Fait le 25 Oct.2021

** Projet Finale Algorithme et Complexité

Ce projet est réalisé par Moustapha Ndiaye Éleve-Ingénieur & Data-scientiste en Ingénierie des données et Intelligence Artificielle

---

Titre du Projet : Webscraping (Collecte Extraction Traitement & Analyse des donnés du site web d'Alibaba (Entreprise Chinoise)

Objectif du Projet : Obtention des données relatives aux produits à partir du sites Web d'Alibaba pour l'amélioration de la chaîne d'approvisionnement

Les professionnels de l'approvisionnement et de vente de produits sont constamment à la recherche de moyens de réduire les coûts d'achat de biens et de services. Mais aussi recherchent en permanence des moyens d'obtenir les données nécessaires et précises avec le moins d'efforts possible. Dans ce contexte, la collecte de données précises sur des sites Web tels qu'Alibaba est une tâche gratifiante et parait difficile. Avec les progrès des applications et des outils de science des données, il est possible d'obtenir des données précises avec beaucoup de facilité par rapport aux efforts requis dans le passé. Le Webscraping ou Grattage des donnés l'aide de diverses bibliothèques Python est une compétence importante à avoir dans ce contexte.

Résultats Attendues :

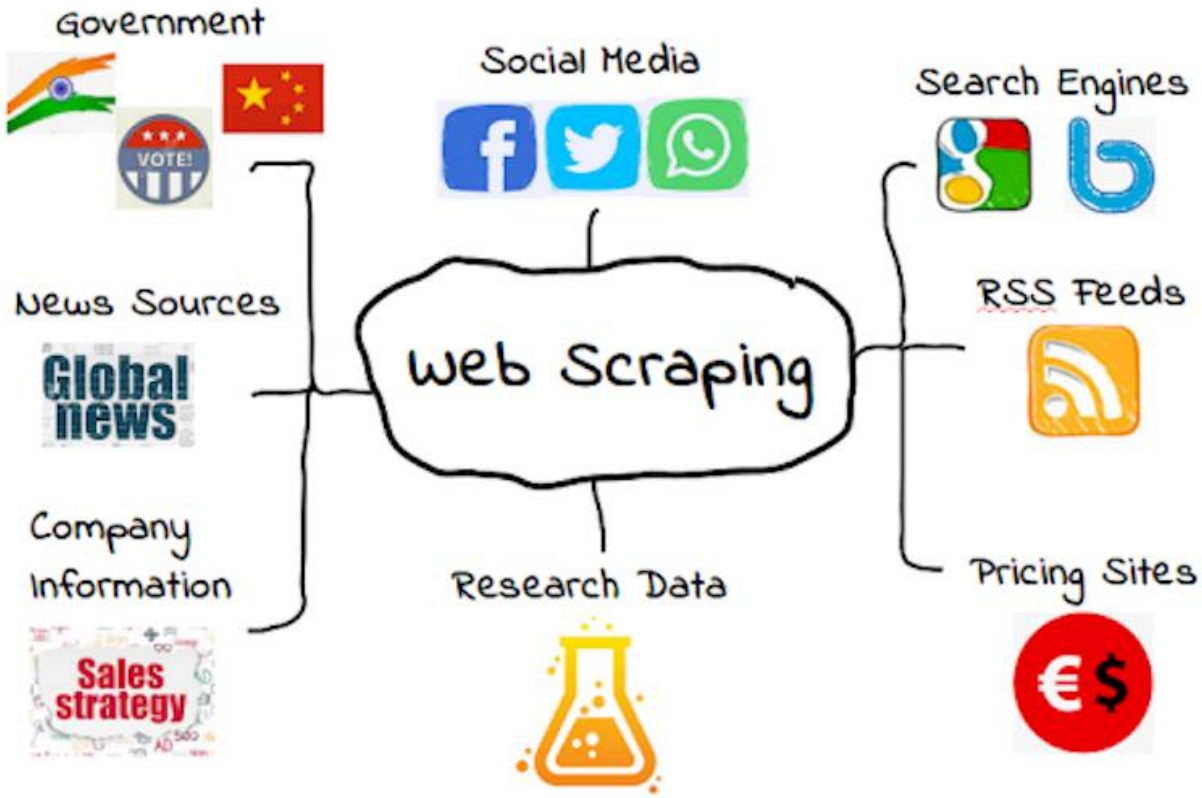
Apres les différentes phases de collectes, d’extraction, d’analyse et de traitement des donnés sur le site web nous obtiendrons enfin dans seul fichiers au format csv les résultats détaillées sur les types d'articles leurs prix leurs évaluation ainsi que leur Url existants dans la plate-forme du site de vente d'Alibaba

