Email Classification System Report

I. OBJECTIVE

The objective of this project was to design and implement an email classification system for a company's support team. The system categorizes incoming support emails into predefined categories while ensuring that personally identifiable information (PII) and Payment Card Industry (PCI) data are masked before processing. After classification, the masked data is restored to its original form for output.

II. PROBLEM STATEMENT

The system was required to include the following key functionalities:

1. Email Classification: Develop a model to classify support emails into categories such as Change, Incident, Problem, and Request.
2. Personal Information Masking (Without LLMs): Mask PII and PCI data (e.g., full name, email, phone number, date of birth, Aadhar card number, credit/debit card number, CVV, and card expiry) using non-LLM methods like Named Entity Recognition (NER), regular expressions (Regex), or custom techniques.
3. API Deployment: Expose the solution as an API that accepts an email, masks PII, classifies the email, and returns the category and demasked email.

-Submission by Shawn Thomas

III. METHODOLOGY

1. Data Collection & Preprocessing

- **Dataset**: The provided dataset (emails_type.csv) contains emails labelled with four categories: **Incident (9586), Request (6860), Problem (5037), and Change (2517**). The dataset was loaded and explored using Pandas in the explore_dataset.ipynb notebook, confirming no missing values and UTF-8 encoding.
- **PII Masking**: The utils.py module implements PII masking using SpaCy for date of birth (dob) detection and Regex for other entities (full_name, email, phone_number, aadhar_num, credit_debit_no, cvv_no, expiry_no). The mask_pii function:
    - Cleans the email by removing subject lines and normalizing whitespace.
    - Uses SpaCy's NER to detect dates, filtering for valid date formats to avoid misclassifying card numbers.
    - Applies Regex patterns with context awareness (e.g., "My name is" for full names) to detect and mask other PII entities.
    - Stores entity details (position, classification, entity value) for later demasking.
    - Example:
        - Input: "My name is John Doe, contact me at +971-50-123-4567."
        - Masked: "My name is [full_name], contact me at [phone_number]."
- **Data Storage**: The original email and masked entities are stored securely in memory during processing, enabling demasking via the demask_email function, which restores the original email by replacing placeholders with stored entity values.
-

2. Model Selection & Training

- **Model Choice**: A BERT-based deep learning model (**bert-base-multilingual-cased**) was selected for classification, as implemented in models.ipynb. BERT was chosen for its ability to handle multilingual text and contextual understanding, suitable for the diverse email dataset.
- **Preprocessing**:
    - Emails were masked using the mask_pii function and cleaned (subject lines removed, whitespace normalized).

  - The dataset was split into 80% training and 20% testing sets, with stratification to maintain class distribution.
  - Labels were encoded using LabelEncoder (Change: 0, Incident: 1, Problem: 2, Request: 3).
- **Training**:
  - The BERT model was fine-tuned with a custom classification head for four labels.
  - Class weights were computed to address class imbalance and applied via a weighted CrossEntropyLoss.
  - Training used the AdamW optimizer, a linear learning rate scheduler, and a batch size of **16 over 5 epochs.**
  - A WeightedRandomSampler ensured balanced sampling during training.
- **Evaluation**:
  - The model achieved an overall accuracy of **78%** on the test set.
  - Classification report:
    - Change: Precision 0.90, Recall 0.94, F1-score 0.92
    - Incident: Precision 0.78, Recall 0.72, F1-score 0.75
    - Problem: Precision 0.53, Recall 0.61, F1-score 0.57
    - Request: Precision 0.94, Recall 0.94, F1-score 0.94

The lower performance on the Problem category reflects its smaller sample size and potential overlap with Incident emails.

