De-identification Tool

**Table of Contents**

Table of Contents --------------------------------------------------------------------------------- 1

Introduction --------------------------------------------------------------------------------------- 2

Technical stack ----------------------------------------------------------------------------------- 3

Architecture --------------------------------------------------------------------------------------- 3

Low-level requirements ------------------------------------------------------------------------- 4

User manual --------------------------------------------------------------------------------------- 5

Troubleshooting ----------------------------------------------------------------------------------- 11

**1. Introduction**

The de-identification tool is designed to remove or obscure personal and sensitive information from datasets, ensuring compliance with privacy regulations and protecting individuals' identities. This guide will walk you through the steps required to effectively use the de-identification tool.

**Scope**

This tool encompasses various aspects related to the sensitive data within datasets. This section outlines the primary objectives, features, use cases to provide a comprehensive understanding of what the tool aims to achieve and its operational boundaries.

**Objectives:**

* **Data Privacy**: Ensure that personal and sensitive information is anonymized to protect individual privacy.
* **Regulatory Compliance**: Assist organizations in complying with data protection regulations such as GDPR, HIPAA, and CCPA.
* **Data Utility**: Maintain the utility of the data for analysis, research, and operational use while safeguarding privacy.

**Key Features:**

1. **Anonymization**: Complete removal of identifiable information.
2. **Pseudonymization**: Replacement of identifiable information with pseudonyms.
3. **Generalization**: Converting specific data points into broader categories.
4. **Masking**: Hiding specific data values or records.
5. **Batch Processing**: Handle large datasets efficiently through batch processing.

**Use Cases:**

1. **Healthcare**: Anonymizing patient records to comply with HIPAA regulations while enabling research and analysis.
2. **Financial Services**: De-identifying customer data to protect privacy and comply with GDPR and other financial regulations.
3. **Marketing**: Removing personally identifiable information (PII) from customer datasets to perform demographic analysis without compromising privacy.
4. **Research**: Providing anonymized datasets to researchers to facilitate studies without exposing sensitive information.

**2. Technical Stack**

The technical stack outlines the technologies and tools used to develop and run the de-identification tool.

**Backend:**

* **Language**: Python
* **Database**: SQLite
* **De-identification Library**: FAKER, SQLite, Pandas, NumPy, tabulate, pyodbc.

**Frontend:**

* **Framework**: .NET Visual Studio 2022 - WinForms

**Others**

* **Version Control**: Git (GitHub)

**3. Architecture**

The architecture section provides an overview of how the components of the de-identification tool interact with each other.

**High-level Architecture**

1. **User Interface (UI)**
   * Users interact with the tool through a web-based interface developed with WinForms.
2. **De-identification Engine**
   * Core logic for de-identification, using libraries like FAKER, SQLite, Pandas, NumPy, tabulate, processes the data according to configured rules.
3. **Database**
   * SQLite stores user data
   * Configurations, and logs are stored in the system.

**Diagram**

A diagram of a software company

Description automatically generated

**4. Low-level Requirements**

This section details the specific, technical requirements necessary to run the de-identification tool.

**Hardware Requirements**

* **Processor**: Quad-core CPU
* **Memory**: Minimum 8 GB RAM
* **Storage**: Minimum 1 GB free disk space

**Software Requirements**

* **Operating System**: Windows 10 or later / macOS 10.15 or later / Linux
* **Java Runtime Environment**: JRE 8 or later (if using ARX)
* **Python**: Version 3.11 or later

**NOTE** : Python will be installed automatically through setup.exe

**Network Requirements**

* **Internet Access**: Required for downloading updates and external libraries

**5. User Manual**

* Installation
* Launching the Tool
* Importing Data
* Configuring Techniques for PII Columns
* De-identifying Data
* Exporting the De-identified Data
* Additional Options

1. **Installation**

**Windows**

1. Download the installer from the setup.exe or mis file.
2. Run the installer and follow the on-screen instructions.
3. Once installed, the tool can be accessed from the Start menu.
4. **Launching the Tool**

After installation, launch the de-identification tool:

* **Windows**: From the Start menu, search for "De-identification Tool" and click to open.

**(OR)**

Find theDe-identification Tool logo on screen and right click to open it.

It will navigate to the below UI  
A screenshot of a computer

Description automatically generated

1. **Importing Data**
2. Open the de-identification tool.
3. Right click on the "**Projects** " button 🡪 Create Project (Give your project a name)🡪 Click on “**Create**”
4. Right click on “Project (Project that was created)”
5. Click on “**Key**” 🡪 Create key(Key is a mandatory value which is used to create a unique value which will be used for maintaining relationships between the tables) and click “**Save**”
6. Again, right click on “**Project**” and Select “**Import**”
7. Select the data file you wish to de-identify (supported formats include CSV, Database, JSON).
   1. **Steps to Import a CSV File:**
      1. **Location**:
         1. Use the "Browse" button to navigate to the location of the CSV file on your system.
         2. The selected file path will be displayed in the "Location" field.
      2. **Delimiter**:
         1. Enter the character that separates the values in the CSV file (e.g., ‘ , ’ for comma, ‘ ; ’ for semicolon).
      3. **Quote**:
         1. Specify the character used for quoting text values in the CSV file (e.g., ‘,").
      4. **Rows Count**:
         1. Enter the number of rows to be inserted from the CSV file
      5. **Table Name**:
         1. Provide a name for the table where the imported data will be stored. This helps in organizing and referencing the data after import.
      6. **Actions**:
         1. **Back**: Click this button to go back to the previous screen if needed.
         2. **Cancel**: Click this button to cancel the import process and discard any entries.
   2. **Steps to import Data from a Database:**
      1. **Connection Details:**
         1. **Type**: Select the type of database (e.g., SQL, MySQL, PostgreSQL) from the dropdown menu.
         2. **Server**: Enter the server address where the database is hosted.
         3. **Username**: Provide the username required to access the database.
         4. **Password**: Enter the corresponding password for the username.
      2. **Additional Details:**
         1. Once the connection details are filled in, click "Next" or "Finish" (depending on the interface) to proceed to the next step.
      3. **Specify Database, Schema, and Table:**
         1. The interface will prompt you to specify the database, schema, and table that needs to be imported.
         2. Database: Select or enter the name of the database from which you want to import data.
         3. Schema: Specify the schema if applicable.
         4. Table: Choose the table that contains the data you wish to import.
      4. **Actions:**
         1. **Back**: Click this button to go back to the previous screen if needed.
         2. **Finish**: Click this button to complete the import process.
         3. **Cancel**: Click this button to cancel the import process and discard any entries.
8. Provide the information required to import the data
9. Click "**finish**" to import the data.
10. After clicking "Finish," the imported data will be stored in a table under the specified project. This will be visible in the project's tree structure on the left side of the interface.
11. Expand the project node (e.g., "Project 1") to see the newly created table (e.g., "Table 1").
12. Ensure that the table is listed, and the data has been imported correctly.

