

Création d'une base de données Elasticsearch

La base a été créée à partir d'un extrait de produits Amazon sous format csv disponible [ici](#)

Structure du fichier d'origine:

'uniq_id', 'product_name', 'manufacturer', 'price', 'number_available_in_stock', 'number_of_reviews',
'number_of_answered_questions', 'average_review_rating',
'amazon_category_and_sub_category', 'customers_who_bought_this_item_also_bought',
'description', 'product_information', 'product_description',
'items_customers_buy_after_viewing_this_item', 'customer_questions_and_answers',
'customer_reviews', 'sellers'

Preprocessing :

La plupart des champs ont dû être retraités pour être exploitables (changement de type, extractions de variables à partir de listes, utilisation d'expressions régulières).

4 index ont ensuite été créés à partir du fichier retraité, pour faciliter les requêtes :

produits *permet d'effectuer des requêtes concernant les produits*

('uniq_id', 'product_name', 'manufacturer', 'price', 'number_of_reviews',
'number_of_answered_questions', 'average_review_rating',
'customers_who_bought_this_item_also_bought', 'description', 'product_information',
'product_description', 'items_customers_buy_after_viewing_this_item', 'customer_reviews',
'stock', 'type_stock', 'categorie_1', 'categorie_2', 'categorie_3', 'categorie_4', 'categorie_5')

reviews *permet d'effectuer des requêtes à propos des notes et reviews clients*

('uniq_id', 'num_review', 'resume', 'note', 'qui', 'commentaire', 'autre_1', 'autre_2', 'autre_3', 'autre_4',
'date_clean')

vendeurs *permet d'effectuer des requêtes concernant les revendeurs*

('revendeur', 'uniq_id', 'prix')

questions *permet d'effectuer des requêtes concernant les questions/reponses*

('uniq_id', 'num_echange', 'conversation')

Les manipulations ont été faites à l'aide de la librairie **pandas**, plus simple d'utilisation que les ingest nodes d'ElasticSearch.

Alimentation de la base avec les données

Le client python **elasticsearch** a été utilisé pour réaliser l'étape de loading, via l'API **_bulk**

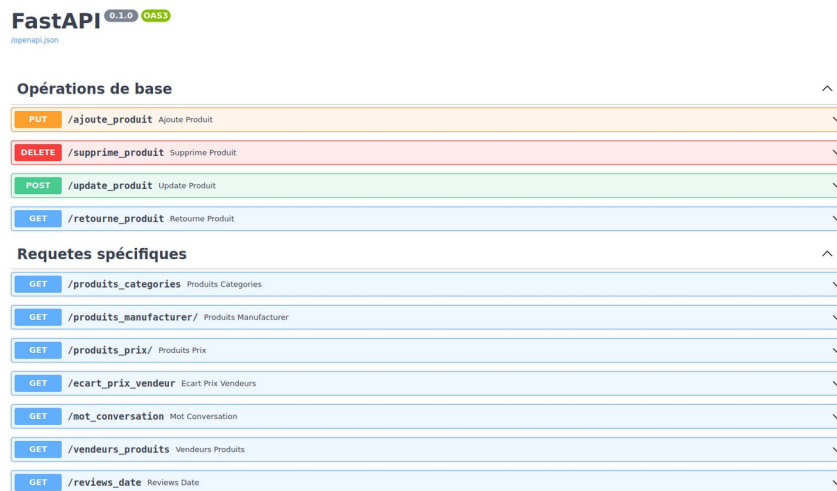
Les étapes de preprocessing et d'alimentation de la base sont codées dans le fichier loading.py

Création de requêtes pour l'utilisation

7 exemples de requêtes ont été créées pour interroger les différents index, à l'aide du langage SQL. Leur explication se trouve dans les docstrings de chaque fonction (une par requête).

API

Une API FastAPI permet de tester les requêtes, ainsi que 4 opérations de base CRUD sur l'index produits. Celle-ci se lance avec la commande `uvicorn api:api`



Les requêtes et l'API sont codées dans le fichier api.py

Dashboard

Un dashboard Kibana a été créé avec 4 visualisations différentes. Il est accessible via ce [lien](#)

