INTERNSHIP TASK REPORT

1. Introduction to the Problem Statement:

- **Problem:**

  Manually managing reports while in the presence of sensitive PII is a huge problem as well as managing the wide range of categories of email is a challenging task.

- **Goal:**

  To solve that problem I had Developed a Automated Email Classification System Where Privacy Wont be a concern and the Personal Information called PII is masked Immediately And Security is a Main Concern To Many People So Masking and Demasking is Done on a API Categorization Analysis and Post Output Application is Achieved.

- **Constraint:**

  No LLMs is used for PII masking.

- **Benefits:**

  Efficiency, privacy, compliance.

2. Approach Taken (PII Masking & Classification)

- **PII Masking & Demasking (in utils.py):**
  - **Method:**

    Used Regular Expressions (Regex) for Masking the Personal Information.

  - **PII Types Handled:**

    Full Name, Email, Phone Number, Date of Birth, Aadhar Card Number, Credit/Debit Card Number, CVV Number, Card Expiry Number.

  - **Process:**

    Used Regex patterns to the identify, extract the entity + position then, replace with placeholder (e.g., [email])…

  - **Key Techniques:**
    - Overlap Resolution: Prioritize longer/more specific regex matches to avoid splitting PII.
    - False Positive Prevention: Refined the use of regexes for CVV (context-dependent like "CVV 123"), Card Expiry (strict year format), and ensured non-PII (like IP addresses, arbitrary dates) are not masked.

- Demasking: Reconstructs original email using stored Masked Data and positions.

- **Email Classification (in models.py):**
  - **Data Preperation:**

    PII masking is applied to emails before the classification.

  - **Feature Extraction:**

    TF-IDF Vectorization (with max_features=5000 and N-grams ngram_range=(1,2) for combining words and finding vectors is performed).

  - **Class Imbalance:**

    SMOTE oversampling is used on training the data to balance the classes so Training can be done properly with no oversampled majority output of one category.

  - **Model:**

    The Random Forest Classifier (n_estimators=100, class_weight='balanced'). Is Used as The Naïve Bayes, SVM, Output Accuracy was low During Testing Phase

3. Model Selection and Training Details:

- **Dataset:**

  The DATASET Used is combined_emails_with_natural_pii (1).csv (24,000 entries).

- **Preprocessing:**

  After the Emails are masked, then the vectorized (TF-IDF), categories label-encoded. The Stop-word removal in TF-IDF was made language-agnostic This is done so that The Main words are given more Priority and the Vectorizer using the n-grams does the Job pretty well for processing while training accurately.

- **Data Split:**

  80% Training and 20% Testing is used.

- **Training Process:**

  TF-IDF fitted on masked training data; SMOTE applied to training data; RandomForestClassifier trained on resampled data.

- **Evaluation:**

  The final Achieved [Your final accuracy from models.py output] on test set on the trained dataset y_pred score.

4. Challenges Faced and Solutions Implemented

- **PII Masking (False Positives/Overlaps):**

    1. **Challenge:**

        The IPs/arbitrary numbers were getting masked with CVV/Expiry dates.

        Solution:

        Continuously Updated the cvv_pattern (context-dependent) and

        The card_expiry_pattern (strict year format) in utils.py.

    2. **Challenge:**

        There was Overlapping matches for Credit Card/Aadhar.

        Solution:

        Implemented a better overlapping resolution logic for mask_email.

- **Demasking Accuracy:**

    1. **Challenge:**

        Reconstruction had to be done as errors due to varying lengths of original PII vs. the original placeholders.

        Solution:

        Used string.replace(..., 1) method with reverse sorting for better demasking.

- **Environment Setup:**

    1. **Challenge:**

        uvicorn, imblearn, fastapi ModuleNotFoundError locally version management is the biggest problem.

        Solution:

        Setted up/activated virtual environment (venv),and managed requirements.txt.

- **Hugging Face Deployment:**

    1. **Challenge:**

        Matching the correct version for deploying in hugging face was the biggest challenge because proper versions were required.

        Solution:

        After lot of trail and error debugging code I was able to do it.

2. **Challenge:**

   I had used Fast Api for deployment and checking different server 500 errors and deployment failed.

   Solution:

   Then I removed all spacy-related code from app.py's startup and used nlp and regular expresions to tackle the n-grams and redeploy it.

- **Model Accuracy Improvement:**

  1. **Challenge:**

     Aim for higher classification accuracy.

     Solution:

     The Integration of N-grams (ngram_range=(1,2)) into TF-IDF and used RandomForestClassifier made the model powerfuk(more powerful ensemble model).

5. Final Output: API Endpoint Details for Testing

- **Deployed API Endpoint:**

  https://chaitanyasaikumar-email-classifier-internship.hf.space/classify

- **Expected Request Body (JSON) Tested via Postman with POST Request:**

  Content-Type: application/json

JSON

{ "input_email_body": "string containing the email" }

- **Expected Response Body (JSON):**

JSON

```
{
  "input_email_body": "string containing the email",
  "list_of_masked_entities": [ { "position": [start, end], "classification": "entity_type", "entity": "original_entity_value" } ],
  "masked_email": "string containing the masked email",
  "category_of_the_email": "string containing the class"
}
```

Example Output:


```
{
  "input_email_body": "Subject: Critical System Outage - Unable to Login\n\nDear Support,\n\nMy name is Emily White. The main production system has been down since 10:00 AM IST. I cannot log in, and it's affecting all operations. My user ID is EW-1234. I'm at 123 Main Street. Please investigate this urgent issue immediately. My credit card is 4567 8901 2345 6789, CVV 567, expiry 05/27. My contact email is emily.w@business.com.",
  "list_of_masked_entities": [
    {
      "position": [382, 402],
      "classification": "email",
      "entity": "emily.w@business.com"
    },
    {
      "position": [355, 360],
      "classification": "card_expiry_number",
      "entity": "05/27"
    },
    {
      "position": [343, 346],
      "classification": "cvv_number",
      "entity": "567"
    },
    {
      "position": [318, 337],
      "classification": "credit_debit_card_number",
      "entity": "4567 8901 2345 6789"
    },
    {
      "position": [77, 88],
      "classification": "full_name",
```

"entity": "Emily White"

}

],

"masked_email": "Subject: Critical System Outage - Unable to Login\n\nDear Support,\n\nMy name is [full_name]. The main production system has been down since 10:00 AM IST. I cannot log in, and it's affecting all operations. My user ID is EW-1234. I'm at 123 Main Street. Please investigate this urgent issue immediately. My credit card is [credit_debit_card_number], CVV [cvv_number], expiry [card_expiry_number]. My contact email is [email].",

"category_of_the_email": "Incident"

}

Followed these guidelines

- **Example Test Command of (PowerShell):**

PowerShell

```
$body = @{

  input_email_body = "Subject: Critical System Outage - Unable to Login`n`nDear Support,`n`nMy name is Emily White. The main production system has been down since 10:00 AM IST. I cannot log in, and it's affecting all operations. My user ID is EW-1234. I'm at 123 Main Street. Please investigate this urgent issue immediately. My credit card is 4567 8901 2345 6789, CVV 567, expiry 05/27. My contact email is emily.w@business.com."

} | ConvertTo-Json

Invoke-RestMethod -Uri "https://chaitanyasaikumar-email-classifier-internship.hf.space/classify" -Method POST -Headers @{"Content-Type" = "application/json"} -Body $body | ConvertTo-Json -Depth 10
```

- **Interactive API Documentation:**

  https://chaitanyasaikumar-email-classifier-internship.hf.space/docs