**MDM MINI PROJECT**

**Fake News Detection Using Machine Learning**

**Submitted by:** Jessica Nayak  
 Registration no. 231091027

Rupali Naresh Thakre  
 Registration no. 231061063

**Technology Stack:** Python, Logistic Regression, Flask API, HTML  
**Institution:** VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE, MUMBAI

**Table of Contents**

 Introduction

 Aim

 Technologies Used

 Dataset Description

 Data Preprocessing

 Model Selection and Training

 Evaluation Metrics

 Deployment

 Future Scope

 Conclusion

 References

**Introduction**

With the exponential rise of digital news platforms and social media, the spread of fake news has become a critical issue globally. Misinformation has the power to mislead the public, influence opinions, and disrupt social harmony. This project aims to address this challenge by building a fake news detection system using machine learning techniques.

**4. Aim**

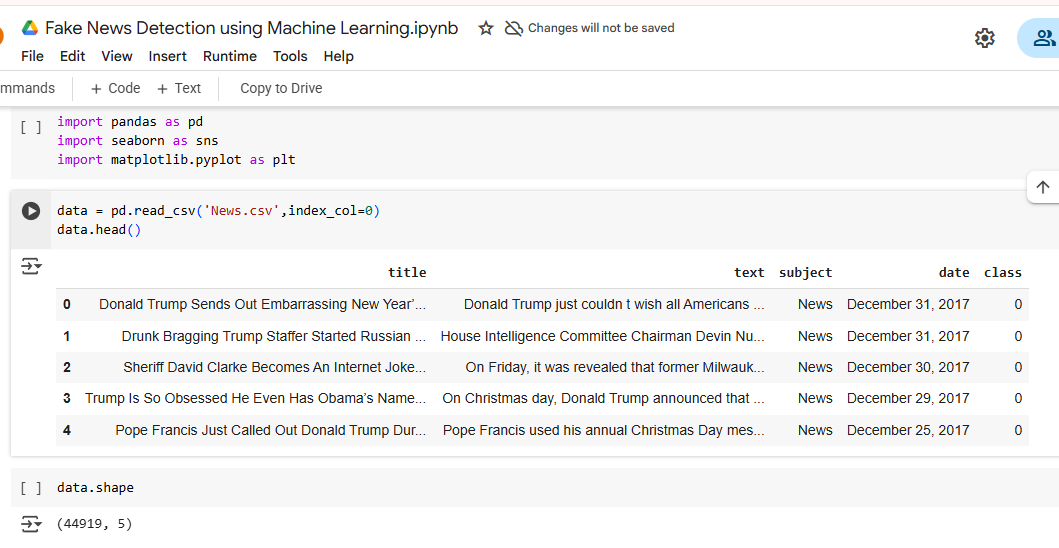
The primary goal of this project is to build a machine learning model that can effectively classify news articles as real or fake based on their textual content. The model is also deployed as a web application to enable real-time user interaction.

**5. Technologies Used**

* **Python:** Core programming language used for data processing and model building.
* **Pandas & NumPy:** For data manipulation and numerical operations.
* **Scikit-learn:** To implement logistic regression and evaluate model performance.
* **Matplotlib & Seaborn:** For data visualization and plotting graphs.
* **Flask:** To build the backend API for deploying the ML model.
* **HTML/CSS:** To create a user-friendly frontend interface for interacting with the model.

### 6. ****Dataset Description****

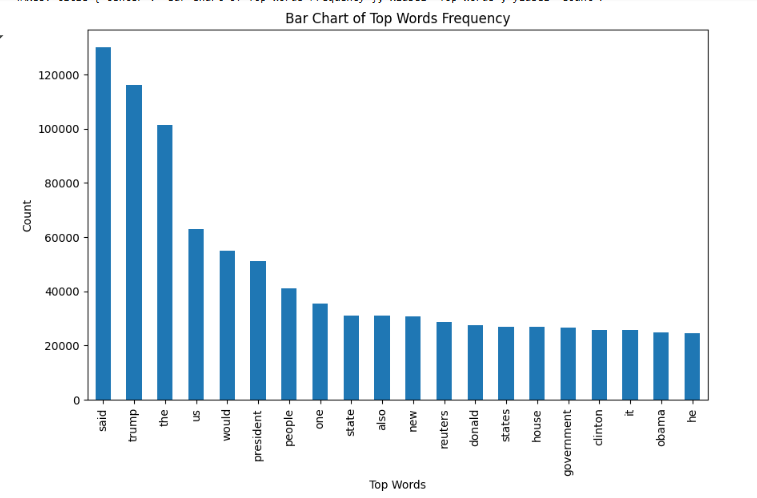
We used a dataset of labeled news articles with two primary labels: REAL and FAKE. The dataset includes the news title and text, which serve as the main features for classification.



### 7. ****Data Preprocessing****

To prepare the data for machine learning, the following preprocessing steps were applied:

* **Text Cleaning:** Removing punctuation, special characters, and converting text to lowercase.
* **Stopword Removal:** Filtering out common but insignificant words (e.g., "the", "is").
* **Tokenization:** Breaking text into individual words or tokens.
* **TF-IDF Vectorization:** Converting text into numerical format using Term Frequency-Inverse Document Frequency (TF-IDF).



