UNIVERSITAT OBERTA DE CATALUNYA

Web scraping

Tipología y ciclo de vida de los datos

Isabel González Valle;Cristina Merino García de la Reina

Abril 2020

Contenido

1. Contexto 2
2. Definir un título para el dataset 3
3. Descripción del dataset 4
4. Representación gráfica 5
5. Contenido 9
6. Agradecimientos 11
7. Inspiración 12
8. Licencia 13
9. Código 15
10. Dataset 20
11. Entrega 21
12. Contribuciones al trabajo 22
Bibliografía 23

1. Contexto

Explicar en qué contexto se ha recolectado la información. Explique por qué el sitio web elegido proporciona dicha información.

Se ha recolectado información de los productos sin gluten que oferta el supermercado Eroski aprovechando que en su web ofrece la posibilidad de visualizar todos los productos que no contienen esta proteína.

Eroski en su supermercado online identifica ciertas marcas para agrupar productos con unas características específicas:

Elegimos extraer la información de todos los alimentos sin gluten que este supermercado online ofrece a sus clientes para poder generar un conjunto de datos que permita conocer la lista de alimentos que se podrían comprar en esta web y de cara a que puede servirnos para realizar comparativas con otros supermercados o crear listas de la compra para personas celiacas o con cierta intolerancia o sensibilidad al gluten. Nos gustaría hacer hincapié en que la web elegida proporciona la lista de alimentos sin gluten de todos los ámbitos de la alimentación (carnes, lácteos, bollería…) y no se centra solamente en productos que tradicionalmente se elaboran con gluten como puede ser la pasta o el pan, como ocurre en la mayoría de secciones sin gluten de otros supermercados, donde la etiqueta “sin gluten” solo aparece en productos específicos.

Dado que la estructura de la página es la misma en todos los casos presentados, con este mismo código se podría llegar a extraer la información del resto de grupos de productos en el caso de que nos interesase a posteriori centrarnos en “Productos ecológicos” o cualquier otro, cambiando únicamente la url de entrada.

2. Definir un título para el dataset

Elegir un título que sea descriptivo.

El título que hemos elegido para este conjunto de datos es: “AlimentosSinGluten”. En principio no se valora incluir el nombre del supermercado del que se extrae la información, puesto que este dataset podría ser ampliado realizando web scraping de otras páginas con información de este tipo de productos.

3. Descripción del dataset

Desarrollar una descripción breve del conjunto de datos que se ha extraído (es necesario que esta descripción tenga sentido con el título elegido).

El conjunto de datos extraído contiene información de detalle de los productos sin gluten que podemos encontrar en este supermercado, tales como datos relativos a cómo se presenta el producto al público, el precio, si el producto se encuentra o no en oferta, valor nutricional y valoración de los clientes para el producto en concreto.

En principio, no se identifica en el título del dataset el supermercado online accedido, puesto que esta información podría llegar a ampliarse con la información de otros supermercados online.

Del conjunto de datos que nos proporciona la página, podemos extraer, entre otras cosas, la siguiente información:

- Product description: Nos ofrece la descripción del producto (nombre + cantidad)
- Product name: Nombre del producto
- Product quantity: Cantidad del producto
- Kilo evaluation: Indica el precio del kilo
- Offer description: Muestra el precio actual del producto y en caso de estar en oferta, también muestra el precio anterior.
- Nutriscore: clasifica los alimentos en función de su calidad nutricional en cinco etiquetas, donde la A es la más saludable hasta la E, menos saludable.

4. Representación gráfica

Presentar una imagen o esquema que identifique el dataset visualmente.

Pensamos que esta fotografía puede definir perfectamente tanto el origen como los datos que pueden estar contenidos en el mismo.

