“Open AI With Python”

TASK REPORT

Submitted by

ASHISH KISHORBHAI PATEL

ABSTRACT

OpenAI, a leader in artificial intelligence research, develops advanced AI models like GPT (Generative Pre-trained Transformer), known for their natural language understanding and generation capabilities. Python, a versatile and easy-to-learn programming language, serves as an ideal tool to interact with OpenAI's APIs, allowing developers to leverage these powerful AI models in their applications. By integrating OpenAI's language models through Python scripts, developers can perform tasks such as text classification, summarization, and content generation. This synergy between OpenAI and Python facilitates the creation of intelligent systems that can understand and respond to human language effectively. The combination of OpenAI's cutting-edge AI technology and Python's simplicity and extensive libraries makes it accessible for developers to build innovative solutions across various domains.

Table of Contents

Chapter 1: Overview of Task

1.1 Introduction

Our company is developing a new service product designed to assist consumers with bulk a email sending: Service to enable clients to send large volumes of emails up to millions hourly, Managing Mail Transfer Agent (MTA) products Providing ongoing management and maintenance of MTA systems to ensure optimal performance, handling configurations, updates, and troubleshooting for MTA products, optimizing email delivery, and overall email system management. Assessment and Monitoring: Continuous assessment of email system performance.

The primary goal is to set up in-house service on client's setup to send Millions of emails hourly efficiently and securely.

Your task is to identify and document companies that offer similar services, focusing on those that provide services rather than just products.

1.2 Objective

- Identify competitors offering similar email services.
- Gather detailed information about each competitor, including company name, website, location (state, city etc.), and a comprehensive list of their services.
- Focus on service companies (like mail monster) or those offering both products and
- services (post mastery service subsection), but NOT product-only companies.
- Do not list the product companies which are email based.
- Raw email services companies or Product + Services.

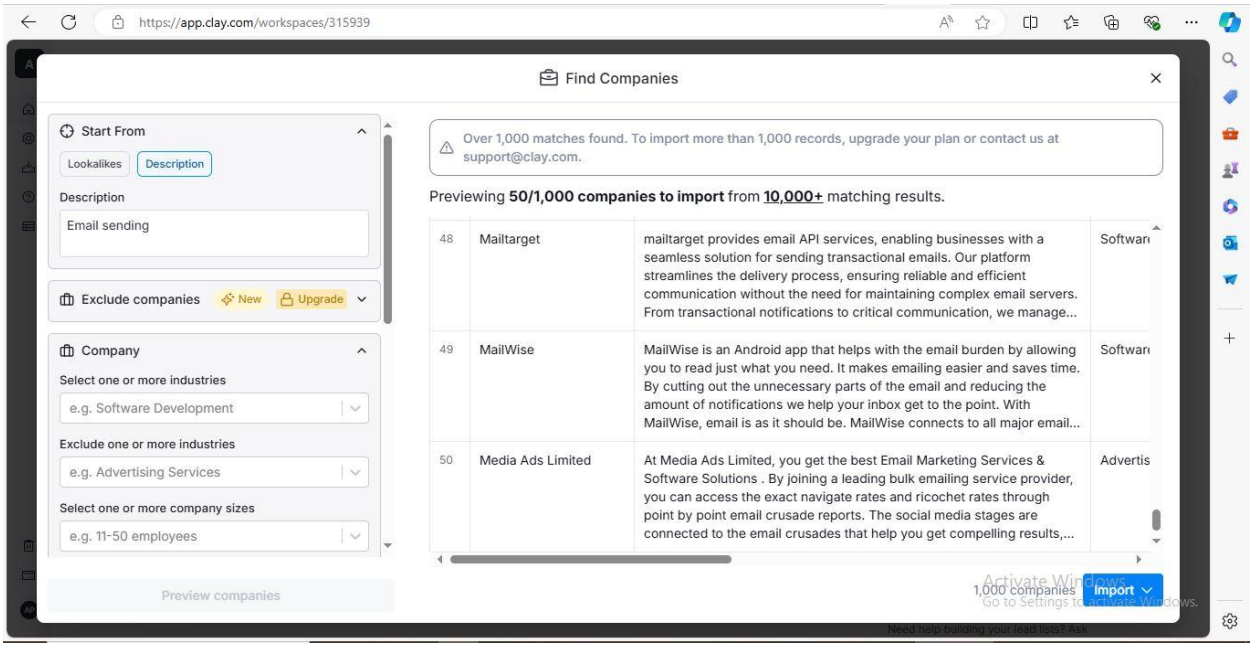
1.3 Tools and Technology Used

- Open AI
- Python For Scripting
- Clay for Data Set
- VS Code
- Jupiter Notebook and Script

