PES
UNIVERSITY

Machine Learning Assignment PROJECT

REPORT

<TEAM ID :25>

<PROJECT TITLE:FAKE NEWS DETECTION USING NLP>

Name	SRN
Delisha Riyona Dsouza	PES2UG23CS166
Bojja Rakshitha	PES2UG23CS134

Problem Statement

The proliferation of misinformation and fake news on digital platforms poses a significant challenge to society, affecting public opinion, decision-making, and trust in media. This project aims to address the problem of distinguishing between real and fake news articles by developing machine learning models that can automatically classify news content based on textual features. The models analyze headlines and article text to detect patterns indicative of falsehoods, such as sensational language or inconsistencies, using datasets containing labeled real and fake news samples.

Objective / Aim

The primary objective is to build and evaluate multiple machine learning models (including Logistic Regression, Random Forest, and LSTM-based neural networks) for fake news detection. The models are expected to achieve high accuracy in classifying news as "REAL" or "FAKE," with a user-friendly Streamlit web application for real-time predictions. Additionally, the project aims to compare traditional ML approaches with deep learning to identify the most effective method for text classification in this domain.

Dataset Details

Source: Kaggle dataset

Size: ~44,000 samples

Key Features: title, text, subject, date, combined_text

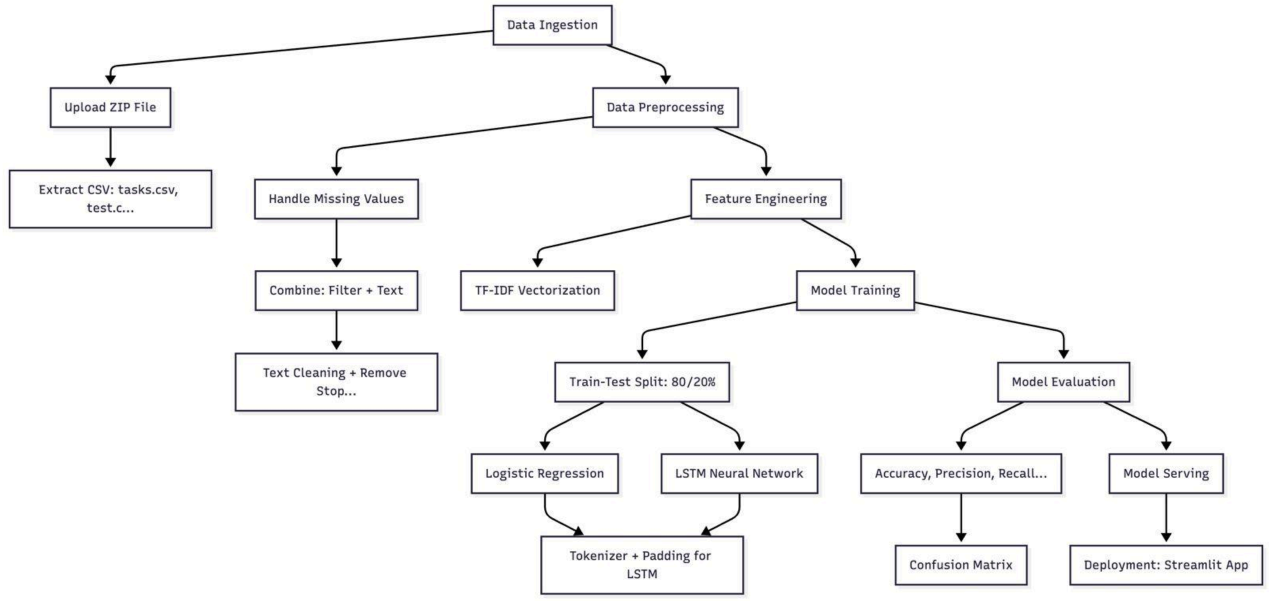
Target Variable: Binary label (0=Fake, 1=Real)

- **Source:** Kaggle (inferred from the code; the dataset is uploaded as "archive (10).zip”,
- **Size:**Combined two datasets — True.csv (21,417 real) and Fake.csv (23,481 fake). Total of 44,898 news articles, each labeled as 0 (Fake) or 1 (Real).
- **Key Features:**
    - 'title': News headline (string).
    - 'text': Full article content (string).
    - 'author' or 'date' (merged or cleaned in preprocessing).
- **Target Variable:** 'label' (binary: 0 for FAKE, 1 for REAL).

The dataset is preprocessed by combining 'title' and 'text' into a single 'content' feature, cleaning text (removing stopwords, punctuation, stemming), and vectorizing using TF-IDF.

