Report: Email Classification & PII Masking System

Introduction

In today's digital landscape, support teams handle large volumes of customer queries through emails. These emails often contain sensitive personal information (PII) and financial data (PCI), which must be secured during processing. The objective of this project is to build a complete pipeline that:

1. Identifies and masks sensitive information in incoming support emails
2. Classifies those emails into predefined support categories (e.g., Billing, Technical Support)
3. Exposes the pipeline as a secure and testable API for real-world use

This solution was designed, implemented, and deployed using Python, FastAPI, spaCy, scikit-learn, and Hugging Face Spaces, adhering strictly to data privacy and performance requirements.

Approach to PII Masking and Classification

PII Masking

To ensure sensitive information is not leaked during classification, a robust masking system was developed using:

- **Regex patterns** for detecting:
  - Email addresses
  - Credit/Debit card numbers
  - Aadhar numbers
  - CVV, expiry dates, and phone numbers
- **spaCy's Named Entity Recognition (NER)** for:
  - Full names
  - Dates of birth

Each identified entity is replaced with a tag (e.g., [full_name], [email], [aadhar_num]) based on its classification. The original values and their character positions are also stored for auditability and potential demasking.

Email Classification

After masking, the cleaned email body is passed to a trained classification model. Steps:

1. **Text Vectorization**: Using TfidfVectorizer to convert masked text into feature vectors
2. **Classification**: A Logistic Regression model predicts one of several support categories (e.g., "Billing", "Request", "Technical")
3. **Label Decoding**: Model predictions are mapped back to human-readable labels via a LabelEncoder

This two-step pipeline ensures both data security and accurate classification.

Model Selection and Training Details

- **Model**: Logistic Regression (chosen for interpretability, speed, and reliability)
- **Feature Extraction**: TfidfVectorizer to extract high-dimensional textual features
- **Label Encoding**: Predefined categories were encoded using LabelEncoder
- **Training Data**: A real-world dataset containing support emails across multiple categories with embedded natural PII
- **Evaluation Metrics**: Accuracy, precision, recall, and F1-score used to assess model performance
- **Pipeline Integration**: Training and vectorizer objects are stored and reused in the FastAPI application

Classification Report:

```
              precision    recall  f1-score   support


      Change       0.95      0.78      0.86       479

    Incident       0.69      0.86      0.77      1920

     Problem       0.57      0.32      0.41      1009

     Request       0.89      0.93      0.91      1392


    accuracy                           0.76      4800

   macro avg       0.78      0.72      0.74      4800

weighted avg       0.75      0.76      0.74      4800
```

Challenges Faced and Solutions Implemented

1. Strict Output Format for API

Challenge: The evaluator system required exact key names and structure for the JSON response. **Solution**: Refactored the output dictionary to match format:

```
{
  "position": [start, end],
  "classification": "entity_type",
  "entity": "original value"
}
```

2. Pickle Compatibility with NumPy in Docker

Challenge: Loading locally trained models inside the Docker container caused ModuleNotFoundError due to binary incompatibility. **Solution**: Retrained the model **inside the Docker build step**, ensuring version compatibility.

3. Deployment on Hugging Face Spaces with FastAPI

Challenge: FastAPI is not natively supported in the SDK list. **Solution**: Used the Docker SDK and built a custom container with uvicorn + FastAPI, successfully exposing /docs for live testing.

4. Entity Overlaps in Regex + NER

Challenge: Overlapping entity matches (e.g., names inside emails) were causing conflicts. **Solution**: Entity detection and replacement were executed in order of appearance in the text, with offsets updated on-the-fly to prevent collision.

