News Article Classification (Fake vs. Real) — Project Summary

Step 1: Dataset Overview

- Datasets Used:
  - True.csv.xlsx: ~21,000 verified real news articles
  - Fake.csv.xlsx: ~20,000 known fake news articles
- Total Articles Combined: ~41,000
- Data Structure:
  - text: Article content
  - label: 1 = Real, 0 = Fake

Step 2: Text Preprocessing

Example (Before & After):

- Original: "President signs a new policy on rural development."
- Cleaned: "presid sign new polici rural develop"

Preprocessing Steps Applied:

- Conversion to lowercase
- Removal of punctuation
- Stopword filtering using NLTK
- Stemming using PorterStemmer

Step 3: Feature Extraction (TF-IDF)

- Method Used: TF-IDF Vectorization
- Features: Top 5000 most frequent terms
- Shape of Feature Matrix: (41,000 rows, 5000 columns)

Step 4: Model Training

- Algorithm: Logistic Regression
- Data Split:
  - 80% Training
  - 20% Testing

Step 5: Model Evaluation

- Accuracy: 98.9%
- Performance Metrics:

Label	Precision	Recall	F1-Score	Support
Fake	0.99	0.99	0.99	4476
Real	0.99	0.99	0.99	4504
Avg	0.99	0.99	0.99	8980

Step 6: Sample Prediction

Input:

"The government announces a new employment policy for rural workers."

Prediction:

REAL

Step 7: Output Files

File Name	Description
news_model.pkl	Trained Logistic Regression model
vectorizer.pkl	TF-IDF vectorizer object

These can be used in Python scripts or deployed via a Streamlit app.