Right click on the “**Table 1**” and select “**View Source Data**” to review the data.

1. **Configuring Techniques for PII Columns**

After importing your data, the next step before de-identification is to configure the techniques for the Personally Identifiable Information (PII) columns. This configuration process allows you to define how each column will be de-identified to ensure privacy and compliance.

**Steps to Configure Techniques:**

* 1. **Navigate to Configuration file:**
* Right click on “**Table**”, Select “**Config**” from the list to configure the techniques on PII columns.
  1. **Define Data Types:**
* For each selected column, choose the appropriate data type from the "**Datatype**" dropdown menu. This helps in selecting the right de-identification technique.
  1. **Choose De-identification Techniques:**
* In the "**Technique**" dropdown menu, select the de-identification method to apply to each column. Common techniques include **Anonymization**, **Pseudonymization**, **Generalization**, and **Masking**.
  1. **Set Key:**
* When set to “**Yes**”, This will generate a hash key of the specified column, which can later be used to maintain relationships between data entries after de-identification.
  1. **Save Configuration:**
* Once all columns are configured, click the "**Save/Finish**" button to save your settings.
  1. **Cancel (if needed):**
* If you need to cancel the configuration, click the "**Cancel**" button to discard the changes.

1. **De-identifying Data**

After configuring the techniques for PII columns and creating the table, the user can proceed to de-identify the data. The de-identification process ensures that sensitive information is anonymized to protect privacy.

* 1. Right-click on the **table** node (e.g., "Table 1") to open the context menu.
  2. **Optional** - If any configurations need to be adjusted, select "**Config**" to update the settings for PII columns.
  3. Click the "**De-Identify**" option in the context menu to start the de-identification process.
  4. The tool will apply the configured techniques to anonymize the data.
  5. Once the de-identification process is complete, you can view the de-identified data by clicking "**View De-identified Data**" in the context menu.
  6. This will display the data with all PII columns de-identified as per the configurations.

1. **Exporting the De-identified Data**

After de-identifying the data, the user can export the de-identified dataset. The export functionality allows users to save the de-identified data either as a CSV file or to a database or as a JSON file.

**Steps to Export De-identified Data:**

1. **Open Project and Table**:
   * In the main interface, navigate to the "**Projects**" tree structure.
   * Expand the project node (e.g., "Project 1") to see the created table (e.g., "Table 1").
2. **Context Menu**:
   * Right-click on the table node (e.g., "Table 1") to open the context menu.
3. **Select Export**:
   * Click on the "**Export**" option in the context menu.
   * A prompt will appear asking for the export format.
4. **Choose Export Format**:
   * 1. **Steps to export a CSV file**
        1. Select "CSV" from the export format options.
        2. **Save CSV File**:
           1. A file dialog will appear, prompting you to select the folder where you want to save the CSV file.
           2. Navigate to the desired folder, provide a file name, and click "Save" to export the de-identified data as a CSV file.
        3. **Confirm Export**:
           1. Ensure that the file is saved in the selected location with the correct name.
     2. **Steps to export data to a Database:**

**NOTE**: Ensure that the table definition is created beforehand in the same order as the de-identified data. This step is crucial for maintaining the integrity and compatibility of the exported data.

* + - 1. Follow the same steps used for importing data from a Database
      2. Ensure that the data is saved in the selected Table under the specified Database.

1. **Additional options:**
2. **Delete**:

* Select "**Delete**" to remove the table or project. A confirmation prompt will appear to ensure you want to delete the selected item.
* Confirm the deletion to remove the table or project from the tool.

1. **Log**:

* Click "**Log**" to view the logs for the de-identification process. Logs provide detailed information about the operations performed and can help in troubleshooting or auditing.
* The log view will display entries such as process start times, completed tasks, and any errors or warnings encountered.

1. **Refresh**:

* Select "**Refresh**" to update the project and table view. This is useful if changes have been made that are not immediately reflected in the interface.
* Refreshing ensures that the latest data and configurations are displayed.

**6. Troubleshooting**

* **Tool not launching**: Ensure your system meets the requirements and try reinstalling the tool.
* **Errors during import**: Check the file format and ensure the data is not corrupted.
* **Slow performance**: Increase your system’s RAM or free up disk space.
* **Inconsistent results**: Verify the configuration settings and re-run the process.

Needs to be added :

After the setup should wait for few minutes