---

Webscraping :

Le Webscraping est le processus de collecte de données Web structurées de manière automatisée. On l'appelle aussi extraction de données Web. Certains des principaux cas d'utilisation du Webscraping incluent la surveillance des prix, la veille sur les prix, la surveillance des actualités, la génération de prospects et les études de marché, entre autres. En général, l'extraction de données Web est utilisée par les personnes et les entreprises qui souhaitent utiliser la grande quantité de données Web accessibles au public pour prendre des décisions plus judicieuses. Dans ce contexte nous élaborerons un processus de collecte d'extraction et d'analyse des données de sites Web d'Alibaba de manière

automatisée à l'aide d'un programme informatique. C'est une technique utile pour créer des ensembles de données pour la recherche et l'apprentissage. Enfin nous suivrons un ensemble étape pour créer notre projet de Webscraping à l'aide de Python et de son écosystème de bibliothèques



** Plan du Projet **

1- Choix d'un site Web et description de notre projet Parcourir le site officiel d'Alibaba. Identifiez les informations que nous souhaiterons retirer du site. Cré un fichier de format CSV après extraction des donnés. Résumez enfin notre idée de projet et décrire notre code avec des explications détaillés dans un cahier Juptyer.

2- Utilisez la bibliothèque de requêtes pour télécharger des pages Web Inspectez la source HTML du site Web et identifiez les bonnes URL à télécharger. Téléchargez et enregistrez des pages Web localement à l'aide de la requests bibliothèque. Créez une fonction pour automatiser le téléchargement pour différents sujets/requêtes de recherche.

3-Utilisez Beautiful Soup pour analyser et extraire des informations Analysez et explorez la structure des pages Web du site téléchargées à l'aide de Beautiful soup. Utilisez les bonnes propriétés et

méthodes pour extraire les informations requises. Créez des fonctions à extraire de la page dans des listes et des dictionnaires. Utilisez une API REST pour acquérir des informations supplémentaires si nécessaire.

4-Cré un fichier CSV avec les informations extraites sur les Articles Créez des fonctions pour le processus de bout en bout de téléchargement, d'analyse et d'enregistrement de fichiers CSV. Exécutez la fonction avec différentes entrées pour créer un ensemble de données de fichiers CSV. Vérifiez les informations contenues dans les fichiers CSV en les relisant à l'aide de Pandas.

Installons d'abord les fonctions utiles de jovian

---

[14]

3 s

```
! pip install jovian --upgrade --quiet
```

---

[15]

0 s

```
import jovian
```

---

[16]

0 s

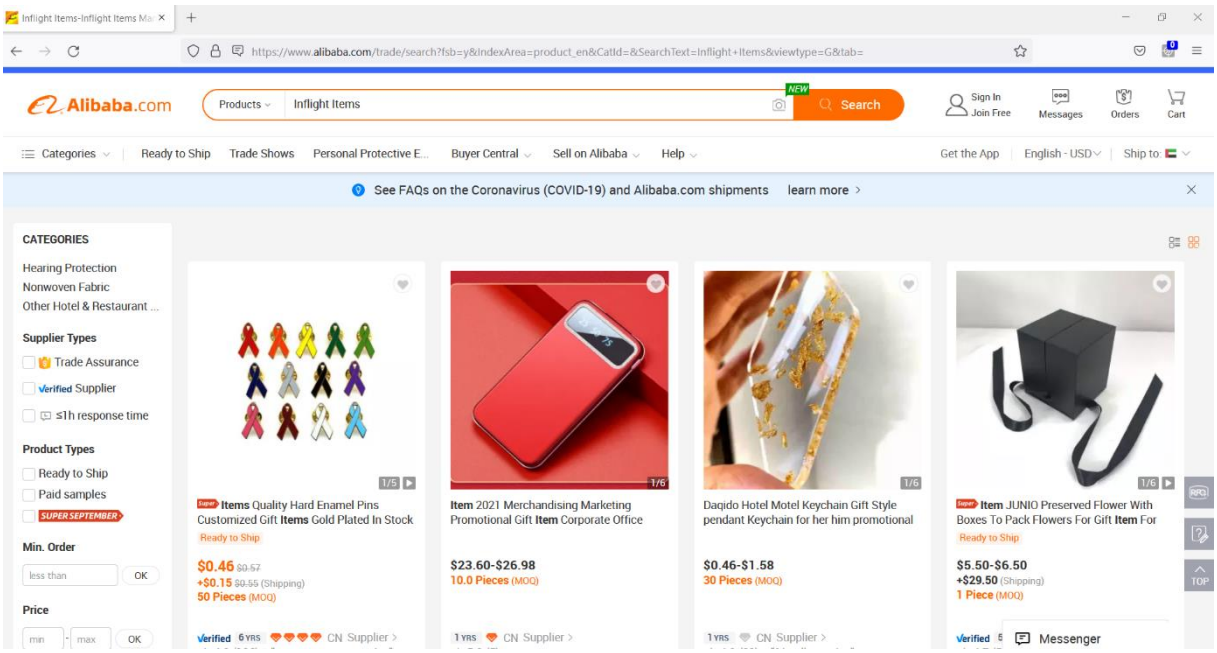
```
jovian.commit(project="Project_Python_Web_scraping_MSDA")
```

```
[jovian] Detected Colab notebook...
[jovian] Error: jovian.commit doesn't work on Colab unless the notebook was created and executed from Jovian.
Make sure to run the first code cell at the top after executing from Jovian.
Alternatively, you can download this notebook and upload it manually to Jovian.
Learn more: https://jovian.ai/docs/user-guide/run.html#run-on-colab
```

