Scraping BBC Sound Effects

January 7, 2020

1 Scraping BBC Sound Effects

1.0.1 Imports

[1]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from io import StringIO
import requests
```

1.0.2 Retrieving Metadata

We start by making a request to the metadata csv file location

[2]:
```
url = 'http://bbcsfx.acropolis.org.uk/assets/BBCSoundEffects.csv'
r = requests.get(url)

r
```

[2]:
```
<Response [200]>
```

We can use StringIO to convert the byte response into a format that can be understoon by pandas

[3]:
```
df = pd.read_csv(StringIO(r.text))

df.head()
```

[3]:

	location	description	secs	\
0	07076051.wav	Two-stroke petrol engine driving small elevato…	194	
1	07076050.wav	Single-cylinder Petter engine, start, run stop…	194	
2	07076049.wav	Start, constant run with engine driving small …	200	
3	07076048.wav	Two false starts, constant run, stop. (2 1/4 h…	195	
4	07076047.wav	An 8 mm projector running at 24 f.p.s.	117	

	category	CDNumber	CDName	tracknum
0	Engines: Petrol	EC117D	Diesel & Petrol Engines	4
1	Engines: Diesel	EC117D	Diesel & Petrol Engines	1
2	Engines: Petrol	EC117D	Diesel & Petrol Engines	3
3	Engines: Petrol	EC117D	Diesel & Petrol Engines	2
4	Cine Projectors	EC6C1	Cameras	4

1.0.3 Downloading the Files

We'll use a lambda function to help form the urls for each file

[4]:
```
location_2_url = lambda location: f'http://bbcsfx.acropolis.org.uk/assets/
↪{location}'

location = '07076051.wav'
location_url = location_2_url(location)

location_url
```

[4]:
```
'http://bbcsfx.acropolis.org.uk/assets/07076051.wav'
```

Finally we'll cycle through each of the files, downloading and saving the data

[6]:
```
df.index.name = 'id'

for file_id, file_metadata in df.iterrows():
    location = file_metadata['location']
    location_url = location_2_url(location)

    r = requests.get(location_url)

    with open(f'data/{location}', 'wb') as f:
        f.write(r.content)
```