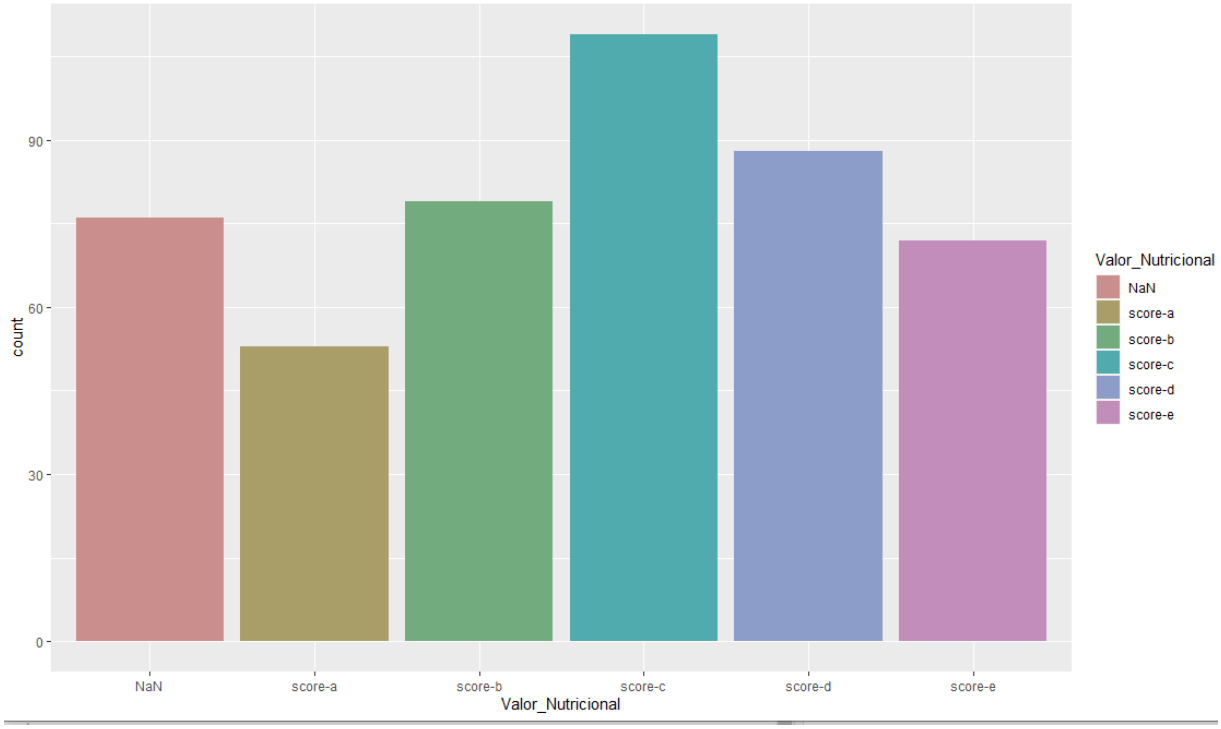
Ilustración 1 - Lineal productos sin gluten

(https://cronicaglobal.elespanol.com/, 2019)

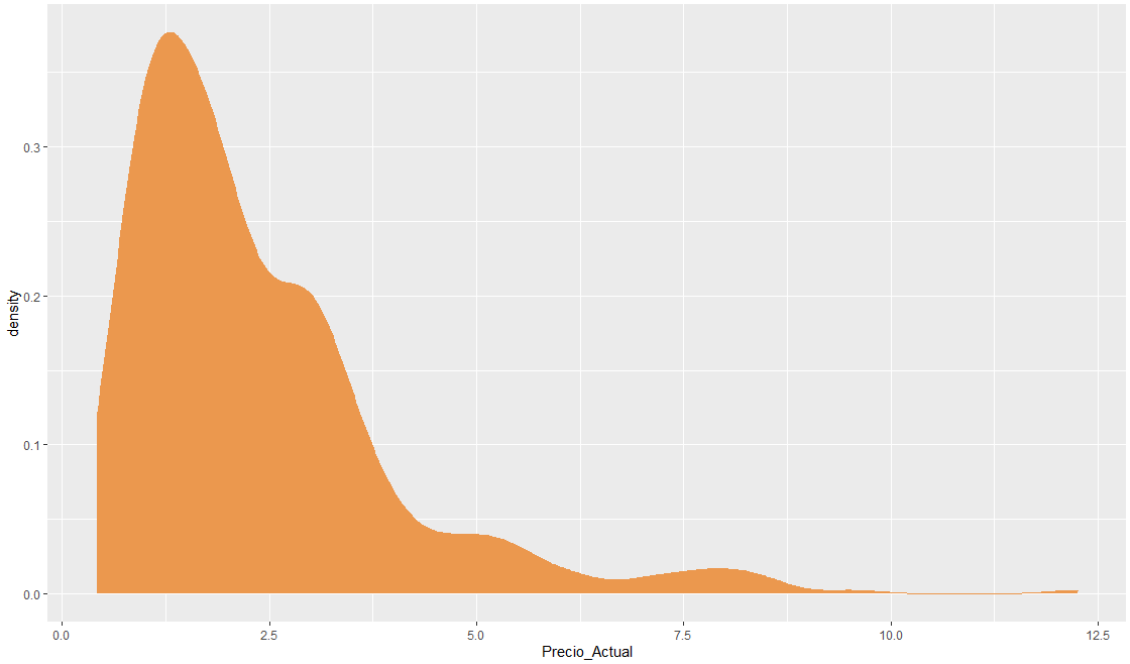
Una vez realizada la extracción de datos, procedemos a cargarlos en R Studio para realizar una breve representación de los mismos mediante estadística descriptiva.

```
Cantidad_Base Precio_Unidad_Base NÃºmero_Valoraciones  ValoraciÃ³n
 KILO  :437    Min.   : 0.58      Min.   : 0.000       Min.   :0.000
 LITRO : 15    1st Qu.: 4.18      1st Qu.: 0.000       1st Qu.:0.000
 NaN   : 24    Median : 8.66      Median : 0.000       Median :0.000
 UNIDAD:  1    Mean   :11.12      Mean   : 1.046       Mean   :1.793
               3rd Qu.:14.17      3rd Qu.: 1.000       3rd Qu.:5.000
               Max.   :98.63      Max.   :20.000       Max.   :5.500
               NA's   :24
 Precio_Anterior Precio_Actual     Fecha_ExtracciÃ³n
 Min.   :1.720   Min.   : 0.420   2020-04-10:477
 1st Qu.:3.092   1st Qu.: 1.190
 Median :3.580   Median : 1.950
 Mean   :3.545   Mean   : 2.362
 3rd Qu.:4.032   3rd Qu.: 2.990
 Max.   :5.300   Max.   :12.250
 NA's   :473
```

