Analisi di dati ricavati dal web

Federico Trotta

PROGETTO WEB SCRAPING

Descrizione del progetto

NECESSITA':
Quando si devono fare delle analisi, spesso non si hanno a disposizione i dati. Una possibile soluzione alla mancanza di dati è quella di ricavare i dati dal web, mediante il cosiddetto 'web scraping', per poterli poi analizzare.

COSA HO FATTO:
Per fare pratica col 'web scraping' ho ricavato i dati da questo sito, li ho salvati in un file CSV che poi ho analizzato in Jupyter Notebook.

Analisi del sito

Il sito 'worldometers' raccoglie statistiche mondiali, molte anche in tempo reale. Per questo progetto mi sono interessato a raccogliere i dati della popolazione mondiale per ogni stato. Cliccando sugli stati, si entra in un'altra pagina con i dettagli di popolazione specifici dello stato in esame. In particolare, mi sono interessato ai valori della popolazione nel tempo (dagli anni '50 ad oggi)

Countries in the world by population (2022)

This list includes both **countries** and **dependent territories**. Data based on the latest *United Nations Population Division* estimates.
Click on the name of the country or dependency for current estimates (live population clock), historical data, and projected figures.
See also: World Population

Search:

#	Country (or dependency)	Population (2020)	Yearly Change	Net Change	Density (P/Km²)	Land Area (Km²)	Migrants (net)	Fert. Rate	Med. Age	Urban Pop %
1	Honduras	**9,904,607**	1.63 %	158,490	89	111,890	-6,800	2.5	24	57 %
2	United Arab Emirates	**9,890,402**	1.23 %	119,873	118	83,600	40,000	1.4	33	86 %
3	Djibouti	**988,000**	1.48 %	14,440	43	23,180	900	2.8	27	79 %
4	Saint Barthelemy	**9,877**	0.30 %	30	470	21		N.A.	N.A.	0 %

Un po' di codice

Ho usato VS CODE per sviluppare la parte di web scraping.

