Pràctica 1
Característiques dels vins venuts per vinissimus.com

1 Context.

En aquesta pràctica aplicarem la tècnica del web scrapping a la pàgina web de vinissimus.com. Vinissimus és una botiga online especialitzada en la venta de vins. Els seus usuaris compren els vins i els valoren i, per tant, la informació que es pot obtenir dels seus vins és àmplia i variada. L'objectiu principal és poder analitzar els gustos dels compradors de vins de la web, quin tipus de raïm és més ben valorat, si coincideixen els gustos dels clients amb les puntuacions de les webs especialitzades, la influència de la forma de l'ampolla en la valoració del vi (analitzant la forma de l'ampolla mitjançant xarxes neuronals), els preus per regions o per país, etc. Múltiples anàlisis que podem extreure d'un datasaet que creiem és molt complert i interessant.

2 Títol.

Característiques dels vins venuts per vinissimus.com.

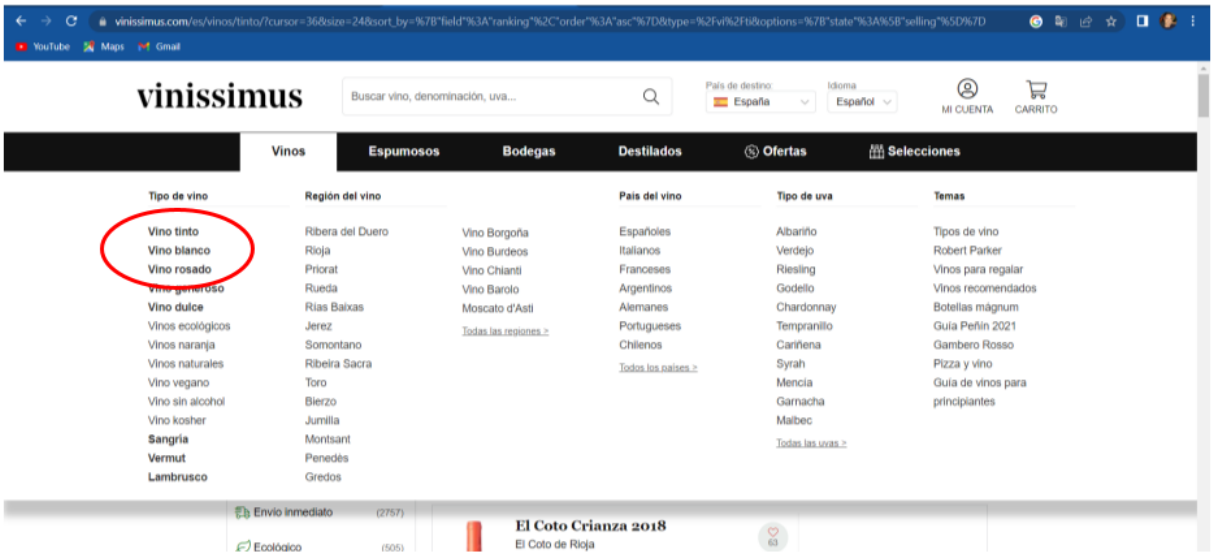
3 Descripció del dataset.

El programa que hem creat recull tota la informació disponible de cada vi que té a la venda la web vinissimus.com. Per tal de no fer un dataset molt gran i poder-hi treballar fàcilment, hem limitat la cerca a les 20 primeres pàgines dels tipus de vins més comuns: negre, blanc i rosat.
Per cada vi en la llista, el programa n'extreu la foto, el nom del vi i l'any de la collita, el productor, la regió i el país. Posteriorment n'extreu totes les varietats de raïm del que es composa i la valoració del vi pels usuaris de vinissimus.com: concretament n'extreu la puntuació en float, les estrelles en un enter d'1 a 5 i, finalment, el nombre d'opinions que el vi ha generat a la web. El programa també extreu el nombre de likes que s'han fet al vi i, en cas que tingui puntuació Parker, Peñín, Suckling o Tim Atkin (webs de referència de puntuació de vins), n'extreu la seva valoració. Així mateix, recopilem si el vi està etiquetat com a ecològic.
Finalment en recopilem el preu tenint en compte que aquest pot tenir un preu normal, un preu amb oferta (per exemple si compres 3 ampolles val X€) i un preu amb

descompte. El programa també recopila si el volum de l'ampolla no és l'estàndard (poden ser també ampolles Magnum d'1,5L o petites de 0,37L).

