Rapport : SAE Collecte Web

Partie Rapport :

Pour ce projet, nous avons adopté une approche de développement en deux étapes distinctes. Dans un premier temps, nous avons élaboré le code sur un fichier initial contenant une quarantaine de lignes. Ensuite, nous avons répliqué le même processus sur un second fichier comprenant l'intégralité des données. Comme les fichiers sources étaient volumineux, nous avons segmenté chacun d'eux en morceaux d'environ un million de lignes. Par la suite, nous avons concaténé ces morceaux respectifs afin de faciliter le traitement des données. Concernant le fichier "Stock Etablissement", le découpage s'est uniquement effectué dans le fichier comportant toutes les lignes. Par la suite, nous avons fusionné les deux ensembles de données à l'aide d'une opération de jointure (left join), en ne conservant que les colonnes jugées pertinentes pour notre analyse.

Les adresses sont prétraitées pour être compatibles avec les API de géolocalisation et de recherche, notamment en concaténant les colonnes nécessaires à la dénomination d'adresses et en remplaçant les caractères spéciaux et les espaces par des formats acceptés. Elles sont utilisées pour interroger l'API Adresse de Data Gouv afin de récupérer les coordonnées géographiques des adresses. Les coordonnées géographiques sont utilisées pour créer des URL de recherche sur Google Maps, permettant ainsi de visualiser l'emplacement des établissements sur une carte. Les URL générées sont ajoutées pour traiter l'ouverture des liens dans un navigateur. Les coordonnées géographiques sont rassemblées dans un Data Frame pour être exportées vers un fichier CSV.

Des listes sont créées pour stocker les informations extraites des pages web, telles que les sites web, les numéros de téléphone, les horaires et les codes supplémentaires. Pour chaque URL dans la liste créée précédemment, le code accède à la page web correspondante. En utilisant BeautifulSoup et des sélecteurs CSS, le code extrait les informations pertinentes telles que le site web, le numéro de téléphone, les horaires et le code supplémentaire à partir du HTML de la page. Les informations extraites sont ensuite ajoutées aux listes correspondantes. Les listes sont utilisées pour créer un Data Frame qui est ensuite exporté vers un fichier CSV.

Enfin, les deux Data Frames finaux crées sont concaténés pour avoir toutes les informations trouvées par ligne, puis il est exporté au format CSV.

Partie Problèmes :

Nous avons rencontré plusieurs difficultés lors de la découpe des fichiers sources. La taille importante des données et les limitations de mémoire de Jupyter ont rendu ce processus complexe. Étant donné le temps considérable nécessaire pour chaque exécution, une erreur dans la découpe retardait significativement notre avancée. De plus, la mise en forme des

données posait également problème parfois. Par exemple, si une adresse ne contenait qu'un code postal et que celui-ci appartenait aux 9 premiers départements de France, le premier zéro disparaissait, entraînant des erreurs dans les résultats. De même, les coordonnées renvoyées par les API étaient dans un ordre incorrect, nécessitant une inversion de la latitude et de la longitude. Nous avons également rencontré des difficultés avec les recherches d'informations sur Google Maps. Certains endroits ne correspondaient pas aux coordonnées fournies, et de nombreuses URL ne renvoyaient aucun résultat pertinent. Cela nous a amenés à remettre en question certaines parties de notre code, soupçonnant des erreurs ou des problèmes de logique dans leur implémentation.

Nous avons tenté d'exécuter le fichier contenant toutes les lignes, cependant, le processus s'avérait trop long. Par conséquent, nous avons opté pour l'affichage du code ainsi que des résultats du programme sur 40 lignes seulement. Le code complet de toutes les lignes est affiché sans les résultats correspondants.