**Phase-2 Submission Template**

**Student Name:Akshaya JS**

**Register Number:** 613023104033

**Institution:** vivekanandha college of technology for women

**Department:** B.E.Computer science and engineering

**Date of Submission:** 30.04.2025

**Github Repository Link:**

*https://github.com/itsakshaya26/Project.git*

### **Problem Statement**

In today's digital era, fake news spreads rapidly on social media and news platforms, leading to misinformation and public confusion. Detecting fake news automatically is crucial to maintaining public trust and preventing societal harm.

This project focuses on building a machine learning model that can accurately classify news articles as *real* or *fake* based on their content using Natural Language Processing (NLP). By using computational methods to analyze news text, we aim to provide a scalable, fast, and reliable method of fake news detection.

 **Problem Type**: *Binary Classification*

 **Relevance:** Reduces workload on human agents, ensures 24/7 support, and enhances customer experience.

### **Project Objectives**

* Develop a machine learning pipeline to detect fake news using NLP techniques.
* Clean and preprocess textual data to make it suitable for model training.
* Implement and compare multiple classification models (e.g., Logistic Regression, Random Forest, SVM).
* Achieve high **accuracy** and **F1-score** in distinguishing fake news from real news.
* Interpret model outputs using feature importance and model explainability techniques.

*The goal evolved after EDA: initial focus was only accuracy, but later interpretability was added to ensure practical real-world usage*

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

### *[Data Collection → Data Preprocessing → EDA → Feature Engineering → Model Building → Evaluation → Interpretation → Final Deployment]*

### **4. Data Description**

* **Dataset Name**: Fake News Dataset
* **Source**: Kaggle (https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset)
* **Type**: Structured text data
* **Records**: ~44,000 news articles
* **Features**: Title, Text, Subject, Date, Label (target)
* **Target Variable**: label (1 = Fake, 0 = Real)
* **Static Dataset**

### **5. Data Preprocessing**

* + Removed missing values and irrelevant columns (date, subject)
  + Combined title and text for better context
  + Cleaned text: lowercasing, punctuation removal, stop word removal, stemming/lemmatization
  + Label encoding for target variable
  + Tokenized text and transformed using TF-IDF Vectorizer

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

### **Univariate**:

* Word cloud and frequency plots revealed common fake news terms.
* Fake news often contains emotionally charged words.

**Bivariate/Multivariate**:

* Term frequency analysis showed stylistic differences between real and fake news.
* Correlation matrix was less applicable due to textual nature; instead, text similarity and n-gram analysis were more insightful.

**Insights**:

* Fake news articles tend to use excessive sensational words and clickbait-like titles.
* Real news is more likely to be published under legitimate "Subject" categories.

### **7. Feature Engineering**

* Combined title and body into a new field: combined text
* Used **TF-IDF** features (unigrams and bigrams)
* Created custom features:
  + Word count
  + Punctuation count
  + Capital letter ratio
* Removed sparse features and reduced dimensionality

### **8. Model Building**

**Models Used**:

* Logistic Regression
* Random Forest
* Support Vector Machine (SVM)

**Reason for Choice**:

* Logistic Regression: Simple and interpretable
* Random Forest: Handles high dimensionality well
* SVM: Effective with sparse and text data

**Train/Test Split**:

* 80/20 stratified split

**Evaluation Metrics**:

* Accuracy
* Precision
* Recall
* F1-Score

*Initial best model: Random Forest with 94% F1-score*

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

###  **Confusion Matrix**: Showed balanced performance with low false positives

 **ROC Curve**: AUC > 0.95 for best model

 **Feature Importance**: TF-IDF terms like "breaking", "shocking", and excessive use of caps were top indicators

 **Residual Analysis**: Few ambiguous articles misclassified by all models

### **10. Tools and Technologies Used**

* **Language**: Python
* **IDE**: Google Colab & Jupyter Notebook
* **Libraries**:
  + pandas, numpy
  + seaborn, matplotlib, plotly
  + scikit-learn, nltk, XGBoost
  + wordcloud, re, string
* **Version Control**: GitHub

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

|  |  |
| --- | --- |
| **Name** | **Role & Contributions** |
| J.S.AKSHAYA | Led Data Collection & Text Preprocessing (including tokenization, stopword removal) |
| B.HARISHMA | Performed Exploratory Data Analysis (EDA) and initial insights visualization |
| V.KEERTHIKA | Handled Feature Engineering and Transformation (TF-IDF, custom features) |
| V.DEJASRI | Developed and Tuned Machine Learning Models (Logistic Regression, SVM, Random Forest) |
| B.LAVANYA | Created Evaluation Visualizations and Contributed to Reporting & Documentation |