Data Preparation

Preprocessing steps for running supervised and unsupervised models

---

Install Dependencies

Using `pip` and `requirements.txt`

```
# Install system package for venv
sudo apt update && sudo apt install python3-venv -y

# Create and activate a virtual environment
python3 -m venv env
source env/bin/activate

# Install Python dependencies
pip install --upgrade pip
pip install -r requirements.txt
```

Using Poetry

```
# Install Poetry using your distro's package manager:
sudo apt install python3-poetry -y

# Navigate to project root, create/activate venv, and install dependencies
python3 -m venv env
source env/bin/activate
poetry install
```

Download Large Files

Some of the files required for this repository are too large to be stored directly in the repository. To download these files, run the following Python script:

```
python download_large_files.py
```

How It Works

- The script downloads the required files from Google Drive.
- By default, the files will be saved to `data_samples/generated`.
- **Custom Destination**: You can specify a custom destination folder by providing it as an argument when running the script. For example:

  ```
  python download_large_files.py --destination /your/custom/path
  ```

Replace `/your/custom/path` with the desired folder path to store the downloaded files.

Sample Data Structure

The `data_samples` directory is organized to streamline access to both raw and processed datasets required for audit and fraud analysis. Below is an outline of its content:

Provided Data: - Data provided by external sources for audits and fraud analysis.

Generated Data: - Processed datasets aggregated from raw data for downstream tasks.

```
data_samples
├── generated
│   ├── full_data_armenia_pre_correction.csv
│   ├── full_data_armenia_pre_correction_complex_only.csv
└── provided
    ├── audit
    │   └── audit_new_database_merged_all_2024_oct_update.xlsx
    ├── corrections_new
    │   ├── Shahutahark
    │   │   ├── section1.csv
    │   │   └── tins and hash_tin_Arsine.xls
    │   └── VAT
    │       ├── mv_vat_ex(2020).xlsx
    │       └── mv_vat_ex(2021,2022,2023).xlsx
    ├── employee
    │   ├── employee_count.csv
    │   └── unique_emp_month 2013-2023.xls
    ├── invoice
    │   ├── invocie buyer 2021-2023.csv
    │   └── invocie supplier 2021-2023.csv
    ├── sector
    │   └── sector2021-2024_lighter.xlsx
    └── tpentry
        └── tpentry.xlsx
```

For clustering, in addition to the files above, the following files are also generated / provided:

```
├── generated
│   ├── clustering_data.pkl
│   ├── location
│   │   ├── TIN_OIN_CRN.csv
│   │   └── region_city.csv
│   └── receipts
│       └── receipt_percentages.csv
└── provided
    ├── geographical
    │   └── address_master_all.csv
    └── receipts
        └── ecr_receipt_subtotal_6month.csv
```

Running script

The `config.py` file provides centralized configurations for the data preparation.

Run `merge_data.py` to merges and processes various datasets for audit and fraud analysis. Set `complex_only = True` for complex audit analysis, and `False` otherwise. It will generate two different datasets depending on that value.

Optional: run `check_data.py` to check the quality of the created data. You can view the results in `public/index.html`.

Run `clustering/data_creation.py` to merges and processes various datasets for clustering. You need to first run `merge_data.py` because it uses the same features there.

`src/merge_data.py`

This script integrates multiple data sources, processes them, and saves merged data for use in analysis and modeling.

Inputs

1. **Audit Data**: Complex audit records.
2. **Financial Data**:
3. VAT correction data
4. Profit correction data (Shahutahark)
5. **Employee Data**: Processed employee records.
6. **Invoice Data**: Aggregated invoice records.
7. **Sector Data**: Processed sector information.
8. **config.py**: `df_path` is the path where to save the final dataset.

Outputs

- **Merged Dataset**: A single processed file saved at the location specified by `df_path` in the configuration, containing combined data from all input sources.

`src/clustering/data_creation.py`

This script integrates additional data sources for clustering/

Inputs

1. **Receipts Data**: Receipts data.
2. **Location Data**: TIN and OIN Location data.

Outputs

- `clustering_data.pkl` that will later be used by the unspervised/clustering analysis.