Taller 8

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 12-oct-2018 11:59 PM

[Juan Camilo Perdomo]

[juan.perdomor@urosario.edu.co]

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller8_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
  1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
  2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

Vamos a hacer "scraping" a esta página: http://archive.ics.uci.edu/ml/datasets.html, que contiene un listado de 360 bases de datos que hacen parte del repositorio de la Universidad de California, Irvine.

Su tarea consiste en crear un "Pandas dataframe" que contenga 360 filas (una por base de datos) y las siguientes columnas:

- Nombre de la base de datos
- Link a la base de datos
- Tipo de datos
- Tipo de tarea a resolver (default task)
- Tipo de las variables
- Número de observaciones
- Número de variables
- Año
- Descricpción de la base (Pista: Utilice la opción list view)

Diviértase.

In [1]:

```
import re
from requests import get
from bs4 import BeautifulSoup
import pandas as pd
```

In [2]:

```
url = 'http://archive.ics.uci.edu/ml/datasets.html'
response = get(url)
html_soup = BeautifulSoup(response.text, 'html.parser')
type(html_soup)
```

Out[2]:

```
bs4.BeautifulSoup
```

In [35]:

```
Data_sets = html_soup.find_all('div', class_ = 'lister-item mode-advanced')

nombre = []
tipo = []

for dataset in Data_sets:
    nombres = [re.findall("<a href=\"datasets/(.+)</a>", str(dataset))]
    nombre = nombre + nombres
    tipos = [re.findall("<p class=\"normal(.+)</p>", str(dataset))]
```

In [37]:

```
nombre = re.findall("<a href=\"datasets/(.+)</a>", response.text)
tipo = re.findall("<p class=\"normal(.+)</p>", response.text)
```

In [38]:

```
print(str(len(nombre)))
print(str(len(tipo)))
```

```
452
4055
```

In [33]:

```
db = {"Nombre":nombre, "Tipo":tipo}
```

In [34]:

```
database = pd.DataFrame(db)
database
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-34-05cf5dbeeea1> in <module>()
----> 1 database = pd.DataFrame(db)
      2 database

~\Anaconda3\lib\site-packages\pandas\core\frame.py in __init__(self, data, index, columns, dtype,
copy)
    346                                 dtype=dtype, copy=copy)
    347         elif isinstance(data, dict):
--> 348             mgr = self._init_dict(data, index, columns, dtype=dtype)
    349         elif isinstance(data, ma.MaskedArray):
    350             import numpy.ma.mrecords as mrecords

~\Anaconda3\lib\site-packages\pandas\core\frame.py in _init_dict(self, data, index, columns,
dtype)
    457             arrays = [data[k] for k in keys]
    458
--> 459         return _arrays_to_mgr(arrays, data_names, index, columns, dtype=dtype)
    460
    461     def _init_ndarray(self, values, index, columns, dtype=None, copy=False):

~\Anaconda3\lib\site-packages\pandas\core\frame.py in _arrays_to_mgr(arrays, arr_names, index,
columns, dtype)
   7313     # figure out the index, if necessary
   7314     if index is None:
-> 7315         index = extract_index(arrays)
   7316
   7317     # don't force copy because getting jammed in an ndarray anyway

~\Anaconda3\lib\site-packages\pandas\core\frame.py in extract_index(data)
   7359             lengths = list(set(raw_lengths))
   7360             if len(lengths) > 1:
```

```
-> 7361                 raise ValueError('arrays must all be same length')
   7362
   7363             if have_dicts:

ValueError: arrays must all be same length
```