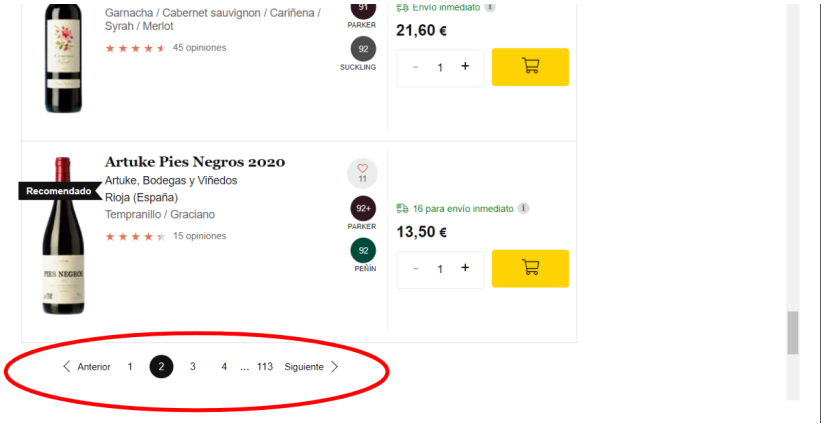
4. Representació gràfica.

La web de Vinissimus està dividida per tipus de vins. El nostre programa només tindrà en compte els vins negres, blancs i rosats per no generar un dataset massa gran. Podem veure la distribució del tipus de vi en la següent imatge:

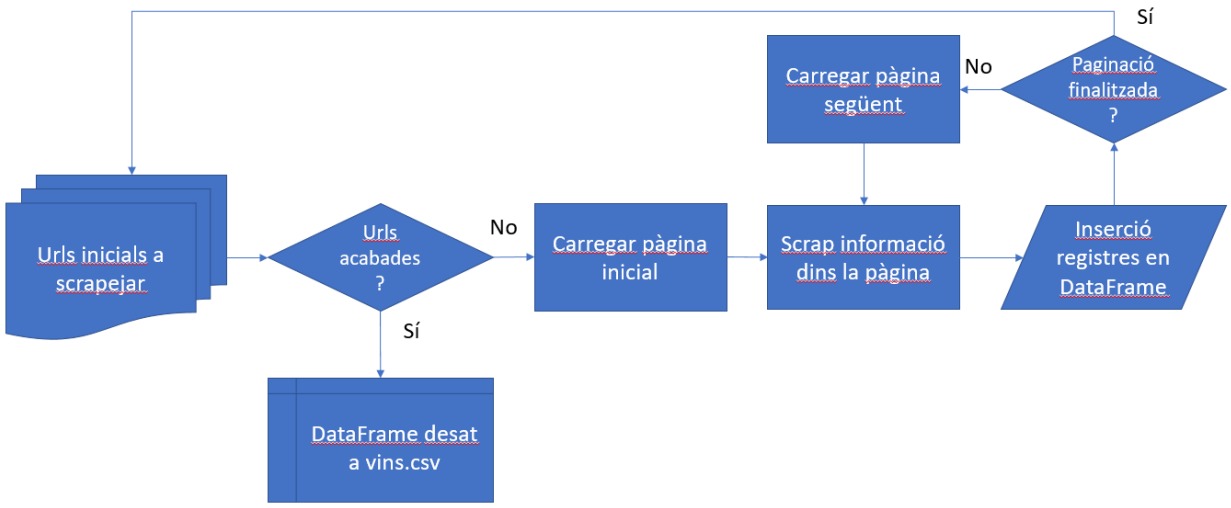


Els vins apareixen llistats en targetes. Cada targeta de vi té tota la informació que volem aconseguir i que està representada en el següent diagrama:

Finalment, cada llistat té una paginació de 24 vins per pàgina. El programa ha d'anar avançant les pàgines que li indiquem. Per tal de no fer un dataset molt gran, hem limitat la paginació a 10 pàgines per tipus de vi, obtenint un total de 1440 registres. La paginació es pot observar en la següent imatge del peu de pàgina de la web:



Per tant, el flux de treball del nostre programa serà el següent:



5. Contingut.

El dataset es genera en un moment concret en el temps i no està pensat per tenir un camp temporal per determinar l'evolució de cap dels paràmetres. L'objectiu d'aquest dataset és poder comparar puntuacions, valoracions, preus, varietats, zones i països i fer un estudi de, per exemple, quins són les regions més valorades pels usuaris de vinissimus.com o quines varietats de raïm obtenen millors puntuacions en webs especialitzades i com aquestes puntuacions es relacionen amb les valoracions per part dels usuaris.

El Dataset conté els següents camps:

- **Type**: String, indica el tipus de vi (al dataset té tres valors, "Vino tinto", "Vino blanco", "Vinos rosados y rosé" perquè hem restringit la cerca a aquests tres tipus de vi)
- **Name**: String, indica el nom del vi.
- **Year**: Integer, indica l'any de la collita. En cas que no hi hagi any especificat, és -1.
- **Cellar**: String, indica el productor del vi.
- **Region**: String, indica la regió del vi.
- **Country**: String, indica el país del vi.

- **Varieties**: String, indica les varietats de raïm que conté el vi. És un camp string que segurament caldrà processar a posteriori per separar cada varietat de raïm. En cas que no hi hagi cap varietat definida, s'omple amb un string buit "".
- **Eco**: String, indica si el vi és ecològic. Si no ho és, el camp s'omple amb un string buit "".
- **Rating**: Float, indica la puntuació del vi segons els usuaris de vinissimus.com. Aquesta puntuació no es veu a la targeta de la pàgina però es pot obtenir de l'arxiu html d'aquesta.
- **Stars**: Integer, ens indica les estrelles (arrodonides) del vi segons els usuaris de la web.
- **Opinions**: Integer, indica el nombre d'opinions publicades a la web sobre el vi en qüestió.
- **Likes**: Integer, indica el nombre de likes que han fet els usuaris al vi en qüestió.
- **Parker**: String, indica la puntuació Parker. En cas que no tingui puntuació Parker el valor és "". (Nota: s'ha canviat a posteriori el tipus d'aquest camp respecte al que s'explica al vídeo)
- **Penin**: String, indicat la puntuació Peñín. En cas que no tingui puntuació Peñín el valor és "". (Nota: s'ha canviat a posteriori el tipus d'aquest camp respecte al que s'explica al vídeo)
- **Suckling**: String, indicat la puntuació Suckling. En cas que no tingui puntuació Suckling el valor és "". (Nota: s'ha canviat a posteriori el tipus d'aquest camp respecte al que s'explica al vídeo)
- **Tim_atkin**: String, indicat la puntuació Tim Atkin. En cas que no tingui puntuació Tim Atkins el valor és "". (Nota: s'ha canviat a posteriori el tipus d'aquest camp respecte al que s'explica al vídeo)
- **Price**: String, indica el preu actual del vi (si no té oferta, el preu en negre de la targeta; si té oferta, el preu vermell de la targeta). Caldrà treballar les dades per obtenir un valor float, suprimint el símbol € de cada registre.
- **Old_price**: String, en cas que el vi estigui d'oferta, indica el preu antic del vi. S'hauran de treballar les dades per obtenir un valor float, suprimint el símbol € de cada registre.
- **Offer**: Boolean, si el vi està en oferta és True, en cas que no és False
- **Volume**: String, indica el volum de l'ampolla. Si no està especificat a la tarja del vi es sobreentén que és de 0,75 l i s'introdueix el string " / bot. 0,75 L ".
- **Image**: bytes, imatge del vi en format de bytes.

