

# Fake News Detection Using ML(SVM)

Presented By

Nikhil Singh : 2202840100137

Nitesh Singh : 2202840100141

Shantanu Singh : 2202840100195

Shashwat Jaiswal : 2202840100198

Under the Guidance of  
Mr. Rohit Mishra  
(Assistant Professor)



Department of Computer Science & Engineering  
United Institute Of Technology, Prayagraj  
(Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow)

2025–2026

# Overview

1 Introduction

2 Problem Statement

3 Existing Solution

4 Methodology

5 SDLC MODEL

6 Iterative Waterfall Model

7 Requirements

8 Design

9 References

# Introduction

In today's era, millions of news stories are being generated every minute around the world, mixed with true and false information, making it difficult to understand which one is true or fake news.

Fake news spreads rapidly across online platforms, leading to misinformation and confusion among the public.

- Manual verification is time-consuming, unreliable, and not scalable.
- The system uses Machine Learning algorithms and NLP techniques to automatically detect and prevent the spread of fake news.
- There is a need for an intelligent system that analyzes text and classifies news accurately.
- Promote Rapid & Authenticated verified news to prevent spreading of fake news among members.

# Objectives

The primary objective of this project is to develop an intelligent system capable of detecting fake news using Machine Learning and Natural Language Processing (NLP) techniques. The specific objectives are as follows:

**1. Algorithmic Accuracy and Analysis:** To design, implement, and evaluate three distinct machine learning algorithms—Support Vector Machine (SVM), Logistic Regression, and Naive Baye.

**2. Integrated Full-Stack Architecture:** To develop a cohesive web application that integrates a React.js frontend (enhanced with HTML5, CSS3, and Bootstrap) with a robust backend infrastructure utilizing both Django and Node.js

**3. Efficient Data Management:** To implement a scalable MongoDB database solution for the efficient storage of news article datasets, training data, and user query logs

**4. User-Centric Interface Design:** To create an intuitive and accessible user interface where users can easily input news URLs or text snippets for verification

# Problem Statement

"The rapid rise of social media has led to an uncontrollable spread of misinformation and fake news, which negatively impacts public opinion and social stability. Manually verifying this vast amount of data is impossible due to its volume and speed. This project addresses this challenge by developing an automated application that uses Machine Learning and Natural Language Processing (NLP) to detect and classify news articles as 'Real' or 'Fake' with high accuracy."

# Existing Solution

**Manual Fact-Checking Organizations:** Currently, fake news is largely identified by human fact-checkers (e.g., Snopes, PolitiFact). While accurate, this process is extremely slow and cannot keep up with the volume of news generated every second.

**Keyword-Based Filtering:** Many existing social media platforms use simple keyword blockers or user reporting mechanisms. These are often ineffective against sophisticated fake news that uses manipulated context rather than just banned words.

**Lack of Accessibility:** Most advanced detection tools are reserved for researchers or large tech companies, leaving the average internet user without a simple, real-time tool to verify the news they read daily.

## **Support Vector Machine (SVM)**

Support Vector Machine is a robust supervised learning algorithm employed in this project for classification tasks. It operates by determining the optimal hyperplane that separates data points of different classes with the maximum possible margin, which enhances the model's generalization capabilities. SVM is particularly advantageous in high-dimensional spaces, making it highly effective for complex feature sets. Additionally, the algorithm utilizes the "kernel trick" to efficiently handle non-linear data relationships by mapping input vectors into higher-dimensional feature spaces.

# Methodology

- **Front-end-** Using HTML5, CSS 3, java script for primary user interface model & React.js framework to improvise UI outlook architecture and structure.
- **Backend-** Using Django Framework backend core functioning & Node js for backend connecting with Django Using RESTFULL api's for communicating with the database section.
- **Database-** Using MongoDB for storing new dataset records after analysing news articles along with timestamps recorded in analyze history.
- **AI/ML-** NumPy, Pandas python libs. and SVM, logistic regression, Naive Bayes algorithms to train and analyze news datasets for further datasets creation.
- **Integration-** integrate the Django backend side with Node js for proper REST Full api communication at end time server communication .
- **Deployment -** Cloud (AWS) / Google Cloud services, Streamlit.