Voici les étapes que nous suivrons : Pour ce faire nous utiliserons ce lien https://www.alibaba.com/trade/search?fsb=y&IndexArea=product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab={page} pour faire une étude d'analyse et d'extraction sur la liste des produits ; spécifiquement des couvertures. Pour chaque article, nous tirerons les détails de base de l'article, le prix de l'article, le prix de l'offre le cas échéant, son évaluation le cas échéant, et les différentes URL de la page de chaque article. Pour chaque article, nous obtiendrons plus de détails sur le produit à partir de la page de l'article. Nous allons créer un fichier avec les détails de l'article. Pour chaque article, nous créerons également un fichier CSV au format indicatif suivant : Titre,

Fait le 25 Oct.2021

ArticlePrix, Étoiles, Site Web



Listons le nombre d'articles au sein du site d'Alibaba

Nous utiliserons la bibliothèque de requests en Python pour télécharger la page ensuite utiliser beautifoul soup pour analyser et extraire des informations convertir en une dataframme de données Pandas Nous exécuterons les étapes ci-dessus comme indiqué

---

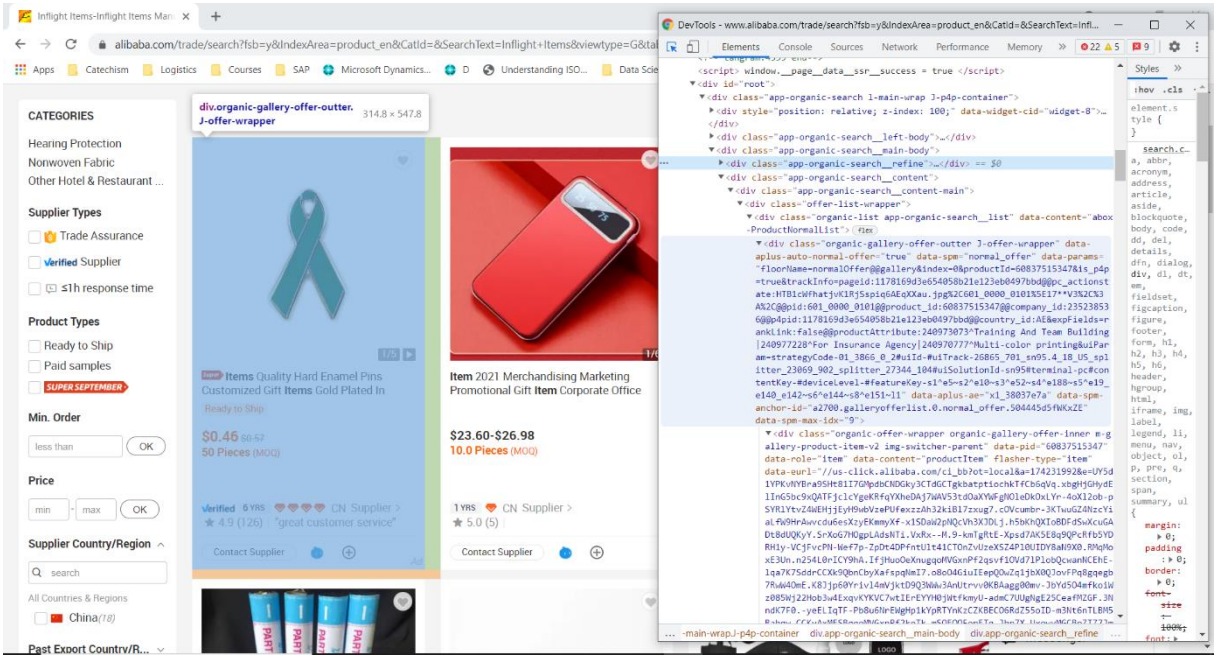
[17]