6. Propietari

Cercant a la web antecedents d'estudis sobre vins obtinguts scrapejant la web de vinissimus, hem trobat el següent codi en javascript[1]. Tot i això no hem trobat cap base de dades associada amb aquest codi, cosa que fa de la nostra base de dades la única que, obtinguda des de la web de vinissimus, que es troba disponible de manera oberta per internet. Buscant conjunts de dades sobre vins disponibles per internet, hem trobat les pàgines descrites a la referència següent[2]. Aquestes es resumeixen en dos grans grups: El primer son les pàgines web comparadores de vins, que tenen una gran base de dades amb informació sobre molts vins, incloent ratings per part de diferents organismes, el segon son les bases de dades, que hem trobat principalment a Kaggle i a data.world, on cal crear un compte per accedir a les dades. Pel que fa a anàlisis previs, se'n poden trobar molts a Kaggle. Aquí ens centrem en l'estudi següent[3] en el que s'utilitza tres models de classificació diferents per predir la qualitat dels vins utilitzant les variables contingudes al conjunt de dades "Red Wine Quality", que té les següents variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol que s'utilitzen per predir la qualitat dels vins que és la variable de sortida utilitzant diferents classificadors. Tot i així, el nostre dataset està enfocat a l'anàlisi de les valoracions i aquest darrer dataset és un estudi de resultats de laboratori.

Com que no hem trobat a la pàgina de vinissimus cap lloc on s'especifiqui cap política de web scraping, hem utilitzat el fet que, com es menciona als apunts de l'assignatura[4], quan un lloc web exposa informació públicament sense ser necessari acceptar una sèrie de termes i condicions, es considera que l'ús moderat de web scraping és adequat. Per navegar per la pàgina veiem que cal acceptar la política de cookies. Llegint-la amb detall hem vist que no hi ha cap menció a la propietat de les dades que es pot extreure de la pàgina web, només es menciona que utilitzen cookies pròpies i de tercers amb finalitats analítiques i per personalitzar l'experiència. Acceptar la política de cookies significa permetre que la pàgina utilitzi cookies de tercers per personalitzar els anuncis que apareixen a la pàgina.

Per tant, considerem que la informació és pública i un scraping moderat a la pàgina està permès. En aquest context, per complir amb els principis ètics del web scraping i

[1] https://gist.github.com/angarg12/9758840604aa053fcdb8cd36d1610b19
[2] https://gist.github.com/EmpeRoar/224b7d525f5bc481421831538e519249 és una petita base de dades a github. https://data.world/datasets/wine Aquesta pàgina conté 22 bases de dades que contenen informació sobre vins. https://www.kaggle.com/search?q=wine conté al voltant de 400 datasets sobre vins. https://www.db.wine/, https://wine81.com/ i https://www.wine-searcher.com/ son exemples de pàgines web que contenen informació sobre vins que es troben disponibles en format web.
[3] https://www.kaggle.com/code/vishalyo990/prediction-of-quality-of-wine
[4] *Web scraping*, L. Subirats Maté, M. Calvo González (2019)

evitar conflictes amb els propietaris de la web hem utilitzat la funció RobotFileParser per comprovar el robots.txt i assegurar que no scrapejem cap pàgina que estigui prohibida per part de la web i hem utilitzat un retard d'1 s entre peticions de descàrrega al servidor per evitar saturar-lo[5]. Considerant aquestes circumstàncies creiem que **els propietaris de la base de dades som els autors del projecte**.

7. Inspiració

Viníssimus és el 51è venedor de vi online a Espanya[6], té un catàleg amb més de 6000 vins i exporta a diferents països. Aquests trets, en particular la gran varietat d'oferta a la web, junt amb les valoracions per part d'organismes independents (Parker, Peñín, Suckin i Tim Atkin), fan de la seva pàgina web una gran font d'informació vinícola.

Tenim planejats els següents anàlisis: Utilitzant xarxes neuronals, extreure la forma de l'ampolla de les imatges i convertir-la en una variable discreta amb diferents valors per a cada tipus d'ampolla (coll estret, clàssica…). Comparar les valoracions de les diferents fonts entre elles i buscar correlacions amb el preu de l'ampolla, amb la forma de l'ampolla o amb la presència de promocions a la pàgina web. Una altra idea seria mirar quines varietats de raïm es troben en els vins més valorats. Aquesta informació podria ser útil per algú que vulgui plantar seps per fer vi i estigui interessat en vendre'l al preu més alt possible. Anàlogament a l'anàlisi que es pot trobar a[3], utilitzar tots els camps que tenim per predir el nombre d'estrelles o la valoració que un vi tindrà a la web per saber si pot ser interessant afegir-lo.

