Fake News Detection Using Random Forest Classifier

# Overview

This project detects fake news using a Random Forest Classifier trained on textual data. It allows users to input a news headline or article and get a prediction of whether the news is real or fake via a simple web interface.

# Project Structure

RF\_FakeNews.ipynb - Jupyter Notebook for training and evaluating the model  
RF\_app.py - Streamlit app for deploying the model  
RFFN\_model.joblib - Trained Random Forest model  
RFFN\_vectorizer.joblib - TF-IDF vectorizer for text input

# 🔧 Features

- Predicts news as Real or Fake  
- Text cleaned with basic preprocessing  
- Vectorization using TF-IDF (with bigrams)  
- Model: Random Forest Classifier  
- User-friendly interface with Streamlit

# Model Training & Evaluation

## 1. Data Preprocessing

- Lowercased all text  
- Removed symbols and punctuation: ? . , ! ; : - ' " &  
- Removed stopwords using ENGLISH\_STOP\_WORDS  
- Vectorized using TfidfVectorizer with:  
 - ngram\_range=(1, 2)  
 - max\_df=0.7  
 - stop\_words='english'

## 2. Model

Algorithm: RandomForestClassifier from scikit-learn

## 3. Model Performance

|  |  |
| --- | --- |
| Metric | Score |
| Accuracy | 91.3% |
| Precision | 90.2% |
| Recall | 92.7% |
|  |  |

# Streamlit App (RF\_app.py)

How it Works:  
- Users input news text.  
- The text is cleaned using clean\_text().  
- The vectorizer transforms the input.  
- The model predicts the class (1 = Real, 0 = Fake).  
- A message is displayed to indicate whether the input is likely real or fake.  
- A confidence score is also displayed

Example Output:  
Prediction Result:  
🟢 Real News ✅ or 🔴 Fake News ❌

# How to Run

Paste this link on your browser: https://rf-fakenews.streamlit.app/