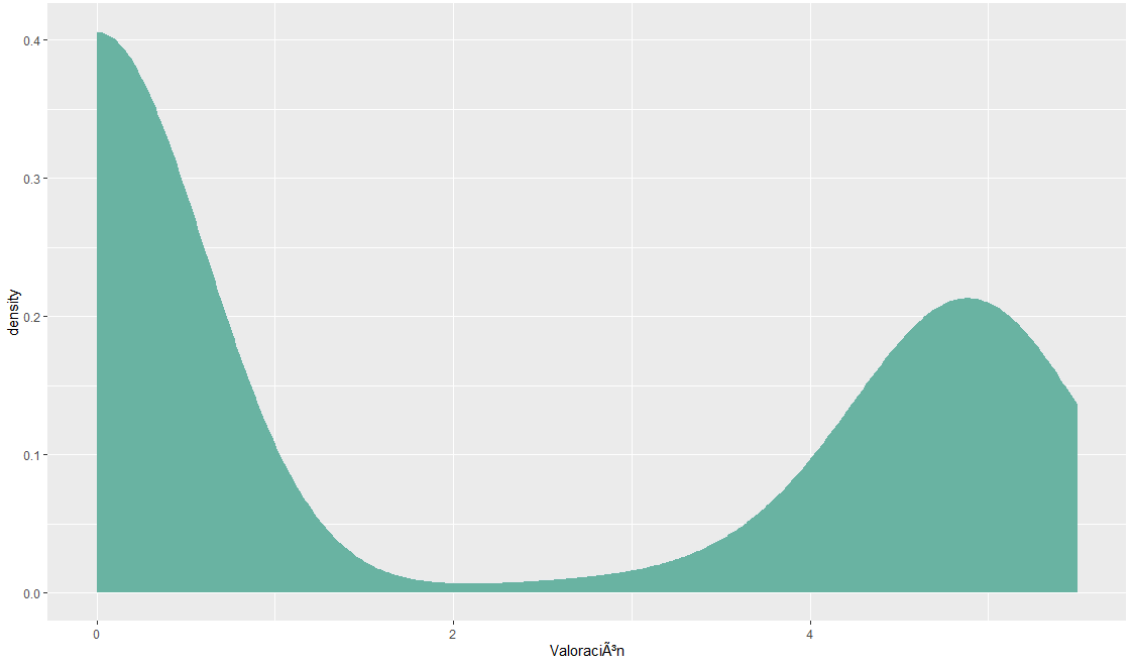
Vemos a continuación un gráfico de barras con la distribución de los valores nutricionales de este tipo de productos. Comprobamos que, aunque podríamos llegar a pensar que la mayor parte de los productos tuviesen valor nutricional entre a y c, al ser productos en principio más saludables, la realidad es que la gran parte de los productos se encuentran en los valores nutricionales intermedios entre el valor c y d. Tenemos además muchos productos que no tienen ese dato informado (esto es así porque no todos los productos ni todas las marcas proporcionan esta información).



