scraping-github-topics-repositories

October 27, 2022

1 Scraping Top Repositories for Topics on GitHub

TODO (Introduction): - Introduction about web scraping - Introduction about GitHub and the problem statement - Mention the tools you're using (Python, requests, Beautiful Soup,Pandas)

Here are the staps we'll follow:

- we're going to scrape https://github.com/topics
- we'll get a list of topics. For each topic we'll get topic title, topic page URL and topic description
- For each topic, we'll get the top 25 repositories in the topic from the topic page
- For each repository we'll grab the Repo name, Username , stars and repo URL

- For each topic we'll create CSV file in the following format:

1.1 Scrape the list of topics from GitHub

Explain how you'll do it.

- use requests to download the pages
- user BS4 to parse and extrac information
- convert to a Pandas dataframe

Let's write a function to download the page.

[8]:
```python
import requests
from bs4 import BeautifulSoup
def get_topics_page():
    topics_url = 'https://github.com/topics'
    response = requests.get(topics_url)
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(topic_url))
    doc = BeautifulSoup(response.text,'html.parser')
    return doc
```

Add some explanation

[9]:
```python
doc = get_topics_page()
```

[10]:
```python
doc.find('a')
```

[10]:
```
<a class="px-2 py-4 color-bg-accent-emphasis color-fg-on-emphasis show-on-focus
js-skip-to-content" href="#start-of-content">Skip to content</a>
```

Let's create some helper function to parse information from the page.

To get topic titles, we can pick `p` tags with the `class`….

[11]:
```
!pip install pandas --quiet
```

[12]:
```
import pandas as pd
```

[13]:
```
def get_topic_titles(doc):
    selection_class = 'f3 lh-condensed mb-0 mt-1 Link--primary'
    topic_title_tags = doc.find_all('p',{'class': selection_class})
    topic_titles = []
    for tag in topic_title_tags:
        topic_titles.append(tag.text)
    return topic_titles
```

get_topic_titles can be used to get the list of titles

[14]:
```
titles = get_topic_titles(doc)
```

[15]:
```
len(titles)
```

[15]:
```
30
```

[17]:
```
titles[:5]
```

[17]:
```
['3D', 'Ajax', 'Algorithm', 'Amp', 'Android']
```

Similarly we have defined functions for description and URLs.

[14]:
```
def get_topic_descs(doc):
    desc_selector = 'f5 color-fg-muted mb-0 mt-1'
    topic_desc_tags = doc.find_all('p',{'class' : desc_selector})
    topic_descs = []
    for tag in topic_desc_tags:
        topic_descs.append(tag.text.strip())
    return topic_descs
```

TODO - example and explanation

[15]:
```
def get_topic_urls(doc):
    topic_link_tags = doc.find_all('a',{'class': 'no-underline flex-grow-0'})
    topic_urls = []
    base_url = 'https://github.com'

    for tag in topic_link_tags:
        topic_urls.append(base_url + tag['href'])
    return topic_urls
```

Let's put this all together into a single function

[16]:
```
def scrape_topics():
    topics_url = 'https://github.com/topics'
    response = requests.get(topics_url)
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(topic_url))
    doc = BeautifulSoup(response.text,'html.parser')
    topics_dict = {
        'title': get_topic_titles(doc),
        'description' : get_topic_descs(doc),
        'url' : get_topic_urls(doc)
    }
    return pd.DataFrame(topics_dict)
```

[17]:
```
!pip install jovian --upgrade --quiet
```

[18]:
```
import jovian
```

[19]:
```
# Execute this to save new versions of the notebook
jovian.commit(project="scraping-github-topics-repositories")
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "mansishah9865/scraping-github-topics-repositories"
on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/mansishah9865/scraping-
github-topics-repositories
```

[19]:
```
'https://jovian.ai/mansishah9865/scraping-github-topics-repositories'
```

1.2 Get the top 25 repositories from a topic page

TODO - explanation and step

[20]:
```
def get_topic_page(topic_url):
     # Download the page
    response = requests.get(topic_url)
    # Check succesful response
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(topic_url))
    # Parse using Beautiful Soup
    topic_doc = BeautifulSoup(response.text,'html.parser')
    return topic_doc
```

[21]:

```
doc = get_topic_page('https://github.com/topics/3d')
```

TODO - talk about the h3 tags

[22]:

```
def parse_star_count(stars):
    stars = stars.strip()
    if stars[-1] == 'k':
        return int(float(stars[:-1])*1000)
    return (int(stars))
```

[23]:

```
def get_repo_info(h3_tag, star_tag):
    base_url = 'https://github.com'
    # returns all the required info about a repository
    a_tags =h3_tag.find_all('a')
    username = a_tags[0].text.strip()
    repo_name = a_tags[1].text.strip()
    repo_url = base_url + a_tags[1]['href']
    stars = parse_star_count(star_tag.text.strip())
    return username, repo_name, stars, repo_url
```