### 8. ****Model Selection and Training****

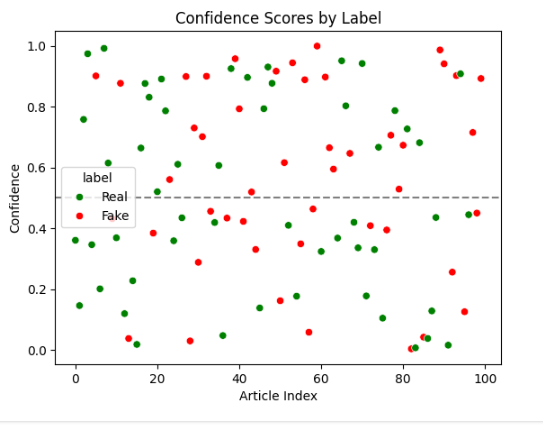
We chose **Logistic Regression** for its simplicity, speed, and effectiveness for binary classification tasks like ours.

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

model.fit(X\_train, y\_train)

The model was trained on 80% of the dataset and tested on the remaining 20% to evaluate performance.

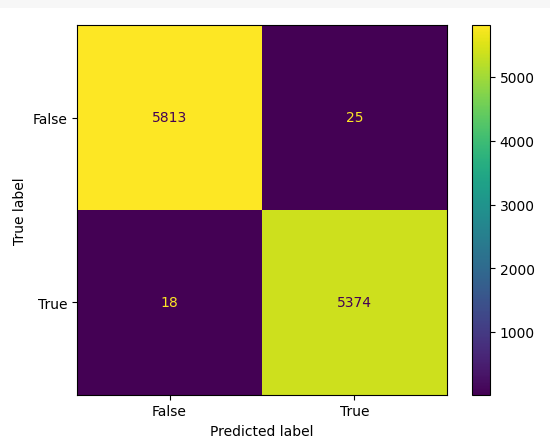


### 9. ****Evaluation Metrics****

* **Accuracy Score:** 95.3%
* **Confusion Matrix:**

This shows:

* True Positives (TP): 535
* True Negatives (TN): 587
* False Positives (FP): 9
* False Negatives (FN): 25

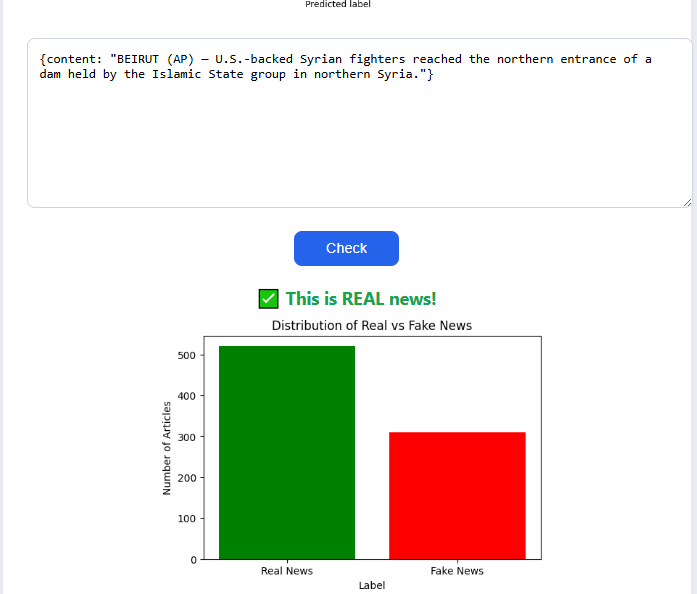


* **Classification Report:**
* **Precision:** 98%
* **Recall:** 95%
* **F1 Score:** 96.4%
* **Visualizations:**
  + Bar graph showing distribution of fake vs real news.
  + Scatter plot showing TF-IDF score vs label.
  + Confusion matrix heatmap using Seaborn.

### 10. ****Deployment****

To make the model accessible, we built a simple web application using Flask and HTML.

* **Backend (Flask):** A RESTful API that receives user input and returns prediction.
* **Frontend (HTML):** A form for users to paste news text and submit for prediction.
* **How it works:**
  1. User inputs news article.
  2. The article is sent via POST request to the Flask API.
  3. The model returns prediction (Real or Fake).
  4. Result is displayed on the frontend.



### 11. ****Future Scope****

There is significant potential for improving and extending this fake news detection system:

* **Multilingual Support:** Extend the model to detect fake news in multiple languages.
* **Real-Time News Scraping:** Integrate news scrapers to automatically classify fresh articles from websites or Twitter.
* **Mobile App Deployment:** Build Android/iOS versions for broader access.
* **Combining Sources:** Use metadata like article source, publisher credibility, or date to enhance predictions.

### 12. ****Conclusion****

This project demonstrates the successful application of machine learning for fake news detection. By using logistic regression and basic NLP techniques, we were able to achieve high accuracy and deploy the model via a user-friendly web interface. The project also opens up several future enhancements to make the system more robust and practical for large-scale use.

### 13. ****References****

* **Fake News Dataset on Kaggle:**  
  <https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset>
* **Fake News Detection Using Machine Learning: An Exhaustive Review**  
  🔗 <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4439846>
* **Detection of Fake News Using Machine Learning and Natural Language Processing**  
  🔗 <https://www.jait.us/issues/JAIT-V13N6-652.pdf>