UND 4

FM

INTRODUÇÃO A BIBLIOTECA PANDAS

Vanessa Cadan Scheffer

WEB SCRAPING

Técnica de extração de dados utilizada para coletar dados de sites através de tags HTML e atributos CSS.

Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

DESAFIO

Em um artigo publicado no dia 06 de março de 2019, no portal Computer World, o autor fala sobre o profissional que deseja seguir a carreira de analista de dados, o qual deve ter habilidades em: filtrar dados, construir APIs, web scraping e ter conhecimento nas linguagens Git, MySQL e Python. (MOTIM, Raphael Bueno da. Carreira de analista de dados oferece salários de até R$ 12,5 mil. 2019. Disponível em: https://computerworld.com.br/2019/03/06/carreira-de-analista-de-dados-oferece-salarios-de-ate-r-125-mil/. Acesso em: 17 jun. 2020).

Como desenvolvedor em uma empresa de consultoria de software, você foi alocado em uma equipe de marketing analítico em uma marca esportiva, que necessita fazer a coleta das principais notícias do mundo de esporte em um determinado portal. O cliente pedi para que o portal https://globoesporte.globo.com/. O cliente deseja um componente capaz de fazer a extração dos dados em forma tabular, com o seguintes campos: manchete, descrição, link, seção, hora da extração, tempo decorrido da publicação até a hora da extração. O Quadro 4.1 apresenta, visualmente, como os dados devem ser organizados e exibidos.

Quadro 4.1 - Resultado esperado

manchete	descrição	link	seção	hora_extração	time_delta
Título da manchete	Descrição da manchete (quando houver)	link para a notícia	Seção que a notícia foi marcada	Data e hora da extração	Quanto tempo se passou da hora da publicação até a extração

Fonte: elaborada pela autora.

O grande desafio no trabalho de web scraping é descobrir qual o padrão nas tags HTML e atributos CSS usados. Pois somente através deles é que conseguiremos alcançar a informação solicitada. Como o cliente já trabalha com o portal de notícias, foram lhe passadas as seguintes informações técnicas que o ajudarão a fazer a extração.

Para extração de todas as informações localize todas as div com atributo 'class':'feed-post-body'. De cada item localizado extraia:

- A manchete que ocupa a primeira posição do conteúdo.
- O link que pode ser localizado pela tag "a" e pelo atributo "href".
- A descrição pode estar na terceira posição conteúdo ou dentro de uma div com atributo 'class':'bstn-related'
- A seção está dentro de uma div com atributo 'class':'feed-post-metadata'. Localize o span com atributo 'class': 'feed-post-metadata-section'.
- O tempo decorrido está uma div com atributo 'class':'feed-post-metadata'. Localize o span com atributo 'class': 'feed-post-datetime'.

Caso tente acessar o texto de uma tag não localizada, um erro é dado, para evitar esses casos, os campos descrição, seção e time_delta devem ser tratados para esses casos, retornando None (nulo). Agora é com você, faça a implementação e gere um DataFrame com as informações solicitadas.

RESOLUÇÃO

Para fazer o web scraping solicitado, vamos utilizar as bibliotecas requests, BeautifulSoup, pandas e datetime. As duas primeiras serão usadas para fazer a captura do conteúdo da página, pandas para entregar os resultados em forma estruturada e datetime para marcar o dia e hora da extração.

In [22]:

```
from datetime import datetime


import requests
from bs4 import BeautifulSoup
import pandas as pd
```

Com as bibliotecas importadas, vamos acessar o portal e utilizar a propriedade text da biblioteca requests para capturar em formato de string. Em seguida, vamos transformar essa string em formato html, para que possamos localizar as tags de nosso interesse. Na linha 2, registramos o horário da extração. Na linha 5, procuramos todas as tags div com o atributo que nos foi indicado. Essa linha retornará uma lista com cada notícia. Veja que na linha 6 imprimimos quantas notícias foram encontradas e na linha 7 imprimimos o conteúdo da primeira notícia. Lembre-se que `contents` transforma cada início e final da div em um elemento da lista.

In [23]:

```
texto_string = requests.get('https://globoesporte.globo.com/').text
hora_extracao = datetime.now().strftime("%d-%m-%Y %H:%M:%S")

bsp_texto = BeautifulSoup(texto_string, 'html.parser')
lista_noticias = bsp_texto.find_all('div', attrs={'class':'feed-post-body'})
print("Quantidade de manchetes = ", len(lista_noticias))
lista_noticias[0].contents
```

```
Quantidade de manchetes =  10
```