TODO - show an example

[24]:

```
def get_topic_repos(topic_doc):
    # Get the h3 tags containing repo title, repo URL and username
    repo_tags = topic_doc.find_all('article',{'class': 'border rounded␣
 ↪color-shadow-small color-bg-subtel my-4'})
    # Get star tags
    star_tags = topic_doc.find_all('span',{'class' : 'Counter js-social-count'})
    topic_repos_dict = {'username' : [],'repo_name' : [],'stars' : [],␣
 ↪'repo_url' : []}

    # Get repo info
    for i in range(len(repo_tags)):
        repo_info = get_repo_info(repo_tags[i],star_tags[i])
        topic_repos_dict['username'].append(repo_info[0])
        topic_repos_dict['repo_name'].append(repo_info[1])
        topic_repos_dict['stars'].append(repo_info[2])
        topic_repos_dict['repo_url'].append(repo_info[3])

    return pd.DataFrame(topic_repos_dict)
```

TODO - show an example

[31]:

```
import os
def scrape_topic(topic_url,path):
    if os.path.exists(path):
        print("The file {} already exists. Skipping...".format(path))
        return
    topic_df = get_topic_repos(get_topic_page(topic_url))
    topic_df.to_csv(path, index = None)
```

TODO - show an example

[ ]:

```

```

1.3 Putting it all together

- We have a function to get the list of topics
- We have a function to create a CSV file for scraped repos from a topics page
- Let's create a function to put them together

[38]:

```
import os
def scrape_topics_repos():
    print('Scraping list of topics')
    topics_df = scrape_topics()

    os.makedirs('data',exist_ok=True)
    for index, row in topics_df.iterrows():
        print('Scraping top repositories for "{}"'.format(row['title']))
        scrape_topic(row['url'],'data/{}.csv'.format(row['title']))
```

Let's run it to scrape the top repos for the topics on the first page of https://github.com/topics

[39]:

```
scrape_topics_repos()
```

```
Scraping list of topics
Scraping top repositories for "3D"
The file data/3D.csv already exists. Skipping…
Scraping top repositories for "Ajax"
The file data/Ajax.csv already exists. Skipping…
Scraping top repositories for "Algorithm"
The file data/Algorithm.csv already exists. Skipping…
Scraping top repositories for "Amp"
The file data/Amp.csv already exists. Skipping…
Scraping top repositories for "Android"
The file data/Android.csv already exists. Skipping…
Scraping top repositories for "Angular"
The file data/Angular.csv already exists. Skipping…
Scraping top repositories for "Ansible"
The file data/Ansible.csv already exists. Skipping…
Scraping top repositories for "API"
The file data/API.csv already exists. Skipping…
Scraping top repositories for "Arduino"
The file data/Arduino.csv already exists. Skipping…
Scraping top repositories for "ASP.NET"
The file data/ASP.NET.csv already exists. Skipping…
```

```
Scraping top repositories for "Atom"
The file data/Atom.csv already exists. Skipping…
Scraping top repositories for "Awesome Lists"
The file data/Awesome Lists.csv already exists. Skipping…
Scraping top repositories for "Amazon Web Services"
The file data/Amazon Web Services.csv already exists. Skipping…
Scraping top repositories for "Azure"
The file data/Azure.csv already exists. Skipping…
Scraping top repositories for "Babel"
The file data/Babel.csv already exists. Skipping…
Scraping top repositories for "Bash"
The file data/Bash.csv already exists. Skipping…
Scraping top repositories for "Bitcoin"
The file data/Bitcoin.csv already exists. Skipping…
Scraping top repositories for "Bootstrap"
The file data/Bootstrap.csv already exists. Skipping…
Scraping top repositories for "Bot"
The file data/Bot.csv already exists. Skipping…
Scraping top repositories for "C"
The file data/C.csv already exists. Skipping…
Scraping top repositories for "Chrome"
The file data/Chrome.csv already exists. Skipping…
Scraping top repositories for "Chrome extension"
The file data/Chrome extension.csv already exists. Skipping…
Scraping top repositories for "Command line interface"
The file data/Command line interface.csv already exists. Skipping…
Scraping top repositories for "Clojure"
The file data/Clojure.csv already exists. Skipping…
Scraping top repositories for "Code quality"
The file data/Code quality.csv already exists. Skipping…
Scraping top repositories for "Code review"
The file data/Code review.csv already exists. Skipping…
Scraping top repositories for "Compiler"
The file data/Compiler.csv already exists. Skipping…
Scraping top repositories for "Continuous integration"
The file data/Continuous integration.csv already exists. Skipping…
Scraping top repositories for "COVID-19"
The file data/COVID-19.csv already exists. Skipping…
Scraping top repositories for "C++"
The file data/C++.csv already exists. Skipping…
```

We can check that the CSVs were created properly

[40]:
```
# read and display a CSV using Pandas
```

[ ]:
```

```

[71]:
```
import jovian
```

[72]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "mansishah9865/scraping-github-topics-repositories"
on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/mansishah9865/scraping-
github-topics-repositories
```

[72]:
```
'https://jovian.ai/mansishah9865/scraping-github-topics-repositories'
```

1.4 References and Future Work

Summary of what we did - ? - ?

References to links you found useful - ? - ?

Ideas for future work - ? - ?

[20]:
```
import jovian
```

[21]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "mansishah9865/scraping-github-topics-repositories"
on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/mansishah9865/scraping-
github-topics-repositories
```

[21]:
```
'https://jovian.ai/mansishah9865/scraping-github-topics-repositories'
```

[ ]:
```

```