ELEVATE LABS

PROJECT

By:

Vandana D

**TABLE OF CONTENTS**

* Introduction
* Abstract
* Tools Used
* Steps Involved
* Results
* Conclusion
* Deliverables

**📰 PROJECT REPORT**

**Title:**

**News Article Classification using Machine Learning**

**Introduction:**

In the current digital landscape, the spread of fake news is a critical problem. With the rapid distribution of unverified information online, distinguishing between legitimate and fake news is crucial. This project implements a machine learning-based approach to classify news articles as either **real** or **fake** using natural language processing (NLP) techniques.

**Abstract:**

This project focuses on building a text classification system that identifies whether a news article is real or fake. Using labeled datasets from Kaggle, we combined real and fake news samples and processed the text data using TF-IDF vectorization. A Logistic Regression model was trained on the vectorized text. The system is capable of predicting the authenticity of news input from users via a terminal interface (no web UI). The model and vectorizer were saved using **joblib**, enabling reuse without retraining.

**Tools Used:**

|  |  |
| --- | --- |
| **Category** | **Tools/Techniques** |
| Programming | Python |
| Libraries | Pandas, Scikit-learn, Joblib |
| IDE | Jupyter Notebook |
| ML Model | Logistic Regression |
| NLP | TF-IDF Vectorizer |
| Dataset | Fake and Real News Dataset (Kaggle) |

**Steps Involved:**

1. **Data Collection:**
   * Downloaded Fake.csv and True.csv from Kaggle.
   * Each file contained news articles labeled as FAKE or REAL.
2. **Data Preprocessing:**
   * Combined both datasets and added a binary label column.
   * Merged title and body into a single content field for classification.
   * Converted text labels into numeric values (REAL = 1, FAKE = 0).
3. **Vectorization:**
   * Used TfidfVectorizer to convert raw text into numerical features.
   * Applied stop\_words='english' and max\_df=0.7 for filtering.
4. **Model Training:**
   * Trained a LogisticRegression model on the training data (80/20 split).
   * Evaluated model performance using accuracy, confusion matrix, and classification report**.**
5. **Model Saving:**
   * Saved the trained model and vectorizer as .pkl files for future use.
6. **Prediction:**
   * Used a simple Python input prompt to let the user enter news content.
   * Model predicts and prints "REAL" or "FAKE" directly in the console or notebook.

**Results:**

* **Accuracy:** ~95% (may vary slightly depending on data split)
* **Evaluation Metrics:**
  + Precision, Recall, and F1-score were all high, especially for the FAKE class.
* **Confusion Matrix:** Clearly separated predictions for real vs fake articles.

**Conclusion:**

This project demonstrated the use of machine learning for fake news detection. A Logistic Regression model combined with TF-IDF vectorization provided high accuracy and fast performance. The project focused on a CLI/terminal-based solution without involving web deployment tools like Streamlit, making it lightweight and easy to execute. This solution can be extended by integrating advanced models or building a user-friendly UI in future versions.

**Deliverables:**

* notebook.ipynb (Model training + prediction)
* news\_model.pkl (Saved model)
* tfidf\_vectorizer.pkl (Saved vectorizer)
* Fake.csv and True.csv (Original datasets)