

Pràctica 1

Comparativa de la qualitat dels suplements proteics

Anna Corral Galeote

Estrella Fernández Pinto

Índex

1.- Context.....	2
2.- Títol	2
3.- Descripció del dataset.....	2
4.- Representació gràfica.....	3
5.- Contingut.....	3
6.- Propietari	4
7.- Inspiració.....	6
8.- Llicència.....	6
9.- Codi	7
10.- Dataset.....	13
11.- Vídeo	13

1. **Context.** Explicar en quin context s'ha recollit la informació. Explicar per què el lloc web triat proporciona aquesta informació. Indicar l'adreça del lloc web.

En aquest projecte, s'ha decidit posar-se a investigar sobre la següent plataforma web d'informació de suplementos nutricionals: <https://supplementdatabase.com/>

Aquesta plataforma neix de la necessitat d'obtenir una informació real sobre la composició de suplementos alimentaris d'una manera clara i senzilla. A més a més, proporciona una classificació d'aquesta informació segons l'objectiu específic que es vulgui assolir com podria ser la pèrdua de greix o l'augment de massa muscular.

Especifiquen que els productes que podem trobar a la seva base de dades han estat investigats i provats científicament per un grup d'experts. Un cop aquests productes passen la primera fase, es testegen els resultats amb models estadístics avançats abans d'arribar a una conclusió final i definitiva perquè sigui tan acurada com sigui possible. Es plantegen tota sèrie de preguntes relacionades amb el producte perquè el consumidor tingui tota la informació que necessita en la seva web, com per exemple; la proteïna del sèrum ajuda a millorar la força?, És veritat que l'extracte d'all ajuda a millorar les funcions immunològiques?, etc. Totes aquestes respostes les podem trobar basades en la informació extreta de 598 estudis.

Tot i que pot haver-hi un gran potencial del públic al qual arribar amb aquesta web, és indispensable no aturar-se en la investigació de nous productes i/o nous ingredients que puguin incorporar-se al mercat perquè és una indústria on el canvi és constant. És per això que aquest treball s'ha escollit pensant fer una comparativa de la qualitat del producte en relació amb el preu que marquen.

2. **Títol.** Definir un títol que sigui descriptiu pel dataset.

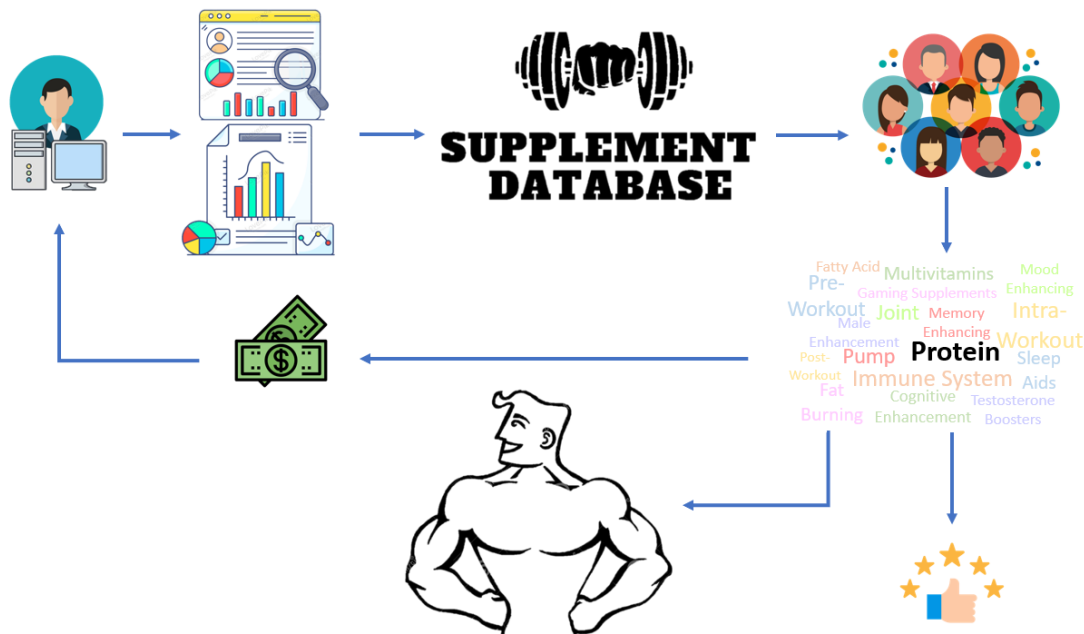
Comparativa de la qualitat dels suplementos proteics

3. **Descripció del dataset.** Desenvolupar una descripció breu del conjunt de dades que s'ha extret. És necessari que aquesta descripció tingui sentit amb el títol escollit.

Tal com especifiquem en el títol del projecte, el conjunt de dades amb el que treballem presenta les dades més importants que hauríem de tenir en compte a l'hora de comparar productes de diferents marques que tenen un objectiu comú i únic. Com podem observar, en el conjunt de dades es presenten les dades més importants pels productes més recomanats dins d'una mateixa categoria. En el nostre cas, s'ha extret la informació de 446 productes de diferents marques que pertanyen a la categoria de suplementos proteics. La descripció més acurada de les característiques extretes les podem veure a l'apartat 5 .

El que podem observar del conjunt de dades també és una falta de processament perquè trobem columnes on hi ha valors numèrics que són representats del tipus string, a causa de la combinació alfanumèrica que té del número del valor amb les seves unitats. És així perquè s'ha extret la informació directament de la pàgina web, però si volguéssim fer un anàlisi més directe hauríem de netejar i processar aquests valors.

4. **Representació gràfica.** Dibuixar un esquema o diagrama que identifiqui el dataset visualment i el projecte escollit.



5. **Contingut.** Explicar els camps que inclou el dataset i el període de temps de les dades.