Chapter 2: Data Collection

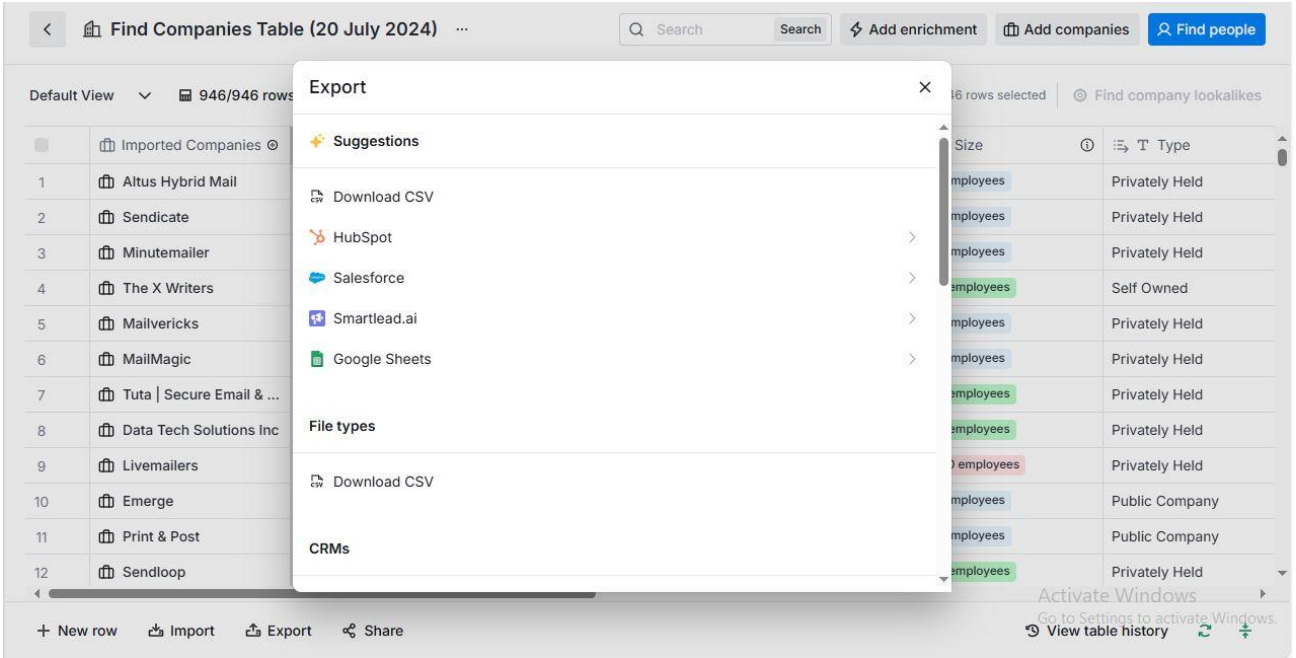
2.1 Using Clay

- Find Competitors' Services and their details (name, location etc.)
- Use Clay to collect the information. After creating a free account, create a new table under find new companies in workspace and here description “Email Sending” Key word used and get List**.**



2.1 Filter Data and Saved as a CSV File

After completing fetching 1000’s of competitors’, download them via CSV by clicking export.