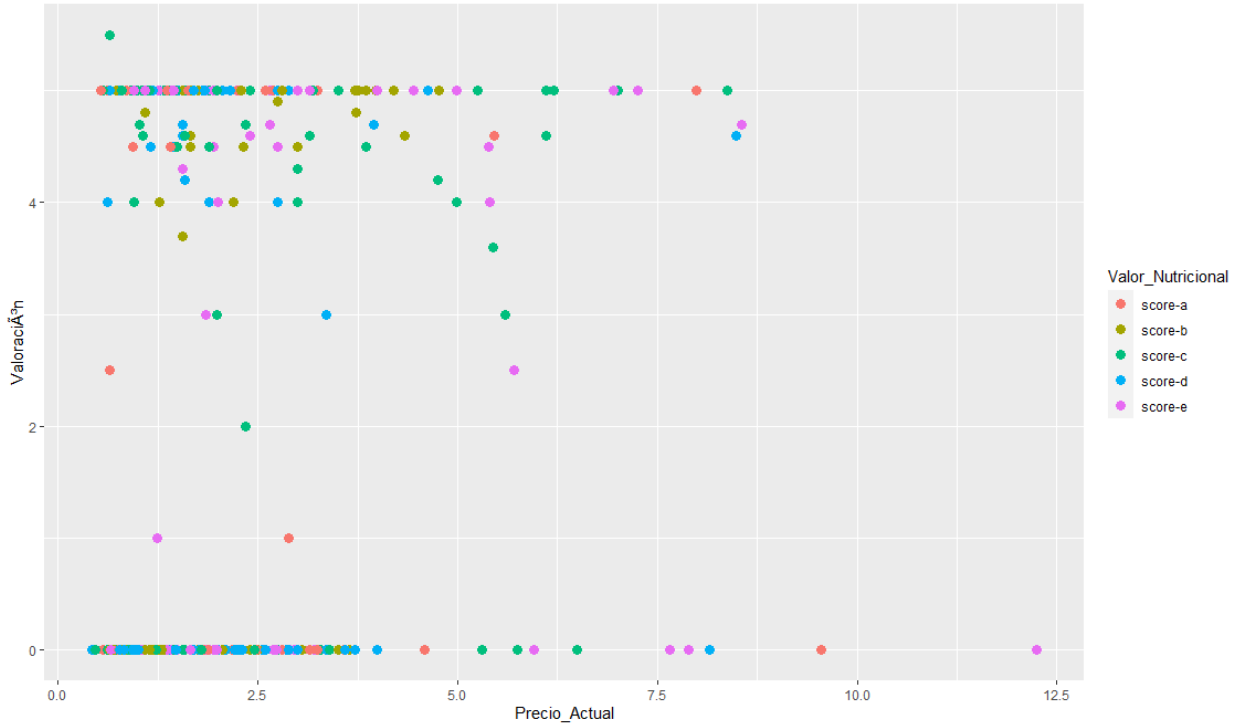
Mostramos también la relación de precios de los productos sin gluten, en este caso mediante un diagrama de densidad. El rango de precios se encuentra entre 0,5€ y 4€, por lo que podemos suponer que son precios normales, aún sin realizar una comparación con el resto de productos “con gluten” de características similares:



En cuanto a la valoración de los clientes, vemos que hay dos tendencias diferenciadas, por un lado, las puntuaciones muy bajas a los productos y, por otro lado, las puntuaciones altas. Es decir, nos puede hacer suponer que la valoración más frecuente se realiza cuando el producto no es del agrado del cliente.



Por último, podemos hacer una comparación entre el precio actual del producto, la valoración de los usuarios y el valor nutricional. No se ve una tendencia o una relación clara entre estos valores, puesto que están distribuidos casi de manera uniforme entre las valoraciones alta y bajas. Podemos identificar que los productos con score-a no son los más caros, ni son los mejor valorados en términos generales.



5. Contenido

Explicar los campos que incluye el dataset, el periodo de tiempo de los datos y cómo se ha recogido.

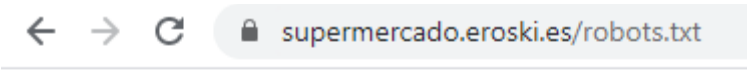
Del conjunto de datos podemos extraer la siguiente información:

