Tipología y ciclo de vida de los datos - PRA 1

Cuestión 1 – Contexto

Dada la situación actual del mercado eléctrico y la fluctuación de los precios de la energía, resulta interesante poder realizar una comparativa entre todas las comercializadoras de electricidad y gas de un código postal.

El sitio web proporciona información detallada de las ofertas de energía que hay disponibles a nivel código postal. Permite al usuario introducir la información que considere relevante o tenga en su poder para realizar una comparativa entre las diferentes comercializadoras y mostrar las que más se ajusten a sus preferencias.

El acceso al comparador de la CNMC es el a través de la dirección web Comparador

Cuestión 2 – Título

Comparativa de ofertas de energía.

Cuestión 3 – Descripción del dataset

Se han generado tres datasets diferentes uno para cada tipología de energías que se ofertan (electricidad, gas y ofertas conjuntas). Para la extracción de estos datasets el proceso ha sido un poco diferente, dado que los atributos de los que se saca la información en cada una de las páginas web son distintos.

La incorporación de la información en un único dataset, así como el enriquecimiento cambiando de código postal o de perfil de clientes, se completará si es necesario en la segunda práctica.

Los tres proporcionan la información de los precios de las diferentes comercializadoras para las diferentes energías (luz, gas o dual).

Cuestión 4 – Representación gráfica

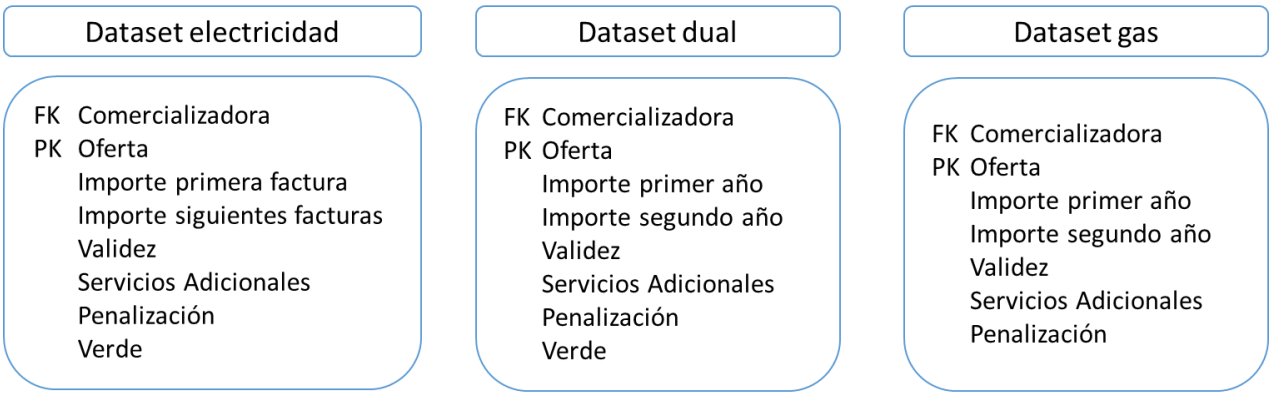


Figura 1. Representación gráfica del dataset

Cuestión 5 – Contenido

Como se ha comentado el dataset es diferente dependiendo de la energía.

Electricidad:

- Comercializadora: empresa que se encarga de la facturación de la energía.
- Oferta: nombre identificativo de la oferta de energía para una determinada comercializadora y con un determinado precio.
- Importe primera factura: precio de la primera factura de electricidad.
- Importe segunda factura: precio que se tendrá desde la segunda factura, incluida.
- Validez: describe a quién se le puede aplicar la oferta.
- Servicios adicionales: describe si la oferta incluye servicios adicionales.
- Penalización: penalización máxima calculada en función de las condiciones de la oferta.
- Verde: indicador que determina si la energía de la oferta es de origen de origen verde, considerando que sea 100% verde o cogeneración de alta eficiencia.

