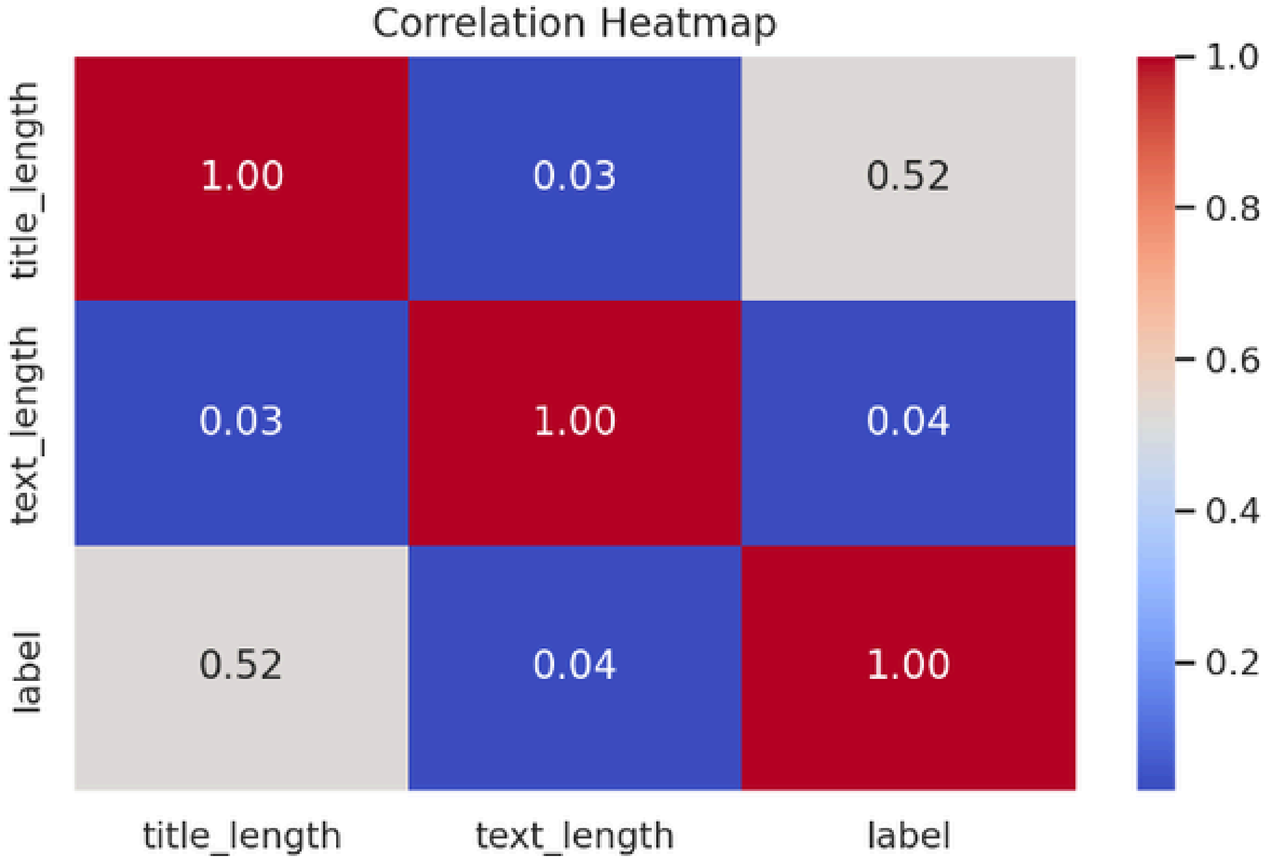
Univariate*: Word clouds, frequency plots*

