**Fake News Detection System**

## ****1. Introduction****

This project aims to build a machine learning model to detect fake news articles using text classification techniques. The system processes news text, extracts features, and classifies it as either real or fake using a Random Forest classifier.

## ****Approach****

### ****2.1 Data Preprocessing:****

1. Loaded the Fake and True news datasets.

2. Removed missing values and ensured all text data was in string format.

3. Converted text to lowercase and removed punctuation.

4. Applied stopword removal, stemming (using PorterStemmer), and lemmatization (using WordNetLemmatizer) for text normalization.

### ****2.2 Feature Extraction:****

Used **TF-IDF Vectorization** to convert textual data into numerical features.

### ****2.3 Model Training:****

* Implemented the **Random Forest Classifier** as the initial model.
* Split the dataset into **80% training and 20% testing**.
* Trained the classifier on the TF-IDF transformed text data.

### ****2.4 Model Evaluation:****

* Evaluated the trained model using **accuracy score**.
* Saved the trained model and vectorizer for future predictions.

### ****2.5 User Input Prediction:****

* Created a function to preprocess user-input news text.
* Transformed input using the saved vectorizer.
* Predicted whether the input text was fake or real using the trained model.

## ****3. Challenges Faced****

### ****1. Handling Missing Values:****

Some text fields contained **NaN values**, leading to errors during preprocessing.

**Solution**: Filled missing values with empty strings.

### ****2. Handling Encoding Errors in CSV Files:****

Encountered encoding issues while reading datasets.

**Solution**: Used **UTF-8 encoding** and handled bad lines using on\_bad\_lines='skip'.

## ****4. Model Performance & Improvements****

The trained Random Forest model achieved **good accuracy** but has room for improvement.

Future improvements:

1. **Try deep learning models** (LSTM, BERT) for better feature extraction.
2. **Improve preprocessing** by handling word contractions and named entity recognition.
3. **Use ensemble learning** for more robust classification.