Gas Natural:

- Comercializadora: empresa que se encarga de la facturación de la energía.
- Oferta: nombre identificativo de la oferta de energía para una determinada comercializadora y con un determinado precio.
- Importe anual primer año: precio del primer año de gas.
- Importe a anual segundo año: precio del segundo año de gas.
- Validez: describe a quien se le puede aplicar la oferta.
- Servicios adicionales: describe si la oferta incluye servicios adicionales.
- Penalización: penalización máxima calculada en función de las condiciones de la oferta.

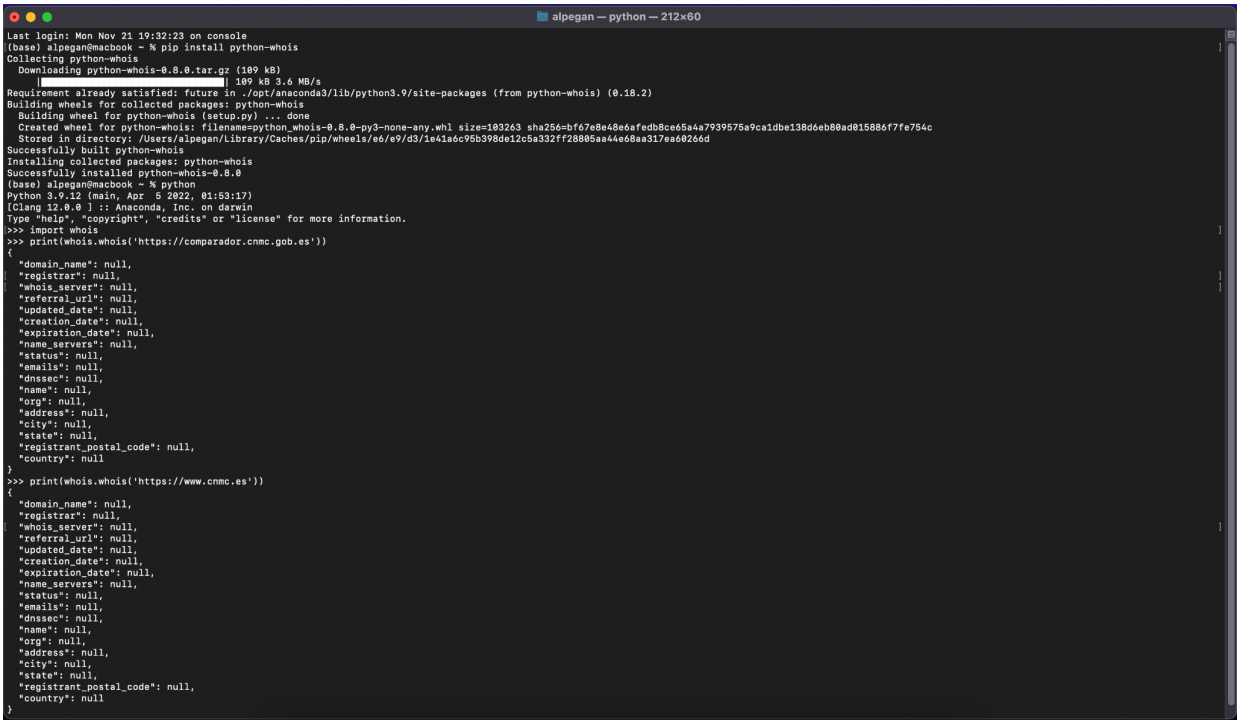
Ofertas conjuntas (dual):

- Comercializadora: empresa que se encarga de la facturación de la energía.
- Oferta: nombre identificativo de la oferta de energía para una determinada comercializadora y con un determinado precio.
- Importe anual primer año: precio del primer año de electricidad y gas.
- Importe a anual segundo año: precio del segundo año de electricidad y gas.
- Validez: describe a quién se le puede aplicar la oferta.
- Servicios adicionales: describe si la oferta incluye servicios adicionales.
- Penalización: penalización máxima calculada en función de las condiciones de la oferta.
- Verde: indicador que determina si la energía de la oferta es de origen de origen verde, considerando que sea 100% verde o cogeneración de alta eficiencia.