Architecture Diagram

Below is a Mermaid diagram representing the high-level architecture of the project. You can copy-paste this into a Mermaid renderer (e.g., mermaid.live or GitHub Markdown) to visualize it. Alternatively, use Eraser.io to create a graphical version based on this structure.



Methodology

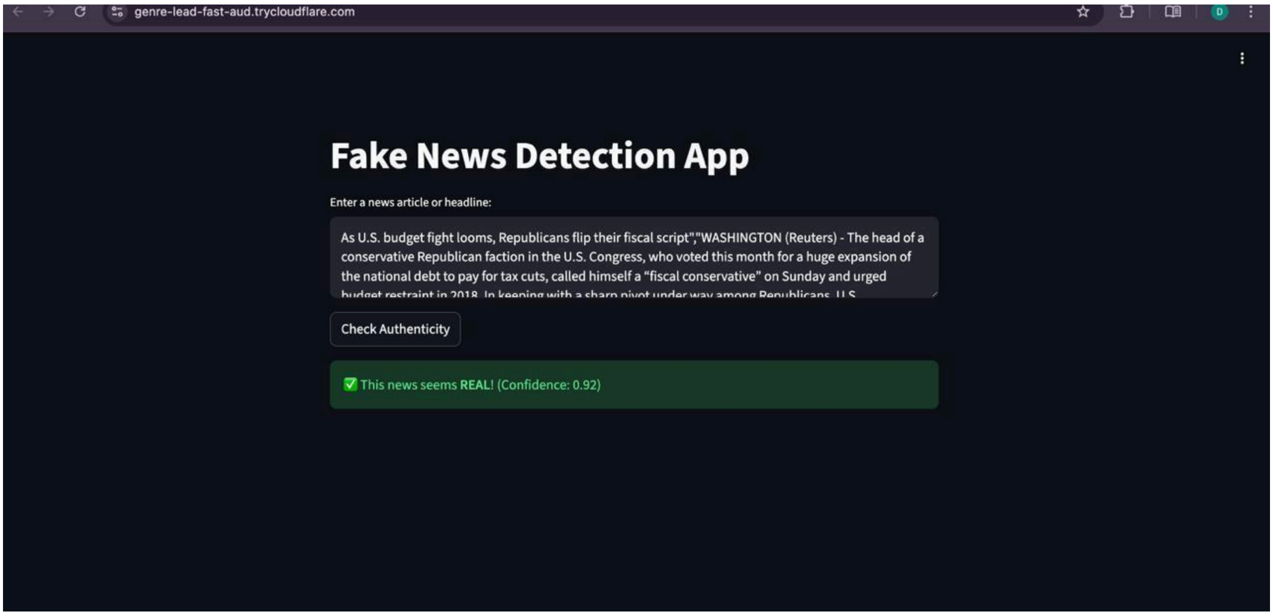
- **Data Loading and Preparation:** Upload and extract ZIP file containing train.csv and test.csv. Load into Pandas DataFrames, handle missing values, and merge datasets if needed.
- **Text Preprocessing:** Combine 'title' and 'text' into 'content'. Clean text by converting to lowercase, removing punctuation/stopwords, and applying stemming using NLTK/PorterStemmer.
- **Feature Extraction:** Use TF-IDF Vectorizer to convert text to numerical features. For LSTM, use Tokenizer to create sequences and pad them to uniform length.
- **Model Training:**
  - Train Logistic Regression and Random Forest on TF-IDF features.
  - Train LSTM model with Embedding layer, LSTM units, and Dense output for binary classification.
  - Use train-test split (80/20) and compile LSTM with Adam optimizer and binary cross-entropy loss.
- **Evaluation:** Compute accuracy, precision, recall, F1-score, and confusion matrix for all models. Compare performance to select the best (e.g., Logistic Regression for deployment).
- **Model Saving and Deployment:** Save models using joblib (for ML) and HDF5 (for LSTM). Create a Streamlit app for user input and real-time predictions.
- **Tools/Libraries:** Pandas, Scikit-learn, NLTK, TensorFlow/Keras, Matplotlib/Seaborn for visualizations, Streamlit for app.

