assignment

May 12, 2020

0.0.1 Note

I got banned from accessing any yelp pages since I was trying the web scrape the first 500 pages (5000 businesses) from the yelp website. Therefore, I only have access to the first few and they have been added to the resulting csv file.

This is because it is against yelp's policy to web scrape their pages, according to https://www.yelp-support.com/article/Can-I-copy-or-scrape-data-from-the-Yelp-site?l=en_US.

The data will be short but can easily be increased to get the data of all businesses.

[1]:
```
from bs4 import BeautifulSoup
import requests
```

[254]:
```
#yelp url to access all businesses in los angeles
print('--------Getting URLs---------')
URL = ['https://www.yelp.com/search?
 ↪find_desc=&find_loc=Los%20Angeles%2C%20CA&ns=1&start={}'.format(i*10) for i␣
 ↪in range(20)]
print('Done.', end='\n\n')

#get the html content of all the pages in the url array
print('--------Getting page contents---------')
pages = [requests.get(url) for url in URL]
print('Done.', end='\n\n')

#set up beautiful soup objects to later analyse the pages
print('--------Setting up beautiful soup ---------')
soups = [BeautifulSoup(page.content, 'html.parser') for page in pages]
print('Done.', end='\n\n')
```

```
--------Getting URLs---------
Done.

--------Getting page contents---------
Done.

--------Setting up beautiful soup ---------
Done.
```

[255]:
```
#get all restaurant divs in each page
#2-dimensional array
business_pages = [soup.find_all("div", {"class": "lemon--div__373c0__1mboc␣
↪container__373c0__3HMKB hoverable__373c0__VqkG7 margin-t3__373c0__1l90z␣
↪margin-b3__373c0__q1DuY padding-t3__373c0__1gw9E padding-r3__373c0__57InZ␣
↪padding-b3__373c0__342DA padding-l3__373c0__1scQ0 border--top__373c0__3gXLy␣
↪border--right__373c0__1n3Iv border--bottom__373c0__3qNtD␣
↪border--left__373c0__d1B7K border-color--default__373c0__3-ifU"}) for soup␣
↪in soups]
```

[248]:
```
#class to store and obtain information about a bussiness
class Business:
    def __init__(self, HTML):
        self.HTML = HTML

    def name(self):
        name = self.HTML.find('a', {"class": "lemon--a__373c0__IEZFH␣
↪link__373c0__1G70M link-color--inherit__373c0__3dzpk␣
↪link-size--inherit__373c0__1VFlE"}).text

        return name

    def rating(self):

        rating = self.HTML.find('div',{"class": "lemon--div__373c0__1mboc␣
↪display--inline-block__373c0__1ZKqC border-color--default__373c0__3-ifU"})
        rating = rating.find('span').find('div').get('aria-label')
        rating = rating.replace(' star rating', '')

        return rating

    def price(self):
        price = self.HTML.find('div',{"class": "lemon--div__373c0__1mboc␣
↪priceCategory__373c0__3zW0R display--inline-block__373c0__1ZKqC␣
↪border-color--default__373c0__3-ifU"})
        price = price.find('span').find('span')
        price = price.text
        return price

    def num_reviews(self):
        reviews = self.HTML.find('div', {'class': "lemon--div__373c0__1mboc␣
↪attribute__373c0__1hPI_ display--inline-block__373c0__1ZKqC␣
↪border-color--default__373c0__3-ifU"})

        reviews = reviews.find('span').text
        return reviews
```

```
    def keywords(self):
        keywords = self.HTML.find('div', {'class': 'lemon--div__373c0__1mboc␣
↪priceCategory__373c0__3zW0R display--inline-block__373c0__1ZKqC␣
↪border-color--default__373c0__3-ifU'})

        keywords = keywords.find_all('span', {'class':␣
↪'lemon--span__373c0__3997G display--inline__373c0__3JqBP␣
↪border-color--default__373c0__3-ifU'})[1]
        keys = []

        for i in range(len(keywords)):
            k = keywords.find_all('span', {'class': 'lemon--span__373c0__3997G␣
↪text__373c0__2Kxyz text-color--black-extra-light__373c0__2OyzO␣
↪text-align--left__373c0__2XGa-'})[i]
            keys += [k.text.replace(', ', '')]

        return keys

    def number(self):
        number = self.HTML.find('p',{"class": "lemon--p__373c0__3Qnnj␣
↪text__373c0__2Kxyz text-color--black-extra-light__373c0__2OyzO␣
↪text-align--right__373c0__1f0KI text-size--small__373c0__3NVWO"})

        return number.text

    def address(self):
        address = self.HTML.find_all('p', {'class': 'lemon--p__373c0__3Qnnj␣
↪text__373c0__2Kxyz text-color--black-extra-light__373c0__2OyzO␣
↪text-align--right__373c0__1f0KI text-size--small__373c0__3NVWO'})[1:]
        address = [ad.text for ad in address]

        address = '; '.join(address)
        return address
```

[256]:
```
f = open("businesses.csv", "w")

header = "Name, Rating, Price, Number of Reviews, Keywords, Phone Number,␣
 ↪Address"

f.write(header)
```

[256]: 71

[257]:
```
for page in business_pages:
    for business in page:
        business = Business(business)

        try:
            name = business.name()
        except:
            name = ""
        try:
            rating = business.rating()
        except:
            rating = ""
        try:
            price = business.price()
        except:
            price = ""
        try:
            num_reviews = business.num_reviews()
        except:
            num_reviews = ""
        try:
            keywords = business.keywords()
        except:
            keywords = ""
        try:
            number = business.number()
        except:
            number = ""
        try:
            address = business.address()
        except:
            address = ""

        f.write('\n' + name + ',' + rating + ',' + price + ',' + num_reviews +␣
↪',' + str(keywords).replace(',', ';') + ',' + number + ',' + address)
```

[258]:
```
f.close()
```