8. Llicència

Per escollir la llicència adequada per a la base de dades hem utilitzat l'ajuda del github[7]. Seguint les recomanacions dels apunts de l'assignatura[4], cal utilitzar la informació extreta de manera justa. Per això hem decidit utiltizar la llicència CC-BY-4.0, que permet que es modifiqui el contingut però requereix que la informació derivada de la base de dades es distribueixi sota la mateixa llicència o una similar o compatible. A més, requereix que, si es comparteix la informació continguda en el repositori es tingui en compte la feina dels autors i es referencii correctament. Es tracta d'una llicència utilitzada freqüentment en mitjans d'informació i material educatiu. És una versió nova de la llicència de Wikipedia[8].

[5] *Web Scraping with Python,* Lawson, R., Packt Publishing Ltd (2015)
[6] https://ecommercedb.com/store/vinissimus.com
[7] https://choosealicense.com/non-software/
[8] https://choosealicense.com/licenses/cc-by-sa-4.0/

9. Codi

Analitzant la web de viníssimus, veiem que hi ha una gran quantitat de vins de cada tipus i cada tipus de vi correspon a una sola url. Per això el link crawler que hem creat només explora els tipus de vi que demanem, especificant la URL de cadascun. En particular, les funcions creades per a fer l'scraping tenen com a input una llista amb les URL a scrapejar, el nombre de pàgines a les que es vol accedir per a cada tipus de vi i el delay entre descarregues d'una mateixa URL. Per realitzar el crawling de la pàgina hem utilitzat el Chrome webdriver de Selenium, que ens permet navegar interactivament per vinissimus.com. Aquesta navegació interactiva consta de diferents passos: El primer és la descàrrega de la pàgina i l'acceptació de la política de cookies. Sense aquesta acceptació no es pot procedir dins la pàgina. El següent pas és comprovar si s'ha introduït un nombre vàlid (>1) de pàgines a scrapejar. En cas que no s'hagi fet, el crawler busca el nombre total de pàgines en la URL a scrapejar i la funció escrapeja la URL sencera. Una vegada hem determinat el nombre de pàgines a scrapejar, el crawler es descarrega l'html de la pàgina i crida l'scraper que n'extreu la informació. La pàgina de vinissimus a vegades crea una finestra on pregunta a l'usuari si es vol crear un compte. Aquesta pàgina s'ha de tancar quan apareixi per poder seguir amb el crawling. Hem realitzat aquest pas identificant l'adreça XML del botó que ens permet tancar la finestra. El següent pas és buscar el botó "Siguiente" a la part inferior de la pàgina i apretar-lo. Una vegada fet això, si el nombre de pàgines scrapejades dins de la mateixa URL és inferior al màxim, es descarrega l'arxiu html de la pàgina següent i es crida l'scraper, que afegeix les dades que extreu al data frame de Pandas on ha abocat els resultats anteriors. El procés es repeteix fins que s'ha scrapejat el nombre màxim de pàgines demanat.

Per evitar ser bloquejats per la pàgina per saturar-ne els servidors, hem utilitzat la classe definida a[5], que permet emmagatzemar el moment de l'última descàrrega de cada URL i esperar el temps desitjat entre descàrregues consecutives. Una altra mesura que hem pres per evitar ser bloquejats ha estat utilitzar el paquet random_useragent per crear user agents aleatòris que canviin per a cada URL diferent que s'executi (considerant que a la llista d'entrada no hi ha URLs repetides).

Per millorar la modularitat del crawler, l'hem implementat de manera que accepti el nom de la funció que farà scraping. Això vol dir que pot funcionar amb qualsevol scraper que tingui només un input obligatòri: L'arxiu html.

