# Akaike Internship Assignment Report

**Title**: Email Classification for Support Team
**Author**: Thumula Praneeth Rao
**Date**: 21-04-2025

*Contents*

# 1. Introduction

This project aims to automate the classification of support emails into predefined categories like **Incident**, **Request**, **Change**, and **Problem**, while also ensuring the privacy of sensitive user data. The system masks **Personally Identifiable Information (PII)** before any processing, classifies the emails, and returns both the result and masked entities in a structured API response.

The entire pipeline has been implemented in Python, trained on real email data, and deployed as an API on **Hugging Face Spaces**.

# 2. PII Masking Approach

To ensure data privacy and meet compliance requirements, PII masking was implemented using a combination of Regex and SpaCy NER (non-LLM).

Entities Masked:

| Entity Type | Placeholder | Method |
|---|---|---|
| Full Name | [full_name] | SpaCy NER |
| Email Address | [email] | Regex |
| Phone Number | [phone_number] | Regex |
| Date of Birth | [dob] | Regex |
| Aadhar Number | [aadhar_num] | Regex |
| Credit/Debit Number | [credit_debit_no] | Regex |
| CVV Number | [cvv_no] | Regex |
| Expiry Number | [expiry_no] | Regex |

Each PII entity detected is recorded with:

- Original value
- Placeholder type
- Position in the email text

## 3. Model Selection & Training

A **Random Forest Classifier** was selected due to its robustness and interpretability. Email content was vectorized using **TF-IDF**, and the model was trained on a labeled dataset containing natural emails and support categories.

Final Model Pipeline:

- TfidfVectorizer(stop_words='english')
- RandomForestClassifier(n_estimators=100)

Model Performance:

| **Category** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Incident | 0.65 | 0.98 | 0.78 |
| Request | 0.86 | 0.92 | 0.89 |
| Change | 0.96 | 0.55 | 0.70 |
| Problem | 0.91 | 0.12 | 0.21 |
| **Accuracy** | | | **74%** |

## 4. API Development & Deployment

The API was built using **FastAPI** and follows strict JSON output format as required:

```
{
  "input_email_body": "original email",
  "list_of_masked_entities": [
    {
      "position": [start, end],
      "classification": "entity_type",
      "entity": "original_value"
    }
  ],
  "masked_email": "masked version",
  "category_of_the_email": "classified category"
}
```

The API was deployed to **Hugging Face Spaces** and is accessible via:
https://pranee31-emailclassification.hf.space/docs

## 5. Challenges & Solutions

| Challenge | Solution |
|---|---|
| Hugging Face rejecting large files | Used **Git LFS** for model, removed dataset |
| Regex inconsistencies for PII | Wrote flexible, multi-format patterns |
| Class imbalance in dataset | Adjusted training, considered oversampling |
| Build error (numpy vs spacy) | Pinned compatible versions in requirements.txt |

## 6. Conclusion

This assignment provided real-world experience in:

- Data privacy via PII masking
- Building and training NLP models
- API development using FastAPI
- Deployment using Hugging Face Spaces

The solution is modular, scalable, and meets all the assignment requirements.

## Github & Hugging Face Space Links

- **GitHub Repository**: https://github.com/praneeth-rao/email-classifier
- **Deployment**: https://pranee31-emailclassification.hf.space/docs