3. System Integration

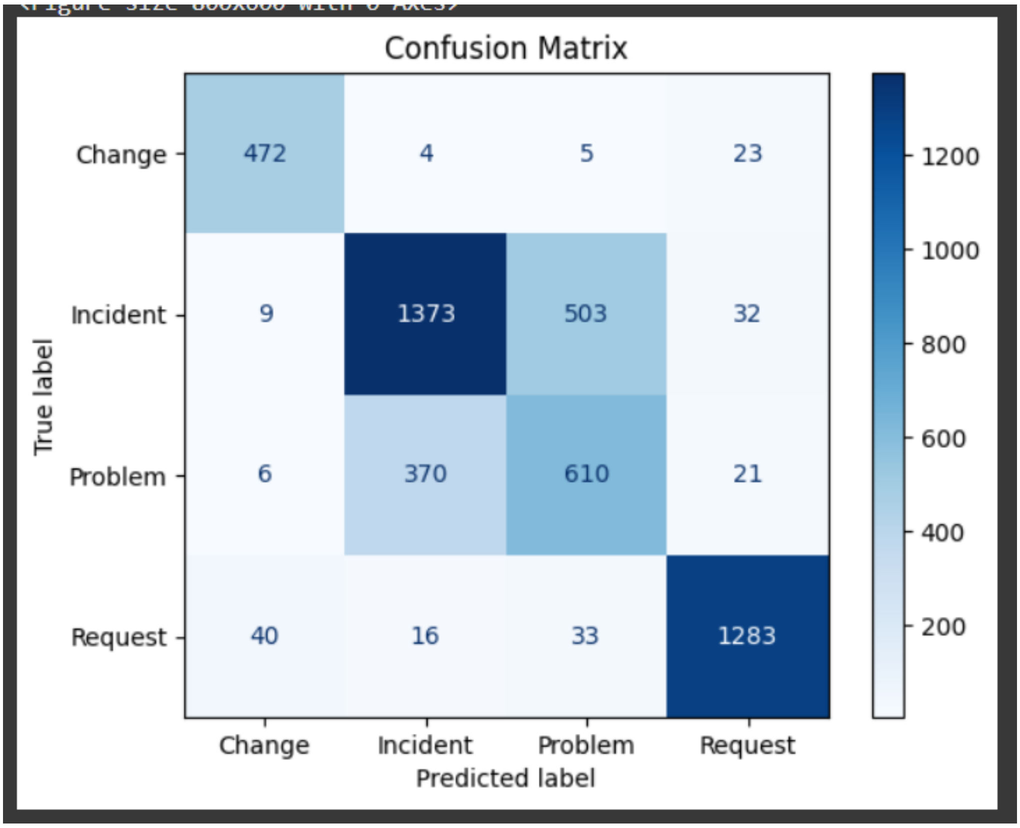
- **Pipeline**:
  - **Input**: Emails are received via the API or web form.
  - **Masking**: The mask_pii function processes the email to mask PII, storing entity details.
  - **Classification**: Masked emails are tokenized using BERT's tokenizer and passed to the fine-tuned BERT model for classification.
  - **Output**: The system returns the original email, masked email, list of masked entities, and the predicted category.

IV. CHALLENGES AND OBSERVATIONS

- **PII Masking**: Regex patterns required careful tuning to avoid false positives (e.g., distinguishing credit card numbers from other numeric sequences). SpaCy's date detection was effective but needed filtering to exclude card-like patterns.
- **Class Imbalance**: The dataset's imbalance (e.g., fewer Change emails) was mitigated using class weights and weighted sampling, but the Problem category's lower F1-score suggests room for improvement, possibly with more data or feature engineering.
- **Model Performance**: BERT's multilingual capabilities were beneficial, but the 78% accuracy indicates potential for further fine-tuning or experimenting with other models (e.g., RoBERTa, traditional ML like SVM).
- **API Robustness**: The API handles malformed JSON and encoding issues gracefully, but scalability for high-throughput environments would require additional optimization (e.g., asynchronous processing, model caching).