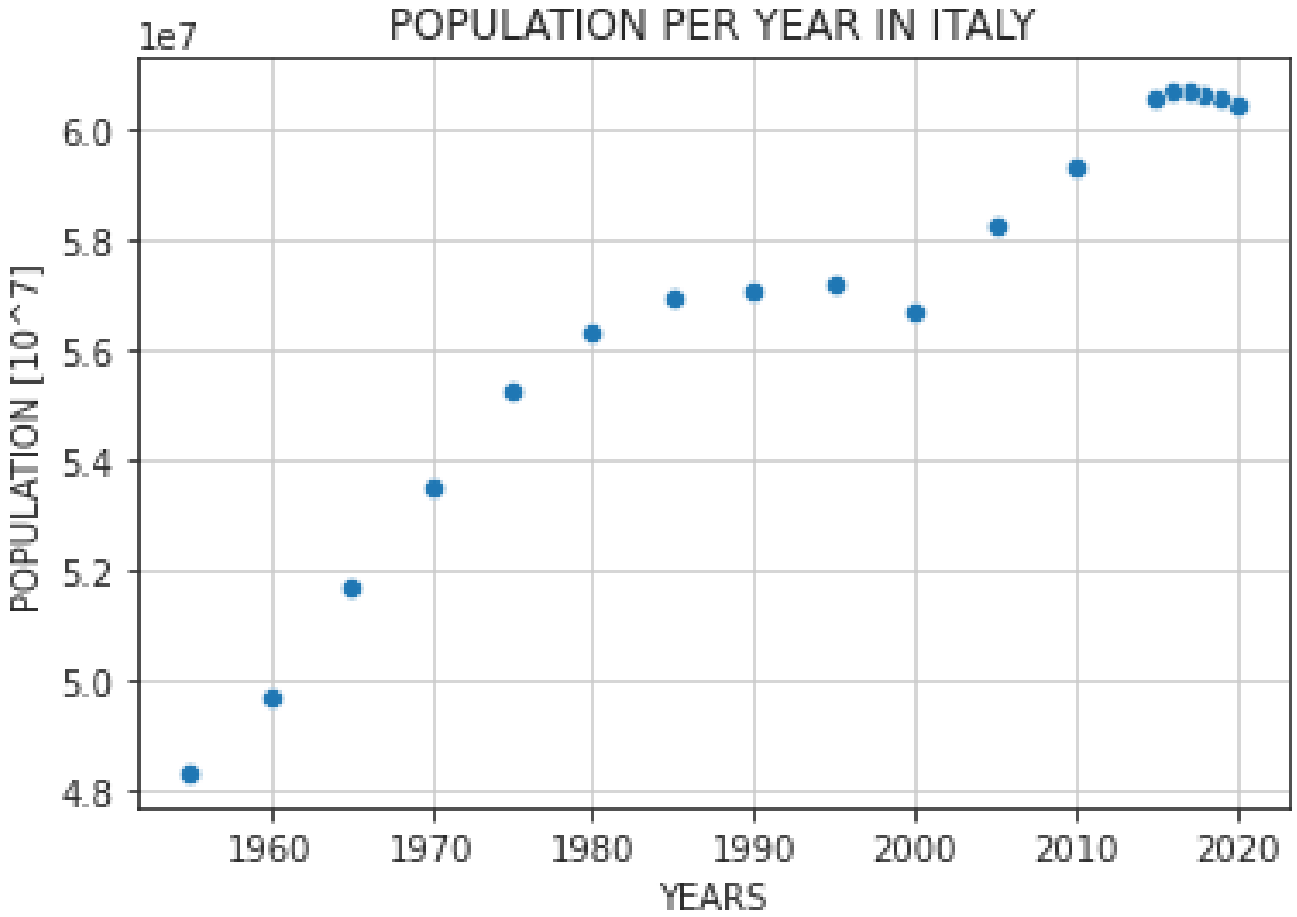
Ho creato un file CVS in cui poter salvare i dati a cui ero interessato, ovvero: "Country" (lo stato), "Population" (il valore di popolazione), "Year" (l'anno di riferimento)

worldometers > spiders > countries.py > ...

```python
import scrapy
import os
import csv

#creating a csv for recap, if it does not exist
if not os.path.exists("recap.csv"):
    recap = open ("recap.csv", "w")
    writer = csv.writer(recap, delimiter=";")
    writer.writerow(["Country", "Population", "Year"]) #defining the header
else:
    recap = open ("recap.csv", "a")
    writer = csv.writer(recap, delimiter=";")

#class by which get the links to get inside and scrape the infos
class CountriesSpider(scrapy.Spider):
    name = 'countries'
    allowed_domains = ['www.worldometers.info']
    start_urls = ['https://www.worldometers.info/world-population/population-by-country/']

    def parse(self, response):
        countries = response.xpath('//td/a') #getting countries
        for country in countries:
            name = country.xpath('.//text()').get() #getting countries names
            link = country.xpath('.//@href').get() #getting link for the next function
```

I dati dell'Italia

A questo punto, si hanno a disposizione i dati da poter analizzare.
Aprendo il CSV in Jupyter Notebook, si possono fare le classiche analisi. In questo caso, ho voluto visualizzare il trend di popolazione dell'Italia.



Conclusioni

Una possibile soluzione per risolvere il problema della mancanza dei dati, come abbiamo visto, è il web scraping che, con una certa facilità, ci permette di accedere ai dati disponibili sul web.

Una volta trovata la fonte dei dati (il sito, oppure più siti) è possibile salvarsi i dati in locale -per esempio, in un CSV, come in questo caso, per poi poterli analizzare.

Per vedere il progetto completo

Puoi vedere il progetto completo (tutto il codice ed anche un rimando ad un articolo che ho scritto su Medium) sulla mia repository di GitHub qui.