Phase-2

Student Name: Selvan samuvel.A

Register Number: 410723106003

Institution: Dhanalakshmi College Of Engineering

Department: Electronics and Communication Engineering

Date of Submission: 06-05-2025

Github Repository Link:

https://github.com/A-SelvanSamuvel/NM_Selvsansamuvel--DS

1.Problem Statement

Exposing the truth with advanced fake news detection powered by natural language processing.

In the age of digital media, the rapid spread of misinformation and fake news poses a significant threat to public opinion, safety, and trust. Fake news, often shared through social media platforms and online news outlets, can influence elections, incite violence, and cause widespread panic. The objective of this project is to develop a machine learning-based system capable of automatically detecting and classifying news articles as either fake or real. This system will analyze the textual content of news using natural language processing (NLP) techniques and predictive models to help mitigate the spread of false information.

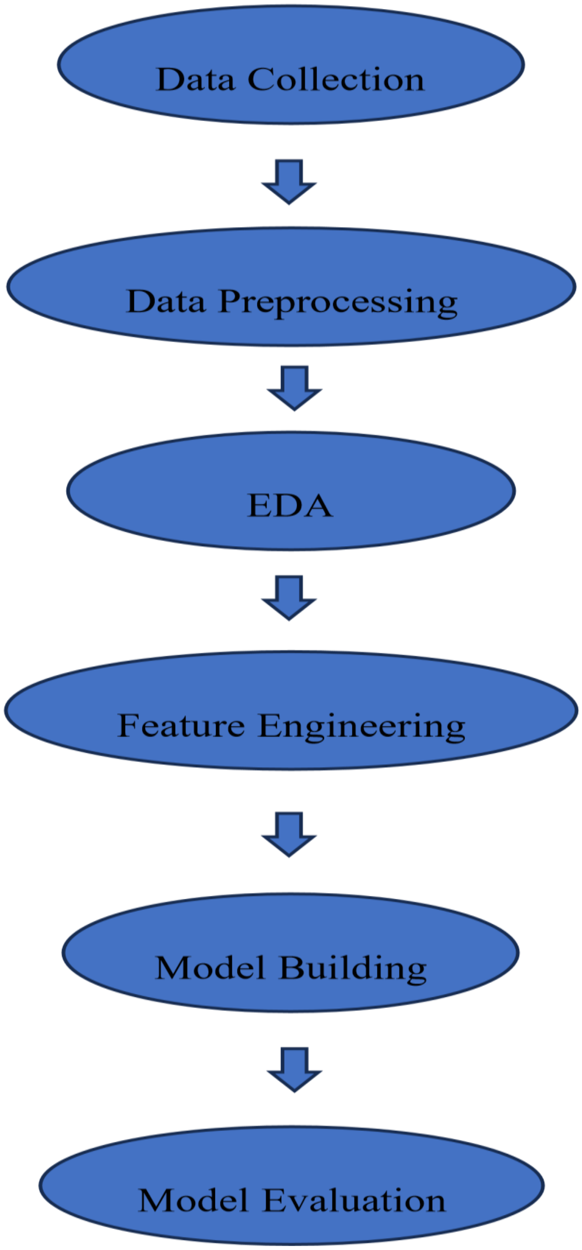
2. Project Objectives

1. To collect and preprocess a dataset of real and fake news articles from reliable sources to ensure balanced and representative input for training and evaluation.

2. To perform exploratory data analysis (EDA) to understand key patterns, trends, and linguistic features that distinguish fake news from real news.

3. To implement natural language processing (NLP) techniques such as tokenization, stemming/lemmatization, and vectorization (e.g., TF-IDF, Word2Vec) for effective feature extraction.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name**: The dataset used is the “fake_news_dataset.csv” dataset from Kaggle.
- **Type**: Unstructured text data+
- **Number of Records**: The dataset contains 4000 fake news with 25 features.
- **Features**: Title, text, label (real/fake)
- **Target Variable**: label (1 = fake, 0 = real)
- **Static Dataset**
- **Data set link:** https://www.kaggle.com/datasets/khushikyad001/fake-news-detection

5.Data Preprocessing

- Missing Values: No missing values were found in the dataset.

```
data.isnull().sum()

data
```

- Duplicate Records: Duplicate rows were checked and removed if present.

```
data.drop_duplicates(inplace=True)

data
```

- Outliers: Detected using boxplots; outliers in plagiarism_score were handled using log transformation.
- Data Types: All features are numeric. No conversion needed.
- Encoding Categorical Variables: Not required as all features are already numerical.

- Normalization: plagiarism_score and clicbait_score wear scaled using standardscaler to bring on the same scale.

```
from sklearn.preprocessing import
StandardScaler
scaler=StandardScaler()
data_scaled=data.copy()

data_scaled[["clickbait_score","plagiarism_score"]]=scaler.fit_transfor
m(data[["cli ckbait_score","plagiarism_score"]]) data_scaled
```

6. Exploratory Data Analysis (EDA)

- **Univariate**:
  - Word clouds for real vs fake news.
  - Histogram of article lengths.
- **Bivariate**:
  - Countplots showing class distribution.
  - Top words by class using TF-IDF weights.
- **Insights**:
  - Fake news tends to have more emotionally charged language.
  - Lengths and vocabulary usage differ slightly between real and fake news.

7. Feature Engineering

- New Features:
  - Created features such as:
  - Text length, title length, and punctuation counts to capture basic stylistic cues.

- Sentiment scores using NLP tools to detect emotional tone.
- Feature Reduction:
    - No dimensionality reduction applid due to prior PCA
- Domain Knowledge Integration:
    - Used pre-trained language models (e.g., BERT embeddings) to capture contextual word meanings.

8. Model Building

- Models Selected:
    - **Logistic Regression:** Random Forest Classifier: Utilized to capture non-linear relationships and feature interactions.
    - **Random Forest Classifier:** Utilized to capture non-linear relationships and feature interactions**.**

- Justification:
    - Both Logistic Regression and Random Forest are well-suited for binary classification problems, especially in scenarios with class imbalance such as fraud detection.
    - Logistic Regression provides a transparent baseline, while Random Forest can capture complex patterns and improve predictive performance.

- Data split:

```
 X = df['text']
 y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=7)
```

9. Visualization of Results & Model Insights

- **Confusion Matrix**: Showed improved precision with Random Forest
- **ROC Curve**: AUC scores > 0.85
- **Feature Importance**: Logistic Regression coefficients and RF feature importance.

- **Insights**:
  - Words like "breaking", "shocking" often appear in fake news
  - Model confidence increases with more context (n-grams).

10. Tools and Technologies Used

- **Programming Language**: Python
- **Notebook**: Google Colab / Jupyter Notebook
- **Libraries**:
  1. **Data Handling**: pandas, numpy
  2. **Visualization**: seaborn, matplotlib
  3. **Modeling**: scikit-learn,imbalanced-learn
- **Visualization Tools**: GitHub.

11.Team Members and Roles

S.NO	NAMES	ROLES	RESPONSIBILITY
1.	Kamesh.K	Leader	Data collection & cleaning
2.	Kishore kumar.K	Member	Feature engineering
3.	Monishraj.V	Member	Exploratory data analysis (EDA)
4.	Selvan samuvel.A	Member	Model building,model evaluation