# Version de python

python 2.7

script pour installer les packages nécessaires (autres que ceux déjà contenu dans anaconda) : pip\_install.sh

# Encoding : UTF-8

Encoding utilisé pour les fichiers : UTF-8.

Une exception pour le fichier téléchargé depuis le site de l’INSEE qui est encodé en iso8859\_15.

# Ordre de lancement des scripts :

python sirenisation\_insee\_main.py

python web\_scraping\_main.py (Optionnel: seulement si l’on veut faire le web scraping)

python add\_tag\_cctx\_main.py

# Liste des packages nécessaires (cf requirements.txt):

os

sys

logging

logging.config

argparse

glob

yaml

ntpath

collections

pandas

fake\_useragent

re

zipfile

unidecode

feather

inspect

urllib.request

bs4

Levenshtein

numpy

csv

json

random

# sirenisation\_insee\_main.py

optional arguments:

-h, --help show this help message and exit

--param\_file PARAM\_FILE

Chemin vers le fichier de paramètre (.yaml)

--download\_new\_db {0,1,2}

Indique la méthode de mise à jour du fichier SIREN

stock depuis le site de l'INSEE - 0: par defaut,

vérifie si un nouveau fichier stock est disponible sur

l'INSEE et le télécharge le cas échéant - 1:

Utilisation du fichier SIREN stock le plus récent du

dossier de données (pas de téléchargement de nouveaux

fichiers sur le site de INSEE) - 2: Force le

téléchargement du fichier le plus récent sur l'INSEE

--overwrite\_csv\_siren {True,False}

Si True, les fichiers csv de résultats déjà générés

seront écrasés et regénérés

# web\_scraping\_main.py

usage: web\_scraping\_main.py [-h] [--param\_file PARAM\_FILE]

[--overwrite {True,False}]

Ce programme prend en entrée les fichiers .csv siren générés par le script

sirenisation\_insee\_main.py. Pour chaque demandeur PM de ces fichiers dont le

niveau de confiance est inférieur au seuil fixé dans le fichier de paramètre,

il envoit une requête au site societe.com pour récupérer des données. Ces

données sont stockées dans des fichiers .csv dans le dossier

webscraping\_results\_path spécifié dans le fichier de paramètre.

optional arguments:

-h, --help show this help message and exit

--param\_file PARAM\_FILE

Chemin vers le fichier de paramètre (.yaml)

--overwrite {True,False}

Si True, les fichiers de résultats de web scraping

déjà générés seront écrasés et regénérés

# add\_tag\_cctx\_main.py

usage: add\_tag\_cctx\_main.py [-h] [--param\_file PARAM\_FILE]

[--overwrite {True,False}]

Ce programme est chargé d'insérer les tags SIREN dans les fichiers json de

jurisprudence se situant dans le dossier input\_decision\_path spécifié dans le

fichier de paramètre. Les fichiers sirénisés sont générés à l'emplacement

output\_decision\_path (fichier de paramètre). Un fichier csv récapitulant les

insertions faites est générés à l'emplacemen output\_csv\_recap (fichier de

paramètre). Le programme va récupérer les résultats des scripts

sirenisation\_insee\_main.py et web\_scraping\_main.py respectivement dans les

dossiers csv\_siren\_path et webscraping\_results\_path (fichier de paramètre) et

insérer les données avec l'indice de confiance le plus élévé. Si les données

de web scraping ne sont pas disponibles, le programme insérera uniquement les

données extraites de la base SIREN.

optional arguments:

-h, --help show this help message and exit

--param\_file PARAM\_FILE

Chemin vers le fichier de paramètre (.yaml)

--overwrite {True,False}

Si True, les fichiers json de jurisprudence sirénisés

déjà générés seront écrasés et regénérés

# Arborescence des fichiers d’input et ouput (paramétrable):

* Dossier Data :
  + Data/decision\_jurisprud : dossier contenant les fichier .json de jurisprudence
  + Data/siren\_data : dossier contenant les fichiers SIREN stock de l’INSEE
* Dossier Output :
  + Output/ output\_siren\_csv : dossier contenant les fichier csv résultats du script « sirenisation\_insee\_main.py »
  + Output/ ouput\_WebScraping : dossier contenant les fichier csv résultats du script « web\_scraping\_main.py »
  + Output/ output\_decision\_jurisprud : dossier contenant les fichiers json de jurisprudence sirenisés.

# Détails sur le calcul de l’indice de confiance :

Toutes les données numériques présentes dans le tableau sont modifiables par l’utilisateur dans les fichiers de paramètres. Le nom « dist », utilisé dans le tableau ci-dessous, est la distance de Levenshtein calculée entre la variable provenant des fichiers de jurisprudence et la variable provenant de la source de données externes (INSEE ou société.com).

|  |  |  |
| --- | --- | --- |
| Champs | Points | Régle |
| Raison Sociale | 50 | Si dist=0 (matching exact), alors Score max (50)  Si dist>0 et subtsring=True, alors 30  Si dist>0 et substring=False, alors  Max(0 ; 50 – alpha \* dist)  Par defaut, alpha = 5 |
| Adresse complète | 30 | Max(0 ; 30-alpha\*dist)  Par defaut, alpha = 3 |
| Ville | 10 | Max(0 ; 10-alpha\*dist))  Alpha = 1 |
| Numéro de voie | 5 | Si match exact -> 5, sinon 0 |
| Code postal | 5 | Si match exact -> 5  Si match exact du prefix -> 3  Sinon -> 0  ‘Préfixe’ correspond au deux premiers chiffres du code postal. |

Le score total de l’indice de confiance est la somme des scores intermédiaires. Dans le cas présenté, le score total maximal est de 100.

# Détails sur le fichier « resultat\_recap\_insertion.csv »

Ce fichier permet à l’utilisateur de voir les décisions prises par le programme pour l’insertion des tags SIREN dans les fichiers json de jurisprudence. On peut notamment y voir les scores intermédiaires pour les résultats provenant de la base INSEE et du web scraping. Une ligne avec uniquement un nom de fichier indique qu’aucun demandeur de type PM n’a été trouvé pour ce fichier.

Voici les principales conventions pour comprendre le nom des colonnes :

* La première colonne indique le nom du fichier traité.
* Les colonnes avec le préfixe « siren@ » sont les colonnes qui proviennent des résultats du script « sirenisation\_insee\_main.py », c’est-à-dire de l’extraction de données du fichier SIREN stock de l’INSEE. Un cas particulier à noter : le préfixe « siren@JsonFeature » indique les features extraites des fichiers json en input.
* Les colonnes avec le préfixe « ws@ » sont les colonnes qui proviennent des résultats du script « web\_scraping\_main.py », c’est-à-dire que ce sont les données récupérées sur le site société.com.
* Les colonnes avec le préfixe « res@ » sont les données qui ont été insérées dans les fichiers json de jurisprudence.