M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 Certified Institution
Thalavapalayam, Karur, Tamilnadu.

IBM

SPAM EMAIL DETECTION

Guided By:

Ms. VAISHNAVI
IBM Corporate Trainer

M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 Certified Institution
Thalavapalayam, Karur, Tamilnadu.

IBM

TEAM LEAD

ARUNAPRABHA K N (927623BAM005)

TEAM MEMBERS

ASIR ROBERTS S (927623BAM006)

BHARATH KUMAR P (927623BAM007)

BOOBALAN M (927623BAM008)

PROBLEM STATEMENT

Email spam remains a major issue, making up over **45% of all email traffic worldwide**. Despite widespread filtering systems, many still fail to catch **evolving phishing techniques** and deceptive content, putting users and organizations at risk.

Users often don’t know the exact reason why an email was marked as spam, and businesses find it hard to adjust standard filters to match their own communication needs and security policies.

INTRODUCTION

- Email spam has become increasingly sophisticated, often slipping past traditional filters and posing serious risks to users and organizations. Basic detection methods are no longer enough to keep inboxes safe.
- This system uses advanced machine learning techniques to detect spam with high accuracy. It provides real-time analysis, detailed visual insights, and a secure, user-friendly experience that empowers users to understand and manage unwanted emails more effectively.

ABSTRACT

This project presents an intelligent email spam detection system built using machine learning and modern full-stack technologies. It aims to enhance traditional spam filters by offering real-time email analysis, transparency in classification, and a user-friendly interface. The system not only identifies spam with high accuracy using advanced models but also provides visual explanations, secure user authentication, and customizable settings - making it suitable for individuals, educators, and organizations seeking smarter, more explainable email filtering solutions.

EXISTING SYSTEM

- **Heavily reliant on static rules** — can't adapt to evolving spam techniques
- **Limited detection scope** — misses subtle threats like spear-phishing and social engineering
- **No explanation mechanism** — users get no insight into why an email was marked as spam
- **One-size-fits-all filtering** — lacks customization based on user or organizational behavior

PROPOSED SYSTEM

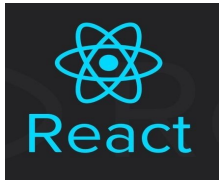
- **Machine Learning Detection** — Accurate spam detection using advanced models like Naive Bayes.
- **Explainable Results** — Users see why an email is flagged with visual insights.
- **Interactive Visualizations** — Confidence scores and charts for better understanding.
- **Secure & Customizable** — JWT-based authentication and personalized settings.
- **Scan History** — Track and review past analyses.
- **Responsive UI** — Optimized for both desktop and mobile devices.

Technology Stack – Frontend & Authentication

Frontend

- **Next.js with React**: Enables fast, SEO-friendly rendering and component-based UI architecture

- **Tailwind CSS**: Modern utility-first CSS framework that ensures responsive and sleek styling



Authentication

- **JWT (JSON Web Token)**: Secure, stateless login system that verifies users and maintains session integrity

Technology Stack – Backend & Data Handling

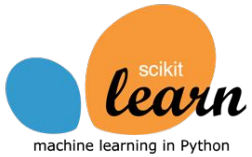
Backend (Python + Flask)

- Lightweight web framework to handle API requests and route predictions from the ML models
- Designed for scalability and easy integration with the frontend

Data Handling & Storage

- **Pickle Files**: Used to serialize and store trained models and scanned data for efficient access
- **LocalStorage (Frontend)**: Retains user session data and interface preferences without repeated server calls

Technology Used - Python & Libraries



scikit-learn: For building and training models like Naive Bayes, TF-IDF vectorization, and evaluation metrics

pandas: For handling and preprocessing email datasets

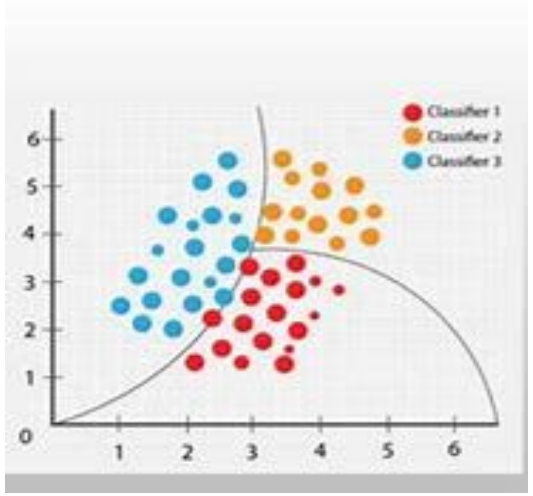
numpy: Supports numerical operations and data transformations