8 s

```
# Installer les librairies
!pip install requests --upgrade --quiet
!pip install beautifulsoup4 --upgrade --quiet
!pip install pandas --quiet
# Importer les librairies
import requests
from random import randint
from time import sleep
from bs4 import BeautifulSoup
import pandas as pd
```

Ici, nous obtenons le contenu de la page au format html à l'aide de requêtes, puis nous l'analyserons à l'aide de Beautiful soup. Dans notre recherche spécifique, nous avons des attributs dont la valeur n’est aucune. Donc, tout d'abord, nous organisons / analysons les données par rapport à la balise de liste d'articles. En

cherchant sur le site Web à l'aide de l'option Inspecter (clic droit sur les éléments), nous avons constaté que l'élément est organisé dans la balise div avec class=organic-gallery-offer-outter J-offer-wrapper. Nous utilisons une boucle for et une méthode d'ajout pour organiser les données en item_list_tags. Nous utilisons également une combinaison de sommeil et de randint sur le site Web. (La fonction randint () choisira un entier aléatoire entre les limites supérieure et inférieure données, dans ce cas, 10 et 2 respectivement, pour chaque itération de la boucle . L'utilisation de la fonction randint () en combinaison avec la fonction sleep () aidera à ajouter des pauses courtes et aléatoires dans la vitesse d'exploration du programme. La fonction sleep () cessera essentiellement l'exécution du programme pendant le nombre de secondes donné. . Ici, le nombre de secondes sera introduit de manière aléatoire dans la fonction sleep en utilisant la fonction randint().



On peut accéder au contenu de la page web en utilisant la propriété .texte de response.

---

[18]

0 s

```
topic_url = 'https://www.alibaba.com/trade/search?fsb=y&IndexArea=product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab={page}'
```

---

[19]

0 s

```
response = requests.get(topic_url)
```

---

[20]

0 s

```
page_contents = response.text
```

---

[21]

0 s

```
len(page_contents)
```

```
430274
```

On peut accéder au contenu de la page web en utilisant la propriété .text de response.

---

[22]

0 s

```
page_contents[:1000]
```

---

[23]

31 s

```
item_list_tags = []
for page in range(1,5):
    items_url = f"https://www.alibaba.com/trade/search?fsb=y&IndexArea=product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab={page}"
    response = requests.get(items_url)
    page_contents = response.text
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(items_url))
    doc = BeautifulSoup(page_contents, "html.parser")
    for item in doc.find_all("div", {'class': "organic-gallery-offer-outter J-offer-wrapper"}):
        item_list_tags.append(item)
    sleep(randint(2,10))
```

- Si nous n'utilisons pas de boucle pour récupérer plus de pages du site Web, cela donne seulement 8 résultats. Dans ce cas; il y a deux options: nous pouvons simplement gratter plus de pages, avec 8 résultats (dans ce cas) dans chacun avec Requests-Beautiful soup.

---

[24]

0 s

```
len(item_list_tags)
```

```
32
```

Comme vous pouvez le voir notre page contient 32 articles Maintenant nous allons cré une fonction pour télécharger les pages et récupéré le nombre d'article contenu dans chaque page

---

[25]

33 s

```
def get_item_list_tags():
    item_list_tags = []
    for page in range(1,5):
        items_url = f"https://www.alibaba.com/trade/search?fsb=y&IndexArea=
product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab={page}"
        response = requests.get(items_url)
        page_contents = response.text
        if response.status_code != 200:
            raise Exception('Failed to load page {}'.format(items_url))
        doc = BeautifulSoup(page_contents, "html.parser")
        for item in doc.find_all("div", {'class': "organic-gallery-offer-outter J-
offer-wrapper"}):
            item_list_tags.append(item)
        sleep(randint(2,10))
        print('Downloading page number', page)
    return item_list_tags
item_list_tags= get_item_list_tags()
len (item_list_tags)
```

```
Downloading page number 1
Downloading page number 2
Downloading page number 3
Downloading page number 4
32
```

Nous allons ensuite cré une fonction d'assistance indépendante qui effectuera la tâche ci-dessus de téléchargement des pages, de vérification de la réponse et d'analyse à l'aide de beautifulsoup

---

[26]

0 s

```
def get_item_page(items_url):
    response = requests.get(items_url)
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(items_url))
    doc = BeautifulSoup(response.text, 'html.parser')
    return doc
```

