Email Classification for Support Team

Submitted by: Vinay Veeragani

Date: 22nd April 2025

1. Introduction

In today's digital age, organizations receive thousands of customer emails that need to be sorted, categorized, and processed efficiently. These emails may contain sensitive personal information, requiring secure handling to ensure user privacy. This project focuses on automating the classification of emails into meaningful categories and detecting Personally Identifiable Information (PII) using state-of-the-art Natural Language Processing (NLP) techniques. It reduces manual effort, ensures data privacy, and enhances the overall workflow of support teams.

2. Objective

The primary goals of the project are:

- Automatically classify emails into categories like Incident, Request, Problem, or Change.

- Detect and mask PII such as names, email addresses, phone numbers, and other sensitive entities.

- Provide a simple, fast, and secure API and UI interface to access these functionalities.

3. Methodology

The project adopts a pipeline approach involving PII detection, masking, embedding generation using SBERT, and email classification using machine learning models.

➤ PII Detection

A hybrid approach was used for PII detection. spaCy's Named Entity Recognition (NER) was utilized to detect person names, while regular expressions were

employed to identify email addresses, card numbers, dates of birth,CVV numbers etc.. Overlapping entities were handled by prioritizing longer matches to avoid partial masking.

➤ PII Masking

Once detected, all PII entities were replaced in the email body with tags like [email], [full_name], [phone_number], ensuring the structure and meaning of the message remain intact while sensitive data is obfuscated.

➤ Embedding Generation

Textual data, especially the masked email content, was converted into dense numerical vectors using the pre-trained Sentence-BERT model `all-mpnet-base-v2`. This model captures the semantic meaning of text and offers superior performance on classification tasks compared to traditional TF-IDF vectors.

➤ Classification Models

Three classification models were trained using the SBERT embeddings: Logistic Regression, Random Forest, and XGBoost. Each model was evaluated based on its accuracy and F1-score on the test dataset. XGBoost showed the best performance in terms of overall accuracy and class-wise F1 balance.

4. Results

The dataset consisted of approximately 24,000 email samples distributed across four classes. After training and evaluation, the following results were obtained:

• Logistic Regression:

- Accuracy: 69.5%

- F1 Score: 0.70

• Random Forest:

- Accuracy: 73.2%

- F1 Score: 0.68

• XGBoost:

- Accuracy: 77.0%

- F1 Score: 0.76

Given its superior balance across metrics, the XGBoost model was chosen for deployment.

5. Challenges & Solutions

- Overlapping Entities:When multiple entities overlapped, priority was given to longer matches using sorted post-processing.
- International Phone Number Detection : Initially, there were challenges in identifying international phone numbers accurately.This was addressed by refining regular expressions to handle a wide range of international formats.
- Full Name Detection: The basic spaCy model was insufficient for precise full name recognition. To improve this, we upgraded to the `en_core_web_md` (medium) model which enhanced detection accuracy.
- Model Training Time: Using Hugging Face transformer models like BERT resulted in long training durations due to the dataset's large size (23,000+ rows).To overcome this, I switched to SBERT embeddings combined with traditional machine learning classifiers like XGBoost, which significantly reduced training time while maintaining high accuracy."
- Performance Bottlenecks: SBERT embedding generation was time-consuming. Solution: embeddings were cached and models saved for reuse.