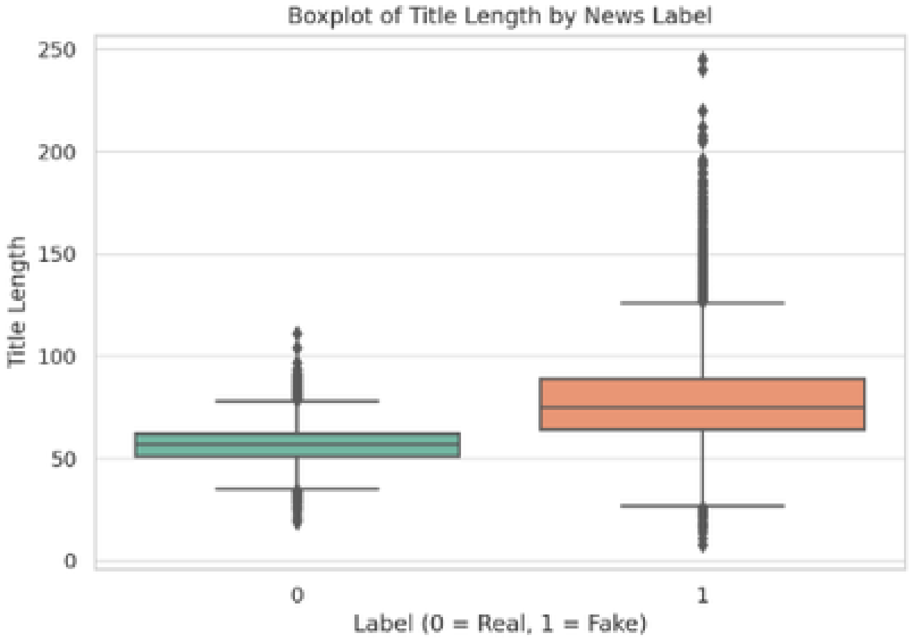
Bivariate*: Subject vs. label bar plots, article length analysis*

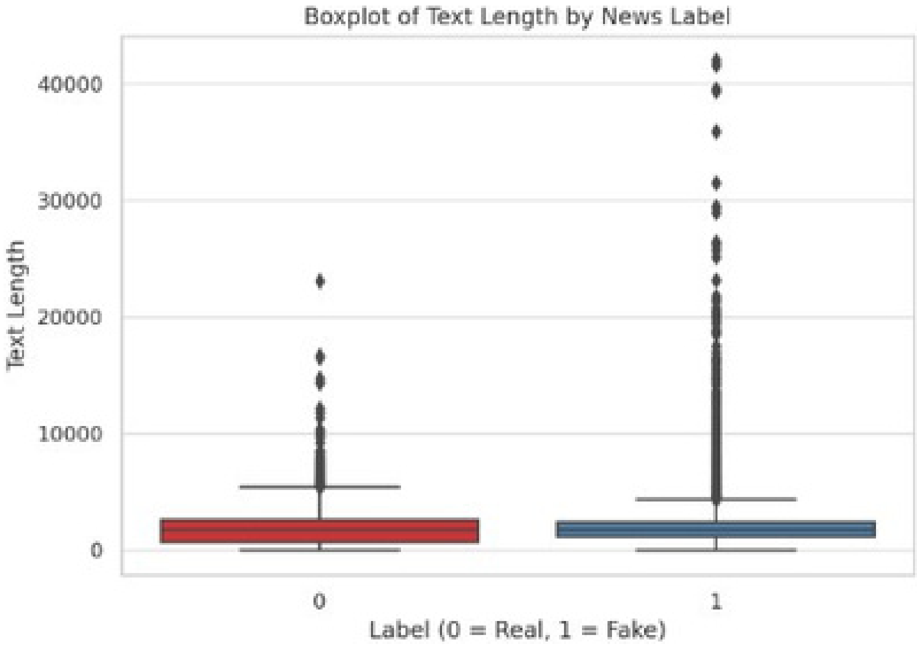
Insights*:*

- *Fake news uses exaggerated terms like "shocking"*
- *Fake news articles tend to have shorter, clickbait-style titles*




Correlation Heatmap
title_length
text_length
label
1.00
0.03
0.52
0.03
1.00
0.04
0.52
0.04
1.00
title_length
text_length
label
1.0
0.8
0.6
0.4
0.2


Boxplot of Title Length by News Label
Title Length
250
200
150
100
50
0
0
1
Label (0 = Real, 1 = Fake)



1. **Feature Engineering**
    - *Added features: article length, title length*
    - *Binary flags for emotional/sensational keywords*
    - *TF-IDF with unigrams and bigrams*
    - *Optional: LSA for dimensionality reduction*
2. **Model Building**

Models Used:

- *Logistic Regression*
- *Support Vector Machine (SVM)*
- *Random Forest Classifier*

Why:

- *SVM performed best on high-dimensional TF-IDF data.*
- *Logistic Regression is simple and interpretable.*
- *Random Forest is robust to noise.*

1. **Model Evaluation**

Metrics:

- ***Accuracy**: 99.55%*
- ***Precision (Fake)**: 99.61%*
- ***Recall (Fake)**: 99.52%*
- ***F1-Score (Fake)**: 99.56%*
- ***AUC (ROC Curve)**: 1.00*

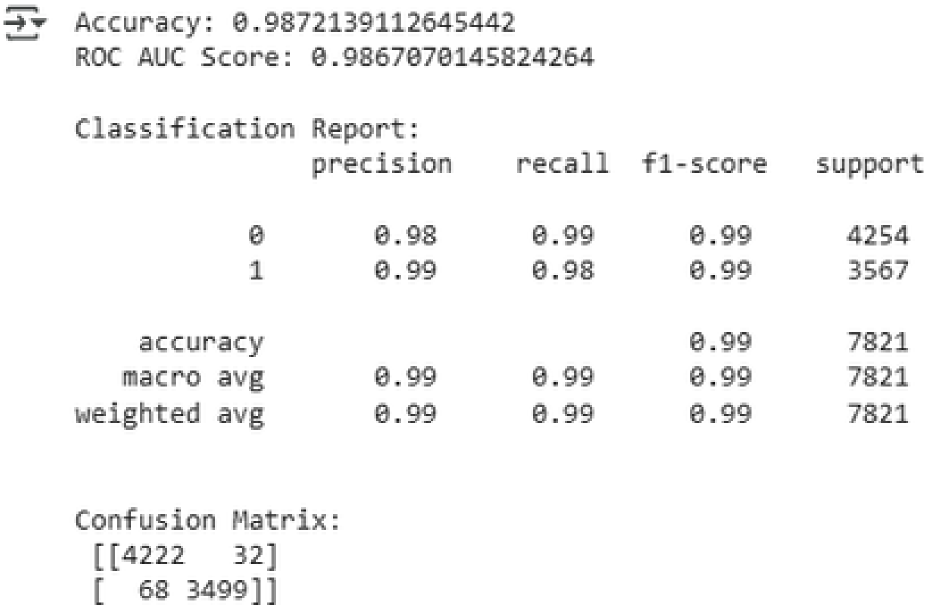
Confusion Matrix:

- *True Real: 4224*
- *True Fake: 4545*
- *False Positives: 18*
- *False Negatives: 22*

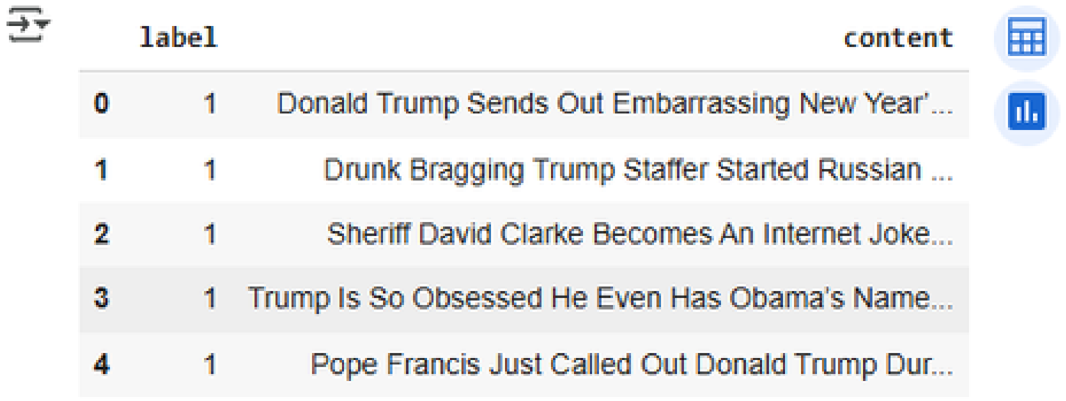
Insights:

- ***SVM with TF-IDF** delivers **exceptional performance** on this dataset.*
- ***Emotional and clickbait-style keywords** in fake news contribute significantly to classification accuracy.*
- *The **ROC curve** shows near-perfect separation between classes (AUC = 1.00), indicating excellent model confidence.*

