NEWS ARTICLE CLASSIFICATION (FAKE/REAL)

PROJECT REPORT (Based on Problem Statement #6)

1. ABSTRACT

Fake news has become one of the biggest challenges in today's digital world, spreading rapidly through social media and online platforms. Manually identifying misleading information is highly unreliable and time-consuming.

This project implements a News Article Classification System using Natural Language Processing (NLP) to automatically classify any news article as Fake or Real. Following the given problem statement, the system uses a labeled dataset from Kaggle, applies NLTK-based text cleaning, converts text into numerical features using TF-IDF, and trains a Logistic Regression model for classification.

A Streamlit-based user interface allows users to paste any news article, run real-time classification, view confidence scores, and see important keywords used in prediction.

The system demonstrates a complete end-to-end implementation of Machine Learning for fake-news detection exactly as described in the project requirement.

2. TOOLS USED

This project was developed using Python 3 within the Visual Studio Code (VS Code) environment, which provided an efficient editor with an integrated terminal for running all scripts. Several libraries were used to build the system: Pandas and NumPy for data handling, NLTK for text cleaning and stopword removal, and Scikit-Learn for TF-IDF vectorization, Logistic Regression modeling, and evaluation metrics. Pickle was used to save the trained model and vectorizer, while Streamlit enabled the creation of an interactive web interface for real-time classification. The PIL library was used to manage images such as the application logo. The dataset used was the Fake and Real News Dataset from Kaggle, created by George McIntire, which includes two labeled files: Fake.csv and True.csv, providing the training data for the model.

3. STEPS INVOLVED IN BUILDING THE PROJECT

The project began with the collection of the Fake and Real News Dataset from Kaggle, which includes two labeled files Fake.csv containing fake news articles and True.csv containing real news articles. These were placed in the news_classifier folder and combined after assigning labels (0 for fake and 1 for real). The next stage involved text preprocessing using NLTK in

data_clean.py, where the news content was converted to lowercase, punctuation was removed, English stopwords were filtered out, and only the essential text and label columns were retained. The cleaned data was saved as cleaned_news.csv.

Following preprocessing, the text was transformed into numerical features using TF-IDF Vectorization with a maximum of 5000 features, allowing important words to be weighted effectively for classification. In the model training phase (train_model.py), an 80–20 train-test split was applied, and a Logistic Regression model (max_iter=1000) was trained on the TF-IDF vectors. Model evaluation was performed using accuracy, precision, recall, and F1-score metrics through the classification_report, ensuring compliance with the problem requirements.

Finally, a user-friendly web interface was developed using Streamlit in app.py, enabling users to paste any news article, analyze it, and receive outputs such as Real/Fake prediction, confidence percentage, and important keywords. Additional features such as example articles and downloadable results were included, fulfilling the deliverable of a functional live demo.

4. CONCLUSION

This project successfully meets all the objectives outlined in Problem Statement #6: News Article Classification (Fake/Real).

By applying NLP techniques such as NLTK-based preprocessing and TF-IDF vectorization, followed by Logistic Regression classification, the system achieves reliable accuracy in detecting fake news.

The Streamlit web application further enhances user experience by providing real-time analysis, confidence scoring, and keyword explanations.

Overall, the project provides a complete end-to-end solution for fake news detection and demonstrates the practical application of Machine Learning and NLP in addressing misinformation challenges.