---

[27]

1 s

```
doc = get_item_page('https://www.alibaba.com/trade/search?fsb=y&IndexArea
=product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab=')
```

Obtenir les détails et la description du titre des Articles

Après l'étape précédente d'extraction de tous les détails de l'élément, nous extrayons tous les titres requis dans un format de dictionnaire, de sorte que la valeur nulle soit également capturée. (Nous pouvons également utiliser un bloc try except qui est une instruction pour faire quelque chose sauf pour aucun cas ; mais nous le ferons sur une autre instance de ce même projet) Veuillez noter que all_item_titledetail_list peut être trouvé à partir de la balise p avec class='elements -title-normal__content medium' lors de l'inspection du site Web comme à l'étape précédente.

---

[28]

0 s

```
all_item_titledetail_list = []
for item in item_list_tags:
    title = item.find ('p', {'class': 'elements-title-normal__content medium'})
    all_item_titledetail_list.append({"Title":title.text.strip()})
```

From experience, we know that a combination of dictionaries may be difficult to handle while making dataframes. So we convert the above list of dictionaries into simple list

---

[29]

0 s

```
itemdetail=[]
for tag in all_item_titledetail_list:
```

```
    itemdetail.append(tag.get('Title'))
itemdetail[:8]
```

```
['Red Packet Cutomized Special theme 2022 Lai see Red pocket lucky money Chinese New Year CNY whloesaler',
 'Items New Promotional Products Crypto Coin Gift Items Best Selling Products 2021 In Usa Amazon',
 'Item JUNIO Preserved Flower With Boxes To Pack Flowers For Gift Item For Coupple Roses',
 'Item CHRISTMAS Theme Decor Birthday Kit De Fiesta Articulos Parties Party Wholesale Party Item',
 'Daqido Hotel Motel Keychain Gift Style pendant Keychain for her him promotional item',
 'Fashionable inflight items plastic sound insulation ear plugs',
 'Cheap Custom Promotional Item,Promotional Product,Promotional Item China',
 'promotional items']
```

Creons une fonction d'assistance qui permet de récupéré le tittre des articles

---

[30]

0 s

```
def get_itemdetail(item_list_tags):
    all_item_titledetail_list = []
    for item in item_list_tags:
        title = item.find ('p', {'class': 'elements-title-normal__content medium'})
        all_item_titledetail_list.append({"Title":title.text.strip()})

    itemdetail=[]
    for tag in all_item_titledetail_list:
        itemdetail.append(tag.get('Title'))
    return itemdetail

itemdetail = get_itemdetail(item_list_tags)
len (itemdetail)
itemdetail
```

```
['Red Packet Cutomized Special theme 2022 Lai see Red pocket lucky money Chinese New Year CNY whloesaler',
 'Items New Promotional Products Crypto Coin Gift Items Best Selling Products 2021 In Usa Amazon',
 'Item JUNIO Preserved Flower With Boxes To Pack Flowers For Gift Item For Coupple Roses',
```

Fait le 25 Oct.2021

'Item CHRISTMAS Theme Decor Birthday Kit De Fiesta Articulos Parties Party Wholesale Party Item',
'Daqido Hotel Motel Keychain Gift Style pendant Keychain for her him promotional item',
'Fashionable inflight items plastic sound insulation ear plugs',
'Cheap Custom Promotional Item,Promotional Product,Promotional Item China',
'promotional items',
'Item 2021 Merchandising Marketing Promotional Gift Item Corporate Office Souvenir Business Gift Business Promotion Luxury Gift Set',
'Items New Promotional Products Crypto Coin Gift Items Best Selling Products 2021 In Usa Amazon',
'Item CHRISTMAS Theme Decor Birthday Kit De Fiesta Articulos Parties Party Wholesale Party Item',
'Red Packet Cutomized Special theme 2022 Lai see Red pocket lucky money Chinese New Year CNY whloesaler',
'Festival Event Party Supplies Chainsaw Monster Weapon Zombie Killer Bloody Stone Foam Horror Halloween Christmas Print Item Die',
'Fashionable inflight items plastic sound insulation ear plugs',
'promotional items',
'Cheap Custom Promotional Item,Promotional Product,Promotional Item China',
'Items Quality Hard Enamel Pins Customized Gift Items Gold Plated In Stock Hard Enamel Awareness Ribbon Lapel Pin',
'Items New Promotional Products Crypto Coin Gift Items Best Selling Products 2021 In Usa Amazon',
'Item CHRISTMAS Theme Decor Birthday Kit De Fiesta Articulos Parties Party Wholesale Party Item',
'Festival Event Party Supplies Chainsaw Monster Weapon Zombie Killer Bloody Stone Foam Horror Halloween Christmas Print Item Die',
'Daqido Hotel Motel Keychain Gift Style pendant Keychain for her him promotional item',
'Fashionable inflight items plastic sound insulation ear plugs',
'Cheap Custom Promotional Item,Promotional Product,Promotional Item China',
'promotional items',
'Items Quality Hard Enamel Pins Customized Gift Items Gold Plated In Stock Hard Enamel Awareness Ribbon Lapel Pin',
'Items New Promotional Products Crypto Coin Gift Items Best Selling Products 2021 In Usa Amazon',
'Item CHRISTMAS Theme Decor Birthday Kit De Fiesta Articulos Parties Party Wholesale Party Item',

```
 'Items Custom Halloween Decorative Items Faceless Doll Decorations Donut
Toys Pumpkin Costume Party Ornaments',
 'Festival Event Party Supplies Chainsaw Monster Weapon Zombie Killer
Bloody Stone Foam Horror Halloween Christmas Print Item Die',
 'Fashionable inflight items plastic sound insulation ear plugs',
 'Cheap Custom Promotional Item,Promotional Product,Promotional Item
China',
 'promotional items']
```

