Email Classification and PII Masking System

INTRODUCTION:

The objective of this project is to build an automated email classification system for a company's support team. The system must handle two key tasks:

1. Detect and mask all Personally Identifiable Information (PII) and Payment Card Industry (PCI) data in incoming emails.
2. Classify the email into one of several predefined categories, such as Billing, Technical Support, Account Management, etc.

This ensures that sensitive user information is protected during processing, and that support requests are routed efficiently to the correct department.

APPROACH:

PII Masking

To mask sensitive information without relying on Large Language Models (LLMs), the following techniques were used:

- Regular Expressions (Regex): Custom patterns were written to detect fields such as:
    - Full Name (full_name)
    - Email Address (email)
    - Phone Number (phone_number)
    - Date of Birth (dob)
    - Aadhar Number (aadhar_num)
    - Credit/Debit Card Number (credit_debit_no)
    - CVV (cvv_no)
    - Expiry Date (expiry_no)

Each match was replaced with a placeholder like [email], while also storing the original text and position for later demasking.

Email Classification

We used a machine learning model trained to classify emails into 4 categories provided in the dataset:

- Incident

- Request
- Change
- Problem

The classification pipeline uses TF-IDF vectorization and a traditional ML model such as RandomForestClassifier. This combination balances speed and accuracy, while keeping the deployment light for Hugging Face Spaces.

MODEL TRAINING AND TESTING DETAILS:

- Dataset: Provided email classification dataset
- Split: 80% training / 20% testing
- Preprocessing: Text cleaned, lowercased, and tokenized
- Vectorization: TF-IDF
- Model: Random Forest Classifier
- Accuracy: ~88% on validation set

Model training was performed locally, and the trained model was saved using joblib and loaded into the API for inference.

CHALLENGES & SOLUTIONS:

CHALLENGE	SOLUTION
Detecting varied formats of PII (e.g., card numbers, phone numbers)	Created precise regex rules and validated them with multiple test cases
Ensuring API follows strict JSON structure	Used Pydantic models in FastAPI to enforce schema
Deployment without frontend	Used Docker and deployed only the backend on Hugging Face Spaces
Hugging Face Docker build timeouts	Optimized dependencies and model loading to reduce build time

FINAL OUTPUT:

- API Endpoint (POST):
  https://sugesh23-email-api.hf.space/classify
- GitHub Repository:
  https://github.com/sugesh233/email-classifier-api

The API accepts a POST request with an input email body, masks PII, classifies the email, and returns the masked version, classification result, and list of masked entities in a structured format.