CTGAN User Interface Documentation

Prepared by: Kamala Viloshini Nadarajan

1. Introduction

The CTGAN UI is a web-based application designed to generate synthetic data using the Conditional Tabular Generative Adversarial Network (CTGAN). This tool enables users to upload a dataset, select specific columns for synthetic data generation, and download the results in a user-friendly interface.

Key Features:

- Upload datasets in CSV format.
- Select specific columns for synthetic data generation.
- Specify the number of synthetic rows to be generated.
- View synthetic data directly on the interface.
- Download generated synthetic data for further use.

2. Technologies Used

- **Backend:** Flask (Python Web Framework)
- **Frontend:** HTML, CSS, Bootstrap, jQuery
- **Data Processing:** Pandas
- **Model Training:** CTGAN
- **Development Tools:** Python 3.9+, IDE (e.g., VSCode, PyCharm)

3. Project Structure

```
UI Project/
│
├── static/
│   ├── styles.css          # General styling for the UI
│
├── templates/
│   ├── index.html          # Main CTGAN interface for synthetic data generation
│
├── app.py                  # Flask application and CTGAN logic
```

4. System Workflow

Overview:

1. **Dataset Upload:** The user uploads a dataset in CSV format.
2. **Column Selection:** The user selects specific columns to include in synthetic data.
3. **Specify Rows:** The user specifies the number of synthetic rows to generate.
4. **Train Model:** The CTGAN model is trained on the uploaded dataset.
5. **Generate Synthetic Data:** Synthetic data is generated and displayed.
6. **Download Results:** The user downloads the synthetic data as a CSV file.

5. Installation and Setup

1. Install Python 3.9+.
2. Navigate to the project directory:

   ```
   cd "UI Project"
   ```

3. Install dependencies:

   ```
   pip install -r requirements.txt
   ```

4. Run the application:

   ```
   Python app.py
   ```

6. Application Interface Overview

- **Homepage:** Users can select CTGAN from the dropdown menu.
- **CTGAN Page:**
  - **Upload Dataset:** Upload a CSV file.
  - **Select Target Columns:** Choose columns for synthetic data generation.
  - **Set Number of Rows:** Specify the number of synthetic rows.
  - **Generate Synthetic Data:** View results in a table and download the output.

7. Scripts and Functionalities

- **Flask.py:** Serves the main application.
- **app.py:** Handles CTGAN-specific requests and model training.
- **ctgan_adapter.py:** Defines CTGAN model logic for training and data generation.

8. Visualizations and Explanations

Generated Data Table:

- Displays synthetic data based on the selected columns and rows.

Download Option:

- Allows users to download synthetic data as a CSV file.

9. Testing and Validation

Functional Testing:

- Ensure dataset upload functionality works.
- Validate column selection and synthetic data generation.
- Confirm synthetic data matches the selected columns.

Error Handling:

- Provide clear messages for errors, such as missing datasets or invalid input.

10. Troubleshooting

1. **Dataset not uploading:**
   - Ensure the file is in CSV format.
   - Verify the file size is within permissible limits.
2. **Synthetic data not generating:**
   - Check for missing or invalid target column selection.
   - Verify that the dataset contains categorical columns.
3. **Download not working:**
   - Ensure Flask's send_file functionality is set up correctly.

11. Future Improvements

- Add support for numerical column generation.
- Implement data visualization for comparing real and synthetic data.
- Add a progress bar for model training.

12. Conclusion

The CTGAN UI provides an intuitive platform for generating high-quality synthetic data tailored to user requirements. This interface empowers users to efficiently create data for testing machine learning pipelines while maintaining simplicity and flexibility.