En aquest conjunt de dades trobem la informació nutricional com les dades de la marca per cadascun dels 50 productes més valorats en la categoria de suplement proteics en un període de temps de dos anys, on la data més allunyada és de l'any 2020 i la més propera és del 2022. Les característiques que hem extret són les següents:

- Product Name. Nom del producte que estem analitzant.
- Manufacturer. Nom de la marca del producte analitzat.
- Manufacturer Website. Adreça web de la marca del producte analitzat.
- Manufacturer Social Media. Adreça web de les xarxes socials de les marques dels productes analitzats.
- Supplement Category. Categoria a la qual pertany el producte analitzat.
- Effectiveness Rating. Valoració de l'efectivitat del producte sobre 3.
- Number of Active Ingredients. Quantitat d'ingredients actius que conté el producte analitzat.
- Research Rating. Valoració de la investigació sobre 100 del producte analitzat.
- Serving Size. Mesura en grams de la quantitat de producte analitzat que necessitem diàriament recomanada.
- Calories per Serving. Quantitat de calories que ens aporta el producte analitzat per cada presa diària.
- Protein per Serving. Quantitat en grams de proteïna que ens aporta cada presa del producte analitzat.
- Protein Percent of Calories. Percentatge de calories en la proteïna del producte analitzat.

- Protein Percent of Serving Size. Percentatge de proteïna en la mesura diària a prendre del producte analitzat.
- Nutrition Label Transparency Score. Percentatge de la transparència de l'etiqueta amb la informació dels valors nutricionals del producte analitzat.
- Nutrition Label Fat Content Transparency Score. Percentatge de la transparència de l'etiqueta nutricional pel que fa al contingut en greix del producte analitzat.
- Ranking within Protein Supplements. Classificació del producte analitzat dins la mateixa plataforma en la categoria de proteïna.
- Ranking within all Supplement Products. Classificació del producte analitzat dins de tots els productes de la plataforma.
- Update Month Day. Dia del mes en què s'ha actualitzat la informació del producte.
- Updated Year. Any en què s'ha actualitzat la informació del producte.
- Fat (%). Percentatge de greix que conté el producte analitzat.
- Carbohydrates (%). Percentatge de carbohidrats que conté el producte analitzat.
- Protein (%). Percentatge de proteïnes que conté el producte analitzat.
- Macros details. Altres detalls del producte analitzat que poden ser d'interès.

Totes aquestes dades van ser recollides a partir de web scrapping utilitzant el llenguatge Python sobre cadascuna de les característiques específiques de cada producte. Primerament, decidim sobre què categoria volem enfocar la nostra informació, en el nostre cas, les proteïnes. Seguidament, s'extrau la informació dels 500 primers productes més ben valorats i afegim la informació i les característiques de cadascun d'ells recorrent cada pàgina individual on trobem la informació. Aquesta és la part del web scrapping on recorrem aquesta informació i la plasmem en un mateix fitxer CSV on guardem les dades que hem obtingut.

6. **Propietari.** Presentar el propietari del conjunt de dades. És necessari incloure cites d'anàlisis anteriors o, en cas de no haver-n'hi, justificar aquesta cerca amb anàlisis similars. Justificar quins passos s'han seguit per actuar d'acord amb els principis ètics i legals en el context del projecte.

El primer que fem és analitzar tant la tecnologia com el propietari del lloc web. Per fer-ho, primer referenciem la pàgina web i després ens centrem en la tecnologia, analitzant la que fa servir el lloc web triat utilitzant la funció "builtwith". Un cop tenim això, ens centrem a analitzar el propietari amb la funció "whois".



```

User-Agent: *
Disallow: /includes/

User-Agent: Mediapartners-Google
Allow: /

Sitemap: https://supplementdatabase.com/sitemap/sitemap.xml

```

```
TECNOLOGIA: Analitzem la tecnologia fet servir a la pàgina web https://supplementdatabase.com fent ús de la funció 'builtwith'.
href_source = 'https://supplementdatabase.com'
builtwith.builtwith(href_source)

[6] ✓ 1.5s

... {'web-servers': ['Nginx'],
     'font-scripts': ['Google Font API'],
     'javascript-frameworks': ['Modernizr', 'jQuery'],
     'widgets': ['OWL Carousel'],
     'photo-galleries': ['jQuery', 'OWL Carousel'],
     'web-frameworks': ['Twitter Bootstrap']}

PROPIETARI: Analitzem qui és el propietari de la pàgina web https://supplementdatabase.com fent ús de la funció 'whois'.
whois.whois(href_source)

[7] ✓ 1.3s

... {'domain_name': ['SUPPLEMENTDATABASE.COM', 'supplementdatabase.com'],
     'registrar': 'NAMECHEAP INC',
     'whois_server': 'whois.namecheap.com',
     'referral_url': None,
     'updated_date': [datetime.datetime(2022, 1, 15, 8, 14, 8),
                     datetime.datetime(2022, 1, 15, 8, 14, 8, 120000)],
     'creation_date': datetime.datetime(2019, 2, 14, 4, 21, 29),
     'expiration_date': datetime.datetime(2023, 2, 14, 4, 21, 29),
     'name_servers': ['NS1.SUPPLEMENTDATABASE.COM',
                     'NS2.SUPPLEMENTDATABASE.COM',
                     'ns1.supplementdatabase.com',
                     'ns2.supplementdatabase.com'],
     'status': 'clientTransferProhibited https://icann.org/epp#clientTransferProhibited',
     'emails': ['abuse@namecheap.com',
                'dfbb3004bffb46c6a1052a42e587d132.protect@withheldforprivacy.com'],
     'dnssec': 'unsigned',
     'name': 'Redacted for Privacy',
     'org': 'Privacy service provided by Withheld for Privacy ehf',
     'address': 'Kalkofnsvegur 2',
     'city': 'Reykjavik',
     'state': 'Capital Region',
     'registrant_postal_code': '101',
     'country': 'IS'}
```

El propietari de les dades i del lloc web és una única persona, en Ken Bendor. És ell mateix també qui facilita les noves dades que van sorgint, els estudis estadístics que hi ha al darrere de cada operació i les anàlisis posteriors juntament amb les conclusions. Podem contactar amb ell si hi ha alguna qüestió no resolta o per qualsevol altre motiu a partir de l'apartat que ens proporciona la mateixa pàgina web d'ajuda (HOME > HELP > CONTACT). Allà ens demana que indiquem el nostre nom, la nostra adreça electrònica, el tema al que està enfocat la nostra pregunta i hi ha un espai també addicional per poder explicar quin és el nostre conflicte.