# SDLC MODEL

The software development lifecycle is a conceptual framework utilized in project management to outline the sequential phases encompassing an information systems development project, commencing with preliminary feasibility studies and culminating in the ongoing maintenance of a completed application. Its purpose is to establish and devise the diverse models governing the software development process.



Fig:1 SDLC Model

# Iterative Waterfall Model

- **Iterative Waterfall Model** "We adopted the Iterative Waterfall Model to allow for feedback and corrections between phases. This flexibility let us revisit previous stages to refine our design or fix bugs without restarting the entire process."

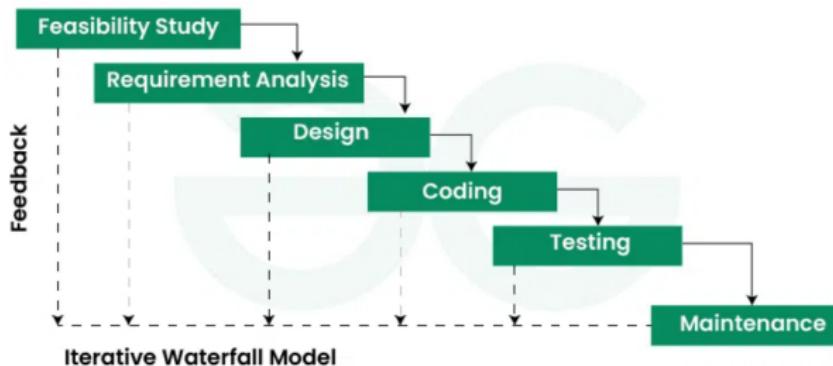


Fig:2 Iterative Waterfall Model

# Requirements Analysis

**Requirement Analysis:** During the requirement analysis phase of our project, we focus on understanding and clearly defining what users want from the software we are developing or modifying. It involves gathering all the necessary information and making sure we meet the users' needs effectively.

- **Pro:** It is flexible; if you find a bug during testing, you can go back and fix the code or design immediately.
- **Con:** It can be hard to manage the timeline because going back to fix things takes extra time and can delay the final launch.

# Requirements

- 1. Dataset Source Type** "We utilised the Kaggle Fake News Dataset, a widely accepted benchmark containing thousands of labelled news articles. It serves as a Supervised Learning resource, providing a balanced collection of 'Real' and 'Fake' instances to effectively train the model."
- 2. Data Format** "The dataset is structured in a CSV (Comma Separated Values) file, consisting of key feature columns: Title, Text, Author, and Label. The Label column acts as the target variable, using binary values (e.g., 0 for Reliable, 1 for Unreliable) to classify the news."

