# Email Classification Project Report with PII Masking

## Problem Statement

Organizations often receive thousands of emails daily. These emails may contain sensitive personal information and cover a wide variety of intents—ranging from service requests to incident reports. The goal of this project is to build an end-to-end system that:

1. Masks Personally Identifiable Information (PII) like names, emails, phone numbers, card numbers, etc.
2. Classifies the email into one of the predefined categories:
    - Request
    - Incident
    - Problem
    - Change
3. Exposes the functionality via a REST API endpoint using FastAPI.

---

## Tech Stack

| Area | Tools/Technologies |
|---|---|
| Programming | Python 3 |
| ML Model | Multinomial Naive Bayes |
| Text Features | TF-IDF Vectorizer |
| NER | spaCy (en_core_web_sm) |
| PII Masking | Regex + spaCy NER |
| API Framework | FastAPI |
| Serving | Uvicorn (local server) |
| Serialization | Joblib |

---

Dataset Overview

- Source: Provided CSV with 24,000 rows
- Columns: email, type
- Classes: Request, Incident, Problem, Change
- Cleaning Steps:
    - Removed "Subject:" prefix
    - Lowercased text
    - Removed null values
    - Balanced class distribution

---

PII Masking Details

Implemented in utils.py:

- Regex patterns used for:
    - Email addresses
    - Phone numbers
    - Aadhar numbers
    - Card numbers
    - CVV and expiry dates
    - Date of birth
- SpaCy used for:
    - Full name recognition via PERSON entities
- Returns:
    - masked_email
    - list_of_masked_entities (with classification, value, and position)

---

Model Details

- Algorithm: Multinomial Naive Bayes
- Vectorizer: TF-IDF with 5,000 features and English stop words
- Train/Test Split: 80/20
- Evaluation Metrics:
    - Accuracy: 71.5% (original), 77%+ (balanced dataset)
    - Precision, Recall, F1 (see full classification report in logs)

---

API Endpoint

/classify-email (POST)

Request Body:

{

"email_body": "Hi, I'm John Doe. My email is john@example.com. I have a request on..."

}

Response:

{

"input_email_body": "Hi, I'm John Doe. My email is john@example.com.",

"list_of_masked_entities": [...],

"masked_email": "Hi, I'm [full_name]. My email is [email]",

"category_of_the_email": "Request"

}

- Automatically documented in Swagger UI (/docs)
- Fully testable with curl, Postman, browser

---

Challenges Solved

- Regex and NLP combined for accurate PII masking
- Handling unbalanced dataset by upsampling classes
- Creating a production-ready API with error handling
- Maintaining strict JSON output format as per assignment

---

Future Improvements

- Integrate transformer-based models like BERT
- Add logging and monitoring to API
- Dockerize the application
- Add rate limiting, authentication (for public use)
- Extend to multi-language support for global use cases

---

Thank You

Thank you for reviewing this submission. I've focused not only on functionality but also on clean code, documentation, and deployment readiness. Looking forward to your feedback and the opportunity to discuss this further.

*— Allwin Raja*