Chapter 3: Open AI

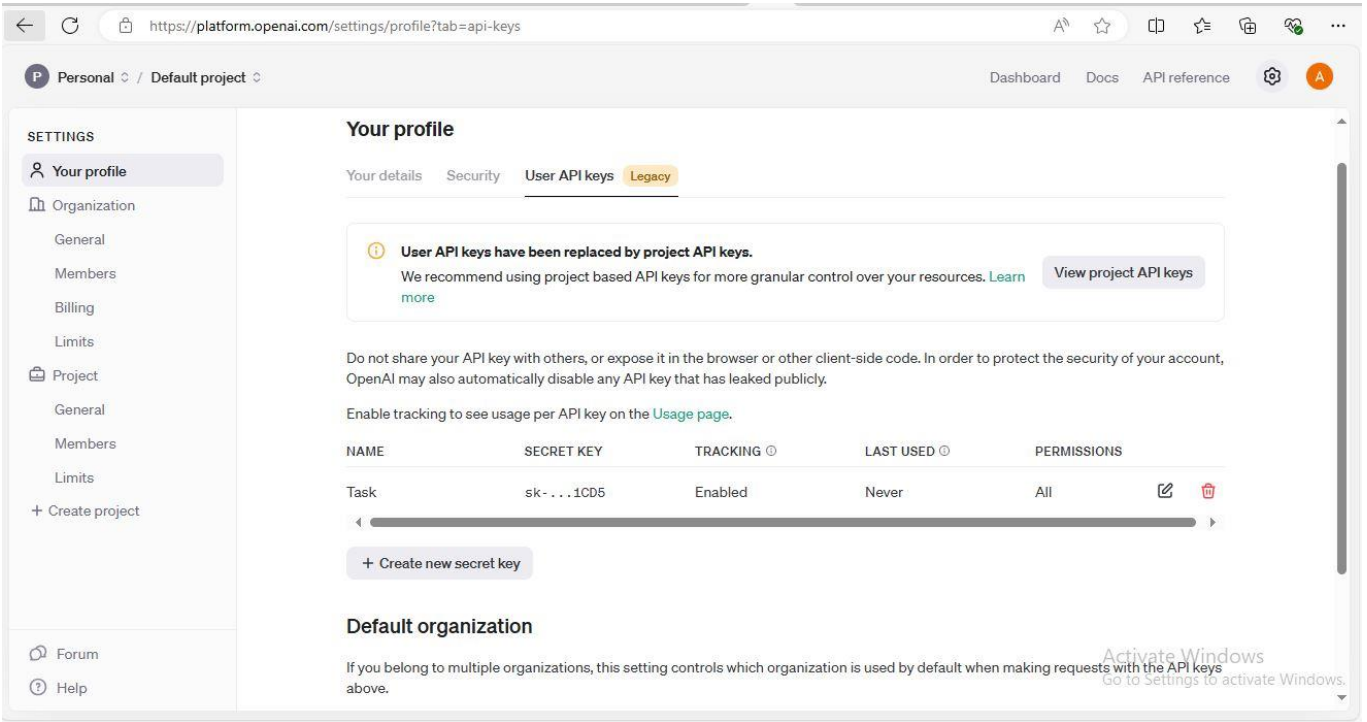
3.1 Overview:

OpenAI is a leading artificial intelligence research lab that develops advanced AI models such as GPT (Generative Pre-trained Transformer). These models excel in natural language understanding and generation, enabling tasks like text completion, translation, and summarization. OpenAI's technology is utilized across various applications, from chatbots to automated content creation. They offer APIs that developers can integrate into their own software to leverage the power of AI. OpenAI is committed to ensuring that artificial intelligence benefits all of humanity.

3.2 Create Account And API Key:

Use the Open AI API to analyze and categorize each company's services by generating a prompt that asks the GPT model to classify the services provided.

For using open ai in python script you should use access key that is to be generated in in OpenAI dashboard.



Chapter 4: Python Script

4.1 Basic of Python:

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's extensive standard library and a vast ecosystem of third-party packages make it highly versatile for tasks ranging from web development to data analysis and machine learning. Created by Guido van Rossum and first released in 1991, Python emphasizes code readability with its use of significant whitespace. It is widely used in both academic and industrial settings.

4.2 Python Script For Task:

- To efficiently categorize and filter competitor information from a CSV file, use a Python script that leverages OpenAI's GPT model.
- Use the OpenAI API to analyze and categorize each company's services by generating a prompt that asks the GPT model to classify the services provided.
- For using open ai in python script you should use access key that is to be generated in in OpenAI dashboard.
- Installing OPEN AI module.
- You can add this secret key in python script.
- And Create Function for Open AI.

```
!pip install openai==0.28 -q
import openai
import csv

# Set up OpenAI API key
openai.api_key = 'sk-None-yoABSsaJvanxScrgTrmGT3BlbkFJ0AxZBhpdaTJiclg91CD5'

def categorize_services(Description):
    response = openai.Completion.create(
        model="gpt-3.5-turbo",
        prompt=f"Classify the following service description into categories
        such as bulk email sending, MTA management, email delivery optimization, etc.:\n\n{description}",
        max_tokens=50
    )
    return response.choices[0].text.strip()
```

- The script adds a new column to the CSV with these categorized results.

```
# Read the CSV file
input_file = '/content/Task/competitors.csv'
output_file = '/content/Task/filtered_competitors.csv'

try:
    with open(input_file, mode='r', encoding='utf-8') as infile, open(output_file, mode='w', encoding='utf-8', newline='') as outfile:
        reader = csv.DictReader(infile)
        fieldnames = reader.fieldnames + ['Categorized Services']
        writer = csv.DictWriter(outfile, fieldnames=fieldnames)

        writer.writeheader()
        for row in reader:
            description = row['Description']  #  CSV column name
            categorized_services = categorize_services(description)
            row['Categorized Services'] = categorized_services
            writer.writerow(row)
except FileNotFoundError:
    print(f"Error: The file {input_file} was not found.")
except Exception as e:
    print(f"An error occurred while reading or writing CSV files: {e}")
```

