

BUT Science des Données - Semestre 3

COMPETENCE 4 : Développer un outil décisionnel - parcours VCOD

Collecte de données web

F.GARNIER



Plan de la Saé

Chapitre 1 : Web Scraping

Chapitre 2 : API

Sujet noté alternants

1) Contexte Web scraping

Pour ce sujet, vous serez libre de choisir le site à scraper. Pensez à analyser les données à traiter, à évaluer la difficulté de récupération de ces données par la technique de web scraping et enfin à imaginer des indicateurs qui aient du sens pour l'application.

TRAVAIL A FAIRE

Etapes à suivre pour réaliser l'application Python :

- Recherche des données et du site à scraper en lien avec notre domaine de compétences et contenant des données géographiques : réfléchir à l'intérêt des données et aux indicateurs potentiels à produire.
- Validation par votre enseignant
- Scraping des données du site web dans un DataFrame.
- Tableau de bord à partir du DataFrame en Python : au moins une carte et un graphique significatif.

2) Contexte API : (Bien faire le cours sur l'API Vélib' auparavant !)

A partir de l'**API Observation météorologique historiques France (SYNOP)**, vous devez répondre à certains nombres de demandes. Vous trouverez des informations sur cette base à cette adresse :

<https://public.opendatasoft.com/explore/dataset/donnees-synop-essentielles-omm/information/?sort=date>

Le code ci-dessous permet de récupérer les premiers éléments :

```
url_base = "https://public.opendatasoft.com/api/explore/v2.1/catalog/datasets/donnees-synop-essentielles-omm/records?limit=1"

requests.get(url_base).json()
```

En utilisant le filtrage de l'API via son URL (voir partie cours) afin de charger au minimum vos dataframes en Python, réaliser le travail suivant :

TRAVAIL A FAIRE

À partir des données collectées, produire au moins 4 **indicateurs visuels** :

a. Carte Folium interactive

Afficher la température moyenne par commune avec des marqueurs Folium :

- couleur = en fonction de classe de température
- info-bulle : nom de la commune et température moyenne.

b. Graphique comparatif

Créer un graphique (matplotlib ou seaborn) comparant par région et par année au moins deux indicateurs (pluviométrie, pression, humidité, nuage, nombre d'observations, nombre de stations, ...)

c. Récupérer les observations à moins de 100km de Poitiers (l'IUT comme point de référence) sur l'ensemble des années. Visuel à définir.

d. Indicateurs libres (à concevoir) et pertinent dans l'application

3) Evaluation

Les deux livrables (WS + API) seront à restituer sur Updago pour le **dimanche 11 janvier** 23h59 au plus tard accompagné d'un pdf succinct (1 recto/verso max) sur ce qui fonctionne / ne fonctionne pas dans votre travail avec la répartition des tâches au sein du groupe.

Une évaluation orale de 10 minutes permettant de juger de l'appropriation de l'application par tous les membres du groupe, sera effectuée la semaine suivant le dépôt. Clairement ceux qui ne comprendront pas les parties générées par l'IA dans leur travail respectif (WS ou API), seront lourdement pénalisés !

Les éléments d'évaluation seront donc les suivants :

- Pour les deux livrables (12 points)
 - Opérationnalité (*réponses aux besoins exprimés et respect du cahier des charges*)
 - Automatisation de l'outil et optimisation, ergonomie,
 - Bonnes pratiques de développement : Commentaires, Indentation, **Découpage modulaire si besoin**, Gestion des erreurs avec try catch si besoin...
 - Qualité des tests (partie fonctionnelle ou non)
- Oral (8 points) :
 - Compréhension de l'ensemble du code WS et API
 - Capacité à identifier les parties de code générées ou non par l'IA dans son travail
 - Capacité à identifier les parties de code issues du support de cours fourni