Projet Techniques web

Scrapping et visualisation

Solveig Poder

Table des matières

Présentation du projet

Ce projet, effectué dans le cadre du Master TAL de l'Inalco, a pour objectif de créer une application Streamlit en langage Python, permettant de visualiser des données issues du scrapping de sites web. Il s'agit de présenter deux domaines à un potentiel investisseur afin de le convaincre de s'y intéresser : les hôtels écologiques et les langues peu dotées.

Pour chacun de ces deux domaines, un site web a été retenu : le site du groupe NH Hotels pour les hôtels écologiques, le dictionnaire collaboratif en ligne de l'association NTeAlan pour les langues peu dotées.

Lancement de l'application

Avant de lancer, créez un environnement virtuel et installez les librairies :

```
python -m virtualenv venv
source venv/bin/activate
pip install -r requirements. txt
```

Pour lancer l'application, placez-vous à la racine du répertoire et tapez `streamlit run app.py` dans le terminal, puis rendez-vous à l'URL qui s'affiche.

L'application est également déployée sur Heroku à cette adresse :

https://streamlit-investment-plan.herokuapp.com/

NH Hotels : hôtels écologiques

Notre application Streamlit possède trois pages en plus de la page de présentation (page de présentation dont le contenu est généré par le script **app.py** qui permet de lancer l'application et qui fait appel à des modules pour les autres pages). Deux de ces pages sont consacrées à la visualisation de données scrappées sur le site de NH Hotels.

Ce site web est de type MPA (Multi Page Application) : le client (le navigateur) charge une page HTML générée par le serveur, et à chaque action utilisateur, une nouvelle page est générée et affichée par le client. En pratique, cela signifie que la bibliothèque Python BeautifulSoup suffit pour récupérer son contenu.

Le fichier robots.txt à la racine du site nous indique que les pages qui nous intéressent (https://www.nh-hotels.fr/hotels et https://www.nh-hotels.fr/hotel) ne sont pas concernées par les limitations imposées aux robots.

1. Statistiques sur les hôtels écologiques NH Hotels

Un premier script, **hotels_scrapping.py**, a été écrit afin de récupérer les informations qui nous semblaient pertinentes et de les stocker dans un fichier au format json. Le fichier **hotels.json** contient la liste de tous les hôtels avec, pour chaque hôtel, le nom de la ville où il se situe, le nombre d'étoiles qui lui sont attribuées et la mention « True » ou « False » selon qu'il s'agisse ou non d'un établissement écologique.

Cela nous a permis de construire quelques graphiques avec la bibliothèque Plotly et d'effectuer quelques analyses statistiques sur les hôtels écologiques de la marque, leurs localisations et leur standing. Ce travail est effectué dans le module **prez_hotels.py** et présenté sur la page *NH Hotels : Chiffres* de l'application Streamlit.

2. Recherches d'hôtels écologiques

Dans un second temps, nous avons mis en place un système de recherche d'hôtels écologiques par scrapping en temps réel sur le site de NH Hotels : l'utilisateur peut entrer un nom de ville dans la barre de recherche de la page *NH Hotels : Recherche* et l'application va chercher sur le site web du groupe tous les hôtels écologiques situés dans cette ville puis afficher pour chaque résultat le nom de l'hôtel, son nombre d'étoiles, une photo et le lien vers la page de l'hôtel sur le site.

Le scrapping en temps réel permet d'avoir des résultats toujours à jour. Cependant, la recherche peut s'avérer un peu longue.

Le module **recherche_hotels.py** effectue cette tâche.

NTeAlan : dictionnaire de langues africaines peu dotées

La dernière page de notre application concerne les langues peu dotées et a nécessité le scrapping du dictionnaire collaboratif en ligne de l'association NTeAlan (New Technologies for African Languages).

Il s'agit cette fois d'une SPA (Single Page Application) : le navigateur charge une seule page web (avec ses CSS et JS nécessaires) qui est capable de se modifier en tout ou en partie, en fonction des actions de l'utilisateur, sans jamais se recharger complètement. Pour scrapper ce type d'application, il est nécessaire d'utiliser un framework de testing Web comme Selenium, qui va permettre de simuler les actions d'un utilisateur sur le site.

Le fichier robots.txt à la racine du site nous informe que seul le robot CuteStat est sujet à des limitations sur ce site. Nous pouvons donc récupérer toutes les données souhaitées en toute tranquillité.

Pour ce faire, nous avons écrit un script **ntealan_scrapping.py** afin de récupérer les cent premiers articles du dictionnaire yemba-français et de les stocker dans un fichier au format json. Pour chaque article, le fichier **ntealan.json** contient l'entrée du dictionnaire avec ses variantes, la catégorie grammaticale du mot et ses traductions en français.

La page *NTeALan*, dont le contenu est affiché grâce au module **prez_ntealan.py**, contient une explication de l'enjeu de créer des ressources pour les langues peu dotées, une présentation de l'association NTeALan et de leur dictionnaire collaboratif et un article de ce dictionnaire pris aléatoirement dans la liste des cent articles scrappés sur le site (l'article change à chaque actualisation de la page).

Organisation du code

À la racine du répertoire se trouve le script **app.py** qui permet de lancer l'application (et fait appel aux modules du répertoire **modules**), ainsi que les fichiers nécessaires au déploiement sur Heroku (**Procfile**, **setup.sh** et **requirements.txt**), le **README.md** et le présent manuel.

Dans le répertoire **data** se trouvent les deux fichiers json où sont stockées les données scrappées par les scripts se trouvant dans le répertoire **scrapping** (répertoire **scrapping** qui contient également le driver Chrome nécessaire au bon fonctionnement de Selenium ainsi que le fichier **.env** qui doit être complété avec ses identifiants NTeALan si l'on souhaite relancer le script **ntealan_scrapping.py**).

Enfin le répertoire **modules** contient, comme son nom l'indique, les différents modules où sont écrites les fonctions appelées dans le script **app.py** pour afficher les différentes pages de l'application, ainsi que le fichier **__init__.py** nécessaire à l'importation des dits modules.

Conclusion

Streamlit est un framework Python très utilisé par les data scientists car il est en effet particulièrement adapté pour intégrer des outils de visualisation de données. Intégrant le langage Markdown et compatible avec de nombreuses librairies, comme Plotly ou Pandas que nous avons utilisées, il permet de créer facilement une application permettant de visualiser des données comme celles que nous avons obtenues par scrapping.

Nous espérons que le travail effectué à l'aide de ce framework saura convaincre de l'intérêt d'investir dans les deux domaines présentés que sont les hôtels écologiques et les ressources pour langues peu dotées, et nous n'hésiterons pas à nous servir à nouveau de cet excellent outil pour nos travaux à venir.