No hem trobat dins la seva pàgina web cap indicati ni declaració que no es pugui realitzar web scrapping així que hem procedit a continuar amb l'operació. De fet, la pròpia interfície de la pàgina web ens fa pensar que ja s'ha fet prèviament un treball de web scrapping per recopilar tota la informació de què disposem amb les característiques pertinents de cada producte. Tot i que potser la nostra web ja tenia fet un web scrapping, nosaltres hem fet un altre web scrapping especificant què és el que estàvem buscant dins del conjunt de dades.

Existeixen diverses anàlisis semblant al que hem pogut realitzar nosaltres buscant per internet. Uns quants exemples són els següents:

- <https://blog.nutritienda.com/mejores-proteinas/>
- <https://fuertesingym.com/mejores-proteinas/>
- <https://saludprev.com/mejores-proteinas-whey/>
- <https://7mejor.top/proteina/>

Tanmateix, aquests enllaços web ens porten a pàgines on es fa una anàlisi de les millors proteïnes que trobem al mercat mostrant simplement els resultats, però no el procediment que hi ha per arribar a aquestes conclusions. Per tant, ens mostren el resultat d'un scrapping. A més a més, a causa del format d'aquestes pàgines, no podem treballar sobre les dades que ens faciliten per realitzar altres tipus d'anàlisi.

7. **Inspiració.** Explicar per què és interessant aquest conjunt de dades i quines preguntes es pretenen respondre. És necessari comparar amb les anàlisis anteriors presentades a l'apartat 6.

L'interès principal a l'hora d'analitzar aquest conjunt de dades es deu a la gran velocitat a la qual surten nous productes al mercat relacionats amb els suplementos nutricionals, ja que formen part de la vida diària de moltes persones. Però el fet de tenir tanta informació sobre tantes marques diferents, pot ser un gran benefici a l'hora de poder escollir entre un gran ventall de possibilitats i alhora pot ser complicat degut al propi gran volum d'informació. Per aquesta raó, i centrant-nos en un objectiu fix (en aquest cas l'obtenció de proteïnes) pot ser important i decisiu tenir un conjunt de dades que ens mostri la qualitat de cada producte d'una manera concisa i clara. Així, la informació que se'ns presenti influenciarà en la nostra presa de decisions depenent de les necessitats nutricionals.

Un dels "inconvenients" que podem trobar en el conjunt de dades extret, és que la informació pot variar en un lapse de temps relativament curt perquè, com ja hem esmentat, és un mercat amb constant canvi i evolució. Per tant, el dataset que hem obtingut fent aquest projecte ens serveix mentre les dades no pateixin canvis o bé. En el suposat cas que els productes patissin modificacions o sorgissin nous productes, caldria anar actualitzant la informació extreta.

Aleshores, les preguntes que ens podríem plantejar a respondre podrien ser: Quins són els productes més ben valorats pels consumidors en la categoria de suplement nutricional proteic? Quin percentatge de proteïnes ens aporten diàriament? Quina quantitat d'ingredients actius tenen? Quin percentatge contenen pel que fa a carbohidrats i greixos?

8. **Llicència.** Seleccionar una d'aquestes llicències pel dataset resultant i justificar el motiu de la seva elecció. Exemples de llicències que poden considerar-se:
 - Released Under CC0: Public Domain License
 - Released Under CC BY-NC-SA 4.0 License.
 - Released Under CC BY-SA 4.0 License
 - Database released under Open Database License, individual contents under Database Contents License.
 - Altres (especificar quina).

La possible llicència que trobem que podria encaixar amb el nostre enllaç web és la de domini públic a CC0: Public Domain License. L'elecció d'aquest tipus de llicència es deu al fet que no hi ha cap mena d'especificació ni avís legal sobre com tractar les dades del lloc web, cosa que ens fa pensar que sigui de domini públic. També, per les clàusules que hi trobem com són la cessió de dades, perquè altres usuaris puguin desenvolupar-les, millorar-les i reutilitzar-les per qualsevol propòsit sense restriccions legals. La diferència que trobem amb les llicències de tipus CC és que les CC0 tenen l'opció de no rebre drets d'autor i protecció de bases de dades perquè aquestes CC0 són un instrument no adaptat a les lleis de cap jurisdicció legal en particular.

El nostre enllaç web també utilitza certificats SSL en concret, el SSL by Default. Aquest certificat neix de la iniciativa de convertir internet en un lloc més segur. Aquest SSL s'ha convertit en una eina indispensable perquè s'encarrega d'establir un vincle segur entre el lloc web que s'estigui visitant i el navegador emprat. Ho converteix en un enllaç encriptat que garanteix la privacitat absoluta mentre ens mantenim en aquell lloc web. En el cas de la nostra web, obtenen el certificat SSL a partir de l'autoritat de certificació Let's Encrypt.

Ens trobem que també es fan servir els HSTS, un mecanisme que ajuda a protegir llocs web de possibles atacs. Aquests HSTS obliga els navegadors a comunicar-se només amb el lloc web mitjançant HTTPS.

