Tipologia i cicle de vida de les dades.

Pràctica 1.

Dataset: Flux dels pacients en espera al servei d'urgències.

Pacients en espera a urgències del Consorci Sanitari de l'Alt Penedès, distribuïts per nivell de triatge.

Context

Aquest dataset es tracta d'una mostra dels pacients en espera al servei d'urgències del Consorci Sanitari de l'Alt Penedès (CSAP). Aquesta informació és actualitzada a la web institucional del CSAP cada cinc minuts, i aporta el número de pacients que estan sent atesos en aquell moment al seu servei d'urgències, distribuïts pel corresponent nivell de triatge. El nivell de triatge és un mètode de classificació de la gravetat de les urgències, essent els valors 1 i 2 els més urgents i que són acompanyats directament al box d'atenció. Els nivells 4 i 5 corresponen als menys urgents, els quals, un cop avaluats i assignats el nivell passen a la sala d'espera fins que puguin ser cridats al box per a la seva atenció.

Contingut

El dataset aportat inclou una fila per a cada captura aconseguida amb èxit. Es realitza un procés de scraping programat cada cinc minuts. El període de temps aportat al dataset d'exemple correspon a les extraccions realitzades cada cinc minuts des de les 21:16 hores del dia 11/04/2018 fins les 23:56 hores del dia 12/04/2018. Els camps que inclou són els següents:

dataActualitzacioWeb: data i hora d'actualització de les dades informada per la pròpia web, en format "Dimecres, 11 d'abril de 2018 a les 21'16h"

dataCaptura: data en format yyyy-mm-dd de la data en la qual ha estat capturada la informació de la fila mitjançant el procés de captura.

horaCaptura: hora en format hh:mm de la data en la qual ha estat capturada la informació de la fila mitjançant el procés de captura.

PacientsNivell1: número de pacients en espera en el moment de la captura per al nivell 1 de triatge.

PacientsNivell2: número de pacients en espera en el moment de la captura per al nivell 2 de triatge.

PacientsNivell3: número de pacients en espera en el moment de la captura per al nivell 3 de triatge.

PacientsNivell4: número de pacients en espera en el moment de la captura per al nivell 4 i 5 de triatge.

TempsEsperaAdults: temps d'espera informat del darrer pacient visitat dels nivells 4 i 5 per a adults.

TempsEsperaPediatric: temps d'espera informat del darrer pacient visitat dels nivells 4 i 5 per a pediatria.

Agraïments

Les dades han estat aconseguides a partir de la informació publicada oficialment per la web institucional del CSAP. Agrair al CSAP la possibilitat de disposar d'aquesta informació per poder analitzar-la.

Inspiració

El rol professional que ostento com a gestor de la informació assistencial d'un consorci sanitari, em porta a intentar identificar aquelles dades que puguin aportar valor a la gestió del dia a dia als nostres centres sanitaris. La informació relativa sobre la situació als serveis d'urgències dels centres propers al nostre servei d'urgències pot ajudar-nos a contextualitzar els moments de sobresaturació al nostre servei d'urgències. D'igual manera pot interferir en la pressa de decisions en el moment de derivar pacients a centres propers. Aquest punt de vista trobo que és igualment vàlid a qualsevol centre sanitari amb servei d'atenció urgent.
Cada cop més les webs institucionals dels centres sanitaris publiquen aquest tipus d'informació per que el usuari pugui prendre la decisió d'on acudir amb coneixement de l'estat del servei als centres.
Des de la perspectiva del usuari, poder seria interessant disposar d'aquesta informació relativa als centres que te al seu abast però de forma unificada, sense haver de realitzar una visita individual a la web de cada centre.

Llicència

Per compartir aquest dataset he optat per alliberar-lo sota la *Atribution Non Comercial International 4.0 (CC BY-NC-SA 4.0) License*. Aquest llicenciament permet compartir i adaptar tot el seu contingut, sense possibilitar el seu ús comercial i atribuint el crèdit al seu autor, proporcionant un enllaç a la llicència i indicant si s'han realitzat canvis. Considero que és el tipus de llicenciament més adequat per aquest tipus de treballs, intentant limitar només el seu comercial.