Pel que fa a la fució scrapper_wine, el primer que fa la funció és construir un objecte soup mitjançant el constructor BeautifulSoup() amb l'html a parsejar i el parsejador, que en nostre cas serà "html.parser". Creem també un pandas DataFrame on anirem guardant la informació parsejada. La funció seguidament localitza l'objecte <div

class=”list large”> que és la taula que conté totes les etiquetes de vins. Aquest objecte serà treballat a posteriori. Seguidament recollim el tipus de vi que estem scrapejant buscant l’objete <h1 class=”section-heading line-bottom”> que és el títol de la pàgina on trobem el tipus de vi (**type_of_wine**), mitjançant la funció find() i extraient-ne el text mitjançant un get_text(). De la taula amb etiquetes s’ha intentat extreure’n cada etiqueta amb tot el contingut per separat, però ens ha resultat impossible. Per obtenir les dades hem hagut d’extreure tres taules per separat (la taula d’imatges, la taula de preus i la taula d’informació) i s’ha pogut comprovar que l’extracció de les dades es pot fer en el mateix ordre en les tres taules i, per tant, podem scrapejar la informació de cada vi en la mateixa posició. Mitjançant la funció find_all extraiem tots els elements de cada taula (etiquetats mitjançant un “div” i distingits per les class "product-image desktop", “quantity-widget small” i “info”).

El scrapper itera llavors posició a posició en les taules per extreure’n el contingut:

- **year, name**: a la taula info_table, fem una cerca mitjançant la funció find() del tag <h2 class=”title heading”> , en recull el text amb get_text(), i mitjançant un split() i rstrip() extraiem un string que conté l’any en els quatre primers caràcters (si en té) i el nom del vi seguidament. Com que hi ha vins que no tenen anyada, la manera de distingir-los ha estat mitjançant un try intentar convertir els quatre primers caràcters a integer. Si dona error, es va a l’except i vol dir que no hi ha any. En cas que no hi hagi any, el valor es fixa a -1.

- **cellar** : per obtenir el productor del vi de la taula info_table, cerquem el tag mitjançant un find() i posteriorment un get_text()

- **region, country**: la regió i el país els localitzem a la taula info_table dins el mateix tag mitjançant un find i posteriorment un get_text(). Obtenim el valor de la regió fent-ne un split pel caracter “(“, agafant-ne e primer string ([0]) i fent un strip(). Obtenim el valor country agafant el segon string del split(“(“) i agafant tot el string excepte el darrer caracter [:-1], que és el parèntesi de tancament

- **varieties**: amb el camp de varietats de raïm hem de tenir en compte que no tots els vins indiquen la seva varietat. Si intentem parsejar el tag “varieties” de la taula info_table i BeautifulSoup no el troba, es produeix una excepció que atura l’execució del programa. Per solucionar aquest problema, utilitzem l’estructura try/except. El programa intenta trobar el tag mitjançant un find() i, si existeix, n’extreu un string utilitzant get_text(). Si no el troba, salta al except i insereix un string buit al camp **varieties**.

- **eco**: de la taula info_table n'obtenim el camp que ens indica si un vi és ecològic o no. En aquest punt ens passa el mateix que en el camp anterior. Pot ser que hi sigui o pot ser que no, cancel·lant l'execució del script. Per evitar-ho utilitzem altra vegada un try/except, que intenta localitzar el tag . En cas que el vi no tingui etiqueta eco, s'introdueix un string buit al camp **eco**.

- **rating, stars**: a continuació cercarem els valors de la puntuació que els usuaris de vinissimus han donat al vi i els localitzarem a la taula info_table. Per tant, primer localitzem el tag utilitzant mitjançant un find. Un cop localitzat aquest tag, cerquem el contingut de l'etiqueta "style" mitjançant attrs["style"]. Obtenim un string amb el format "--rating:4.42857; --numStars:5;". Per obtenir la puntuació (rating), primer fem un split del string i el separem utilitzant els caracters "--". Obtenim tres strings, dels quals el primer (pos 0) està buit. En el segon (pos 1) apliquem un regex per localitzar el float i n'obtenim el valor de **rating**. En el tercer (pos 2) apliquem un regex per obtenir-ne el integer que ens donarà el valor de **stars**. Cal notar que el valor **rating** és un valor ocult que no es motra a la pàgina web.