Extraction du prix des l'articles, des évaluations de chaque articles et des URL

---

[31]

0 s

```
 item_price_tags = []
 for item in item_list_tags:
    item_price = item.find ('span', {'class': 'elements-offer-price-
 normal__price'})
    if item_price == None:
       item_price_tags.append({"ItemPrice":item_price})
    else:
       item_price_tags.append({"ItemPrice":item_price.get_text(strip=True)})
```

---

[32]

0 s

```
 len(item_price_tags)
```

```
32
```

Définissions de la fonction pour récupérer le prix de trois articles dans notre page web

---

[33]

0 s

```
 itemprice=[]
 for tag in item_price_tags:
    itemprice.append(tag.get('ItemPrice'))
 itemprice[:3]
```

```
['$0.99-$4.99', '$0.23-$0.55', '$5.50-$6.50']
```

---

[34]

0 s

```
#nous allons résumer les étapes ci-dessus pour obtenir le prix dans une fonction
def get_itemprice(item_list_tags):
    item_price_tags = []
    for item in item_list_tags:
        item_price = item.find ('span', {'class': 'elements-offer-price-
normal__price'})
        if item_price == None:
            item_price_tags.append({"ItemPrice":item_price})
        else:
            item_price_tags.append({"ItemPrice":item_price.get_text(strip=True)})

    itemprice=[]
    for tag in item_price_tags:
        itemprice.append(tag.get('ItemPrice'))
    return itemprice

itemprice = get_itemprice(item_list_tags)
len(itemprice)
```

32

Le premier prix apparaît Aucun car il s'agit d'un article avec un prix d'offre et nous devons donc afficher le prix d'offre. (Pour le deuxième article, le prix d'offre affichera une valeur Aucun puisque le prix normal est valide)

Extraction des évaluations d'articles

Nous extrairons les évaluations des articles comme indiqué ci-dessous avec cette fonction :

---

[35]

0 s

```
star_tags = []
for item in item_list_tags:
    star = len(item.find_all('i', {'class': "iconfont iconzuanshi seller-star-
level__dm dm-orange"}))
    star_tags.append({"Stars":star})
```

---

[36]

0 s

```
len(star_tags )
```

32

---

[37]

0 s

```
star=[]
for tag in star_tags:
    star.append(tag.get('Stars'))
```

---

[38]

0 s

```
star[:5]
```

```
[0, 2, 3, 3, 1]
```

---

[39]

0 s

```
star_tags[:6]
```

```
[{'Stars': 0},
 {'Stars': 2},
 {'Stars': 3},
 {'Stars': 3},
 {'Stars': 1},
 {'Stars': 0}]
```

---

[40]

0 s

```
#nous allons créer une petite fonction pour faire les étapes ci-dessus
def get_Stars(item_list_tags):
    star_tags = []
    for item in item_list_tags:
        star = len(item.find_all('i', {'class': "iconfont iconzuanshi seller-star-
level__dm dm-orange"}))
        star_tags.append({"Stars":star})
    star=[]
    for tag in star_tags:
        star.append(tag.get('Stars'))
    return star
star = get_Stars(item_list_tags)
star
```

```
[0,
 2,
 3,
```

3,
1,
0,
3,
0,
0,
2,
3,
0,
1,
0,
0,
3,
4,
2,
3,
1,
1,
0,
3,
0,
4,
2,
3,
0,
1,
0,
3,
0]

