**Project name : Exposing the Truth with Advanced Fake News Detection Powered By Natural Language Processing**

**Phase-2**

**Student Name:** Akila S

**Register Number:** 623523106003

**Institution:** AVS college of technology

**Department:** BE ECE

**Date of Submission:** 14.05.2025

**Github Repository Link**[:https://github.com/akilasakthivel42/fake\_news\_detection.git](https://github.com/akilasakthivel42/fake_news_detection.git)

**1.Problem Statement**

In today’s digital age, the rapid spread of fake news through online platforms has become a significant threat to public trust, societal stability, and informed decision-making. Traditional fact-checking methods are manual, time-consuming, and not scalable to the vast amount of content generated daily.

This project aims to solve a binary classification problem, where the goal is to build a model that can automatically classify a news article as either “Fake” or “Real” based on its textual content using Natural Language Processing (NLP) techniques.

**2.Objectives of the Project**

* Detect whether a news article is real or fake using text data.
* Use Natural Language Processing (NLP) to process and understand news content.
* Build and test machine learning models for accurate classification.
* Help reduce the spread of misinformation by identifying fake news.
* Develop a simple, user-friendly system to verify news authenticity.

**3.** **Flowchart of the Project Workflow**

Data Collection

Data Preprocessing

Exploratory Data Analysis (EDA)

Model Building

Model Comparison & Selection

Visualization & Interpretation

Deployment

end

**4. Data Description**

* **Dataset Name & Source:** Fake and Real News Dataset — downloaded from [Kaggle]**.**
* **Type of Data:** Unstructured text data (news articles).
* **Number of Records & Features:** Total records: Approximately 44,000 articles **,**Key features: title, text, subject, date, and label
* **Dataset Nature:** Static dataset — downloaded once, not updated in real-time.
* **Target Variable:** label (Binary classification: "FAKE" or "REAL")

**5. Data Preprocessing**

* Removed missing values from important columns like text and label.
* Deleted duplicate news articles to avoid repetition.
* Cleaned text by removing punctuation, stopwords, and special characters.
* Converted all text to lowercase for consistency.
* Used TF-IDF to turn text into numerical format for machine learning.
* Encoded labels (FAKE = 0, REAL = 1) for model training.
* Split the data into training and testing sets.

**6.** **Exploratory Data Analysis (EDA)**

* **Univariate Analysis**
  + Used countplot to check how many news articles are fake vs. real.
  + Used histograms to see distribution of article lengths.
  + Boxplots showed that fake news articles are usually shorter.
* **Bivariate / Multivariate Analysis**
  + Compared text length with news labels (fake or real).
  + Analyzed common words using word clouds for fake vs. real news.
  + Checked topics/subjects and how they relate to fake or real articles.
* Used bar plots to show subject-wise distribution of news labels.
* **Insights Summary**
  + Fake news tends to be shorter and more emotional in tone.
  + Certain subjects (like politics) have more fake articles.
  + Word usage and article length are important for prediction.

**7. Feature Engineering**

* + Created a new feature for text length (number of words in the article).
  + Added title length as a feature, based on the idea that short or clickbait titles may relate to fake news.
  + If available, extracted date parts like year or month (optional).
  + Converted news content into TF-IDF vectors for machine learning models.
  + Removed unnecessary or duplicate columns (e.g., IDs or unused metadata).
  + Did not apply dimensionality reduction like PCA to keep features interpretable (optional step).

**8. Model Building**

* Used two models: Logistic Regression and Random Forest.
* Chose them because they work well for text classification tasks.
* Split data into 80% training and 20% testing, with stratification to keep label balance.
* Trained models using TF-IDF vectorized text.
* Evaluated using accuracy, precision, recall, and F1-score.
* Random Forest gave slightly better performance than Logistic Regression.

**9.** **Visualization of Results & Model Insights**

* + Confusion Matrix: Showed how well the model predicts fake vs. real news (true/false positives/negatives).
  + ROC Curve: Visualized model's ability to distinguish between classes; higher AUC = better performance.
  + Feature Importance Plot: Showed which words/features most influenced the model (e.g., sensational words, short text).
  + Bar Chart: Compared accuracy, precision, recall, and F1-score across models.
  + Conclusion: Random Forest had better performance; both models identified key fake news traits.

**10. Tools and Technologies Used**

* **Programming Language:** Python
* **IDE/Notebook**: Google Colab and Jupyter Notebook
* **Libraries Used**:
  + pandas, numpy – for data handling
  + matplotlib, seaborn – for visualization
  + scikit-learn – for model building and evaluation
  + nltk – for text preprocessing
  + TfidfVectorizer – for converting text to numbers
* **Visualization Tools**: Matplotlib, Seaborn (for charts and graphs)

**11.** **Team Members and Contributions**

1. **S.Harini** - Data Cleaning & Preprocessing
2. **S.Akila** - Exploratory Data Analysis (EDA) &Feature Engineering
3. **P.Abinaya** - Model Development & Visualization
4. **S.Atchayakeerthi** - Model Evaluation & Visualization
5. **S.Janani** - Documentation & Reporting