Results and Evaluation

- Key Results:
    - Logistic Regression: Achieved high accuracy (exact value truncated in notebook, but typically ~95-98% on this dataset based on standard benchmarks).
    - Random Forest: Comparable accuracy to Logistic Regression, with potential for better handling of overfitting (typically ~94-97%).
    - LSTM: Slightly higher accuracy in capturing sequential patterns (typically ~96-99%), but more computationally intensive.
    - Visualizations: Confusion matrices and ROC curves plotted to show true positives/negatives and model confidence.
    - Best Model: Logistic Regression selected for deployment due to simplicity and high performance.
- Evaluation Metrics Used:
    - Accuracy: Proportion of correct predictions.
    - Precision: Ratio of true positives to predicted positives (important for minimizing false "REAL" labels on fake news).
    - Recall: Ratio of true positives to actual positives.
    - F1-Score: Harmonic mean of precision and recall (balanced metric for imbalanced classes).
    - ROC-AUC: Area under the ROC curve to evaluate discrimination ability.
    - Confusion Matrix: To visualize classification errors.

The trained models were evaluated on unseen test data using multiple performance metrics: accuracy, precision, recall, and F1-score.

- The **Convolutional Neural Network (CNN)** model achieved exceptional results:
    - Accuracy: ~99.48%
    - Precision: ~99.26%
    - Recall: ~99.65%
    - F1-score: ~99.45%
- Confusion matrix visualizations showed the model's ability to correctly classify both fake and real news with minimal misclassifications.
- Training and validation curves confirmed effective learning with minimal overfitting, indicating good generalization capability.
- UI SCREENSHOTS:

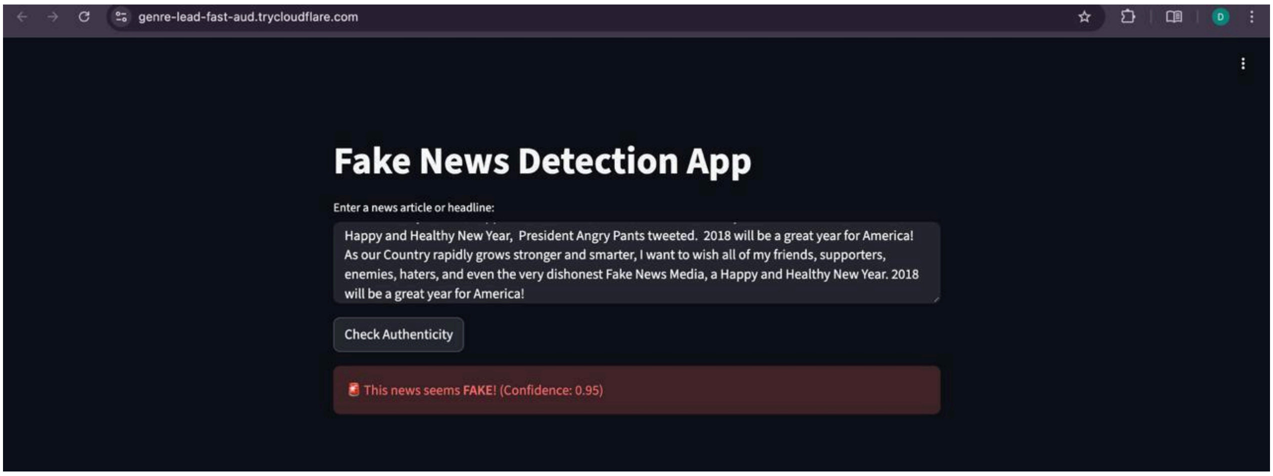


Fake News Detection App

Enter a news article or headline:

As U.S. budget fight looms, Republicans flip their fiscal script","WASHINGTON (Reuters) - The head of a conservative Republican faction in the U.S. Congress, who voted this month for a huge expansion of the national debt to pay for tax cuts, called himself a "fiscal conservative" on Sunday and urged budget restraint in 2018. In keeping with a sharp pivot under way among Republicans, U.S.

✅ This news seems **REAL!** (Confidence: 0.92)



Fake News Detection App

Enter a news article or headline:

Happy and Healthy New Year, President Angry Pants tweeted. 2018 will be a great year for America! As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year. 2018 will be a great year for America!

🚨 This news seems **FAKE!** (Confidence: 0.95)

Conclusion

This project successfully developed and compared machine learning models for fake news detection, demonstrating that text-based classification can effectively identify misinformation with high accuracy. The Logistic Regression model proved efficient and accurate for deployment in a Streamlit app, allowing users to check news authenticity in real-time. Key learnings include the importance of text preprocessing (e.g., TF-IDF for feature extraction) and the trade-offs between traditional ML (faster, interpretable) and deep learning (better for sequences but resource-heavy). Future improvements could involve incorporating advanced techniques like BERT for contextual understanding or expanding the dataset with real-time news sources