nltk / re: For natural language processing, tokenization, and regex-based email content cleaning

pickle: For saving and loading trained models efficiently

Flask: To create a REST API and connect the ML model to the frontend

ALGORITHMS USED



1.Naive Bayes Model

Based on Multinomial Naive Bayes with TF-IDF weighting.

Trained on preprocessed spam/ham dataset.

Key Features:

- 30+ spam-indicating keywords and patterns
- Weighted word frequencies for better accuracy
- High performance on short-text classification

ALGORITHMS USED

2. Heuristic-Based Classifier

Custom rule-based model designed to catch edge cases

Analyzes:

- Excessive **capitalization**
- Abnormal **punctuation** (!!!, ???)
- Presence of **URLs**, suspicious links, and symbols
- Keyword density and placement

ALGORITHMS USED

3. Word Influence Analysis

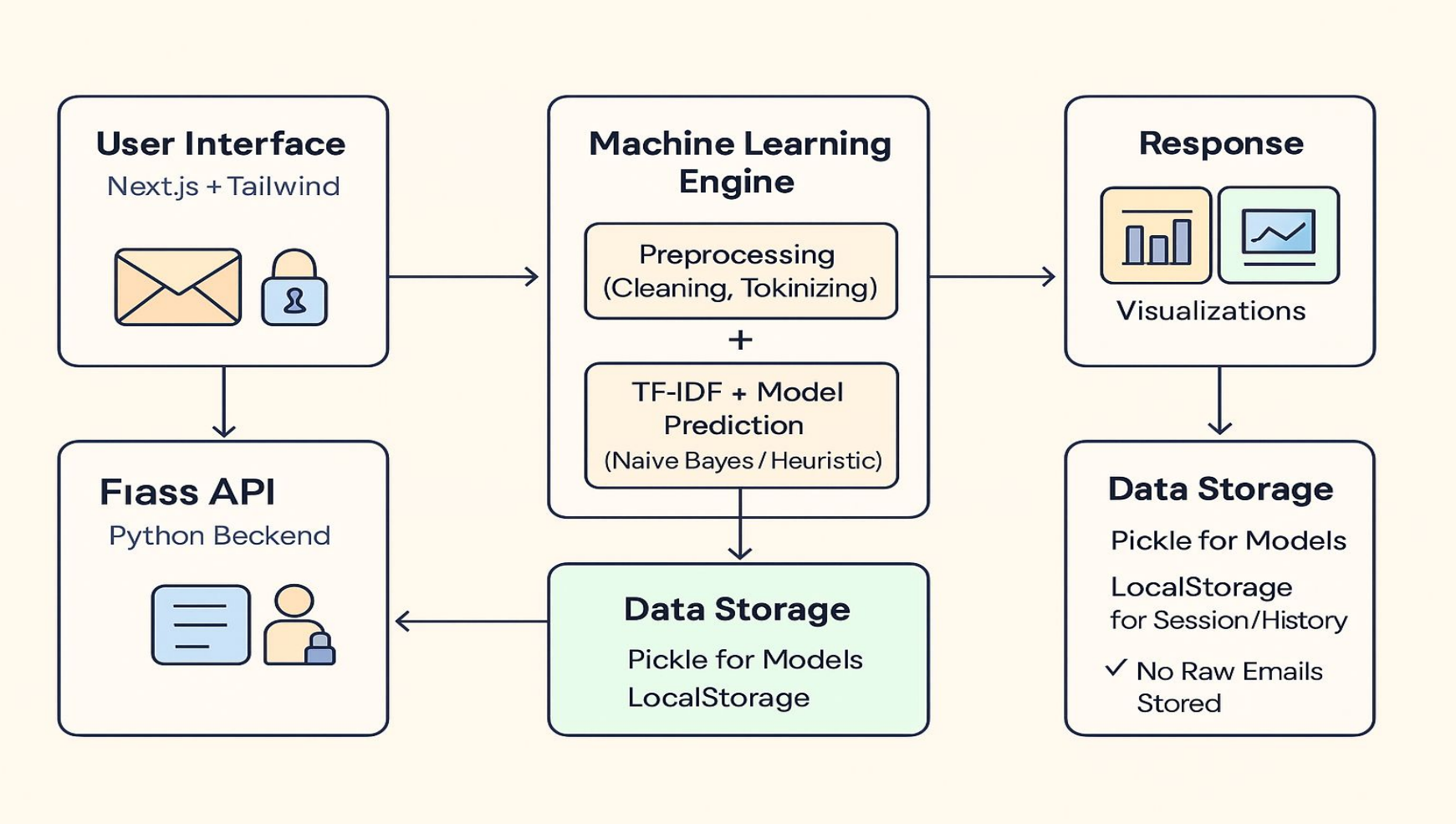
Tracks the top contributing words behind the spam/ham classification