Extraction des pages URL des Articles

Nous allons extraire les pages URL avec les fonctions comme indiqué ci-dessous

---

[41]

0 s

```
website_tags = []
base_url = 'https:'
for item in item_list_tags:
```

```
    website = item.find ('a', {'class': 'organic-gallery-offer__img-section'})
    website_tags.append({"Website":base_url+website['href']})
```

---

[42]

0 s

```
len(website_tags)
```

```
32
```

---

[43]

0 s

```
URLS=[]
for tag in website_tags:
    URLS.append(tag.get('Website'))
```

---

[44]

0 s

```
#nous résumerons les étapes ci-
dessus pour obtenir des URL a l'aide de la fonction ci dessous
def get_URLS(item_list_tags):
    website_tags = []
    base_url = 'https:'
    for item in item_list_tags:
        website = item.find ('a', {'class': 'organic-gallery-offer__img-section'})
        website_tags.append({"Website":base_url+website['href']})
    URLS=[]
    for tag in website_tags:
        URLS.append(tag.get('Website'))
    return URLS
URLS = get_URLS(item_list_tags)
```

---

[45]

0 s

```
URLS[:9]
```

```
['https://www.alibaba.com/product-detail/Red-Packet-Cutomized-Special-theme-2022_10000003513025.html?s=p',
 'https://www.alibaba.com/product-detail/Items-New-Promotional-Products-Crypto-Coin_1600213826443.html?s=p',
 'https://www.alibaba.com/product-detail/Item-JUNIO-Preserved-Flower-With-Boxes_62339028105.html?s=p',
 'https://www.alibaba.com/product-detail/Item-CHRISTMAS-Theme-Decor-Birthday-Kit_1600334607321.html?s=p',
```

'https://www.alibaba.com/product-detail/Daqido-Hotel-Motel-Keychain-Gift-Style_1600329298987.html?s=p',
'https://www.alibaba.com/product-detail/Fashionable-inflight-items-plastic-sound-insulation_60383877450.html',
'https://www.alibaba.com/product-detail/Cheap-Custom-Promotional-Item-Promotional-Product_905019274.html',
'https://www.alibaba.com/product-detail/promotional-items_60602317457.html',
'https://www.alibaba.com/product-detail/Item-2021-Merchandising-Marketing-Promotional-Gift_1600324447530.html?s=p']

Compilations des informations extraites dans un dictionnaire

Maintenant, nous allons cré un dictionnaire avec tous les détails des articles extraites

---

[46]

0 s

```
Dict={"Title":itemdetail,"ItemPrice":itemprice,"Stars":star,"Website":URLS}
```

We can organise the data in a dataframe as shown below using pandas

---

[47]

0 s

```
items_df = pd.DataFrame(Dict)
```

---

[48]

0 s

```
items_df
```

Nous pouvons maintenant crés le fichier csv requis à partir du cadre de données créé comme indiqué ci-dessous

---

[49]

0 s

```
items_df.to_csv('items.csv', index=None)
```

Nous avons maintenant résumé en tout une fonction de Webscraping complet ci-dessous

[50]

23 s

```
def get_item_list_tags():
    item_list_tags = []
    for page in range(1,5):
        items_url = f"https://www.alibaba.com/trade/search?fsb=y&IndexArea=
product_en&CatId=&SearchText=Inflight+Items&viewtype=G&tab={page}"
        response = requests.get(items_url)
        page_contents = response.text
        if response.status_code != 200:
            raise Exception('Failed to load page {}'.format(items_url))
        doc = BeautifulSoup(page_contents, "html.parser")
        for item in doc.find_all("div", {'class': "organic-gallery-offer-outter J-
offer-wrapper"}):
            item_list_tags.append(item)
        sleep(randint(2,10))
        print('Downloading page number', page)
    return item_list_tags
item_list_tags= get_item_list_tags()

def get_itemdetail(item_list_tags):
    all_item_titledetail_list = []
    for item in item_list_tags:
        title = item.find ('p', {'class': 'elements-title-normal__content medium'})
        all_item_titledetail_list.append({"Title":title.text.strip()})

    itemdetail=[]
    for tag in all_item_titledetail_list:
        itemdetail.append(tag.get('Title'))
    return itemdetail
itemdetail = get_itemdetail(item_list_tags)

def get_itemprice(item_list_tags):
    item_price_tags = []
    for item in item_list_tags:
        item_price = item.find ('span', {'class': 'elements-offer-price-
normal__price'})
        if item_price == None:
            item_price_tags.append({"ItemPrice":item_price})
        else:
            item_price_tags.append({"ItemPrice":item_price.get_text(strip=True)})
```

```
    itemprice=[]
    for tag in item_price_tags:
        itemprice.append(tag.get('ItemPrice'))
    return itemprice
itemprice = get_itemprice(item_list_tags)

def get_Stars(item_list_tags):
    star_tags = []
    for item in item_list_tags:
        star = len(item.find_all('i', {'class': "iconfont iconzuanshi seller-star-
level__dm dm-orange"}))
        star_tags.append({"Stars":star})
    star=[]
    for tag in star_tags:
        star.append(tag.get('Stars'))
    return star
star = get_Stars(item_list_tags)

def get_URLS(item_list_tags):
    website_tags = []
    base_url = 'https:'
    for item in item_list_tags:
        website = item.find ('a', {'class': 'organic-gallery-offer__img-section'})
        website_tags.append({"Website":base_url+website['href']})
    URLS=[]
    for tag in website_tags:
        URLS.append(tag.get('Website'))
    return URLS
URLS = get_URLS(item_list_tags)
```

```
Downloading page number 1
Downloading page number 2
Downloading page number 3
Downloading page number 4
```