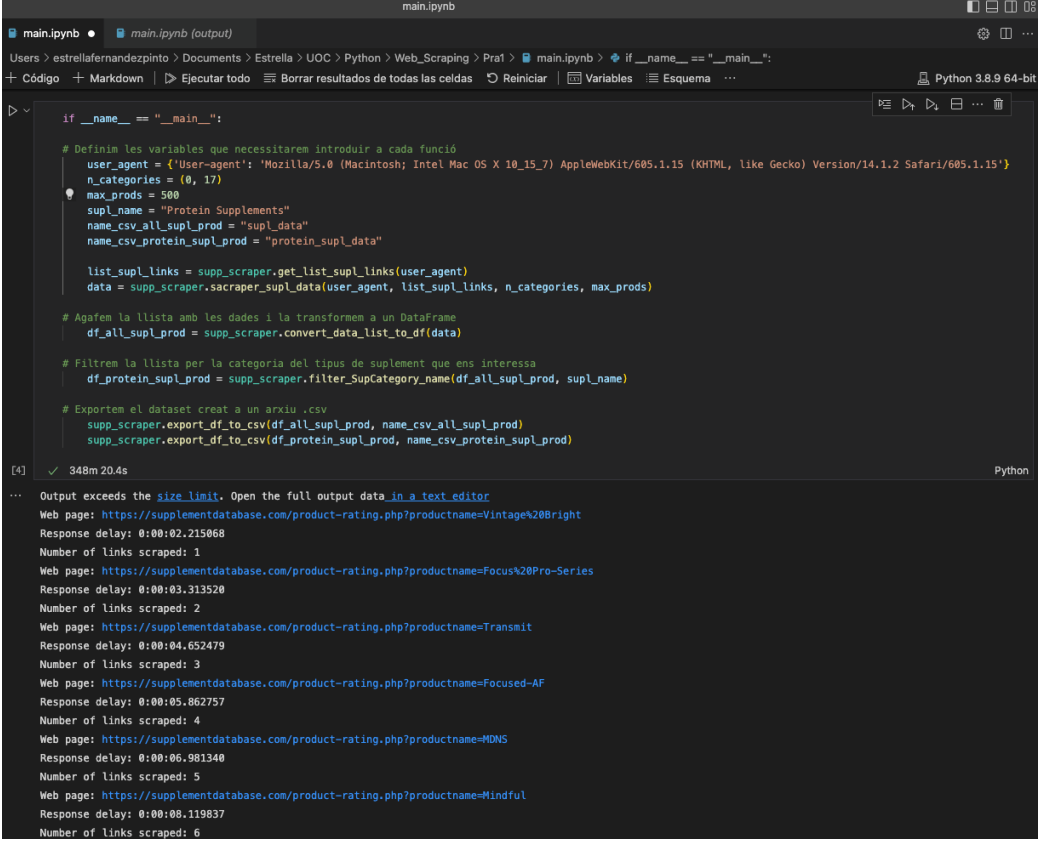
9. **Codi.** Codi amb el qual s'ha generat el dataset, preferiblement en Python o, alternativament, en R.

Resultat de l'execució del codi des del Jupyter notebook per extreure un màxim de 500 productes per cadascuna de les 17 categories de suplementos. Podem observar que s'ha generat correctament el conjunt de dades i l'exportació de les dades a un format .csv. Trobem tot el codi dins la carpeta source del Git i els .csv resultants a la carpeta dataset.

El que fem al document main.ipynb és primerament definir les variables que després s'inclouran dins les funcions, com és l'agent d'usuari (user agent), el nombre de categories, el màxim de productes que volem obtenir, etc.

Fem una llista de la funció que utilitzarem per obtenir els enllaços dels productes i definim la variable data amb els components que la formen. Seguidament, convertim aquesta llista en un DataFrame per després filtrar-la segons la categoria que ens interessen.

Finalment, exportar el DataFrame creat a un .csv



```
if __name__ == "__main__":  
  
    # Definim les variables que necessitem introduir a cada funció  
    user_agent = {'User-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.2 Safari/605.1.15'}  
    n_categories = (0, 17)  
    max_prods = 500  
    supl_name = "Protein Supplements"  
    name_csv_all_supl_prod = "supl_data"  
    name_csv_protein_supl_prod = "protein_supl_data"  
  
    list_supl_links = supp_scraper.get_list_supl_links(user_agent)  
    data = supp_scraper.scraper_supl_data(user_agent, list_supl_links, n_categories, max_prods)  
  
    # Agafem la llista amb les dades i la transformem a un DataFrame  
    df_all_supl_prod = supp_scraper.convert_data_list_to_df(data)  
  
    # Filtrarem la llista per la categoria del tipus de suplement que ens interessa  
    df_protein_supl_prod = supp_scraper.filter_SupCategory_name(df_all_supl_prod, supl_name)  
  
    # Exportem el dataset creat a un arxiu .csv  
    supp_scraper.export_df_to_csv(df_all_supl_prod, name_csv_all_supl_prod)  
    supp_scraper.export_df_to_csv(df_protein_supl_prod, name_csv_protein_supl_prod)  
  
[4] ✓ 348m 20.4s Python  
... Output exceeds the size limit. Open the full output data in a text editor  
Web page: https://supplementdatabase.com/product-rating.php?productname=Vintage20Bright  
Response delay: 0:00:02.215068  
Number of links scraped: 1  
Web page: https://supplementdatabase.com/product-rating.php?productname=Focus20Pro-Series  
Response delay: 0:00:03.313520  
Number of links scraped: 2  
Web page: https://supplementdatabase.com/product-rating.php?productname=Transmit  
Response delay: 0:00:04.652479  
Number of links scraped: 3  
Web page: https://supplementdatabase.com/product-rating.php?productname=Focused-AF  
Response delay: 0:00:05.862757  
Number of links scraped: 4  
Web page: https://supplementdatabase.com/product-rating.php?productname=MDNS  
Response delay: 0:00:06.981340  
Number of links scraped: 5  
Web page: https://supplementdatabase.com/product-rating.php?productname=Mindful  
Response delay: 0:00:08.119837  
Number of links scraped: 6
```



```
Web page: https://supplementdatabase.com/product-rating.php?productname=Cerebral
Response delay: 0:00:10.338956
Number of links scraped: 7
Web page: https://supplementdatabase.com/product-rating.php?productname=TB12%20Perform
Response delay: 0:00:11.480486
Number of links scraped: 8
Web page: https://supplementdatabase.com/product-rating.php?productname=Optimus
...
Number of links scraped: 3381
List transformed to DataFrame successfully!
File 'supl_data.csv' created successfully!
File 'protein_supl_data.csv' created successfully!
```