V. SCREENSHOTS

```
Epoch 1
Epoch 1: 100%|██████████| 1200/1200 [14:10<00:00,  1.41it/s, loss=0.34]
Epoch 2
Epoch 2: 100%|██████████| 1200/1200 [14:16<00:00,  1.40it/s, loss=0.283]
Epoch 3
Epoch 3: 100%|██████████| 1200/1200 [14:15<00:00,  1.40it/s, loss=0.294]
Epoch 4
Epoch 4: 100%|██████████| 1200/1200 [14:16<00:00,  1.40it/s, loss=0.141]
Epoch 5
Epoch 5: 100%|██████████| 1200/1200 [14:16<00:00,  1.40it/s, loss=0.258]
```



- The confusion matrix evaluates the BERT model's performance in classifying emails into four categories: *Change, Incident, Problem, and Request*.
- **Strong accuracy for Change with 472 out of 504** correctly classified (93.7% recall) and minimal misclassifications (e.g., 23 as Request).
- **High accuracy for Request with 1283 out of 1372** correctly classified (93.5% recall) and few errors (e.g., 40 as Change).
- Significant **confusion between Incident and Problem, with 503 Incident emails misclassified** as Problem and 370 Problem emails as Incident.
- Incident recall is 71.6% (1373/1917 correct), while Problem recall is 60.6% (610/1007 correct), indicating challenges in differentiation.
- **Likely causes of confusion include overlapping features or data imbalanc**e (1917 Incident vs. 1007 Problem emails).

-Submission by Shawn Thomas

Email Classification System

```
Subject: Verbesserung des Projekt-Managements mit Ubuntu 20.04 LTS und Redis 6.2

Sehr geehrte Kundenservice, ich schreibe in diesem Brief, um zu erkundigen, wie ich Ubuntu 20.04 LTS mit Redis 6.2 integrieren kann, um die
Skalierbarkeit und Leistung des Projekt-Managements zu verbessern.Ich denke, dass diese Integration unser Team in der Lage setzen wird, grÃ¶ÃŸere
Workloads zu verarbeiten und unseren Kunden bessere Dienstleistungen anzubieten. KÃ¶nnten Sie mir bitte mehr Informationen dazu geben, wie ich
diese Integration umsetzen kann? Ich wÃ¼rde jede von Ihnen bereitgestellte Ratgebersupport oder Ressourcen sehr schÃ¤tzen You can reach me at
david.kim@corp.kr.. ZusÃ¤tzlich wÃ¼rde ich gerne wissen, ob es bestimmte Systemanforderungen oder potenzielle Herausforderungen gibt, die ich
beachten sollte. Ich freue mich darauf, von Ihrer UnterstÃ¼tzung zu hÃ¶ren. Vielen Dank fÃ¼r Ihre Zeit und Ihre Hilfe. Ich freue mich darauf, von
Ihnen bald zu hÃ¶ren. Mit freundlichen GrÃ¼ÃŸen, [Ihr Name] My name is Elena Ivanova.
```

Classify Email

Result:


```
{
  "input_email_body": "Subject: Verbesserung des Projekt-Managements mit Ubuntu 20.04 LTS und Redis 6.2\r\n\r\nSehr geehrte Kundenservice, ich schreibe in d
  "list_of_masked_entities": [
    {
      "position": [
        930,
        943
      ],
      "classification": "full_name",
      "entity": "Elena Ivanova"
    },
    {
      "position": [
        571,
        590
      ],
      "classification": "email",
      "entity": "david.kim@corp.kr.."
    }
  ],
  "masked_email": "Sehr geehrte Kundenservice, ich schreibe in diesem Brief, um zu erkundigen, wie ich Ubuntu 20.04 LTS mit Redis 6.2 integrieren kann, um d
  "category_of_the_email": "Change"
}
```


Email Classification System

```
Subject: Security Measures for Medical Data in Hospital Infrastructure

Could you provide detailed solutions for securing medical data within our
hospital's infrastructure?. My contact number is +1-555-123-4567.
```

Classify Email

Result:


```
{
  "input_email_body": "Subject: Security Measures for Medical Data in Hospital Infrastructure\r\n\r\nCould you provide detailed solutions for securing medical data
  "list_of_masked_entities": [
    {
      "position": [
        126,
        138
      ],
      "classification": "phone_number",
      "entity": "555-123-4567"
    }
  ],
  "masked_email": "Could you provide detailed solutions for securing medical data within our hospital's infrastructure?. My contact number is +1-[phone_number].",
  "category_of_the_email": "Request"
}
```