Tipología y Ciclo de Vida de los Datos

Máster de Ciencia de Datos de la UOC

Práctica 1

Enrique Javier Andrés Orera

Sergio Fernández Bertolin

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

La finalidad del proyecto es educativa y de investigación, no comercial. Idealista, el sitio web "raspado", es uno de los portales inmobiliarios de referencia en el contexto estatal y probablemente uno de los mayores portales de compra venta en España. Esta web pone al alcance del usuario multitud de información inmobiliaria para hacer más fácil al usuario su experiencia en la compraventa de inmuebles.

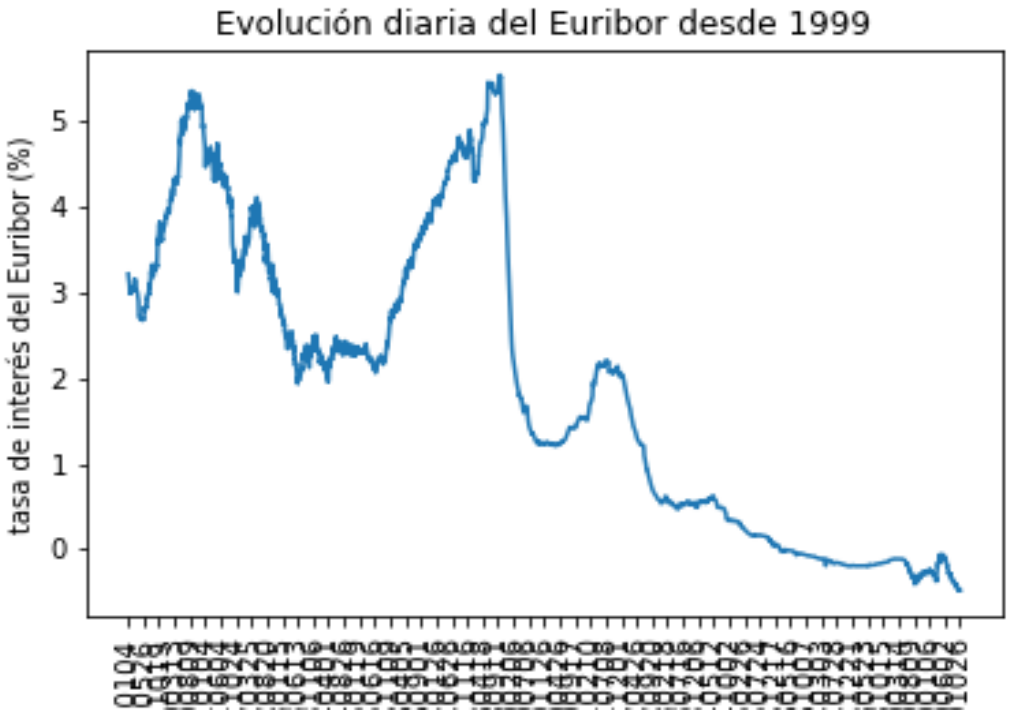
2. **Definir un título para el dataset. Elegir un título que sea descriptivo.**

Llamaremos al dataset "Euribor diario"

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).**

Creación de un dataset con los valores diarios del índice del Euribor desde 1999 hasta el día actual con fines educativos. Se obtienen las cotizaciones con un programa en python mediante técnicas de webscraping contra el portal inmobiliario www.idealista.com

4. **Representación gráfica. Presentar una imagen o esquema que identifique el dataset visualmente**



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

El Dataset generado tiene el siguiente formato:

Index	Dia	Valor
0	19990104	0.345
1	19990105	0.367

Donde **Index** es un valor entero auto incremental, empezando en cero

Dia es el dia en el que se ha registrado el valor, con el formato AAAAMMDD, donde las cuatro primeras cifras son el año, las dos siguientes el mes en formato numérico y las dos últimas el día del mes

Valor es el valor del índice del Euribor registrado ese día, en formato decimal

El periodo de tiempo d erecogida es desde el 4 de enero de 1999 hasta el 30 de octubre de 2020 y se ha recogido mediante un raspado recurrente de las páginas de idalista con la información diaria y mensual del índice Euribor.

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

Idealista es uno de los portales inmobialiarios de referencia en el contexto estatal y probablemente uno de los mayores portales de compra venta en España.

Incluir citas de estudios con datos de Idealista (TODO)

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

En el momento actual, dentro de la mayor pandemia del siglo, hay una incertidumbre creciente en todos los aspectos financieros, de los que no queda exento el mercado inmobiliario.

Mediante esta recolección de datos se pretende obtener una herramienta que permita analizar la tendencia del principal índice inmobiliario en Europa, el Euribor, para poder realizar tanto análisis como predicciones.

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**

Hemos escogido la licencia **CC BY-NC-SA 4.0 License**, porque creemos que recoge adecuadamente el espiritu de la creación del dataset. Un dataset creado para un uso educativo, no comercial.