A continuació, podem observar el final de l'output complet i el resultat exitós de l'execució dintre del Jupyter notebook.

```
10103 Response delay: 5:48:02.329419
10104 Number of links scraped: 3368
10105 Web page: https://supplementdatabase.com/product-rating.php?productname=APEX%20Male
10106 Response delay: 5:48:03.828151
10107 Number of links scraped: 3369
10108 Web page: https://supplementdatabase.com/product-rating.php?productname=Alpha%20King
10109 Response delay: 5:48:05.114948
10110 Number of links scraped: 3370
10111 Web page: https://supplementdatabase.com/product-rating.php?productname=Jumpstart%20EC
10112 Response delay: 5:48:06.433282
10113 Number of links scraped: 3371
10114 Web page: https://supplementdatabase.com/product-rating.php?productname=1Alpha
10115 Response delay: 5:48:07.443931
10116 Number of links scraped: 3372
10117 Web page: https://supplementdatabase.com/product-rating.php?productname=FNG
10118 Response delay: 5:48:08.800459
10119 Number of links scraped: 3373
10120 Web page: https://supplementdatabase.com/product-rating.php?productname=Male%20Balance
10121 Response delay: 5:48:10.822591
10122 Number of links scraped: 3374
10123 Web page: https://supplementdatabase.com/product-rating.php?productname=Built
10124 Response delay: 5:48:11.882119
10125 Number of links scraped: 3375
10126 Web page: https://supplementdatabase.com/product-rating.php?productname=Free%20Testosterone%20Booster
10127 Response delay: 5:48:12.898652
10128 Number of links scraped: 3376
10129 Web page: https://supplementdatabase.com/product-rating.php?productname=Singularity
10130 Response delay: 5:48:14.040546
10131 Number of links scraped: 3377
10132 Web page: https://supplementdatabase.com/product-rating.php?productname=T-Volve
10133 Response delay: 5:48:15.166423
10134 Number of links scraped: 3378
10135 Web page: https://supplementdatabase.com/product-rating.php?productname=Antler%20Test
10136 Response delay: 5:48:16.441553
10137 Number of links scraped: 3379
10138 Web page: https://supplementdatabase.com/product-rating.php?productname=Test%20X180
10139 Response delay: 5:48:18.306119
10140 Number of links scraped: 3380
10141 Web page: https://supplementdatabase.com/product-rating.php?productname=Todd%20Lee%20MD%20Test%20Booster
10142 Response delay: 5:48:19.325014
10143 Number of links scraped: 3381
10144 List transformed to DataFrame successfully!
10145 File 'supl_data.csv' created successfully!
10146 File 'protein_supl_data.csv' created successfully!
10147
```

A continuació podem observar el codi de l'arxiu "main.py" que és el que s'executarà en el terminal per poder obtenir els conjunt de dades d'interès i l'arxiu "supp_scraper.py" que conté totes les funcions que es fan servir en el primer arxiu:

```
main.py
main.py (output)
main.py x
Users > estrellafernandezpieto > Desktop > Pral_Supl_Scraping > main.py > ...
1 from urllib.request import urlopen
2 from bs4 import BeautifulSoup
3 import requests
4 import csv
5 import re
6 import pandas as pd
7 import time
8 from datetime import datetime
9 import whois
10 import builtwith
11 import supp_scraper
12 import sys
13
14
15 if __name__ == "__main__":
16
17     # Definim les variables que necessitarem introduir a cada funció
18     user_agent = {'User-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.2 Safari/605.1.15'}
19     n_categories = (0, 17)
20     max_prods = int(sys.argv[1])
21     suppl_name = "Protein Supplements"
22     name_csv_all_supl_prod = "supl_data"
23     name_csv_protein_supl_prod = "protein_supl_data"
24
25     list_supl_links = supp_scraper.get_list_supl_links(user_agent)
26     data = supp_scraper.scrapper_supl_data(user_agent, list_supl_links, n_categories, max_prods)
27
28     # Afehem la llista amb les dades i la transformem a un DataFrame
29     df_all_supl_prod = supp_scraper.convert_data_list_to_df(data)
30
31     # Filtem la llista per la categoria del tipus de suplement que ens interessa
32     df_protein_supl_prod = supp_scraper.filter_SupCategory_name(df_all_supl_prod, suppl_name)
33
34     # Exportem el dataset creat a un arxiu .csv
35     supp_scraper.export_df_to_csv(df_all_supl_prod, name_csv_all_supl_prod)
36     supp_scraper.export_df_to_csv(df_protein_supl_prod, name_csv_protein_supl_prod)
37
```

Dins l'arxiu "supp_scraper.py", carreguem les llibreries necessàries i comencem a definir les funcions del nostre codi. La primera funció, anomenada `get_page`, és la que s'encarrega de fer la petició a l'URL que definim introduint l'agent d'usuari com a paràmetre d'entrada. Fent ús de la llibreria BeautifulSoup, podem extreure l'HTML, que és el paràmetre de sortida.

La segona funció definida com a `get_list_supl_links`, també té com a paràmetre d'entrada l'agent d'usuari perquè dintre de la funció aplicarem la primera funció i, per tant, s'haurà de realitzar una petició al servidor de la pàgina web. En aquest cas, volem que la funció ens retorni una llista dels enllaços de totes les categories de tipus de suplementes que trobem a l'HTML extret de la primera funció.

La tercera funció definida com a `scrapper_supl_data`, conté com a paràmetres d'entrada: l'agent d'usuari, la llista de les categories de suplementes creada amb l'anterior funció, el rang dels ítems de la llista de tipus de suplement que ens interessa extreure definit com `n_categories` i el màxim nombre de productes que es vol extreure per cada tipus de suplement definit com a `max_prod`. Aquesta funció anirà creant un diccionari amb tota la informació d'interès que va capturant per cada enllaç al qual accedim de cadascun dels productes. Després, cada diccionari creat, que correspondrà a cadascuna de les files del conjunt de dades resultant que ens interessa obtenir, s'anirà afegint a una llista. Per tant, dins l'element "data" hi haurà la informació extreta de cada producte que es trobin dintre del rang de tipus de categoria d'interès i abans del valor màxim de productes especificats.

La quarta funció definida com a `convert_data_list_to_df`, té com a paràmetre d'entrada la llista creada anteriorment ("data") Aquesta funció convertirà la llista introduïda a DataFrame.

La cinquena funció definida com a `filter_SupCategory_name`, s'encarrega de filtrar el DataFrame introduït com a paràmetre d'entrada per la categoria del tipus de suplement especificat també en el segon paràmetre d'entrada de la funció. La sortida d'aquesta funció serà, per tant, un DataFrame amb les dades de la categoria que s'ha especificat, en aquest cas pels suplementes proteics.

