**ASSIGNMENT REPORT**

**INTRODUCTION - UNDERSTANDING THE PROBLEM STATEMENT**

The problem statement given was to develop an **email classification system** with key functionalities centered around automating the categorization of support emails and ensuring privacy through personal information masking. The first part of the requirement involved building a model that could accurately classify incoming support emails into different predefined categories such as *Incident, Request*, etc. For this, I was free to choose from a variety of modeling approaches including traditional machine learning, deep learning, or even Large Language Models (LLMs).

The second part of the task was focused on **Personal Information Masking**, where all Personally Identifiable Information (PII) and Payment Card Industry (PCI) data needed to be identified and masked **without using LLMs**. This included entities like full names, email addresses, phone numbers, dates of birth, Aadhar numbers, card numbers, CVV numbers, and card expiry details. The masking had to be performed using methods such as regular expressions, Named Entity Recognition (NER), or other custom approaches excluding LLMs. For example, an input like “Hello, my name is John Doe, and my email is johndoe@example.com” should be transformed to “Hello, my name is [full\_name], and my email is [email].”

**THOUGHT PROCESS:**

**Approach taken for PII masking and classification:**

For the first part of the project, the goal was to effectively mask all sensitive information (PII and PCI data) from the incoming email content. Since the problem statement explicitly mentioned that Large Language Models (LLMs) should not be used for this task, I focused on leveraging traditional techniques like **regular expressions (regex)** to detect and mask sensitive information. I started by identifying the types of data that needed to be masked, including full names, email addresses, phone numbers, credit card details, and more. By designing specific regex patterns to match these data types (e.g., email addresses, phone numbers, Aadhar numbers), I could easily replace the sensitive data with placeholders such as [full\_name], [email], and [credit\_debit\_no]. This allowed me to preprocess the emails efficiently while ensuring that the sensitive information was completely removed before further processing.

**Model Selection and Training Details**

For the email classification task, the primary challenge was to handle both the multilingual nature of the dataset and the complexity of semantic relationships within the content. Given that the dataset contained a combination of English and German emails, traditional machine learning models like SVM or Naive Bayes were less suitable, as they rely on vectorization techniques like TF-IDF that may not capture the subtleties of text in multiple languages. Additionally, these models struggle with understanding the contextual meaning of words across different languages and contexts, which is critical for accurate classification.

To overcome this limitation, I selected XLM-RoBERTa, a state-of-the-art multilingual transformer model. XLM-RoBERTa is an extension of the RoBERTa model, which has been trained on multiple languages and is highly effective in tasks such as text classification, sentiment analysis, and question-answering. The model's transformer architecture excels at capturing the semantic relationships and contextual understanding within text, making it an ideal choice for this task. Unlike traditional models, XLM-RoBERTa doesn't rely on predefined word vectors but instead learns contextual embeddings for each word in relation to its surrounding words, allowing it to better understand the meaning behind sentences and phrases in both English and German.

To train the model, I used **Hugging Face’s transformers and datasets libraries**, which made the process efficient and seamless. The dataset was preprocessed to remove PII information, and the cleaned text data was then tokenized for input into the model. The model was fine-tuned on the labeled email dataset, where the goal was to optimize its ability to classify the emails into their appropriate categories. Fine-tuning was done using a classification head added on top of the pre-trained XLM-RoBERTa model, which was trained to predict the category label based on the input text. The model's performance was evaluated using standard metrics such as accuracy, precision, recall, and F1-score to ensure that it could classify the emails accurately.

In the training process, special attention was given to handling the multilingual aspect of the data and ensuring that the model could capture the context and intent of the emails, regardless of whether they were written in English or German. This allowed the model to make accurate predictions, even when the email content switched languages or contained a mixture of languages.

**CHALLENGES FACED**

1. **PII Masking Using Regex**: While regex was effective for most of the entities, I faced some challenges when masking certain types of PII, particularly **full names**. The regex pattern I used to detect names was based on capitalized words, but this led to issues in some cases. For instance, other words in the email, such as proper nouns or places, were also capitalized at the beginning, causing false positives where non-name entities were incorrectly identified as full names. This made it tricky to accurately mask names in such cases. Refining the regex patterns to handle these edge cases proved to be a challenge and required additional logic to differentiate between actual names and other capitalized words.
2. **API Deployment**: Deploying the model as an API was another significant challenge, particularly since this was my first time working with **Hugging Face's model deployment**. While Hugging Face provides great tools for training and fine-tuning models, I faced some difficulties in setting up the API for deployment. Issues arose around packaging the model and ensuring it was accessible through the API. I had to deal with configuring the model server and handling model inference in an efficient way, which required learning how to work with **FastAPI** and **Hugging Face's Inference API**. Although Hugging Face simplifies many aspects of model deployment, I still encountered challenges with making sure the API responded quickly and managed resources well for real-time email classification.

**CONCLUSION**

In this project, I built an email classification system that masks sensitive personal information and classifies the email into the correct category. I started by understanding the complete pipeline — first masking PII using regular expressions without using any LLMs, and then classifying the cleaned content using a multilingual transformer model. For classification, I chose XLM-RoBERTa because it could handle both English and German emails and understand the meaning behind the text.

Throughout the project, I learned a lot, especially when it came to handling edge cases in PII masking and deploying the model as an API for the first time. It was a great learning experience and gave me a better understanding of how to handle real-world text data.

Thank you for giving me this opportunity. It really helped me grow my skills and confidence in solving end-to-end machine learning problems.