Project Report: Data Masking Tool

Introduction

The Data Masking Tool project aimed to develop a user-friendly application that allows users to mask sensitive information in CSV and Excel files. The primary goal was to simplify the masking process, providing a graphical user interface (GUI) for users to interact with and observe the masking logs.

Approach

1. Tool Design and Features

The tool was designed to incorporate the following features:

GUI Interface: Developed using Tkinter, the GUI allows users to select a CSV or Excel file, choose columns for masking, and initiate the masking process.

Masking Logic: Implemented a simple masking logic to replace 10-digit phone numbers with 'XXX-XXX-XXXX'.

Support for CSV and Excel Files: Utilized the Pandas library to read and manipulate CSV and Excel files.

Threading for Asynchronous Operation: Incorporated threading to run the masking process asynchronously, preventing the GUI from freezing during file processing.

2. File Handling

The tool accepts both CSV and Excel files. For CSV files, it reads them directly using Pandas `read_csv`. For Excel files, it uses `read_excel` with the 'openpyxl' engine.

3. Masking Logic

The masking logic focuses on identifying and masking 10-digit phone numbers. It extracts phone numbers using regular expressions and replaces them with 'XXX-XXX-XXXX'. This logic is customizable to accommodate additional masking requirements.

4. User Interaction

The GUI provides a user-friendly interface, allowing users to:
- Select a file through a file dialog.
- Choose columns to mask from a listbox.
- Initiate the masking process with a button click.
- View real-time logs of the masking process.

Results

1. File Conversion

The tool successfully converts both CSV and Excel files. For Excel files, it uses the 'openpyxl' engine, ensuring compatibility with modern Excel formats.

2. Masking Operation

The masking logic effectively replaces phone numbers with 'XXX-XXX-XXXX'. The tool iterates through the selected columns, applies the masking logic, and generates a new masked file.

3. User Experience

The GUI enhances user experience by providing a simple and intuitive interface. Users can visualize the logs in real-time, making the process transparent and user-friendly.

4. Output

The output file, containing the masked information, is saved in CSV format. The tool removes unnecessary columns, such as 'Unnamed' columns, providing a clean output.

Conclusion

The Data Masking Tool project successfully achieves its objectives by providing a user-friendly solution for masking sensitive information in CSV and Excel files. The tool's modular design allows for easy customization of the masking logic, making it adaptable to different use cases. The implementation of a GUI enhances usability and transparency during the masking process.

Future improvements may involve extending the masking logic to handle additional types of sensitive information, providing more customization options for users.

Overall, the Data Masking Tool project demonstrates an effective approach to simplifying a complex task and making it accessible to a broader audience.

Video Link :
https://www.loom.com/share/0f399021bfc94d9593f6ec843c78e404?sid=1d44ce35-a844-4553-9025-e0cef2186629