- **Id:** Identificador secuencial del producto. (Numérico)
- **Artículo**: Nos ofrece la descripción del producto (nombre + cantidad). (Carácter)
- **Nombre**: Identifica el producto. Es la descripción del producto. (Carácter)
- **Presentación**: Indica el formato en el que se presenta el producto. Puede ser peso o unidades. (Carácter)
- **Valor_Nutricional**: Etiquetado internacional que indica el valor nutricional de los alimentos, se representa con 5 colores, desde el verde al rojo, asociados a 5 letras, de la A a la E. (Factor)
- **Cantidad_Base**: Peso base de referencia (Kilo, Litro, unidad, …) (Factor)
- **Precio_Unidad_Base**: Precio de la cantidad base para poder comparar en el mismo producto o tipo de producto los precios en diferentes presentaciones. (Float)
- **Número_Valoraciones**: Número total de valoraciones de usuarios. (Numérico)
- **Valoración**: Puntuación dada por los consumidores (del 1 al 5) (Float)
- **Precio_Anterior**: Si el producto se encuentra en oferta, se muestra el precio anterior. (Float)
- **Precio_Actual**: Precio de venta actual. (Float)
- **Fecha_Extracción**: Fecha en la que se realiza el volcado de datos. (Date)

Como este dataset tiene dependencia temporal se incluye un campo fecha que va a identificar la fecha de la extracción de los datos.

Los datos sin información se han rellenado a “NaN”.

Para la recogida de datos primero se realizó una visualización de la página web completa, identificando la información contenida en la misma. Se revisó el fichero robots.txt, para confirmar que el acceso a la página elegida podía realizarse sin problemas. Adicionalmente se revisaron los términos y condiciones de la plataforma.




```
User-agent: *
Disallow: /*/procesoEnvio*
Disallow: /*/SendImage/*
Disallow: /*/home-no-logado.shtml.html
Disallow: /*/ficha-producto.shtml.html
Disallow: /*/mycart/
Disallow: /*/search/
Disallow: /*/mylists/
Disallow: /*/listdetail/
Disallow: /*/filter/*/
Disallow: /*/bookingdelivery/
Disallow: /checkout/
Disallow: /myfavourites/
Disallow: /*/myorders/
Disallow: /*/combineoffer/*/
Allow: /
```

La página de productos sin gluten se va cargado de manera dinámica al realizar el scroll-down, por ese motivo, para llegar a cargar toda la información se determina el uso de la librería Selenium. En el código se ha hecho uso del navegador Chrome de Google y por lo tanto se utilizan los drivers de este y la opción de ejecución en segundo plano. Podría utilizarse cualquier otro navegador, como por ejemplo Firefox, descargando y llamando a los drivers correspondientes.

Para la extracción de la información, una vez recorrida toda la página y habiendo recopilado toda la información de los productos, se ha utilizado la librería Beautifulsoup. Esta librería permite hacer el parseo, de manera sencilla, de documentos en HTML. Se ha buscado la etiqueta que nos permite sacar la información del producto y se realizado el bucle correspondiente para recorrer el total de ítems y recuperar los datos necesarios, navegando en las etiquetas tanto hacia abajo (drill-down) como a nivel de padres o hermanos. La información recuperada se traslada a una variable de diccionario.

Por último, se ha trasladado la información recuperada en la variable diccionario a un dataset, para posteriormente volcarlo a un fichero csv.

6. Agradecimientos

Presentar al propietario del conjunto de datos. Es necesario incluir citas de investigación o análisis anteriores (si los hay).

El propietario del conjunto de datos es EROSKI. S. COOP.

El Grupo Eroski es una empresa cooperativa de distribución de las más importantes de España perteneciente a la Corporación Mondragón. Tiene su sede en la localidad vizcaína de Elorrio en el País Vasco, España.

Se fundó en el año 1969 y es una de las empresas de distribución más importante de España contando con una plantilla de más de 35 000 trabajadores repartidos por toda España. La empresa cuenta con alrededor de 2000 establecimientos de diferentes marcas, entre las que se incluyen los hipermercados "Eroski", supermercados "Eroski City" y "Eroski Center", supermercados "Eroski Merca", supermercados "Cash Record", supermercados "Caprabo", supermercados "Familia", autoservicios "Aliprox", "Eroski Viajes", "Viajes Caprabo", "Eroski Óptica", "Estaciones de Servicio Eroski" y "Tiendas de Deporte FORUM". (Wikipedia, s.f.)

7. Inspiración

Explique por qué es interesante este conjunto de datos y qué preguntas se pretenden responder.

La Enfermedad Celíaca (EC) es una enfermedad multisistémica con base autoinmune provocada por el gluten. Se estima que el 1% de la población la padece aunque diversos estudios indican que sobre el 75% de los casos están sin diagnosticar.

A día de hoy no existe cura para esta enfermedad, por lo que el único tratamiento para este colectivo es seguir una dieta sin gluten de por vida.

Aunque cada vez es más fácil encontrar alimentos sin gluten, no todos los supermercados disponen de un etiquetado reconocible para los productos que no son elaborados específicamente para los celiacos, por lo hacer la compra no resulta una tarea sencilla.