---

[51]

25 s

```
def get_web_scraping():
    get_item_list_tags()

    Dict1 = {
        'title': get_itemdetail(item_list_tags),
        'itemprice': get_itemprice(item_list_tags),
        'starrs': get_Stars(item_list_tags),
```

```
        'url': get_URLS(item_list_tags)
    }
    return pd.DataFrame(Dict1)

function_based_df = get_web_scraping()
function_based_df[:5]
```

---

[52]

0 s

```
item_page_url = URLS[0]
```

---

[53]

0 s

```
URLS[0:9]
```

['https://www.alibaba.com/product-detail/Red-Packet-Cutomized-Special-theme-2022_10000003513025.html?s=p',
'https://www.alibaba.com/product-detail/Items-Custom-Halloween-Decorative-Items-Faceless_1600320901257.html?s=p',
'https://www.alibaba.com/product-detail/Items-New-Promotional-Products-Crypto-Coin_1600213826443.html?s=p',
'https://www.alibaba.com/product-detail/Item-JUNIO-Preserved-Flower-With-Boxes_62339028105.html?s=p',
'https://www.alibaba.com/product-detail/Item-CHRISTMAS-Theme-Decor-Birthday-Kit_1600334607321.html?s=p',
'https://www.alibaba.com/product-detail/Fashionable-inflight-items-plastic-sound-insulation_60383877450.html',
'https://www.alibaba.com/product-detail/Cheap-Custom-Promotional-Item-Promotional-Product_905019274.html',
'https://www.alibaba.com/product-detail/promotional-items_60602317457.html',
'https://www.alibaba.com/product-detail/Item-2021-Merchandising-Marketing-Promotional-Gift_1600324447530.html?s=p']

Résumer de notre travail

Tout d'abord, nous avons essayé d'extraire les données sur le site Web d'Alibaba en utilisant Request, HTML Parser et Beautiful soup. Mais après avoir compris que nous n'obtenons pas de données correctes au-delà des huit premiers, nous avons effectué le grattage à l'aide d'un environnement de bibliothèque de python (Panda) qui est un outil plus puissant. Donc, en bref, la combinaison request-html parser-Beautiful soup est facile à utiliser et peut gérer tout projet qui ne

nécessite aucune interaction avec la page pour rendre le JavaScript (en termes simples pages statiques) et le

Références aux liens utiles :

- The Data Wrangling Workshop - Deuxième édition Par Brian Lipp, Shubhadeep Roychowdhury, Dr. Tirthajyoti Sarkar
- Exploitation des médias sociaux : trouver des histoires dans les données Internet par Lam Thuy Vo
- Construisons un projet de webscraping avec Python à partir de zéro | Tutoriel pratique par Aakash N S, PDG, Jovian
- https://www.analyticsvidhya.com/blog/2020/08/web-scraping-selenium-with-python/
- https://towardsdatascience.com/how-to-use-selenium-to-web-scrape-with-example-80f9b23a843a
- Tutoriel de grattage Web (https://docs.python-guide.org/scenarios/scrape/)
- Documentation HTML (https://developer.mozilla.org/en-US/docs/Web/HTML)
- Documentation des demandes (https://www.tutorialspoint.com/python_network_programming/python_http_requests.htm)
- Documentation BeautifulSoup4 (https://www.crummy.com/software/BeautifulSoup/bs4/doc/)
- Documentation Selenium (https://selenium-python.readthedocs.io/)