**Final Project Report: Support Email Classification with PII/PCI Masking**

**Introduction**

This project focuses on the secure processing and classification of support emails. It ensures that all Personally Identifiable Information (PII) and Payment Card Industry (PCI) data is masked before classification. The final output adheres to a strict JSON format, as required by the assignment specifications.

**Objectives**

* Mask PII/PCI entities in email bodies using deterministic pattern recognition.
* Classify masked emails into predefined support categories.
* Output the result in a strict, evaluation-compatible JSON format.
* Build a deployable and testable API and UI using FastAPI and Gradio.

**Methodology**

**PII/PCI Masking:**

A custom masking function was developed using Python's re (regex) module to identify and replace the following entities:

**PII:** full\_name, email, phone\_number, dob, aadhar\_num

**PCI:** credit\_debit\_no, cvv\_no, expiry\_no

**Pattern-Based Detection:**

Regex patterns were built to match:

* Contextual identifiers like "my name is", "reach me at", "my contact number is"
* Common date, card, and number formats
* Name following greetings like "Regards," and "Sincerely,"

**Email Classification:**

Masked emails were vectorized using TfidfVectorizer and classified using a LinearSVC model. This model was chosen over regular SVM due to its superior speed and scalability on large, high-dimensional datasets typical of text classification.

**Output Format:**

Each processed email produces a JSON object containing:

* Original email text
* Masked version of the email
* List of masked entities with positions
* Predicted category of the email

**Deployment:**

The solution was packaged as a FastAPI backend with a Gradio UI for interactive testing and submission via Hugging Face Spaces.

**Challenges Faced**

**Contextual Name Detection:**

* The phrase "my name is" often worked well, but variations or multi-line formats were missed initially.
* Adding logic for names after "Regards," or "Sincerely," required careful newline handling and whitespace trimming.

**Misclassification of CVV:**

* Initially, standalone 3-digit numbers were incorrectly masked as CVVs.
* This resulted in masking contact numbers or product names (e.g., "Norton 360").
* Fix: CVV masking was changed to only occur when "cvv" or related keywords are nearby (within 10-15 words).

**Pattern Interference:**

* Some fields (like phone numbers and card numbers) had overlapping patterns.
* Refined and prioritized the order of regex matching to minimize conflicts.

**Results**

* All entities are masked accurately based on the evaluation criteria.
* The classifier demonstrates reliable performance and acceptable accuracy.
* The API returns output in the exact required format.

**Conclusion**

The solution is secure, scalable, and accurate for email classification with strict masking requirements. The use of deterministic patterns for masking ensures compliance, and LinearSVC offers a balanced performance for classification. The solution is ready for real-world evaluation or production usage.

**Recommendations**

* Extend masking logic to support more natural language variations.
* Consider integrating Named Entity Recognition (NER) for advanced masking (if allowed).
* Fine-tune the classifier with more labeled examples for better accuracy.