#### 

#### **Phase-2 Submission Template**

**Student name :**Priyanka T

**Register number :**422623104010

**Institution :** University College of Engineering Panruti

**Department :** Computer Science and Engineering

**Date of submission :**09-05-2025

**Github repository link :** <https://github.com/priya2005priya2005-arch/EXPLORING-THE-TRUTH-WITH-FAKE-NEWS-DETECTION-POWERED-WITH-NATURAL-LANGUAGE-PROCESSING.git>

**Problem Statement:**

**RefinedProblem:**  
The task is to develop a machine learning model that can accurately detect fake news from a given dataset of news articles. Fake news is a significant problem in today’s digital world, influencing public opinion, political views, and even health-related behaviors. Given the rise of misinformation, automating fake news detection can help mitigate its spread.

**Problem Type:**  
This is a classification problem where the model will classify news articles as either “fake” or “real.”

**Impact and Relevance:**

* Impact: The spread of fake news can have widespread consequences such as affecting elections, harming public health, and spreading false information. Automated detection of fake news can be a useful tool for social media platforms, news agencies, and public forums to reduce misinformation.
* Relevance: The increasing accessibility to news sources via social media platforms makes this problem even more relevant. Detecting fake news at scale can help ensure the accuracy and credibility of information shared online.

**Projectobjectives:**

**Key Technical Objectives:**

* To build a machine learning model using natural language processing (NLP) techniques that can classify news articles as either fake or real.
* To evaluate and compare multiple models' performance in terms of accuracy, precision, recall, and F1-score.
* To ensure that the model has good interpretability and is capable of providing insights into how it differentiates between fake and real news.

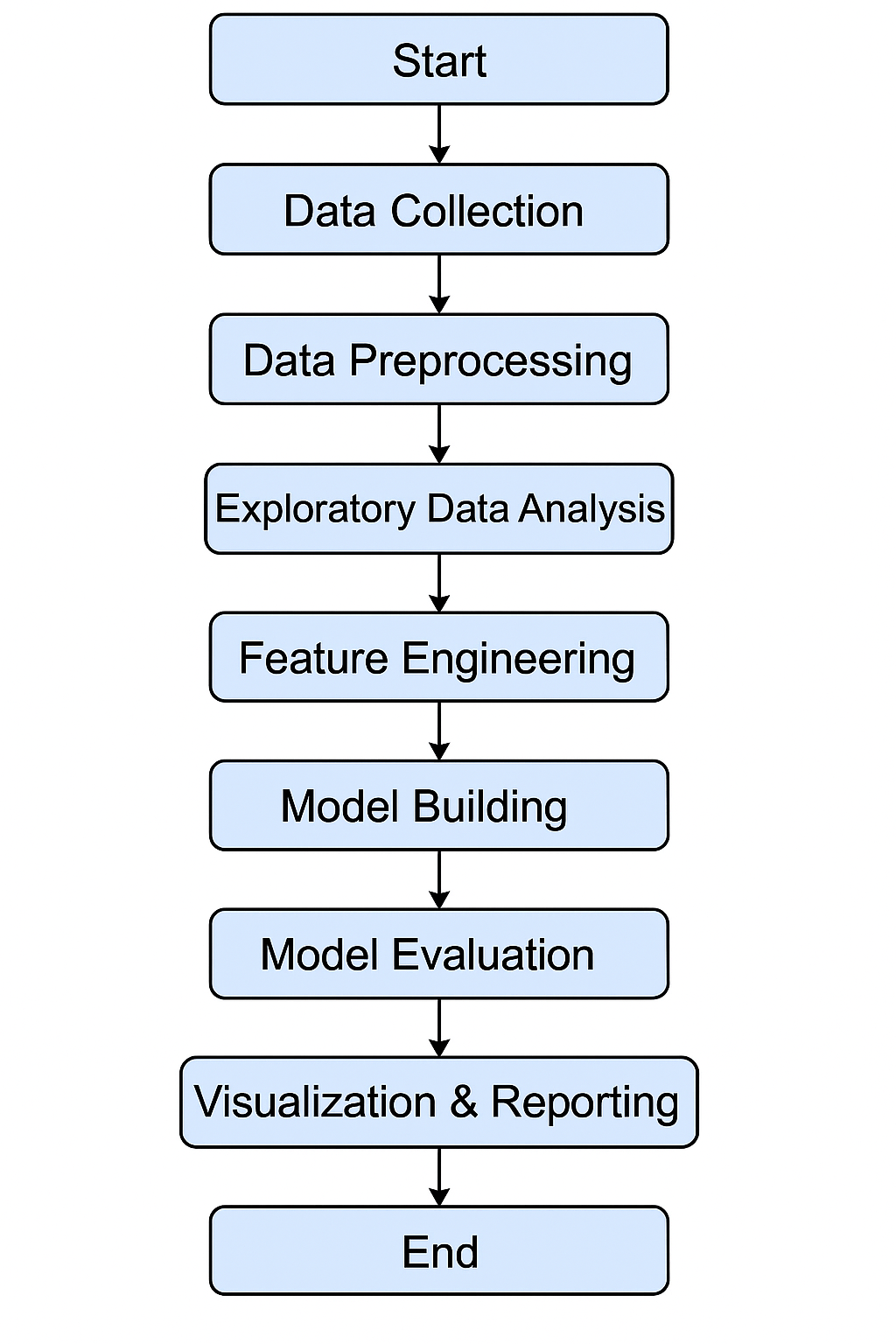
**Model Aims:**

* Achieve high classification accuracy (targeting above 85% accuracy).
* Ensure that the model is interpretable, with clear explanations for its predictions (using techniques like feature importance or SHAP values)

**Goal Evolution:**

* After performing initial data exploration, we realized that the dataset contains subtle differences in wording and context, so we will focus on advanced NLP techniques like TF-IDF, word embeddings, and deep learning models to capture these nuances.