Además de la falta del etiquetado sin gluten, debemos sumar otro problema más: el precio.

La Federación de Asociaciones de Celiacos de España, más conocida por sus siglas FACE, elabora cada año un informe de precios en el que se muestra con carácter semanal, mensual y anual el gasto extra que supone para una persona celíaca seguir la dieta sin gluten.

Las conclusiones del informe de precios 2020 son:

> *"Teniendo como base los resultados obtenidos se puede concluir que una familia con una persona celiaca entre sus miembros, que tenga un patrón alimentario con aporte calórico de 2000 a 2200 kcal, tendrá un* ***incremento estimado en la adquisición de la cesta de la compra de 18,97€ a la semana, 75,89 € al mes, y de 910,73 € al año****, en relación con otra familia que adquiera productos con gluten."* (FACE, 2020)

Por lo tanto, nuestro conjunto de datos pretende facilitar la compra al colectivo celiaco identificando qué productos son los que pueden consumir con total tranquilidad e informando de su precio posibilitando así la realización de comparativas con otros supermercados.

8. Licencia

Seleccione una de estas licencias para su dataset y explique el motivo de su selección:

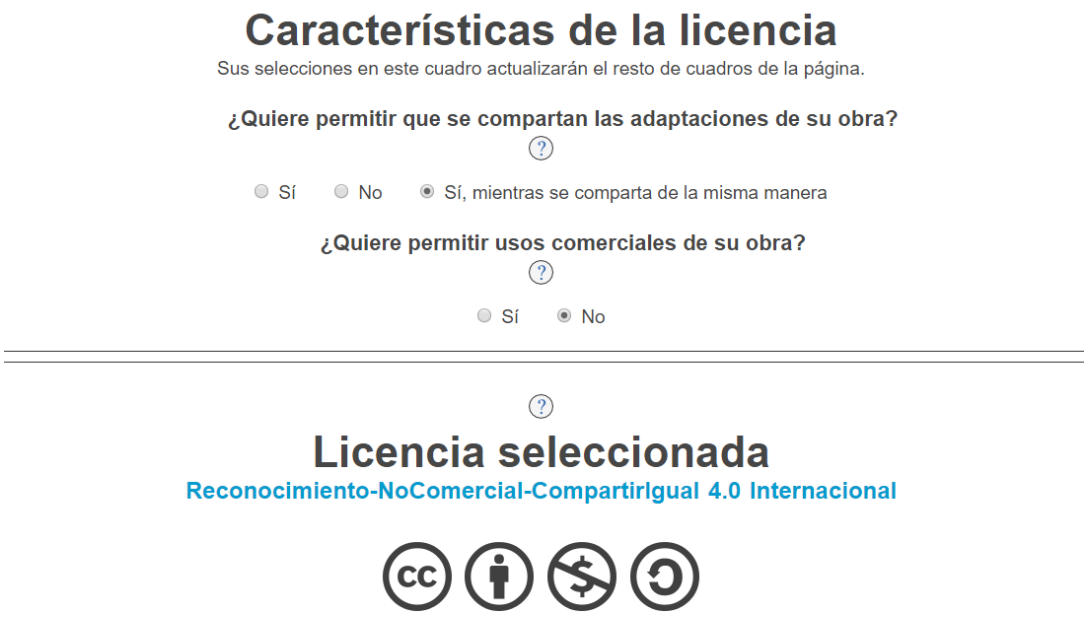
- Released Under CC0: Public Domain License
- Released Under CC BY-NC-SA 4.0 License
- Released Under CC BY-SA 4.0 License
- Database released under Open Database License, individual contents under Database Contents License
- Other (specified above)
- Unknown License

El propietario de los datos es EROSKI. Incluimos a continuación los términos y condiciones.

https://www.eroski.es/terminos-y-condiciones-de-uso/

La licencia que se ha elegido para la creación y publicación del conjunto de datos es la CC BY-NC-SA 4.0 por ser la que, bajo nuestro punto de vista, más se ajusta al trabajo que se ha realizado.

Para identificar que dicha licencia es la que mejor se adapta a nuestro trabajo, hemos realizado la encuesta que se incluye en Creative Commons. Este es el resultado en base a las características que creemos más adecuadas:



- CC BY-NC-SA 4.0
  Esta licencia permite a otros distribuir, remezclar, retocar, y crear a partir de tu obra de modo no comercial, siempre y cuando te den crédito y licencien sus nuevas creaciones bajo condiciones idénticas.