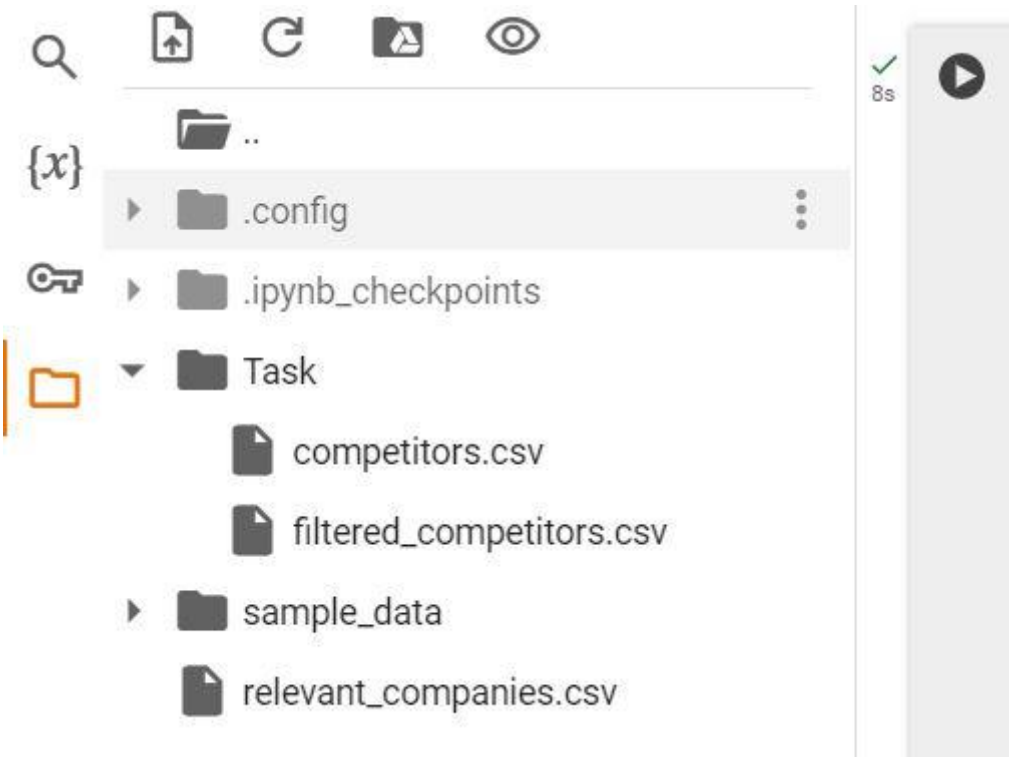

- Subsequently, we filter the data to retain only the companies offering relevant services, such as Email, Bulk email sending, United States, MTA management, and email delivery optimization.

```
# Filter the CSV to retain relevant services
try:
    with open(output_file, mode='r', encoding='utf-8') as infile:
        reader = csv.DictReader(infile)
        relevant_companies = [row for row in reader if any(keyword in row['Categorized Services'].lower()
        for keyword in ['Email','Bulk Email','United States','email sending', 'email delivery'])]
```

- The filtered results are saved to a new CSV file as Relevant Companies:

```
    # Save the filtered results to a new CSV file
    final_output_file = 'relevant_companies.csv'
    with open(final_output_file, mode='w', encoding='utf-8', newline='') as outfile:
        writer = csv.DictWriter(outfile, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(relevant_companies)

    print(f"Filtered relevant companies saved to {final_output_file}")
except FileNotFoundError:
    print(f"Error: The file {output_file} was not found.")
except Exception as e:
    print(f"An error occurred while filtering CSV files: {e}")
```



- As Result I Got Relevant_companies.csv File we saw in this Image.

5 Overall Learning From Task:

- Clay : Its used for Data Collection.
- Python : Python used for Script writing for the Task.
- Open AI Setup : Installs necessary libraries, sets up the OpenAI API key.
- Categorization: Defines a function to use OpenAI's model to categorize service descriptions.
- CSV Processing: Reads an input CSV, categorizes services, and writes to an output CSV.
- Filtering: Filters the output CSV to retain only relevant companies and saves the results to a final CSV.
- This script automates the process of categorizing and filtering companies based on their email services, leveraging OpenAI's capabilities to enhance data management.