Lecture15_WebData

April 30, 2024

1 Web data in Python

- Download datasets from the internet
- Web scraping
- HTML: HyperText Markup Language
- NLTK: Natural Language ToolKit: a suite of Python libraries and programs for symbolic and statistical natural language processing for English

1.1 Downloading data from the internet

- It is not reproducible to download a dataset from the internet and read in locally.
- Downloading lots of data from the internet and saving locally before loading in to Python can take up a lot of time.
- We can use the urllib and requests packages to read a dataset from the internet into Python.

- URL: Uniform Resource Locator: a web resource specifying its location and a mechanism for retrieving it.
- A URL is effectively a web address.
- urlopen() accepts URLs as arguments instead of filenames.
- The next example is reproducible and saves time.
- It also makes it easy to download the other datasets from the same website, or to re-download the dataset in the case that it may have been updated.

1.2 Example of reading in a data file from the internet

[15]:

```
import pandas as pd

from urllib.request import urlretrieve

url = 'https://www.football-data.co.uk/mmz4281/2324/E0.csv'

import os

os.chdir("C:/Users/DKITStaff/OneDrive - Dundalk Institute of Technology/DKIT/
  ↪Programming for Data Analytics/Programming_2024_25/Datasets")

urlretrieve(url, 'pl_2324.csv')
```

```
pd.read_csv('pl_2324.csv')
```

```
[15]:
    Div        Date   Time       HomeTeam        AwayTeam  FTHG  FTAG FTR  \
0    E0  11/08/2023  20:00        Burnley        Man City     0     3   A
1    E0  12/08/2023  12:30        Arsenal   Nott'm Forest     2     1   H
2    E0  12/08/2023  15:00    Bournemouth        West Ham     1     1   D
3    E0  12/08/2023  15:00       Brighton           Luton     4     1   H
4    E0  12/08/2023  15:00        Everton          Fulham     0     1   A
..   ..         …    …            …             …   …   …  ..
341  E0  27/04/2024  17:30        Everton       Brentford     1     0   H
342  E0  27/04/2024  20:00    Aston Villa         Chelsea     2     2   D
343  E0  28/04/2024  14:00    Bournemouth        Brighton     3     0   H
344  E0  28/04/2024  14:00      Tottenham         Arsenal     2     3   A
345  E0  28/04/2024  16:30  Nott'm Forest        Man City     0     2   A

     HTHG  HTAG  … AvgC<2.5  AHCh  B365CAHH  B365CAHA   PCAHH  PCAHA  \
0       0     2  …     2.28  1.50      1.95      1.98    1.95   1.97
1       2     0  …     2.63 -2.00      1.95      1.98    1.93   1.97
2       0     0  …     2.12  0.00      2.02      1.91    2.01   1.92
3       1     0  …     2.48 -1.75      2.01      1.92    2.00   1.91
4       0     0  …     1.71 -0.25      2.06      1.87    2.04   1.88
..    …   …  …      …   …       …       …     …    …
341     0     0  …     1.93  0.00      2.10      1.80    2.16   1.78
342     2     0  …     2.86 -0.50      2.05      1.85    2.06   1.88
343     1     0  …     2.99 -0.50      1.95      1.95    1.98   1.95
344     0     3  …     2.58  0.75      1.92      2.01    1.93   1.99
345     0     1  …     2.55  1.50      2.03      1.90    2.04   1.90

     MaxCAHH  MaxCAHA  AvgCAHH  AvgCAHA
0        NaN      NaN     1.92     1.95
1       2.01     2.09     1.95     1.92
2       2.06     1.96     1.96     1.91
3       2.14     1.93     2.00     1.86
4       2.08     1.99     1.98     1.88
..       …      …      …      …
341     2.17     1.85     2.09     1.79
342     2.09     1.89     2.06     1.83
343     2.01     2.00     1.96     1.92
344     2.01     2.01     1.93     1.93
345     2.06     1.93     2.01     1.87

[346 rows x 106 columns]
```

1.3 Note

Notice that I specified the working directory into which the url dataset will be retrieved. Could also have specified the full address of the file in urlretrieve and in pd.read_csv.

2 Exercise

Find a dataset online that can be downloaded. Write code to automatically download the data and read it into Python.

3 HTML

3.1 HTTP, HTTPS, HTML definitions

- HTTP: HyperText Transfer Protocol
- HTTPS is a more secure form of HTTP.
- HTML: HyperText Markup Language

3.2 Extracting HTML from a webpage using urlopen, Request from urllib.request

[16]:

```
from urllib.request import urlopen, Request

url = 'https://www.autocarindia.com/bikes/bikes-under-2-lakhs/'

request = Request(url)

response = urlopen(request)

html = response.read()

response.close()
```

3.3 Will not output for the cell below in the notes because the output is too long.

[17]:

```
# html
```

3.4 Extracting HTML from a webpage using the Requests package

This saves the HTML as a string in the name 'html_text'.

[18]:

```
import requests

url = 'https://www.autocarindia.com/bikes/bikes-under-2-lakhs/'

r = requests.get(url)

html_text = r.text
```