9. Código

Adjuntar el código con el que se ha generado el dataset, preferiblemente en Python o, alternativamente, en R.

Se incluye el código Python correspondiente a esta práctica y el repositorio GitHub en el que se encuentra.

```
#Cargamos las librerías necesarias
import requests
from urllib.request import urlopen
from bs4 import BeautifulSoup
from urllib.error import HTTPError
import pandas as pd
from datetime import date
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.firefox.options import Options
import time

#Creamos una función para la gestión de errores

def getTitle(url):
    try:
        html = urlopen(url)
    except HTTPError:
        return None
    try:
        bsObj = BeautifulSoup(html.read(), "html.parser")
        title = bsObj.body.h1
    except AttributeError:
        return None
    return title


#Identificamos la url que vamos a rastrear
str = "https://supermercado.eroski.es/es/supermercado/SinGluten/"

# Comprobamos el acceso y sacamos el itulo de la página por pantalla
title = getTitle(str)
if title == None:
    print("Title could not be found")
else:
    print(title)
```

```
#podríamos meter en el else toda la lógica


#Creamos la función de scroll infinito (cuando llamemos a esta función re
correrá todas las páginas disponibles al hacer el scroll hacia abajo
def scroll(driver, timeout):

    last_height = driver.execute_script("return document.body.scrollHeigh
t")
    # el tiempo de espera vendrá definido como uno de los valores de la f
unción
    Scroll_Wait = timeout

    i=0 # contador de scroll
    while True:
        # execute script to scroll down the page
        driver.execute_script("window.scrollTo(0, document.body.scrollHei
ght);")

        time.sleep(Scroll_Wait)
        # Calculate new scroll height and compare with last scroll height
        new_height = driver.execute_script("return document.body.scrollHe
ight")
        i=i+1
        print ("scroll número", i)
        if new_height == last_height:
            break
        last_height = new_height
    print ("fin del scroll")


#Por si por tiempo de espera no funcionase el While, podemos forzar la le
ctura total con un bloque for.
#def scroll(driver, timeout):

#    for i in range (1,40):
#        # execute script to scroll down the page
#        driver.execute_script("window.scrollTo(0, document.body.scrollHe
ight);var lenOfPage=document.body.scrollHeight;return lenOfPage;")
        # sleep
#        time.sleep(timeout)

# ****** Ejecución con PhantomJS y Firefox *******
#driver = webdriver.PhantomJS("C:/phantomjs.exe")
```

```
#options = Options()
#options.headless = True #No abrimos el navegador
#driver = webdriver.Firefox(firefox_options=options, executable_path = "C
:/geckodriver.exe")
# *************

# ******** Ejecución con Chrome *******
#Utilizamos las opciones de Chrome para abrir el navegador en segundo pla
no
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument("headless")

driver = webdriver.Chrome("C:/chromedriver.exe", chrome_options=chrome_op
tions) #la ruta donde tengamos el ejecutable

#Creamos una espera previa en el caso de que nos devuelva algún error dar
 un error
driver.implicitly_wait(100)
#abrimos la página
driver.get(str)

#Llamamos a la función de scroll con una espera entre scroll hacia abajo
de 7 sg
scroll (driver, 7)

#Parseamos el html resultante con toda la información
soup = BeautifulSoup(driver.page_source, "html.parser")

#cerramos el driver
driver.close()


#Extraer el texto de la etiqueta que estamos buscando
nameList = soup.findAll("div", {"class":"product-description"})

#Inicializamos el diccionario de datos donde cargaremos la información
data = []
#recorremos todas las etiquetas para crear el dataset
for i, name in enumerate(nameList):
    #Sacar el valor nutricional. Hay que buscarlo en el texto
    #<a class="nutriscore score-
c" que está un nivel por encima de donde estamos ahora
```

```
    nutrition = name.find("div", {"class":"description-
text"}).parent.parent
    if nutrition.find("a", {"class":"nutriscore"})==None:
        nutrition_score = "NaN"
    else:
        nutrition = nutrition.find("a", {"class":"nutriscore"})
        nutrition_score = nutrition.get("class")[1]

    #Sacamos la información del producto (nombre y presentación)
    title = name.find("h2", {"class":"product-title product-title-
resp"}).text
    title = title.strip()
    product_name = title.split(',')[0] #Nos devuelve el nombre sin el pes
o (Filetes de lomo adobado de cerdo extrafino EROSKI)
    product_name = product_name.strip()
    # product quantity es presentación del producto
    product_quantity = title.split(' ',1)[1].split(',')[1].strip() #Nos d
evuelve la cantidad (bandeja 300 g)
    product_quantity = product_quantity.strip()

    # Las siguientes etiquetas no se encuentran informadas en todos los c
asos, lo que hacemos es,
    #en el caso de que no existan poner valos NaN y en otro caso recupera
r la información
    if name.find("span", {"class":"quantity-product"})==None:
        quantity_product = "NaN"
    else:
        quantity_product = name.find("span", {"class":"quantity-
product"}).text
        quantity_product = quantity_product.split(' ')[1].strip()

    if name.find("span", {"class":"price-product"}) == None:
        price_product = "NaN"
    else:
        price_product = name.find("span", {"class":"price-product"}).text
        price_product = price_product.split(' ')[0].replace(",",".").stri
p()

    #Rating_stats - número valoraciones usuarios
    rating_stats = name.find("div", {"class":"ratingTitle"}).span.text
    rating_stats = rating_stats.replace ("(","").replace(")","")

    #Rating - Valoración del producto
    rating = name.find("div", {"class":"ratingSubtitle"}).get_text()
```

```
    rating = rating.split('de')[0]
    rating = rating.replace(",",".").strip()
    #Porcentaje de valoración de los clientes.


    #Podemos incluir aquí las ofertas
    #En los precios tenemos que dejar únicamente el precio
    if name.find("span", {"class":"price-offer-before"})==None:
        price_before = "NaN"
    else:
        price_before = name.find("span", {"class":"price-offer-
before"}).get_text()
        price_before = price_before.replace(",",".").strip()

    price_now = name.find("span", {"class":"price-offer-
now"}).get_text()
    price_now = price_now.replace(",",".").strip()
    #print(name.get_text())

    #append dict to array
    data.append({"Id": i, "Artículo" : title, "Nombre" : product_name, "P
resentación" : product_quantity, "Valor_Nutricional" : nutrition_score,
                 "Cantidad_Base" : quantity_product, "Precio_Unidad_Base"
 : price_product,"Número_Valoraciones" : rating_stats,
                 "Valoración": rating, "Precio_Anterior": price_before, "
Precio_Actual" : price_now, "Fecha_Extracción": date.today()})

#print (data)


#Como último paso trasladamos los datos a un dataframe para poder volcarl
os a un csv
df=pd.DataFrame(data)
df.to_csv('AlimentosSinGluten.csv', index=False, encoding='utf-8')
```