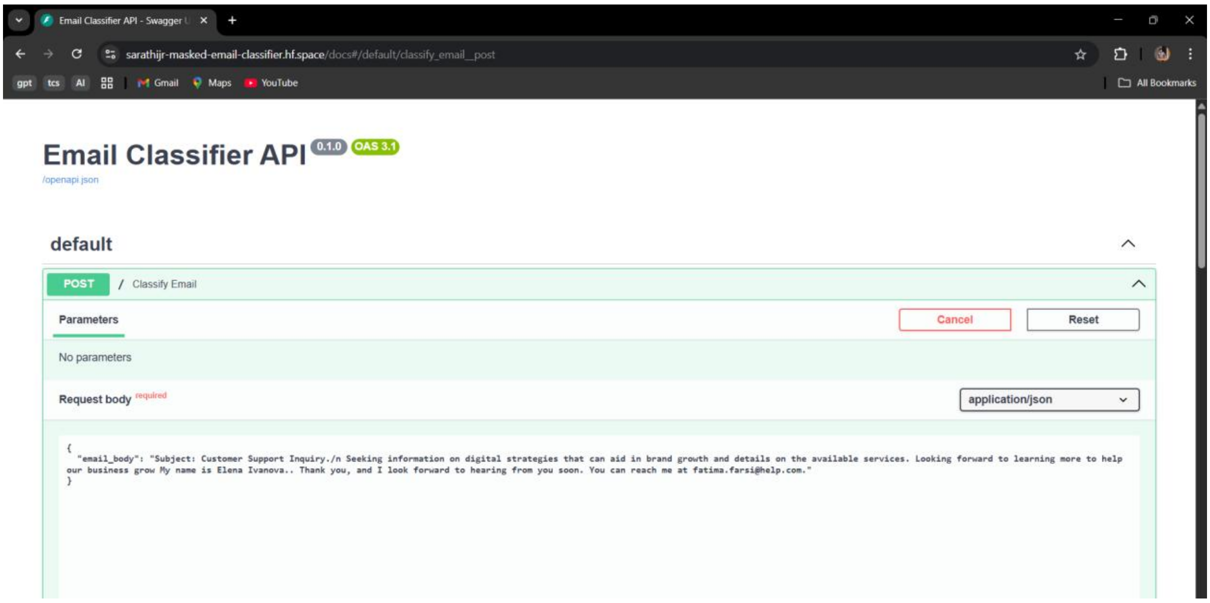
Outcome

This project resulted in a production-ready, API-based solution for secure email classification. It masks sensitive data, classifies content, and returns structured output via a POST endpoint with Swagger UI support for easy testing and evaluation.

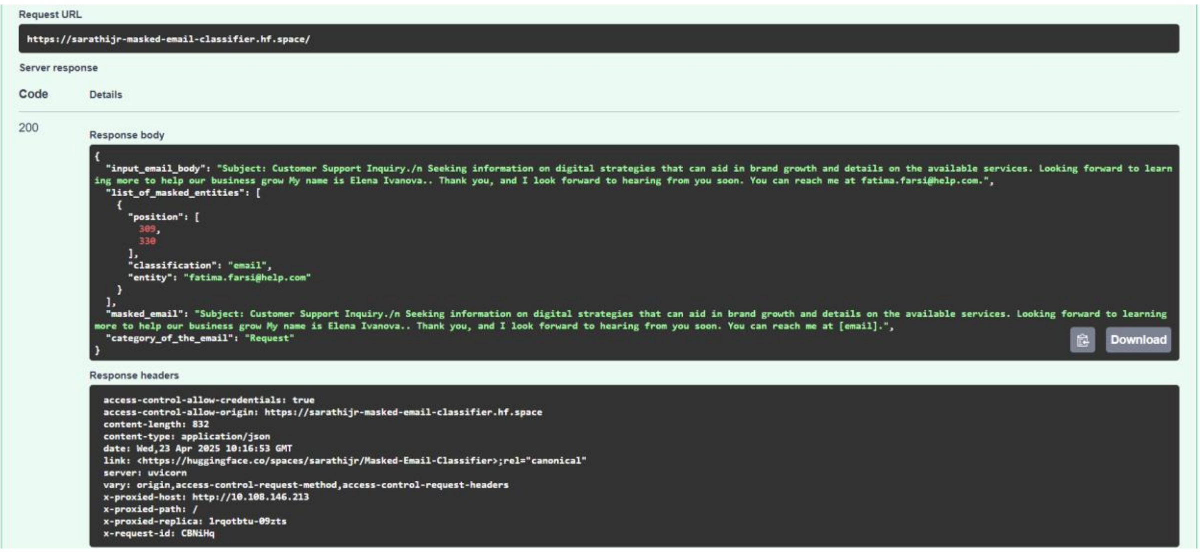
The deployed solution is accessible at:

https://sarathijr--masked-emailclassifier.hf.space/docs

ScreenShots:



```
{
  "email_body": "Subject: Customer Support Inquiry./n Seeking information on digital strategies that can aid in brand growth and details on the available services. Looking forward to learning more to help
our business grow My name is Elena Ivanova.. Thank you, and I look forward to hearing from you soon. You can reach me at fatima.farsi@help.com."
}
```



```
https://sarathijr-masked-email-classifier.hf.space/
```

```
{
  "input_email_body": "Subject: Customer Support Inquiry./n Seeking information on digital strategies that can aid in brand growth and details on the available services. Looking forward to learn
ing more to help our business grow My name is Elena Ivanova.. Thank you, and I look forward to hearing from you soon. You can reach me at fatima.farsi@help.com.",
  "list_of_masked_entities": [
    {
      "position": [
        309,
        330
      ],
      "classification": "email",
      "entity": "fatima.farsi@help.com"
    }
  ],
  "masked_email": "Subject: Customer Support Inquiry./n Seeking information on digital strategies that can aid in brand growth and details on the available services. Looking forward to learning
more to help our business grow My name is Elena Ivanova.. Thank you, and I look forward to hearing from you soon. You can reach me at [email].",
  "category_of_the_email": "Request"
}
```

```
access-control-allow-credentials: true
access-control-allow-origin: https://sarathijr-masked-email-classifier.hf.space
content-length: 832
content-type: application/json
date: Wed,23 Apr 2025 10:16:53 GMT
link: <https://huggingface.co/spaces/sarathijr/Masked-Email-Classifier>;rel="canonical"
server: uvicorn
vary: origin,access-control-request-method,access-control-request-headers
x-proxied-host: http://10.108.146.213
x-proxied-path: /
x-proxied-replica: 1rqotbtu-09zts
x-request-id: CBNiHq
```