Codi

A continuació s'adjunta el codi dissenyat en llenguatge python que implementa el procés descrit:

```
import urllib.request as urllib2
import datetime
import csv
import time
import re
import os
from bs4 import BeautifulSoup

#Controlem el path de l arxiu en base al directori actiu on es trobi l arxiu py
directoriActiu = os.path.dirname(__file__)
nomArxiuCSV = "CSAP.csv"
pathArxiu = os.path.join(directoriActiu, nomArxiuCSV)

#Definim una funcio que estableixi els parametres de connexio amb la web i controli els reintents i els
possibles errors de connexio

def download(url, frequenciaSegons, user_agent='wswp', num_retries=2):
    print ('Downloading from', url, 'every', int(frequenciaSegons/60) , 'minutes.' )
    headers = {'User-agent': user_agent}
    request = urllib2.Request(url, headers=headers)
    try:
        html = urllib2.urlopen(request).read()
    except urllib2.URLError as e:
        print ('Download error:', e.reason)
        html = None
        if num_retries > 0:
            if hasattr(e, 'code') and 500 <= e.code < 600:
                return download(url, user_agent, num_retries-1)
    return html

#Declarem la url de la web i la llista de variables que extreurem

url = 'http://www.csap.cat/ciutadania/temps-espera.html'
filaNova=[]
llistaVariables=['dataActualitzacioWeb', 'dataCaptura','horaCaptura', 'PacientsNivell1',
'PacientsNivell2', 'PacientsNivell3', 'PacientsNivell4','tempsEsperaAdults','tempsEsperaPediatrics']
L=locals()

#Realitzem una primera connexio amb l arxiu per crear-lo si no existeix i incorporar els noms de les
```

```
variables

with open(pathArxiu, 'w', newline='') as csvFile:
    writer = csv.writer(csvFile)
    writer.writerow(llistaVariables)

#Utilitzem el metode input() per obtenir els parametres d inici i finalitzacio del proces, així com la frequencia de captura

dataHora_inici = input('Indiqueu una data i hora per iniciar el procés d\'extracció en format DD-MM-AAAA-HH-MM, si voleu començar ara escriviu ARA: ')

if dataHora_inici=="ARA":
    dataHora_inici=datetime.datetime.now()
    day=dataHora_inici.day
    month=dataHora_inici.month
    year=dataHora_inici.year
    hour=dataHora_inici.hour
    minute=dataHora_inici.minute
else:
    day, month, year, hour, minute = map(int, dataHora_inici.split('-'))

data1 = datetime.datetime(year, month, day, hour,minute)
dataHora_fi = input('Indiqueu una data i hora per finalitzar el procés en format DD-MM-AAAA-HH-MM: ')
day, month, year, hour, minute = map(int, dataHora_fi.split('-'))
data2=datetime.datetime(year, month, day, hour, minute)

frequenciaSegons=int(input('Amb quina frequencia voleu realitzar la captura? (informeu en minuts) : '))*60

print(f"Es realitzarà l'extracció des del {datetime.datetime.strftime(data1, '%d/%m/%Y a les %H:%M')} hores fins al {datetime.datetime.strftime(data2, '%d/%m/%Y a les %H:%M')} hores, cada {int(frequenciaSegons/60)} minuts.")

# Comprovem cada minut si el proces ha de continuar en base a les dates informades

while data1>datetime.datetime.now():
    time.sleep(60)
while data1<=datetime.datetime.now() and data2>=datetime.datetime.now() :
    #Iniciem el proces de scraping

    html = download(url, frequenciaSegons)
    soup = BeautifulSoup(html,"html5lib")

    ul =soup.find(attrs={'class':'caixes temps clearfix'})
    li = ul.find(attrs={'class':'caixa caixa-1'})
    span = li.find(attrs={'class':'caixa-data'})
    tempsEsperaAdults=span.text
    li = ul.find(attrs={'class':'caixa caixa-2'})
    span = li.find(attrs={'class':'caixa-data'})
    tempsEsperaPediatrics=span.text

    ol =soup.find(attrs={'class':'caixes num-pacients clearfix'})

    for i in range(1,5):
```

```
        li = ol.find(attrs={'class':'caixa caixa-%d'%i})
        span = li.find(attrs={'class':'caixa-data'})
        L["PacientsNivell%d"%i] = re.sub(" pacients", "", span.text)
    p=soup.find(attrs={'class':'avis'})
    darreraActualitzacio=p.find(attrs={'id':'esperaData'})
    dataActualitzacioWeb=darreraActualitzacio.text
    dataCaptura=datetime.datetime.now().date()
    horaCaptura=str(datetime.datetime.now().hour)+":"+str(datetime.datetime.now().minute)


    filaNova=[dataActualitzacioWeb, dataCaptura, horaCaptura, PacientsNivell1, PacientsNivell2,
PacientsNivell3, PacientsNivell4,tempsEsperaAdults,tempsEsperaPediatrics]

    #Afegim una nova fila a l arxiu amb les noves dades capturades

    with open(pathArxiu, 'a', newline='') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerow(filaNova)

    #Retenim la nova captura durant el temps de frequencia informat
    time.sleep(frequenciaSegons)

del writer
csvFile.close()
```

Dataset.

Es presenta una petita capçalera del dataset, el qual s'adjunta complert al repositori Gihub.

dataActualitzacioWeb	dataCaptura	horaCaptura	PacientsNivell1	PacientsNivell2	PacientsNivell3	PacientsNivell4	tempsEsperaAdults	temps
Dimecres, 11 d'abril de 2018 a les 21'16h	2018-04-11	21:16	0	4	8	11	1h. 24min.	1h. 0
Dimecres, 11 d'abril de 2018 a les 21'21h	2018-04-11	21:21	0	4	8	12	1h. 24min.	1h. 0
Dimecres, 11 d'abril de 2018 a les 21'26h	2018-04-11	21:26	0	4	8	9	0h. 51min.	1h. 1
Dimecres, 11 d'abril de 2018 a les 21'31h	2018-04-11	21:31	0	4	8	10	1h. 14min.	1h. 1
Dimecres, 11 d'abril de 2018 a les 21'36h	2018-04-11	21:36	0	4	8	12	1h. 14min.	1h. 1
Dimecres, 11 d'abril de 2018 a les 21'41h	2018-04-11	21:41	0	4	7	13	1h. 14min.	1h. 1

Referències:

Richard Lawson. Web Scraping with Python. Packt Publishing Ltd, October 2015. 174 p. ISBN 9781782164371

Simon Munzert, Christian Rubba, Peter Meißner, Dominic Nyhuis. Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons, 2015. ISBN 9781118834732