Cuestión 6 – Propietario

El propietario de los datos es la Comisión Nacional de los Mercados y la Competencia, CNMC, que es un organismo regulador encargado de asegurar que el funcionamiento, la transparencia y la competitividad sean correctos en los diferentes mercados para beneficio de los consumidores y usuarios.

Por otro lado, para conocer el propietario del sitio web, se ha realizado la siguiente secuencia de comandos desde terminal con el objetivo de conocer el propietario.



Se ha empleado la librería python-whois tal y como se explica en los recursos de web scraping de la asignatura sin éxito, ya que no se logra conocer el propietario por este método.

Por lo tanto, como se menciona anteriormente, el propietario de la página se considera a la propia CNMC.

Finalmente, en cuanto a los pasos seguidos para actuar dentro de los principio éticos, se ha comprobado que no se violan los siguientes aspectos:

- Terminos y condiciones de la página.
- Infracción de derechos de autor, véase los términos anteriores.
- Ley de fraude y abuso informático.
- Allanamiento de morada.
- Protocolo de exclusión de robots tras la revisión del archivo robots.txt adjunto con el resto del trabajo.
- Ley de derechos de autor del milenio digital y ley CAN-SPAN

Asimismo, el presente trabajo ha contado con la consideración de los siguientes principios clave:

- Verificación de las condiciones de uso.
- Rastreo solo de información pública.
- No sobrecargar el servidor, ya que únicamente se realizan tres búsquedas, para cada una de las opciones de suministro.
- La información extraída no se utiliza con fines comerciales.

Cuestión 7 – Inspiración

Como se ha comentado en la cuestión 1, la volatilidad de los precios y los mercados de electricidad y gas hacen interesante el análisis de las diferentes ofertas. Una posible mejora sería poder recopilar

todas las ofertas de todos los códigos postales de España para diferentes perfiles de consumidores e incluyendo servicios adicionales.

La principal pregunta que se pretende recoger es que tipo de suministro, para cada tipo de cliente es la más rentable en cada código postal. También podría resultar interesante de cara al estudio de la competencia, para saber qué ofertas y precios tienen el resto de comercializadoras de energía.

Cuestión 8 – Licencia

En este caso y dado que son datos de carácter público que se pueden obtener directamente de la web del comparador de la CNMC, la licencia que se escogería sería la Release CC BY. Esta licencia permite la utilización y modificación de las obras siempre que se haga referencia a la fuente y autores originales, no se restringen los fines de uso, pudiendo ser utilizada para fines comerciales.

