**Email Classification and Data Extraction for Commercial Bank Lending Service Team**

**Overview**

This project automates the classification of servicing requests received via email by a Commercial Bank Lending Service Team. The solution reads and interprets both email content and attachments, identifying the intent of the email and classifying the request into predefined request types and sub-request types. Additionally, it extracts key attributes required for service request automation.

**Problem Statement**

* The bank's servicing requests arrive via email, often with attachments.
* The manual classification and data extraction process is time-consuming and prone to errors.
* The goal is to automate email classification and attribute extraction using Generative AI (LLMs), improving efficiency, accuracy, and turnaround time.
* The extracted information will be used to populate service requests, which will later be routed to the correct processors using skill-based routing.

**Solution Approach**

* Fetch emails from the inbox using **IMAP**.
* Extract email content and process attachments (PDFs, Word documents, Excel files).
* Use **OpenAI's GPT-3.5-turbo** to classify emails into predefined request types and sub-request types.
* Extract key attributes such as **Customer Name, Loan ID, Amount, Request Date**, etc.
* Output structured data for further processing and automation.

**Technology Stack**

* **Python** (Primary Language)
* **IMAP** (Email retrieval)
* **BeautifulSoup** (HTML parsing)
* **pdfminer** (Extracting text from PDFs)
* **python-docx** (Extracting text from Word documents)
* **pandas** (Processing Excel files)
* **OpenAI API** (LLM-based classification and data extraction)

**Setup Instructions**

1. Clone the repository:
2. git clone https://github.com/your-repo/email-classification.git
3. cd email-classification
4. Install required dependencies:
5. pip install imapclient beautifulsoup4 pdfminer.six python-docx pandas openai
6. Update the **configuration** in the script:
   * Set your **email credentials** in EMAIL\_ACCOUNT and EMAIL\_PASSWORD.
   * Set your **OpenAI API key** in OPENAI\_API\_KEY.
7. Run the script:
8. python Everest\_hackathon2025.py

**How It Works**

1. The script connects to an email inbox and retrieves unread emails.
2. It extracts email content and processes attachments.
3. The extracted data is sent to OpenAI's API for classification and attribute extraction.
4. The structured data is printed or stored for further automation.

**Future Enhancements**

* Integrate with a **Service Request Management System**.
* Implement **Skill-Based Routing** for auto-assigning requests.
* Support additional attachment formats (e.g., CSV).
* Improve classification accuracy with **custom fine-tuned models**.

**Contributors**

* [Your Name]
* [Your Team Members]

Let me know if you need any modifications! 🚀