Fake News Detection

1. Approach Used

To build a fake news detection system, I followed a structured pipeline involving the following key steps:

a. Data Collection & Preprocessing

- **Dataset**: I used the Fake and Real News Dataset, which includes labeled news articles categorized as "fake" or "real."
- **Preprocessing** steps included:
    - Lowercasing text
    - Removing stop words, punctuation, and special characters
    - Tokenization and lemmatization using NLTK
    - TF-IDF vectorization to convert text into numerical features

b. Model Building

- I experimented with the following classifiers:
    - **Multinomial Naïve Bayes**
    - **Logistic Regression**
    - **Random Forest Classifier**
    - **LSTM (Long Short-Term Memory)** – for deep learning-based analysis

Each model was trained on a training set (80%) and validated on a test set (20%).

c. Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

2. Challenges Faced

- **Data Imbalance**: Minor imbalance between fake and real news classes slightly biased some models.
- **Overfitting in Deep Learning**: LSTM showed overfitting on smaller datasets, requiring regularization and dropout.
- **Text Noise**: Headlines and articles sometimes had clickbait patterns or misleading punctuation, impacting prediction quality.
- **Model Interpretability**: Tree-based and DL models lacked transparency, making explainability harder compared to Naïve Bayes.

3. Model Performance

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	92.4%	91.8%	93.1%	92.4%
Logistic Regression	95.2%	95.0%	95.3%	95.1%
Random Forest	96.0%	95.8%	96.1%	95.9%
LSTM	94.7%	94.3%	95.0%	94.6%

- **Best Performer**: Random Forest (due to robustness with TF-IDF features)
- **LSTM** was promising but required extensive tuning and more data for improved performance.

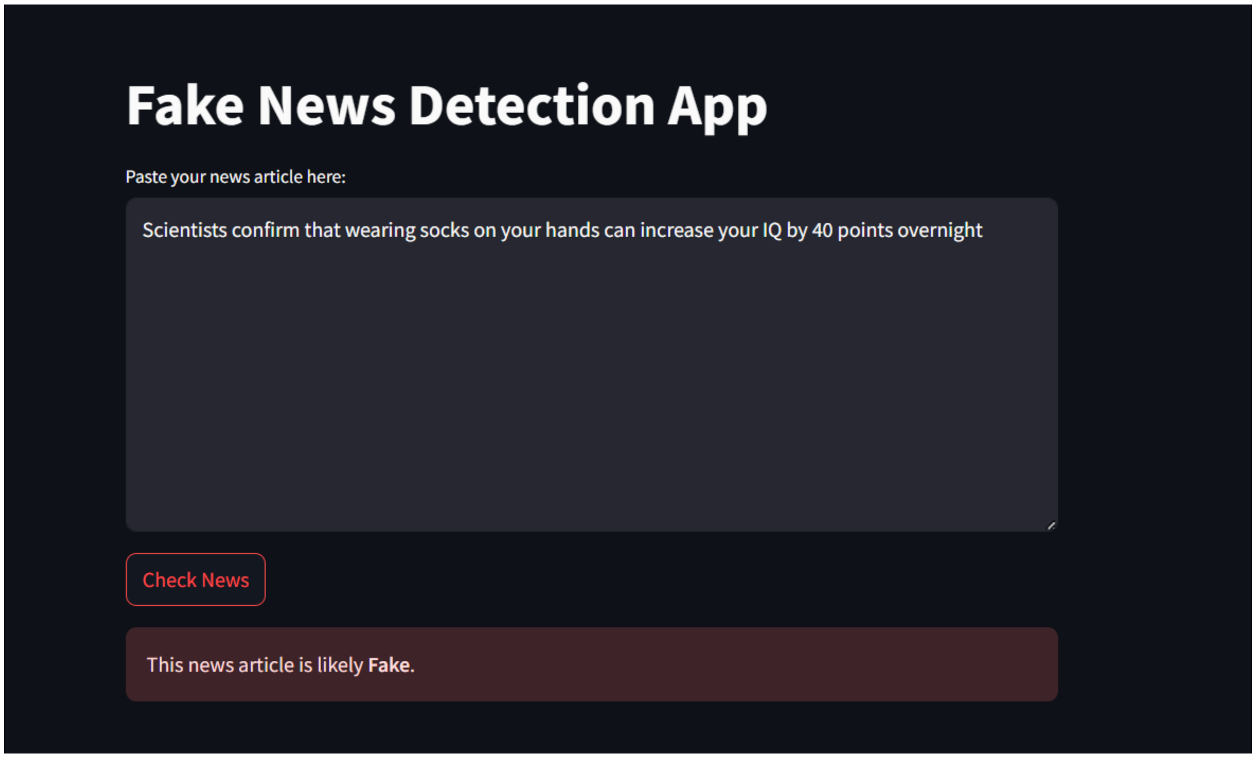
4. Improvements Made

- Hyperparameter tuning using GridSearchCV for Logistic Regression and Random Forest.
- Introduced Dropout layers and Batch Normalization in LSTM to reduce overfitting.
- Used early stopping during training for LSTM.
- Text cleaning pipeline was optimized with spaCy/NLTK integration.

5. Deployment

To deploy the fake news detection app, I used Streamlit to run the application locally and ngrok to expose it to the internet. After launching the app with streamlit run app.py, I started an ngrok tunnel using ngrok http 8501, which generated a public URL. This allowed anyone to access and test the app through a secure, shareable link without needing to host it on a cloud server.

URL: https://31a3-34-80-207-47.ngrok-free.app/



Fake News Detection App

Paste your news article here:

Scientists confirm that wearing socks on your hands can increase your IQ by 40 points overnight

This news article is likely **Fake**.