Cuestión 9 – Código

```
# Importación de liberías
import pandas as pd
from time import sleep
import sys
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By # Para poder usar el By

#Inicialización driver
def init_driver(url):
    driver = webdriver.Chrome()
    driver.get(url)
    return driver

#Cierre driver
def close_driver(driver):
    driver.close()

#Aceptación cookies
def accept_cookies(driver):
    driver.find_element(By.CLASS_NAME,'cookiesjsr-btn.important').click()

#Abrir lista de opciones
def selecct_sum(driver):
    driver.find_element(By.ID, "input-47").click()

#Selección de ofertas de electricidad en desplegable
def select_elect(driver):
    driver.find_element(By.ID, "list-item-59-0").click()

#Selección de ofertas de gas en desplegable
def select_gas(driver):
    driver.find_element(By.ID, "list-item-59-1").click()
```

```
#Selección de ofertas conjuntas en desplegable
def select_both(driver):
    driver.find_element(By.ID, "list-item-59-2").click()

#Selección de boton "Iniciar"
def iniciar(driver):
    driver.find_element(By.ID, "Iniciar").click()

#Escribe un código postal en formulario electricidad
def type_postal_code(driver):
    driver.find_element(By.NAME, "codigoPostal").click()
    driver.find_element(By.NAME, "codigoPostal").send_keys("08035")

#Continúa desde el formulario hasta la siguiente página
def continue_to_page(driver, gas=False):
    driver.find_element(By.ID, "Continuar").click()
    sleep(1)

    if gas == False:
        driver.find_element(By.CLASS_NAME,
"v-input--selection-controls__ripple").click()
        sleep(1)
        driver.find_element(By.XPATH, "(.//*[normalize-space(text()) and
normalize-space(.)='He leído este aviso'])[1]/following::span[2]").click()
        return driver

    else: return driver

#Navegación hasta página de ofertas de electricidad
def get_comparator_elect(url_base):
    driver = init_driver(url_base)
    accept_cookies(driver)
    selecct_sum(driver)
    select_elect(driver)
    iniciar(driver)
    sleep(3)
    type_postal_code(driver)
    driver = continue_to_page(driver)
    return driver

#Naveción hasta página de gas
def get_comparator_gas(url_base):
    driver = init_driver(url_base)
    accept_cookies(driver)
    selecct_sum(driver)
    select_gas(driver)
    iniciar(driver)
    sleep(3)
    type_postal_code(driver)
    driver = continue_to_page(driver, gas=True)
    return driver
```

```
#Navegación hasta página de ofertas combinadas
def get_comparator_both(url_base):
    driver = init_driver(url_base)
    accept_cookies(driver)
    selecct_sum(driver)
    select_both(driver)
    iniciar(driver)
    sleep(3)
    type_postal_code(driver)
    continue_to_page(driver)
    return driver

#Extraccion de tabla de ofertas de electricidad
def get_elect_table(driver):

    html = driver.page_source
    soup = BeautifulSoup(html, "html.parser")
    tables = soup.find_all('table', attrs={'class': ''})

    for table in tables[1]:

        rows = table.find_all('tr', attrs={'class': ''})
        table_list = []

        for row in rows:

            data = row.find_all('td')
            row_list = []

            for elem in data:

                if data[0] == elem:

                    company = elem.find('img').attrs['alt']
                    row_list.append(company)
                elif data[1] == elem:

                    oferta = row.find('a').text.lstrip()
                    row_list.append(oferta)
                elif data[5] == elem:

                    servicios_adicionales = elem.find('div').string
                    row_list.append(servicios_adicionales)
                elif data[7] == elem:

                    eco = elem.find('i').attrs['class'][-1]

                    if eco == 'grey--text':
                        row_list.append(False)
                    elif eco == 'green--text':
```

```
                        row_list.append(True)
                elif data[8] == elem:
                    break
                else:
                    row_list.append(elem.string)

            table_list.append(row_list)

    df_elect = pd.DataFrame(table_list)
    df_elect.columns = ['Comercializadora', 'Oferta', 'Importe primera
factura',
    'Importe siguientes facturas', 'Validez', 'Servicios adicionales',
'Penalizacion', 'Verde']

    return df_elect

#Extracción de tabla de ofertas conjuntas
def get_both_table(driver):
    html = driver.page_source
    soup = BeautifulSoup(html, "html.parser")
    tables = soup.find_all('table')

    for table in tables[0]:

        rows = table.find_all('tr', attrs={'class': ''})
        table_list = []
        for row in rows:

            data = row.find_all('td')
            row_list = []

            for elem in data:

                if data[0] == elem:

                    company = elem.find('img').attrs['alt']
                    row_list.append(company)
                elif data[1] == elem:

                    oferta = row.find('a').text.lstrip()