https://github.com/igonzalezvalle/PRA1-Web-Scraping/blob/master/src/scraping.py

10. Dataset

Publicación del dataset en formato CSV en Zenodo con una pequeña descripción.

La descripción elegida para publicar en Zenodo ha sido la siguiente:

Collection of information on gluten-free products offered by the Eroski supermarket, taking advantage of the fact that on its website it offers the possibility of displaying all products that do not contain this protein

Referencia a la publicación:

https://zenodo.org/record/3748725#.XpLRacgzbIU

11. Entrega

Presentar el trabajo con el DOI del dataset en Github.

DOI generado por Zenodo: **10.5281/zenodo.3748725**

Se deja reflejo de esta información en la WIKI de GitHub.

12. Contribuciones al trabajo

Contribuciones	Firma
Investigación previa	IGV, CMGR
Redacción de las respuestas	IGV, CMGR
Desarrollo código	IGZ, CMGR

Bibliografía

@hellomrspaceman, H. . (s.f.). Obtenido de https://dev.to/hellomrspaceman/python-selenium-infinite-scrolling-3o12

Eroski. (s.f.). *https://supermercado.eroski.es/*. Obtenido de https://supermercado.eroski.es/es/supermercado/SinGluten/

FACE. (2020). *Informe de precios sobre productos sin gluten 2020.* Madrid: Federación de Asociaciones de Celiacos de España (FACE).

https://cronicaglobal.elespanol.com/. (27 de 05 de 2019). *https://cronicaglobal.elespanol.com/*. Obtenido de https://cronicaglobal.elespanol.com/vida/gluten-free-pueblos-ciudades-pequenas_247779_102.html

Lawson, R. (2015). Web Scraping with Python. .

Masip, D. (2010). El lenguaje Python. Editorial UOC.

Parker, K. (8 de 11 de 2018). *Towards data science*. Obtenido de https://towardsdatascience.com/data-science-skills-web-scraping-javascript-using-python-97a29738353f

Simon Munzert, C. R. (2015). Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining.

Subirats, L. C. (2019). Web Scraping. Editorial UOC.

Tutorial de Github. (s.f.). Obtenido de https://guides.github.com/activities/hello-world.

Wikipedia. (s.f.). Obtenido de https://es.wikipedia.org/wiki/Eroski