# Email Spam Classifier with PII Masking

## Data Scientist Internship Submission — Akaike Technologies

---

### 1. Introduction to the Problem Statement

With email being a fundamental mode of communication in both professional and personal domains, the rise of spam and phishing emails presents a growing concern. These unsolicited emails not only cause inconvenience but also pose serious security threats, such as fraud, malware delivery, and identity theft.

Traditional spam filters are often rule-based and rigid, making them vulnerable to bypass techniques. Furthermore, in real-world datasets, **Personally Identifiable Information (PII)** like email addresses, names, and Aadhar numbers are embedded in the content, raising concerns about **privacy**, **compliance**, and **ethical AI practices**.

This project addresses both concerns:

- **Classifying emails** into spam and non-spam using machine learning.
- **Masking PII** within the text to ensure data security and ethical model training.

---

### 2. Approach for PII Masking & Classification

🔹 **PII Masking Strategy**

Before performing any analysis or model training, PII was removed or masked using **pattern-based techniques** (primarily regular expressions):

- **Email addresses**, Aadhar numbers, names, and personal identifiers were replaced with placeholder tokens like <EMAIL>, <AADHAR>, and <NAME>.
- This step ensured the model focused on contextual and structural text features rather than memorizing sensitive or irrelevant entities.

**Example:**

Original: "My name is Ravi Sharma. My Aadhar is 1234 5678 9012. I need help with billing."

Masked: "My name is [full_name]. My Aadhar is [aadhar_num]. I need help with billing."

This also helps the model generalize better when deployed in real-world environments where PII is dynamically different.

---

### 3. Model Selection & Training Details

🔹 **Why Logistic Regression?**

The model of choice was **Logistic Regression**, well-known for:

- Its simplicity and interpretability.
- High performance in binary classification tasks.
- Speed and low computational overhead—ideal for quick iterations and scalable APIs.

🔹 **Text Preprocessing Pipeline**

The text processing and model training pipeline includes:

1. **Text Cleaning**: Lowercasing, punctuation removal.
2. **Stopword Removal**: Eliminated common words that carry minimal meaning.
3. **Tokenization**: Text split into meaningful components.
4. **Vectorization**: Text converted into numerical features using **TF-IDF** (Term Frequency–Inverse Document Frequency) to capture word importance.
5. **Model Training**: Logistic Regression fitted on the TF-IDF vectors.

🔹 **Tools & Libraries Used**

- **Python**, **Scikit-learn**, **Pandas**, **Regex**, **NLTK**
- **TF-IDF Vectorizer** (max_features=5000)
- **Model Evaluation** using accuracy, confusion matrix, and classification report

🔹 **Evaluation Outcome**

The trained model achieved:

- ✅ **Accuracy**: 74.42%

This shows that the model correctly classified over 74% of the emails. It effectively learns the distinguishing features of spam, even after PII removal, which confirms the robustness of the preprocessing approach.

---

#### 4. Deployment Architecture & API Design

To enable real-time inference and API-based access, the trained model was deployed using **FastAPI**, a modern, high-performance web framework.

🔹 **Key Features of Deployment:**

- **Model Loading Automation**: On launch, the app checks if email_classifier.pkl exists. If not, it triggers training.
- **API Interface**: The api.py file exposes endpoints for classification using HTTP POST requests.
- **Uvicorn Server**: Used for running the API locally and supports async operation for high-speed inference.

🔹 **Workflow Summary:**

1. **Startup** → Check model → Train if needed.

2. **Inference** → Accept email text → Predict → Return spam / not spam.

---

**5. Challenges Faced & Solutions Implemented**

**Challenge 1: PII Leakage During Preprocessing**

**Solution**: Used regex-based masking to redact sensitive content before model training, ensuring ethical AI and GDPR/DPDP readiness.

**Challenge 2: Overfitting on Common Spam Words**

**Solution**: TF-IDF vectorization reduces emphasis on overly frequent terms and balances rare ones for contextual accuracy.

**Challenge 3: Deployment Reliability**

**Solution**: Added automated model training check on server startup, reducing dependency on manual steps and improving resilience.

---

**6. Business & Internship Relevance**

This project reflects multiple skills that align directly with the work at Akaike Technologies:

- **Data Sanitization & Compliance**: Handling sensitive data in a privacy-preserving manner.
- **Practical ML Engineering**: Model building, evaluation, and real-time deployment.
- **API Design for Scalability**: Packaging ML into deployable and scalable systems.
- **Production Readiness**: Automation, modular structure, and testing built-in.

---

**7. Final Thoughts**

This project successfully bridges the gap between ethical data handling and intelligent automation. By combining **PII masking** with an **accurate spam detection system**, and wrapping it all in a production-ready API, it demonstrates the ability to deliver complete, real-world data science solutions.

---