Python & Automation Report

Objective

The objective of this assignment was to evaluate automation skills by working with real event data. The task involved writing Python scripts to clean the dataset, generate personalized messages, and optionally automate message delivery through email or Telegram.

Dataset Overview

The dataset contained over 600 rows and the following columns:

- name, first_name, last_name, email, created_at
- approval_status, has_joined_event
- amount, amount_tax, amount_discount, currency, ticket_name
- Job Title, LinkedIn profile

Step 1: Data Cleaning

Approach

- Removed rows with duplicate emails to ensure data uniqueness.
- Normalized the has_joined_event column by converting values such as "Yes"/"No" into boolean True/False.
- Identified and flagged records where the LinkedIn profile was missing or incomplete (empty strings or invalid URLs).
- Flagged rows with missing or blank job titles.
- Saved the cleaned dataset to a new CSV file for further use.

Challenges

- Variations in "Yes"/"No" strings required normalization logic.
- Handling empty or malformed LinkedIn URLs needed careful validation.

Outcome

- Cleaned data saved as cleaned_output.csv.
- The cleaning script was modular and reusable (clean_data.py).

Step 2: Auto-Personalized Messaging

Approach

- Developed a script to generate customized messages based on:
    - Whether the participant joined the event (has_joined_event)
    - Their job title
    - Their first name
    - Presence or absence of a LinkedIn profile
- Created different message templates for joined and not joined participants, incorporating job title and LinkedIn profile availability for personalization.

Examples

- Joined:
  "Hey Venkatesh, thanks for joining our session! As a freelance developer, we think you'll love our upcoming AI workflow tools. Want early access?"
- Not joined:
  "Hi Arushi, sorry we missed you at the last event! We're preparing another session that might better suit your interests as a Product Manager."

Output

- Messages were saved in a CSV file with two columns: email and message.
- Optionally, messages were stored individually in .txt files per user for easy retrieval.

Step 3: Optional Automation (Bonus)

Approach

- Added an email sending function using Python's smtplib for SMTP integration (Gmail SMTP or dummy SMTP server).
- Alternatively, created a batch-ready script to push personalized messages into a Telegram bot queue for automated dispatch.

Benefits

- Demonstrated capability to integrate messaging automation with real-time delivery channels.

Tools and Technologies Used

- Python 3.12
- Libraries: pandas, requests, smtplib (optional), re for regex validations
- Telegram Bot API (optional)
- CSV for input and output data formats

Conclusion

This assignment enhanced my skills in data cleaning, string manipulation, and conditional logic in Python. It also provided practical experience in automating personalized communication workflows using real-world data. The optional automation step demonstrated how messaging platforms like email or Telegram can be integrated into workflow pipelines.

GITHUB: https://github.com/Sanskruti-Gohil/python-automation

LINKEDIN: https://www.linkedin.com/in/sanskruti-gohil-4b9401294/