Project Report: Email Classification System for Support Team

PHILIP SIMON DEROCK P
philipsimonderock@gmail.com
8300057632

Github repo link: https://github.com/simon-derock/email-class.git
Hugging Face Spaces: https://huggingface.co/spaces/philip11/email-classification-system

Introduction:

In today's fast-paced digital world, support teams are inundated with emails, many of which contain sensitive personal information and diverse query types.
Manually sorting through these emails is time-consuming and inefficient, often leading to delayed responses. To address this challenge, our project the
Email Classification System for Support Team
was developed. This system automates the
classification of support emails into predefined categories and
masks personally identifiable information (PII), ensuring data privacy while streamlining workflow.

Objectives:

Automatically classify incoming emails into categories such as Billing Issues, Technical Support, and Account Management.
Detect and mask PII including names, phone numbers, email addresses, card details, and Aadhar numbers.
Provide a RESTful API to make this functionality accessible and integratable with other systems.
Deploy the system online for easy testing and demonstrations via Hugging Face Spaces.

System Features:

1. Email Classification
The heart of this system lies in its ability to categorize emails using machine learning techniques. By leveraging TF-IDF vectorization and Naive Bayes
classification, the model analyzes the text content and determines the appropriate category.

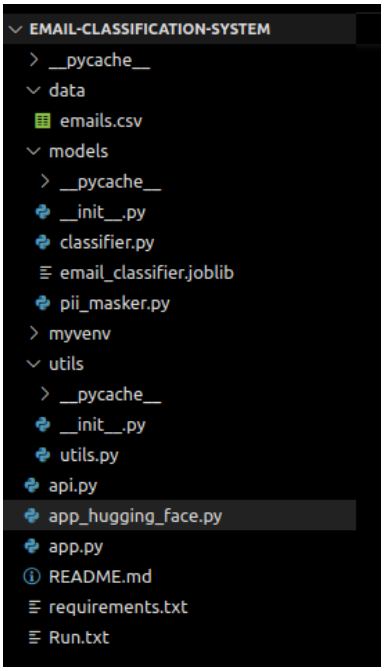
2. PII Masking
Data privacy is a core component of the solution. Using Pythons regular expressions (regex), thesystem identifies and replaces personal information such as:
- Names (e.g., "John Doe")
- Emails (e.g., "john.doe@example.com")
- Phone numbers

- Birth dates
- Government IDs (e.g., Aadhar numbers)
- Credit/debit card numbers, CVV, and expiry dates

3. API Deployment
A FastAPI-powered RESTful interface enables users to interact with the system programmatically. The API can receive an email body as input and return the
classification result along with masked text and PII metadata. It is hosted locally via Uvicorn, making it accessible on http://localhost:8000.

Project Architecture:

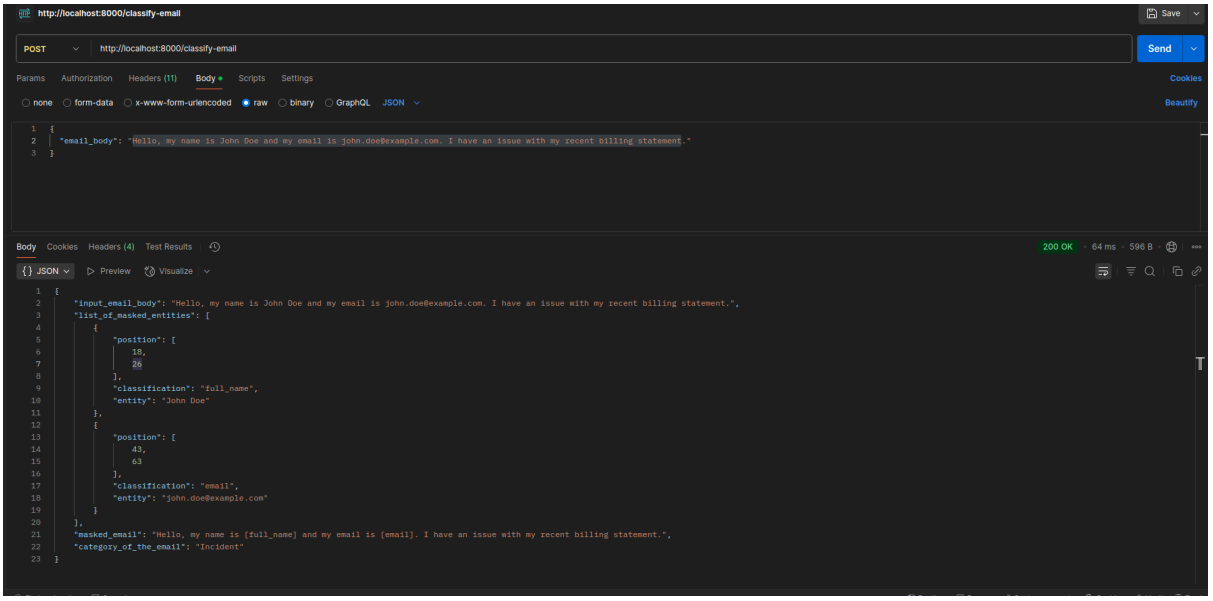
The project follows a well-structured format:
email-classification-system/
data/
Contains sample datasets
models/
utils/
Classifier and PII masking modules
Helper utilities
app.py# Training and testing scripts
api.py# FastAPI implementation
requirements.txt
List of dependencies
Each module has a dedicated function, making the system easy to understand, extend, or maintain.



Installation and Usage:

Setting up the system requires minimal effort:
1. Clone the repository.
2. Install dependencies using pip install -r requirements.txt.
3. Train the model using:
python app.py --train --data-path data/emails.csv
4. Test PII masking with:
python app.py --test-masking
5. Start the API:
uvicorn api:app --host 0.0.0.0 --port 8000 --reload
Deployment and Technologies
The system is also deployed on Hugging Face Spaces, allowing for public interaction and testing via a web interface.
Key Technologies:
- scikit-learn for machine learning
- Pandas & NumPy for data handling
- FastAPI for building the API
- Regex for PII masking

To Run Localy:
> uvicorn api:app --host 0.0.0.0 --port 8000 --reload



Conclusion

This project bridges two crucial domains automation and data privacy. By reducing manual efforts and ensuring sensitive data is not exposed,
the Email Classification System enhances the productivity of support teams while adhering to privacy best practices. Its modular design and API-first
approach make it ideal for integration into larger enterprise systems or use as a standalone support assistant.Further improvements could include integrating deep learning models for improved accuracy,
multilingual support, and deploying on scalable cloud
infrastructure for production use.