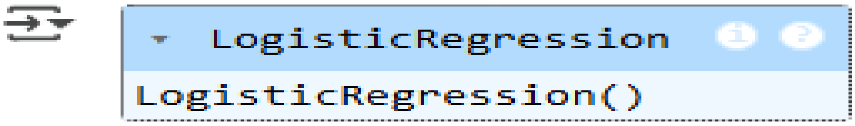
***Sample Output**:*



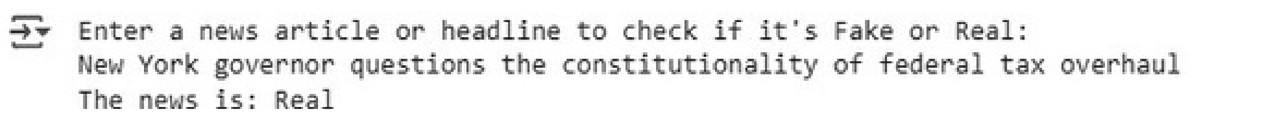
```
Accuracy: 0.9872139112645442
ROC AUC Score: 0.9867070145824264

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4254
           1       0.99      0.98      0.99      3567

    accuracy                           0.99      7821
   macro avg       0.99      0.99      0.99      7821
weighted avg       0.99      0.99      0.99      7821


Confusion Matrix:
 [[4222   32]
 [  68 3499]]
```



	label	content
0	1	Donald Trump Sends Out Embarrassing New Year'...
1	1	Drunk Bragging Trump Staffer Started Russian ...
2	1	Sheriff David Clarke Becomes An Internet Joke...
3	1	Trump Is So Obsessed He Even Has Obama's Name...
4	1	Pope Francis Just Called Out Donald Trump Dur...



```
LogisticRegression
LogisticRegression()
```



```
Enter a news article or headline to check if it's Fake or Real:
New York governor questions the constitutionality of federal tax overhaul
The news is: Real
```

1. Source code

```
df_true = pd.read_csv("True.csv")

df_fake['label'] = 1

df_true['label'] = 0

df = pd.concat([df_fake, df_true], ignore_index=True) df = df[['title', 'text',
'label']]

df['content'] = df['title'] + " " + df['text'] df.drop(['title', 'text'], axis=1,
inplace=True)
```

```
df.dropna(inplace=True) df.drop_duplicates(inplace=True) df.head()
sns.countplot(x='label', data=df)

plt.title("News Label Distribution") plt.show()



text_fake = ' '.join(df[df['label'] == 1]['content']) wordcloud =
WordCloud(width=800, height=400, background_color='black').generate(text_fake)
plt.figure(figsize=(10,5))

plt.imshow(wordcloud, interpolation='bilinear') plt.axis('off')

plt.title("Fake News WordCloud") plt.show()

X = df['content'] y = df['label']



vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7) X_tfidf =
vectorizer.fit_transform(X)



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

model = LogisticRegression() model.fit(X_train, y_train) y_pred =
model.predict(X_test)



print("Accuracy:", accuracy_score(y_test, y_pred)) print("ROC AUC Score:",
roc_auc_score(y_test, y_pred))

print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

def predict_news(text):

vector = vectorizer.transform([text]) pred = model.predict(vector)[0] return "Fake"
if pred == 1 else "Real"
```

1. Future scope

- *Integrate deep learning models like BERT or LSTM for better performance.*
- *Extend support for multiple languages.*
- *Include multimedia verification (images, videos).*
- *Add browser plugin for live detection while reading articles.*

1. Team Members and Roles

S.no	***Name***	***Role***
1	*Priyadharshan P*	*Data Cleaning*
2	*Bindhiya T.*	*Model Development*
3	*Akhilan B.*	*Feature Engineering*
4	*James Aathithyan A.*	*Exploratory Data Analysis*
5	*Anish M.*	*Documentation and Reporting*

```
!pip install nltk spacy wordcloud plotly import pandas as pdimport numpy as npimport
matplotlib.pyplot as plt import seaborn as snsfrom wordcloud import WordCloud import
plotly.express as px import nltkimport spacyfrom sklearn.model_selection import
train_test_splitfrom sklearn.feature_extraction.text import TfidfVectorizer from
sklearn.linear_model import LogisticRegressionfrom sklearn.metrics import
accuracy_score, classification_report, confusion_matrix,
roc_auc_scorenltk.download('stopwords') nltk.download('punkt')!python -m spacy
download en_core_web_sm from google.colab import filesuploaded = files.upload()df_fake
= pd.read_csv("Fake.csv")
```

```
user_input = input("Enter a news article or headline to check if it's Fake or
Real:\n")result = predict_news(user_input) print(f"The news is: {result}")
```