Projekt CSV_DataCleaner

Allgemeines:

Mit der in der Datei DataCleaner.py enthaltenen Klasse CleanerCSV können csv-Dateien eingelesen und nach eigenen Vorgaben bereinigt werden. Die Optionen der Datenverarbeitung werden in einer Spezifikationsdatei im JSON-Format vorgenommen. Dort wird auch ausgewählt, ob und unter welchem Namen die bereinigten Daten im csv-Format gespeichert werden sollen und ob Profiling Reports gewünscht sind.

Installationshinweise

Installation der notwendigen Module erfolgt mittels

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

Ordnerstruktur:

In dem Projektordner müssen folgende Elemente existieren:

- ein Ordner `data`, in welchen die zu verarbeitende csv-Datei gespeichert wird.
  Die bereinigten Daten werden anschließend ebenfalls in diesem Ordner gespeichert.
- Falls Profile der ursprünglichen und der bereinigten Daten erstellt werden, befinden sich diese in dem Ordner `profiles`.
- Debugging-Informationen (default der Stufe debug) über den Ablauf des ausgeführten Programmes sowie eventuelle Hinweise zu fehlerhaften Eingaben in der Spezifikationsdatei werden im Ordner `log` festgehalten. Weiterhin wird die Datei `logging.ini` benötigt.
- die Datei `DataCleaner.py`, welche alle nötigen Werkzeuge für die Bereinigung der Daten enthält
- die Datei `app.py` bietet ein Beispiel zur Ausführung des Reinigungsvorganges
- neben der bereits enthaltenen Datei `default_specs.json` wird im Projektordner eine eigene Spezifikationsdatei im json-Format erstellt (z.B `my_specs.json`).
- die Datei `requirements.txt` enthält eine Liste der benötigten, nicht in der Standardbibliothek enthaltenen Module

Spezifikationsdatei:

Die json Datei zur Bereinigung der Daten muss die Form und die Schlüsselwörter der Datei **default_specs.json** besitzen:

```
{
    "input_file" : "data/00_origin_data.csv",
    "output_file" : "data/01_cleaned_data.csv",
    "delimiter": ",",
    "cleaning_info_file": "data/cleaning_info.txt",

    "export_output_file": true,

    "create_profiles": true,
    "input_file_profile": "profiles/profile_in.html",
    "output_file_profile": "profiles/profile_out.html",

    "drop_double_headers": true,
    "drop_duplicates" : true,
    "drop_na" : true,
    "drop_na_how" : {"all":[], "any":"all"},

    "drop_row_title": true,
    "drop_col" : [],

    "str_columns_upper" : [],
    "float_col": ["Price Each"],
    "int_col": ["Quantity Ordered"],
    "numeric_col": [],
    "datetime_col": ["Order Date"],

    "outliers_col": [],

    "replace_char_details": [
      {"col": ["Product", "Purchase Address"], "change": {"@":""}},
      {"col": ["Product"],                     "change": {" ":"_"}} ]
}
```

- in der Datei können verschiedene cleaning-Methoden mittels `true`/`false` ausgewählt werden.
- Einige Bereinigungsmethoden benötigen Spaltennamen. Die **Spaltenauswahl** erfolgt dann über eine **Liste** der gewünschten Spaltenbezeichnungen (leere Liste, wenn nichts ausgewählt werden soll) bzw. über das Literal **"all"**, falls die Methode für alle Spalten gelten soll.
- Falls das Löschen von Zeilen mit fehlenden Daten gewünscht ist (**"drop_na"** : true), muss das Dictionary für **"drop_na_how"** angegeben werden: Die Werte zu den Keys "all" und "any" müssen hier ebenfalls die Struktur einer leeren Liste, einer Liste mit den gewünschten Spaltennamen oder des Literals "all" besitzen. Ist einer der beiden Werte eine leere Liste, gilt folgendes: Der Key "all" entspricht hier einer Löschung der Zeile, wenn alle Zellen (entsprechend der angegebenen Liste) naN-Werte sind, bei "any" wird die Zeile gelöscht, wenn mindestens eine Zelle (entsprechend der angegebenen Liste) ein naN-Wert ist. Werden zu beiden Keys nichtleere Listen angegeben, entspricht dies einer **UND**-Verknüpfung: Falls alle Zellen zu Liste1 und mindestens eine Zelle zu Liste2 einen naN-Wert besitzt, wird die entsprechende Zeile gelöscht.

- Falls in Spalten Zeichenersetzungen durchgeführt werden sollen, werden diese in einer Liste von Dictionaries eingetragen ("replace_row_char_details"): Jeder Auftrag wird dazu mit den keys "col" und "change" angegeben. "col" : Liste der betreffenden Spaltennamen, "change" : Dictionary mit den Einträgen "alt":"neu"

How To:

Eine einfache Anwendung bildet die Datei `app.py`:

```
from DataCleaner import CleanerCSV
cleaner = CleanerCSV(specification_file=my_specs.json)
cleaner.specific_cleaning()
```

Um die Klasse CleanerCSV zu nutzen, wird diese importiert.
Mit dem Namen der vorgegebenen oder einer eigenen Spezifikationsdatei (z.B. 'my_specs.json'), welche die obigen Anforderungen erfüllt, wird das Objekt der CleanerCSV-Klasse anschließend instanziiert.
Die komplette Datenbereinigung nach angegebener Spezifikation kann anschließend mit der Methode `specific_cleaning` durchgeführt werden.

Zugriff auf vorhandene Informationen:

Auf die originalen bzw. **bereinigten Daten**, sofern sie im Programmverlauf weiter benötigt werden, kann außerdem über das Attribut cleaner.df_origin bzw. `cleaner.df` zugegriffen werden. Falls in der Spezifikationsdatei `export_output_file : true` eingestellt ist, wird die Veränderung der Daten im Ordner `data` default unter `cleaning_info.txt` gespeichert. Andernfalls kann man auf den Infostring mittels `cleaner.info()` zugreifen.
Die Anzahl der gelöschten Zeilen durch eine bestimmte Methode erhält man über die folgenden Attribute:

- cleaner.n_double_headers
- cleaner.n_nan_rows
- cleaner.n_duplicates
- cleaner.n_outlier_rows

sowie die Grenzen für die outliers mittels cleaner.whiskers
und die Datentypen nach der Bereinigung wie üblich über cleaner.df.dtypes

Die Profiling Reports werden standardmäßig im Ordner profiles gespeichert und sind mit einem Browser zu öffnen.

Weitere Optionen:

Falls unter bestimmten Umständen die Reihenfolge der Bereinigungsmethoden geändert werden muss, kann dies nach dem Vorbild der Methode cleaner.specific_cleaning() erfolgen, welche die verschiedenen Reinigungsprozesse nach Baukastenprinzip aneinanderreiht. Diese sind so konzipiert, dass innerhalb der einzelnen Methoden geprüft wird, ob die jeweilige Bereinigung in der Spezifikationsdatei aktiviert ist. Nur dann werden die Transformationen durchgeführt.