La sisena funció definida com a `export_df_to_csv`, s'encarrega d'exportar el DataFrame introduït com a paràmetre d'entrada a un arxiu .csv. El segon paràmetre d'entrada de la funció correspon al nom que tindrà l'arxiu exportat.

```

1 from urllib.request import urlopen
2 from bs4 import BeautifulSoup
3 import requests
4 import csv
5 import re
6 import pandas as pd
7 import time
8 from datetime import datetime
9 import hashlib
10 import builtwith
11
12 def get_page(user_agent: str):
13     page_all = requests.get('https://supplementdatabase.com/suppl-products.php', headers=user_agent)
14     soup_all = BeautifulSoup(page_all.content, 'html.parser') # Parsing content using BeautifulSoup
15     return soup_all
16
17 def get_list_suppl_links(user_agent: str):
18     list_suppl_links = []
19     soup_all = get_page(user_agent)
20     all_supl_prod = soup_all.find("aside", class_="sidebar pb-4")
21     all_supl_prod = all_supl_prod.select("aside div a")
22
23     for ref in all_supl_prod:
24         word = "supplement-product-filter"
25         if word in ref["href"]:
26             list_supl_links.append(ref["href"])
27
28     return list_supl_links
29
30 def scraper_suppl_data(user_agent: str, list_supl_links: list, n_categories: tuple, max_prods: int):
31     start = datetime.now()
32     n_rows = 0
33     data = []
34
35     # Passen per cada un dels links de cada categoria de tipus de suplement
36     for type in list_supl_links[n_categories[0]:n_categories[1]]:
37         try:
38             page = requests.get(type, headers=user_agent, timeout=10)
39             soup = BeautifulSoup(page.content, 'html.parser')
40
41             # Creem una llista amb tots els productes pel tipus de suplement
42             all_products = soup.select("table tbody tr td a")
43             list_prod_links = []
44             max_val = 1
45
46             for product in all_products:
47                 if max_val <= max_prods:
48                     page_link = "https://supplementdatabase.com/" + product["href"]
49                     list_prod_links.append(page_link)
50                     max_val = max_val + 1
51             except requests.exceptions.Timeout:
52
53     pass
54
55 # Entrem a cada un dels links dels productes de la llista per obtenir la informació d'interès
56 for link in list_prod_links:
57     row = {}
58     try:
59         prod_page = requests.get(link, headers=user_agent, timeout=10)
60         prod_soup = BeautifulSoup(prod_page.content, 'html.parser')
61
62         # Get list of elements
63         elements = prod_soup.find("ul", class_="list list-icons list-icons-sm")
64         for ele in elements:
65             text = ele.get_text()
66             if len(text) > 1:
67                 key, value = text.split(": ", 1)
68                 if key == "Manufacturer Website":
69                     r = ele.find("a")["href"]
70                     row[key] = r
71                 elif key == "Manufacturer Social Media":
72                     r = ele.find("a")["href"]
73                     row[key] = r
74                 elif key == "Research Rating":
75                     r = re.search("(=)(above 60 indicates sufficient research)", value).groups()
76                     row[key] = r[0]
77                 elif key == "Serving Size":
78                     if "(" in value:
79                         r = re.search("(\\w\\s\\w+ \\((.*)\\)", value).groups()
80                     else:
81                         row[key] = value
82                 elif key == "Calories per Serving":
83                     r = re.search("(=) calories", value).groups()
84                     row[key] = r[0]
85                 elif "Ranking within" in key:
86                     if key == "Ranking within all Supplement Products":
87                         row[key] = value
88                     else:
89                         row["Ranking within Supplement Category"] = value
90                 else:
91                     row[key] = value
92
93         # Get Updated date
94         updated_date = prod_soup.find("i", class_="fas fa-clock").next_element
95         updated_date = re.search("Updated (.+), (.+)", updated_date).groups()
96         row["Updated Month Day"] = updated_date[0]
97         row["Updated Year"] = updated_date[1]
98
99         # Protein Sup. only
100         if row["Supplement Category"] == "Protein Supplements":
101             # Get Macros percentage in Protein Sup.

```

```
102 info = prod_soup.find("div", class_="col-lg-9")
103 for i in info:
104     i = str(i)
105     if "libfat" in i:
106         fat = re.search("(libfat: (.+)%/lib", i).groups()
107         carbs = re.search("(libcarbohydrates: (.+)%/lib", i).groups()
108         prot = re.search("(libprotein: (.+)%/lib", i).groups()
109         row.update({"fat": fat[0], "Carbohydrates": carbs[0], "Protein": prot[0]})
110
111 # Get Macros information in Protein Supp.
112 macros_info = str(prod_soup.find("p", id="macros").next_element.next_element.next_element.next_element.next_element)
113 macros_info = re.search("<sup>.</sup>", macros_info).groups()
114 row["Macros details"] = macros_info[0]
115
116 # Afegeix cada fila d'informació extreta a la llista "data"
117 data.append(row)
118
119 # Anem visualitzant cada link executat, el temps de resposta total i el nombre de links als qual s'han accedit
120 n_rows = n_rows+1
121 add_one_row = datetime.now()
122 resp_delay = str(add_one_row-start)
123 print("Web page: " + link)
124 print("Response delay: " + resp_delay)
125 print("Number of links scraped: " + str(n_rows))
126
127 except requests.exceptions.Timeout:
128     pass
129 except requests.exceptions.RequestException:
130     pass
131
132 return data
133
134 def convert_data_list_to_df(data):
135     df_all_supl_prod = pd.DataFrame(data)
136     message = "List transformed to DataFrame successfully!"
137     print(message)
138     return df_all_supl_prod
139
140 def filter_SupCategory_name(df_all_supl_prod: pd.DataFrame, supl_name: str):
141     df_protein_supl_prod = df_all_supl_prod[df_all_supl_prod["Supplement Category"] == supl_name]
142     return df_protein_supl_prod
143
144 def export_df_to_csv(df: pd.DataFrame, name_csv: str):
145     df.to_csv(name_csv+".csv", index=False)
146     message = "File " + name_csv + ".csv" created successfully!"
147     print(message)
148     return
149
```

S'ha de tenir en compte que l'script està en un llenguatge Python, aleshores, mostrarem el resultat de les execucions des del terminal dintre del VirtualBox amb sistema operatiu Ubuntu.