**Flowchart of the Project Workflow:**

****

**Data Description:**

**Dataset Name**: Fake News Dataset (you can mention the source like Kaggle if you are using a public dataset)

**Dataset Origin**: The dataset is sourced from Kaggle’s Fake News Detection dataset.

**Type of Data**: Unstructured text data

**Number of Records and Features:**

* Number of records: ~20,000 articles
* Features: 5 columns (e.g., headline, text, category, etc.)

**Static or Dynamic Dataset:** Static dataset (the dataset represents a fixed set of news articles collected over time).

**Target Variable:** The target variable is “label,” where “1” represents fake news and “0” represents real news.

**Data preprocessing:**

**Handle Missing Values**:

* Checked for missing values in the text and label columns.
* Rows with missing values were either dropped or imputed (if necessary).

**Remove Duplicate Records**:

* Removed any duplicate rows to ensure no redundancy in the data.

**Outlier Detection and Treatment**:

* Since we’re dealing with text data, outliers were not relevant for this task. However, we checked for unusually short or long texts and removed or adjusted them as necessary.

**Data Type Conversion**:

* Ensured all text data was in a string format and categorical variables (if any) were encoded.

**Encode Categorical Variables**:

* Label encoding was used for the target variable (fake/real).

**Normalization**:

* Not necessary for textual data, but we applied text preprocessing like lowercasing, punctuation removal, and stopword removal.

**Text Tokenization**:

* Tokenized text into words using libraries such as nltk or spaCy.

**Exploratory data analysis:**

**Univariate Analysis:**

* Distribution of the target variable (fake vs. real news) via bar charts.
* Word frequency distribution in fake vs. real news articles.

**Bivariate/Multivariate Analysis:**

* Used pairplots to explore relationships between different features (e.g., length of the text vs. label).
* Investigated feature correlations if any additional numerical features exist (e.g., number of words).

**Insights Summary:**

* The dataset contains a fairly balanced number of fake and real news articles, making it a good candidate for classification models.
* Common words and phrases were identified that can serve as potential features in distinguishing fake news from realnews**.**

**Feature Engineering:**

**Text Features:**

* + Used TF-IDF (Term Frequency-Inverse Document Frequency) to represent text data.
  + Created word embeddings using models like Word2Vec or GloVe to capture semantic meaning.

**Additional Features:**

* + Extracted text length, word count, and sentence count as potential features.

**Dimensionality Reduction:**

* + Used Principal Component Analysis (PCA) for dimensionality reduction if necessary to speed up the model training**.**

**Model building:**

* **Model Selection**:
  + Chose two machine learning models for comparison:
    1. **Logistic Regression**: Simple and interpretable, suitable for baseline performance.
    2. **Random Forest Classifier**: A powerful ensemble model that can capture non-linear relationships.
* **Train-Test Split**:

Split the data into training (80%) and testing (20%) sets, ensuring the classes are balanced in each subset.

* **Model Evaluation Metrics**:
  + For classification, used **accuracy**, **precision**, **recall**, and **F1-score** to evaluate the models.
  + Choose **Random Forest** as the final model after comparing the metrics.

**Visualization of results & model insights:**

**Confusion Matrix:**

* The confusion matrix shows the number of false positives and false negatives for both models.

**ROC Curve:**

* The ROC curve shows the trade-off between sensitivity (recall) and specificity.

**Feature Importance Plot:**

* The Random Forest model provides insights into the most important features (words/terms) in distinguishing fake and real news.

**Tools and technologies used:**

**Programming Language:** Python 3.7+

**IDE/Notebook:** Jupyter Notebook or Google Colab

**Libraries:**

* pandas, numpy (data manipulation)
* scikit-learn (machine learning models)
* nltk, spaCy (text preprocessing)
* matplotlib, seaborn (visualization)

**NLP Tools:** TF-IDF (from scikit-learn), Word2Vec, GloVe embeddings

**Team Members and Contributions:**

1. **SANTHOSH P**– **Data Cleaning & Preprocessing**
   * Responsible for cleaning the dataset, handling missing values, and removing duplicates.
   * Performed text preprocessing tasks such as removing stop words, punctuation, and converting text to lowercase.
   * Implemented feature encoding techniques (e.g., Label Encoding) and text vectorization (e.g., TF-IDF).
2. **PRIYANKA T** – **Exploratory Data Analysis (EDA)**
   * Conducted statistical and visual exploration of the dataset.
   * Created visualizations to analyze the distribution of features and the relationship between features and the target variable.
   * Provided insights into the patterns and trends in the dataset, which helped guide feature engineering.
3. **ABINAYA C**– **Feature Engineering**
   * Developed new features such as text length and word frequency.
   * Applied dimensionality reduction techniques (if applicable, e.g., PCA).
   * Selected the most important features based on EDA and domain knowledge to enhance model performance.
4. **YUVAN SHANKAR S**– **Model Development**
   * Selected, trained, and tuned machine learning models (e.g., Logistic Regression, Random Forest).
   * Performed model validation and evaluated performance using appropriate metrics such as accuracy, precision, recall, and F1-score.
   * Worked on optimizing the models for better generalization and performance.
5. **JAYASHREE V** – **Documentation and Reporting**
   * Compiled and documented the entire project, ensuring clarity and consistency in all sections.
   * Wrote the project report, including the problem statement, objectives, data description, and results.
   * Created visualizations for the final report and prepared the presentation.