                    row_list.append(oferta)
                elif data[5] == elem:

                    servicios_adicionales = elem.find('div').string
                    row_list.append(servicios_adicionales)
                elif data[7] == elem:

                    eco = elem.find('i').attrs['class'][-1]

                    if eco == 'grey--text':
                        row_list.append(False)
```

```
                    elif eco == 'green--text':
                        row_list.append(True)
                elif data[8] == elem:
                    break
                else:
                    row_list.append(elem.string)

            table_list.append(row_list)
    df_both = pd.DataFrame(table_list)
    df_both.columns = ['Comercializadora', 'Oferta', 'Importe primer año',
    'Importe 2o año', 'Validez', 'Servicios adicionales', 'Penalizacion',
'Verde']

    return df_both

#Extraciión de tabla de gas
def get_gas_table(driver):
    html = driver.page_source
    soup = BeautifulSoup(html, "html.parser")
    body = soup.find_all('tbody', attrs={'class': ''})
    head = soup.find_all('thead', attrs={'class': 'v-data-table-header'})
    for table in head:
        rows = table.find_all('th')
        headers_list = []
        for row in rows:
            headers = row.find('span').text
            headers_list.append(headers)
    for table in body:
        rows = table.find_all('tr', attrs={'class': ''})
        table_list = []
        for row in rows:
            row_list = []
            data = row.find_all('td',attrs={'class': 'text-center'})
            for elem in data:
                if data[0] == elem:
                    if elem.find('img'):
                        company = elem.find('img')['alt']
                        row_list.append(company)
                    else:
                        company = elem.find('p').text.lstrip().rstrip()
                        row_list.append(company)
                elif data[1] == elem:
                        oferta = elem.find('a').text.lstrip().rstrip()
                        row_list.append(oferta)
                elif data[2] == elem:
                        price = elem.find('p').text.lstrip().rstrip()
                        row_list.append(price)
                elif data[3] == elem:
                        price_2 = elem.find('div').text.lstrip().rstrip()
                        row_list.append(price_2)
                elif data[4] == elem:
```

```
                        validez = elem.find('div').text.lstrip().rstrip()
                        row_list.append(validez)
                elif data[5] == elem:
                        servicios_adicionales =
elem.find('div').text.lstrip().rstrip()
                        row_list.append(servicios_adicionales)
                elif data[6] == elem:
                        penality = elem.find('div').text.lstrip().rstrip()
                        row_list.append(penality)
                else:
                        row_list.append(elem.string)
            table_list.append(row_list)

    df_gas = pd.DataFrame(table_list)
    df_gas.columns= headers_list
    df_gas.drop(df_gas.columns[-1], axis=1, inplace=True)
    return df_gas

#Scraping CNMC ofertas electricidad
def elect_scrap(url_base):

    driver_elect = get_comparator_elect(url_base)
    sleep(5)
    df_elect = get_elect_table(driver_elect)
    sleep(5)
    close_driver(driver_elect)
    return df_elect

#Scraping CNMC ofertas gas
def gas_scrap(url_base):
    driver_gas = get_comparator_gas(url_base)
    sleep(5)
    df_gas = get_gas_table(driver_gas)
    sleep(5)
    close_driver(driver_gas)
    return df_gas

#Scraping CNMC ofertas conjuntas
def both_scrap(url_base):

    driver_both = get_comparator_both(url_base)
    sleep(5)
    df_both = get_both_table(driver_both)
    sleep(5)
    close_driver(driver_both)
    return df_both

#Convierte dataframe a csv
def data2csv(df, name):

        df.to_csv(name, encoding='utf-8')
```

```
def main():

# Definición de la url base
    url_base = 'https://comparador.cnmc.gob.es/'

    df_elect = elect_scrap(url_base)
    df_gas = gas_scrap(url_base)
    df_both = both_scrap(url_base)

    data2csv(df_elect, 'comparador_electricidad.csv')
    data2csv(df_gas, 'comparador_gas.csv')
    data2csv(df_both, 'comparador_conjuntas.csv')


if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(0)
```

Asimismo, el código en Python es subido a Github.

Cuestión 10 – Dataset

Se han publicado los datasets en Zenodo. Se puede acceder a dichos datos mediante el siguiente enlace: Zenodo

Cuestión 11 – Vídeo

Se puede acceder al vídeo explicativo en el siguiente enlace: Video

En la siguiente tabla se muestran las contribuciones de cada uno de los participantes.

Contribuciones	Firma
Investigación Previa	Alberto Pérez, Patricia García
Redacción de las respuestas	Alberto Pérez, Patricia García
Desarrollo del código	Alberto Pérez, Patricia García
Participación en el vídeo	Alberto Pérez, Patricia García