Está licencia tiene los siguientes términos:

Atribución : debe otorgar el crédito correspondiente , proporcionar un enlace a la licencia e indicar si se realizaron cambios . Puede hacerlo de cualquier manera razonable, pero no de ninguna manera que sugiera que el licenciante lo respalda a usted o su uso.

No comercial: no puede utilizar el material con fines comerciales .

ShareAlike : si remezcla, transforma o construye sobre el material, debe distribuir sus contribuciones bajo la misma licencia que el original.

Terminos extraídos de https://creativecommons.org/licenses/by-nc-sa/4.0/

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

```
"""
Script para extraer el euribor diario de la página web de Idealista
Allí hay almacenados los datos del Euribor desde Enero de 1999 hasta
el mes anterior a la extracción
"""

# Importación de librerías necesarias

import locale
import datetime
import random
from bs4 import BeautifulSoup as bsoup
import requests
import calendar
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Necesario para calendar (configuración local a hora española)
locale.setlocale(locale.LC_ALL, 'es_ES.utf8')

# Lista de user agents

userAgents = [
    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/60.0.3112.113 Safari/537.36',
    'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/60.0.3112.90 Safari/537.36',
    'Mozilla/5.0 (Windows NT 5.1; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/60.0.3112.90 Safari/537.36',
    'Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/60.0.3112.90 Safari/537.36',
    'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/44.0.2403.157 Safari/537.36',
    'Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/60.0.3112.113 Safari/537.36',
    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/57.0.2987.133 Safari/537.36',
    'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/57.0.2987.133 Safari/537.36',
    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/55.0.2883.87 Safari/537.36',
```

```
    'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/55.0.2883.87 Safari/537.36',
    'Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1)',
    'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko',
    'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)',
    'Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko',
    'Mozilla/5.0 (Windows NT 6.2; WOW64; Trident/7.0; rv:11.0) like Gecko',
    'Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like
Gecko',
    'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0)',
    'Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko',
    'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)',
    'Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like
Gecko',
    'Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64;
Trident/6.0)',
    'Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)',
    'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET
CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)'
]

# Fecha actual
now = datetime.datetime.now()

anyo = int(now.strftime("%Y"))
mes = int(now.strftime("%m"))
dia = int(now.strftime("%d"))

controlmesactual = True

# Creamos listas donde guardaremos los días y los valores
dias= []
valores =[]
# Para cada año
while anyo >= 1999:
    # Para cada mes
    while mes > 0:
        if controlmesactual:
            url = "https://www.idealista.com/news/euribor/mensual/mes-
actual/"
            controlmesactual = False
        else:
            url = "https://www.idealista.com/news/euribor/mensual/%s-%d/" %
(calendar.month_name[int(mes)], anyo)
```

```
            # Seleccionamos user agent aleatoriamente
            userAgent = random.choice(userAgents)

            # Cargamos cabeceras por defecto
            headers = requests.utils.default_headers()

            # Actualizamos cabeceras con el User-Agent aleatorio
            headers.update({'User-Agent': userAgent})

            # Descargamos el sitio web de interés
            html = requests.get(url, headers=headers)

            soup = bsoup(html.content)

            # Se crea una lista vacía y mediante un bucle for, guardamos los
títulos de la tabla que se quiere almacenar

            tablehead = []

            for header in soup.body.thead.tr.children:
                tablehead.append(header.text)

            contador = 0

            for dato in soup.body.tbody.find_all('td'):
                contador = contador + 1
                if contador % 2 == 1:
                    # fecha
                    fecha = "%d%s%s" % (anyo, '{:02d}'.format(mes),
'{:02d}'.format(int(dato.string)))
                    # Añadimos la fecha a la lista
                    dias.append(fecha)
                    print(fecha)
                else:
                    # euribor
                    euribor = dato.string[:-1].replace(",", ".")
                    # Añadimos el euribor a la lista
                    valores.append(float(euribor))
                    print(float(euribor))

            mes = mes - 1

        anyo = anyo - 1
        mes = 12

# Creamos un DataFrame para almacenar los pares de conjuntos días, valores
```

```
euribordf = pd.DataFrame(list(zip(dias[::-1], valores[::-1])),
columns=['Dia', 'Valor'])

# Almacenamos los resultados de nuestro dataset en un csv
euribordf.to_csv('euribordiario.csv')

# Representamos gráficamente la evolución temporal del Euribor
print(euribordf)

f, ax = plt.subplots()
ax.plot(euribordf.index, euribordf.Valor)
ax.set(xlabel='Fecha (AñoMesDia)', ylabel='tasa de interés del Euribor
(%)', title='Evolución diaria del Euribor desde 1999')
plt.xticks(np.arange(euribordf.shape[0])[::100], euribordf.Dia[::100],
rotation=90)

plt.show()

# Almacenamos la gráfica
f.savefig("euribor.png")
```

10. Dataset. Publicación del dataset en formato CSV en Zenodo (obtención del DOI) con una breve descripción.