Primer s'ha creat un nou entorn virtual anomenat "venv":

```
Scraping - File Manager Terminal - datasci@datasci: ~
Terminal - datasci@datasci: ~/UOC/Scraping
File Edit View Terminal Tabs Help
datasci@datasci:~/UOC/Scraping$ ls
main.py proves requeriments.txt supp_scraper.py
datasci@datasci:~/UOC/Scraping$ virtualenv venv
created virtual environment CPython3.8.10.final.0-64 in 196ms
creator CPython3Posix(dest=/home/datasci/UOC/Scraping/venv, clear=False, no_vcs_ignore=False, global=False)
seeded FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/datasci/.local/share/virtualenv)
added seed packages: pip==22.3.1, setuptools==65.5.1, wheel==0.37.1
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
datasci@datasci:~/UOC/Scraping$
```

A continuació instal·larem els diferents mòduls necessaris per executar el codi que es troben dintre de l'arxiu "requeriments.txt".

```
Scraping - File Manager Terminal - datasci@datasci: ~
Terminal - datasci@datasci: ~/UOC/Scraping
File Edit View Terminal Tabs Help
datasci@datasci:~/UOC/Scraping$ pip install -r requeriments.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: appdirs==1.4.4 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 1)) (1.4.4)
Requirement already satisfied: apturl==0.5.2 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 2)) (0.5.2)
Requirement already satisfied: astroid==2.3.3 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 3)) (2.3.3)
Requirement already satisfied: astropy==4.0.1.post1 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 4)) (4.0.1.post1)
Requirement already satisfied: attrs==19.3.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 5)) (19.3.0)
Requirement already satisfied: Automat==20.2.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 6)) (20.2.0)
Requirement already satisfied: backcall==0.1.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 7)) (0.1.0)
Requirement already satisfied: beautifulsoup4==4.11.1 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 8)) (4.11.1)
Requirement already satisfied: bleach==3.1.5 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 9)) (3.1.5)
Requirement already satisfied: blinker==1.4 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 10)) (1.4)
Requirement already satisfied: bs4==0.0.1 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 11)) (0.0.1)
Requirement already satisfied: builtwith==1.3.4 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 12)) (1.3.4)
Requirement already satisfied: catfish==1.4.13 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 13)) (1.4.13)
Requirement already satisfied: certifi==2019.11.28 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 14)) (2019.11.28)
Requirement already satisfied: chardet==3.0.4 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 15)) (3.0.4)
Requirement already satisfied: command-not-found==0.3 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 16)) (0.3)
Requirement already satisfied: configobj==5.0.6 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 17)) (5.0.6)
Requirement already satisfied: constantly==15.1.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 18)) (15.1.0)
Requirement already satisfied: cryptography==2.8 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 19)) (2.8)
Requirement already satisfied: cssselect==1.1.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 20)) (1.1.0)
Requirement already satisfied: cupshelpers==1.0 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 21)) (1.0)
Requirement already satisfied: cyclus==0.10.0 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 22)) (0.10.0)
Requirement already satisfied: dbus-python==1.2.16 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 23)) (1.2.16)
Requirement already satisfied: decorator==4.4.2 in /home/datasci/.local/lib/python3.8/site-packages (from -r requeriments.txt (line 24)) (4.4.2)
Requirement already satisfied: defer==1.0.6 in /usr/lib/python3/dist-packages (from -r requeriments.txt (line 25)) (1.0.6)
```

```
Scraping - File Manager  Terminal - datasci@datasci: ... 17 nov, 11:26
Terminal - datasci@datasci: ~/UOC/Scraping
File Edit View Terminal Tabs Help
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/utils/entrypoints.py", line 58, in get_best_invocation_for_this_pip
    if found_executable and os.path.samefile(
File "/usr/lib/python3.8/genericpath.py", line 101, in samefile
    s2 = os.stat(f2)
FileNotFoundError: [Errno 2] No such file or directory: '/usr/bin/pip3.8'
Call stack:
File "/home/datasci/.local/bin/pip", line 8, in <module>
    sys.exit(main())
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/cli/main.py", line 70, in main
    return command.main(cmd_args)
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/cli/base_command.py", line 101, in main
    return self._main(args)
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/cli/base_command.py", line 223, in _main
    self.handle_pip_version_check(options)
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/cli/req_command.py", line 148, in handle_pip_version_check
    pip_self_version_check(session, options)
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/self_outdated_check.py", line 237, in pip_self_version_check
    logger.info("[present-rich] %s", upgrade_prompt)
File "/usr/lib/python3.8/logging/_init_.py", line 1446, in info
    self._log(INFO, msg, args, **kwargs)
File "/usr/lib/python3.8/logging/_init_.py", line 1589, in _log
    self.handle(record)
File "/usr/lib/python3.8/logging/_init_.py", line 1599, in handle
    self.callHandlers(record)
File "/usr/lib/python3.8/logging/_init_.py", line 1661, in callHandlers
    hdlr.handle(record)
File "/usr/lib/python3.8/logging/_init_.py", line 954, in handle
    self.emit(record)
File "/home/datasci/.local/lib/python3.8/site-packages/pip/_internal/utils/logging.py", line 179, in emit
    self.handleError(record)
Message: '[present-rich] %s'
Arguments: (UpgradePrompt(old='22.1.2', new='22.3.1'),)
datasci@datasci:~/UOC/Scraping$
```

Abans d'executar l'arxiu "main.py", observarem les línies de codi que hi ha dins.

```
Terminal - datasci@datasci: ~/UOC/Scraping
File Edit View Terminal Tabs Help
datasci@datasci:~/UOC/Scraping$ cat main.py
from urllib.request import urlopen
from bs4 import BeautifulSoup
import requests
import csv
import re
import pandas as pd
import time
from datetime import datetime
import whois
import builtwith
import supp_scraper
import sys

if __name__ == "__main__":
    # Definim les variables que necessitem introduir a cada funció
    user_agent = {'User-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.2 Safari/605.1.15'}
    n_categories = (0, 17)
    max_prods = int(sys.argv[1])
    supl_name = "Protein Supplements"
    name_csv_all_supl_prod = "supl_data"
    name_csv_protein_supl_prod = "protein_supl_data"

    list_supl_links = supp_scraper.get_list_supl_links(user_agent)
    data = supp_scraper.scraper_supl_data(user_agent, list_supl_links, n_categories, max_prods)

    # Agafem la llista amb les dades i la transformem a un DataFrame
    df_all_supl_prod = supp_scraper.convert_data_list_to_df(data)

    # Filtem la llista per la categoria del tipus de suplement que ens interessa

    # Filtem la llista per la categoria del tipus de suplement que ens interessa
    df_protein_supl_prod = supp_scraper.filter_SupCategory_name(df_all_supl_prod, supl_name)

    # Exportem el dataset creat a un arxiu .csv
    supp_scraper.export_df_to_csv(df_all_supl_prod, name_csv_all_supl_prod)
    supp_scraper.export_df_to_csv(df_protein_supl_prod, name_csv_protein_supl_prod)
datasci@datasci:~/UOC/Scraping$
```