# DFD 0 / 1 level Diagram

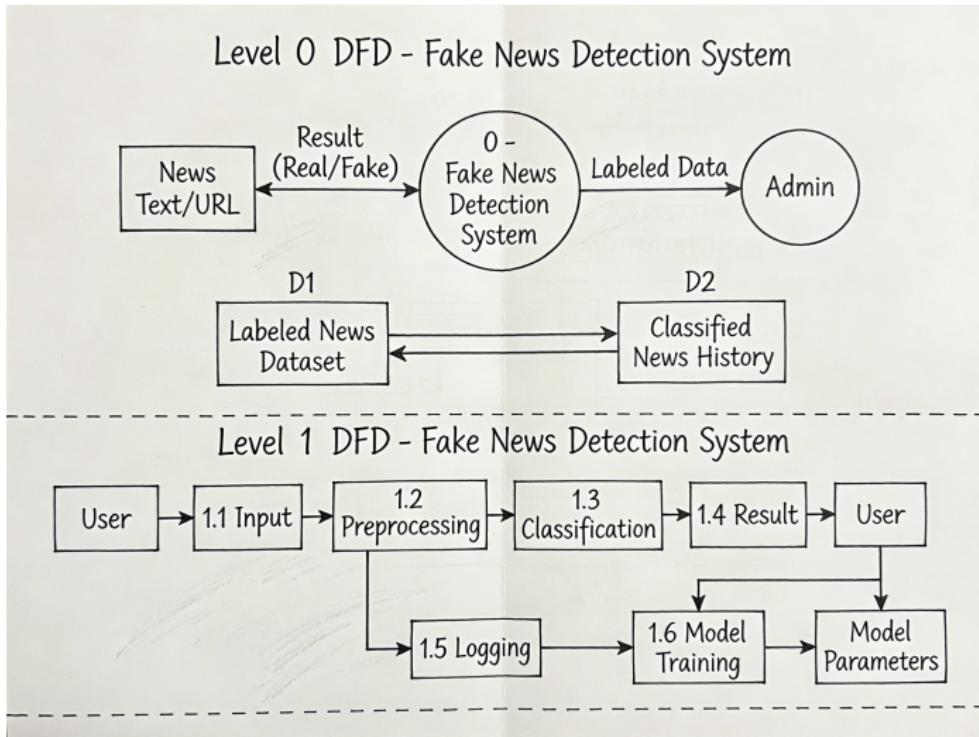


Fig:3 Level 0 and 1 Data Frame Diagram

# ER Diagram



Fig: 4 ER Diagram

# Flow Chart

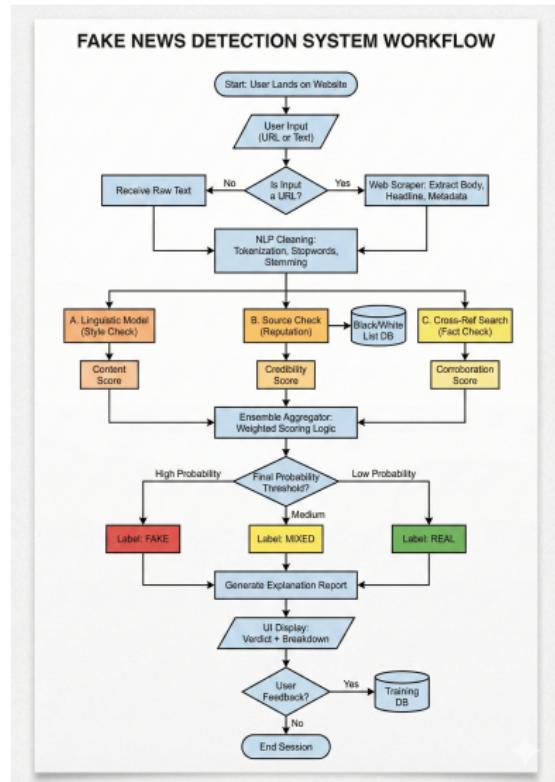


Fig: 5 Flow Chart

# Gantt Chart

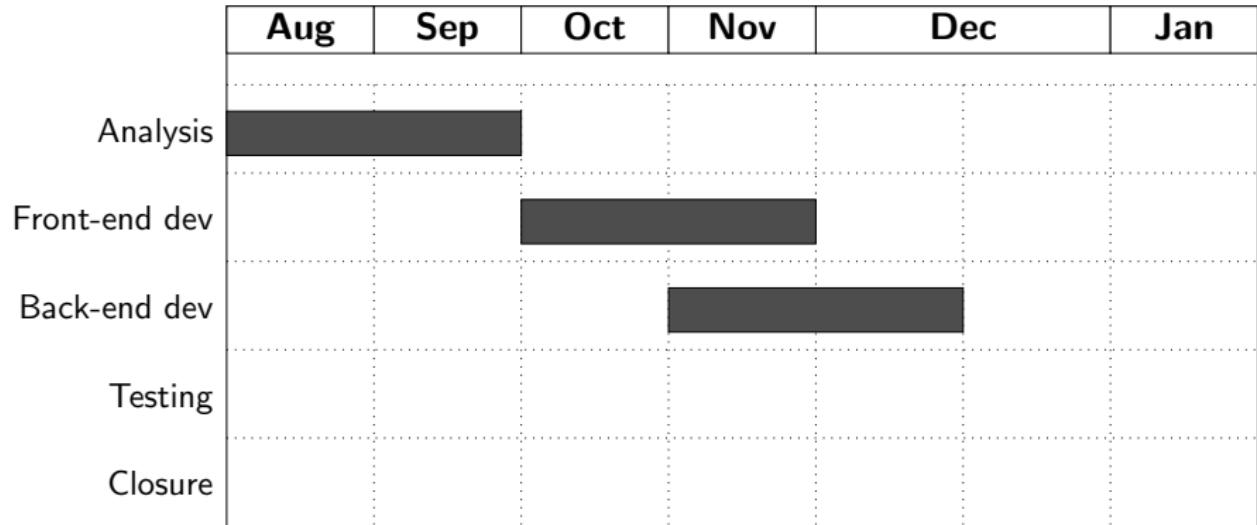


Fig: 6 Gantt Chart- Project Timeline (Aug – Jan)

# Data visualization



Fig:7 Bar Chart

# Data visualization

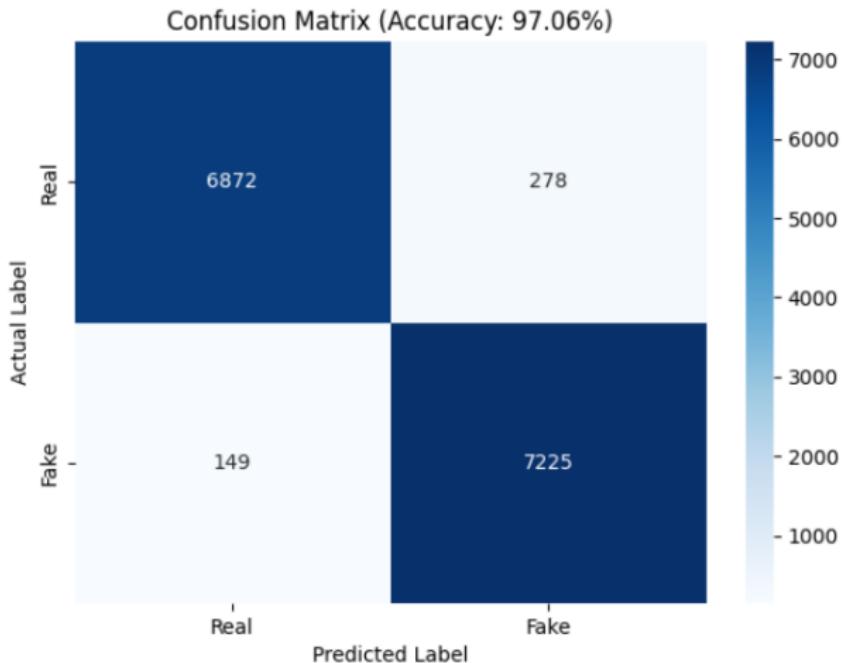


Fig:8 Confusion-Matrix.

# References

-  M. D. P. I. Journals, "Fake news detection using machine learning and deep learning algorithms: A comprehensive review and future perspectives," *MDPI*, 2025.
-  Fundamental Research, "An overview of fake news detection: From a new perspective," *Fundamental Research*, 2025.
-  arXiv, "Misinformation detection using large language models with explainability," *arXiv preprint*, 2025.
-  AAAI/ACM Conference, "Too focused on accuracy to notice the fallout: Towards socially responsible fake news detection," *Proceedings of the AAAI/ACM Conference*, 2025.
-  V. Singh, R. Dasgupta, and D. Choudhury, "Fake news detection using machine learning," *International Journal of Research in Applied Science and Engineering Technology*, vol. 11, no. 6, pp. 1450–1458, 2023.

# The End

# Thank-You Teacher's