**EMAIL CLASSIFICATION FOR SUPPORT TEAM**

**Introduction to Problem Statement**

Design an Email Classification System for a Company’s support team. The system should be able to mask the personal information(pii) and categorise the email accordingly for the further processing of the email. After classification the email should return to it’s original format i.e., the masked data should be unmasked and the data should be returned.  
  
**Approach taken for PII masking and Classification**

1. **PII Masking**

A rule based approach was used for PII detection that is Regular Expression(regex).

|  |  |  |
| --- | --- | --- |
| **PII Entity** | **Label** | **Method** |
| Full Name | full\_name | Pattern should match for capitalized words |
| Email | email | Regex match |
| Phone Number | phone\_number | Regex Match with country codes |
| Date of Birth | dob | DD/MM/YY |
| Aadhar card Number | aadhar\_num | 12 digit format |
| Cerdit/Debit number | credit\_debit\_no | 16 digit format |
| CVV number | cvv\_no | 3 / 4 digit format |
| Expiry date | expiry\_date | MM/YY or MM/YYYY |

Each matched entity is masked using a specific entity based placeholder to protect the data and maintain semantic relation.

1. **Email Classification**

I went with the approach of Machine Learning Model with a pipelining system as following:

* Text Vectorization – TF-IDF vectorizer for converting email body into numerical features
* Classifier – Support Vector Machine(SVM)

I used scikit-learn and joblib for the pipelining and deployment

**Model Selection and Training**  
Data Source was the given dataset (combined\_emails\_with\_natural\_pii.csv) which and 2 columns that is email (containing the email body) and the type (label for the category)

1. Preprocessing

Tokenization and lowercasing was done by TF-IDF vectorizer.

1. Training Configuration

Vectorizer : TF-IDF

Model : SVM

Test Split : 80% training and 20% testing

1. Model Saving

Vectorizer + Classifier was saved as svm\_classifier.pkl using joblib

**Challenges and Solutions**

1. Missing or invalid App file

Hugging face showed error “Space is missing an app file”.

Solution:

* ensuring main.py has FastAPI() function.
* Ensuring no conflicts between default app files.

1. Regular Expressing not generalizing

Some PII (dates and phone numbers) varied in format and were missed by early regex

Solution:

* Improving iteratively the expressions to handle common features.
* Added space tolerance and optional country codes.