Helps users understand **why** an email was flagged as spam

Visual Tools:

- Bar charts showing word weights and their contribution

WORKFLOW


User Interface
Next.js + Tailwind
Machine Learning Engine
Preprocessing (Cleaning, Tokinizing)
+
TF-IDF + Model Prediction
(Naive Bayes / Heuristic)
Response
Visualizations
Fıass API
Python Beckend
Data Storage
Pickle for Models
LocalStorage
Data Storage
Pickle for Models
LocalStorage for Session/History
✓ No Raw Emails Stored

WORKFLOW

User Interface

- Built with **Next.js** and **Tailwind CSS** for responsive, sleek, and interactive design.
- Users input email content for spam detection.

Flask API

- A **Python backend** API that connects the frontend with the ML engine.
- Handles routing, data exchange, and authentication.

Machine Learning Engine

- **Preprocessing**: Cleans and tokenizes input text.
- **TF-IDF Vectorization + Model Prediction**: Transforms text and applies a trained model (Naive Bayes or heuristic rules) for spam classification.

WORKFLOW

Response & Visualization

- Outputs are shown with **clear visualizations**, like charts and graphs, to help users interpret predictions.

Data Storage

- **Pickle files** store trained ML models securely.
- **LocalStorage** saves session data/history in-browser—**no raw email content is stored**, ensuring user privacy.

Privacy & Security

- Ensures **secure interactions** with no sensitive data stored or transmitted beyond what's necessary.

BENEFITS

- **High-Accuracy Detection** with advanced Naive Bayes
- **Real-Time Protection** from spam and malicious content
- **Minimal False Positives** for smoother communication
- **Explainable AI** with confidence scores & word impact
- **Responsive, User-Friendly Design** across devices
- **JWT Security & API Integration** for seamless use
- **Insightful Analytics** on spam trends and model performance

FEATURES

- **Spam Detection**: ML analysis, real-time classification, confidence scores
- **Explainable AI**: Word influence, top contributors, model insights
- **User Management**: JWT authentication, personalized profiles
- **History & Analytics**: Scan history, detailed reports
- **Technical**: RESTful API, responsive design

APPLICATIONS

- **Email Services:** Used in popular email platforms (e.g., Gmail, Outlook) to filter out spam.
- **Corporate Security:** Helps organizations protect their inboxes from phishing and malicious emails.
- **Personalized Email Management:** Enables users to manage their inboxes by sorting emails into relevant categories.
- **Educational Use:** Can be used for teaching purposes in machine learning and natural language processing courses.

FUTURE GOALS

- **Enhanced Accuracy:** Experiment with advanced models like **Deep Learning** for better spam detection.
- **Integration with Other Platforms:** Integrate the system with popular email clients and apps for automatic filtering.
- **Continuous Learning:** Implement online learning to allow the system to adapt to new spam trends in real-time.
- **Multi-language support:** To detect spam across diverse languages.

CONCLUSION

The Spam email detection revolutionizes email security by combining advanced machine learning with transparent explanations. Using Naive Bayes classification with TF-IDF vectorization, the system not only identifies spam with high accuracy but explains its decisions through word influence analysis. The modern React/Next.js frontend and secure Flask backend create a seamless user experience, while JWT authentication ensures data protection. By providing insights into spam patterns and maintaining comprehensive history tracking, it empowers users to better understand and combat email threats.


THANK
YOU!