- **opinions**: a continuació scrapegem de la taula info_table el nombre d'opinions d'usuaris de vinissimus sobre aquest vi. Localitzem el tag i, mitjançant un regex, n'extraiem l'enter que inserirem al camp.

- **likes**: de la taula info_table també n'extraurem el nombre de likes del vi. Aquests es troben en el tag <div class="class":"styles_likes__Jvb7B"> mitjançant un find() i un get_text(). Com que pot ser que el tag no existeixi, utilitzarem un try/except i, si no existeix, fixarem el valor **likes** = 0.

- **parker, penin, suckling, tim_atkin**: a continuació, de la taula info_table extreurem les puntuacions de les webs especialitzades en puntuar vins. Els tags que contenen aquesta informació poden ser-hi o no, per això en cada cas utilitzarem un try/except i si no hi ha el tag, s'inserirà un string null. Cercarem els següents tags:
    - **parker**: <span style="'--score-color:var(--pk);'>
    - **penin**: <span style="'--score-color:var(--penin);'>
    - **suckling**: <span style="'--score-color:var(--suckling);'>
    - **tim_atkin**: <span style="'--score-color:var(--atkins);'>

En tots es casos farem un find() i un get_text() per obtenir-ne el valor en format String.

- **price, previous, offer**: aquests atributs de preu els treurem de la taula price_table. La web indica si es tracta d'un preu estàndard o d'una oferta. En el segon cas, se'ns indica el preu antic i el preu d'oferta. Hem agafat la següent definició:
    - Preu normal: Si no hi ha cap oferta, el preu està inclòs en el tag <p class="price uniq small">. Aquest tag, en cas que hi hagi oferta, no existeix. Per tant, hem d'aplicar un try/except. Dins el try, si trobem aquest tag, introduim el valor del get_text() a la variable **price** i marquem com a False la variable booleana **offer** i introduim un text null a la variable **previous**.
    - Oferta: Si hi ha oferta, el primer que fem és determinar el preu antic. Intentem mitjançant un try localitzar el tag <p class="price old small"> i introduim el seu contingut a la vairable **previous**. Si no existeix entrem al except i hi introduim un text buit. A continuació mirem de localitzar un dels dos tags de preus en oferta (mitjançant un try/except, ja que si no hi ha un hi haurà l'altre, perque estem dins del except que ens indica que hi ha oferta). Hi ha dos tipus de preus en oferta, que no distingirem en el dataset, indicats en els tags <p class="dto small"> i <p class="special small">. Agafarem el text del tag que correspongui mitjançant get_text() i l'introduirem a la variable **price** i seguidament marcarem la vairable **offer** com a True.

- **volume**: de la taula price_table també n'extrurem el volum de l'ampolla. Per defecte, el volum estàndard és de 750 ml. Si el volum és diferent, està marcat en el tag . Mitjançant un try/except intentem obtenir aquest tag. Si el trobem mitjançant un find(), n'extraiem el contingut utilitzant el mètode get_text() i l'insertem a la variable **volume**. En cas que no hi sigui, insertem el text en el mateix format però indicant que és de 0,75l " / bot. 0,75 L".

- **image**: finalment recollirem la imatge del vi de la taula images_table. Mitjançant requests, cercarem la url de la imatge utilitzant la comanda següent:
  get(images_table[0].find('img')['src'])
  i aplicarem el mètode content per obtenir-ne la imatge en format bytes, que inserirem a la variable **image**.

Un cop tenim totes les variables que volem, les inserim a una nova fila en el DataFrame pandas, que es va omplint fins que tots els elements de les taules han estat scrapejats i, llavors, retorna el dataframe a la funció crawler_buttons.

Enllaç a la pàgina de github on es troba el projecte:
https://github.com/PepIngla/webScrapingWines

10. Dataset

DOI del dataset a Zenodo: https://doi.org/10.5281/zenodo.7339749

11. Vídeo

Enllaç que permet accedir al vídeo emmagatzemat a Google Drive:
https://drive.google.com/file/d/14zDtcctewImmgJvKfWloYTcFXyinrqrR/view?usp=sharing

Taula de contribucions:

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