Finalment, executem el codi de l'arxiu especificant que volem extreure un producte com a valor màxim per cada categoria de suplement. Es pot observar cada pàgina web a la qual està entrant, el temps de resposta total que va acumulant amb l'extracció de les dades de cada enllaç per cada producte i el nombre total de productes (enllaços) als quals ha accedit i ha extret dades. També, podem observar que el conjunt de dades s'ha creat correctament, així com l'exportació de les dades als arxius d'extensió .csv.


```
Scraping - File Manager Terminal - datasci@datasci: ~/UOC/Scraping
datasci@datasci:~/UOC/Scraping$ python3 main.py 1
Web page: https://supplementdatabase.com/product-rating.php?productname=Vintage%20Bright
Response delay: 0:00:05.710675
Number of links scraped: 1
Web page: https://supplementdatabase.com/product-rating.php?productname=Eviscerate
Response delay: 0:00:09.228978
Number of links scraped: 2
Web page: https://supplementdatabase.com/product-rating.php?productname=Morphomega
Response delay: 0:00:12.693314
Number of links scraped: 3
Web page: https://supplementdatabase.com/product-rating.php?productname=Universal%20Soldier
Response delay: 0:00:18.531788
Number of links scraped: 4
Web page: https://supplementdatabase.com/product-rating.php?productname=Zinc%20Defender
Response delay: 0:00:20.510426
Number of links scraped: 5
Web page: https://supplementdatabase.com/product-rating.php?productname=Cluster%20Bomb
Response delay: 0:00:25.958640
Number of links scraped: 6
Web page: https://supplementdatabase.com/product-rating.php?productname=Triple%20Flex
Response delay: 0:00:30.803209
Number of links scraped: 7
Web page: https://supplementdatabase.com/product-rating.php?productname=Libido%20Boost%20Powder
Response delay: 0:00:32.944339
Number of links scraped: 8
Web page: https://supplementdatabase.com/product-rating.php?productname=Prevagen%20Regular%20Strength
Response delay: 0:00:34.518844
Number of links scraped: 9
Web page: https://supplementdatabase.com/product-rating.php?productname=Relax%20Day
Response delay: 0:00:38.580574
Number of links scraped: 10
Web page: https://supplementdatabase.com/product-rating.php?productname=Cardio%20Daily%20RX
Response delay: 0:00:40.840919
Response delay: 0:00:32.944339
Number of links scraped: 8
Web page: https://supplementdatabase.com/product-rating.php?productname=Prevagen%20Regular%20Strength
Response delay: 0:00:34.518844
Number of links scraped: 9
Web page: https://supplementdatabase.com/product-rating.php?productname=Relax%20Day
Response delay: 0:00:38.580574
Number of links scraped: 10
Web page: https://supplementdatabase.com/product-rating.php?productname=Cardio%20Daily%20RX
Response delay: 0:00:40.840919
Number of links scraped: 11
Web page: https://supplementdatabase.com/product-rating.php?productname=Crea%20Plus
Response delay: 0:00:43.184449
Number of links scraped: 12
Web page: https://supplementdatabase.com/product-rating.php?productname=Performance%20One
Response delay: 0:00:47.805339
Number of links scraped: 13
Web page: https://supplementdatabase.com/protein-rating.php?productname=Protein%20Creations&manu=NutraOne
Response delay: 0:00:49.899742
Number of links scraped: 14
Web page: https://supplementdatabase.com/product-rating.php?productname=Gains%20Candy%20Hydroprime
Response delay: 0:00:53.741773
Number of links scraped: 15
Web page: https://supplementdatabase.com/product-rating.php?productname=Relax%20Night
Response delay: 0:00:57.492997
Number of links scraped: 16
Web page: https://supplementdatabase.com/product-rating.php?productname=Genius%20Test
Response delay: 0:01:00.905959
Number of links scraped: 17
List transformed to DataFrame successfully!
File 'supl_data.csv' created successfully!
File 'protein_supl_data.csv' created successfully!
datasci@datasci:~/UOC/Scraping$
```

10. **Dataset.** Publicar el dataset obtingut en format CSV a Zenodo, incloent-hi una breu descripció. Obtenir i adjuntar l'enllaç del DOI del dataset. El dataset també haurà d'incloure's a la carpeta **/dataset** del repositori.

Enllaç del DOI del dataset a Zenodo:

Target URL:

<https://doi.org/10.5281/zenodo.7339942>

DOI:

10.5281/zenodo.7339942

[Upload](#)
[Communities](#)

anna29695@gmail.com

November 20, 2022

Dataset Open Access

Comparativa de la qualitat dels suplementes

Anna Corral Galeote; Estrella Fernández Pinto

Diferents paràmetres per analitzar la qualitat de tots els tipus de suplementes (supl_data.csv) i dels suplementes proteics en concret (protein_supl_data.csv).

Product Name	Manufacturer	Manufacturer Website	Supplement Category	Serving Size	EF
Protein Creations	NutraOne	https://nutraonenutrition.com	Protein Supplements	32.7g	2.1

0 views

0 downloads

See more details...

Indexed in

OpenAIRE

Publication date:

November 20, 2022

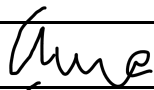
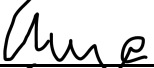
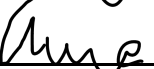
DOI:

DOI 10.5281/zenodo.7339942

Keyword(s):

supplement proteins

11. **Vídeo.** Realitzar un breu vídeo explicatiu de la pràctica (màxim 10 minuts), que haurà de comptar amb la participació dels dos integrants del grup. Al vídeo s'haurà de realitzar una presentació del projecte, destacant els punts més rellevants, tant de les respostes als apartats com del codi utilitzat per a extreure les dades. Indicar l'enllaç del vídeo (<https://drive.google.com/>...), que haurà d'estar al Google Drive de la UOC.

Contribucions	Signatura
Investigació prèvia	
Redacció de les respostes	
Desenvolupament del codi	
Participació al vídeo	