Out[23]:

```
[<div class="feed-post-header"></div>,
 <div class="_label_event"><div class="feed-post-body-title gui-color-primary
gui-color-hover"><div class="_ee"><a class="feed-post-link gui-color-primary
gui-color-hover" href="https://globoesporte.globo.com/futebol/futebol-
internacional/futebol-italiano/jogo/17-06-2020/napoli-juventusita.ghtml">VICE
SENHORA</a></div></div></div>,
 <div class="_label_event"><div class="feed-post-body-resumo">Napoli vence
Juventus nos pênaltis e leva Copa da Itália</div></div>,
 <div class="feed-media-wrapper"><div class="_label_event"><a class="feed-
post-figure-link gui-image-hover"
href="https://globoesporte.globo.com/futebol/futebol-internacional/futebol-
italiano/jogo/17-06-2020/napoli-juventusita.ghtml"><div class="bstn-fd-item-
cover"><picture class="bstn-fd-cover-picture"><img alt="Foto: (Alberto
Lingria/Reuters)" class="bstn-fd-picture-image"
src="https://s2.glbimg.com/BeTGAixT5O_Cvs4hQA88PdHiCsY=/0x0:5406x3041/540x304/
06-17t212444z-1091152315-rc29bh9icqss-rtrmadp-3-soccer-italy-nap-juv-
report.jpg"
srcset="https://s2.glbimg.com/BeTGAixT5O_Cvs4hQA88PdHiCsY=/0x0:5406x3041/540x3
06-17t212444z-1091152315-rc29bh9icqss-rtrmadp-3-soccer-italy-nap-juv-
report.jpg 1x,https://s2.glbimg.com/A6dTbIFD8sDl_t7eMHvA-
2ONF0Y=/0x0:5406x3041/810x456/smart/https://i.s3.glbimg.com/v1/AUTH_bc8228b667
06-17t212444z-1091152315-rc29bh9icqss-rtrmadp-3-soccer-italy-nap-juv-
report.jpg 1.5x,https://s2.glbimg.com/n_XVqiX_3nn_wSar4FYy5I-
cPUw=/0x0:5406x3041/1080x608/smart/https://i.s3.glbimg.com/v1/AUTH_bc8228b6673
06-17t212444z-1091152315-rc29bh9icqss-rtrmadp-3-soccer-italy-nap-juv-
report.jpg 2x" title="Foto: (Alberto Lingria/Reuters)"/></picture></div></a>
</div></div>,
 <div class="feed-post-metadata"><span class="feed-post-datetime">Há 3
horas</span><span class="feed-post-metadata-section"> futebol italiano
</span></div>]
```

Dentro dessa estrutura, procurando pelas tags corretas, vamos encontrar todas as informações que foram solicitadas. Pela saída anterior podemos ver que a manchete ocupa a posição 2 da lista de conteúdos, logo para guardar a manchete devemos fazer:

In [24]:

```
lista_noticias[0].contents[1].text.replace('"',"")
```

Out[24]:

```
'VICE SENHORA'
```

Para extração do link para notícia, como ele se encontra também na posição 1 da lista, vamos utilizar o método find('a') para localizá-lo e extrair da seguinte forma:

In [25]:

```
lista_noticias[0].find('a').get('href')
```

Out[25]:

```
'https://globoesporte.globo.com/futebol/futebol-internacional/futebol-
italiano/jogo/17-06-2020/napoli-juventusita.ghtml'
```

Para a descrição, como ela pode estar na terceira posição ou em outra tag, vamos ter que testar em ambas e caso não esteja, então retornar None (nulo). Veja a seguir.

In [26]:

```
descricao = lista_noticias[0].contents[2].text
if not descricao:
    descricao = noticia.find('div', attrs={'class': 'bstn-related'})
    descricao = descricao.text if descricao else None # Somente acessará a
propriedade text caso tenha encontrado ("find")
descricao
```

Out[26]:

```
'Napoli vence Juventus nos pênaltis e leva Copa da Itália'
```

Para extração da seção e do tempo decorrido, vamos acessar primeiro o atributo `'feed-post-metadata'` e guardar em uma variável, para em seguida, dentro desse novo subconjunto, localizar os atributos `'feed-post-datetime'` e `'feed-post-metadata-section'`. Como existe a possibilidade dessa informação não existir, precisamos garantir que somente acessaremos a propriedade text (linhas 6 e 7) caso tenha encontrando ("find"). Veja a seguir

In [27]:

```
metadados = lista_noticias[0].find('div', attrs={'class':'feed-post-
metadata'})

time_delta = metadados.find('span', attrs={'class': 'feed-post-datetime'})
secao = metadados.find('span', attrs={'class': 'feed-post-metadata-section'})

time_delta = time_delta.text if time_delta else None
secao = secao.text if secao else None

print('time_delta = ', time_delta)
print('seção = ', secao)
```

```
time_delta =  Há 3 horas
seção =   futebol italiano
```

Veja que para a notícia 0 extraímos todas as informações solicitadas, mas precisamos extrair de todas, portanto cada extração deve ser feita dentro de uma estrutura de repetição. Para criar um DataFrame com os dados, vamos criar uma lista vazia e a cada iteração apendar uma tupla com as informações extraídas. Com essa lista, podemos criar nosso DataFrame, passando os dados e os nomes das colunas. Veja a seguir:

In [28]:

```
dados = []

for noticia in lista_noticias:
    manchete = noticia.contents[1].text.replace('"',"")
    link = noticia.find('a').get('href')

    descricao = noticia.contents[2].text
    if not descricao:
        descricao = noticia.find('div', attrs={'class': 'bstn-related'})
        descricao = descricao.text if descricao else None

    metadados = noticia.find('div', attrs={'class':'feed-post-metadata'})
    time_delta = metadados.find('span', attrs={'class': 'feed-post-
datetime'})
    secao = metadados.find('span', attrs={'class': 'feed-post-metadata-
section'})

    time_delta = time_delta.text if time_delta else None
    secao = secao.text if secao else None

    dados.append((manchete, descricao, link, secao, hora_extracao,
time_delta))

df = pd.DataFrame(dados, columns=['manchete', 'descrição', 'link', 'seção',
'hora_extração', 'time_delta'])
df.head()
```

Out[28]:

	manchete	descrição	link	seção	hora_extração
0	VICE SENHORA	Napoli vence Juventus nos pênaltis e leva Copa...	https://globoesporte.globo.com/futebol/futebol...	futebol italiano	17-06-2020 18:58:17
1	ESPERA AÍ, LIVERPOOL	Em noite trágica de David Luiz, Manchester Cit...	https://globoesporte.globo.com/futebol/futebol...	futebol inglês	17-06-2020 18:58:17
2	BASTIDORES CONTURBADOS	João Doria só libera volta aos treinos a parti...	https://globoesporte.globo.com/sp/futebol/camp...	campeonato paulista	17-06-2020 18:58:17

	manchete	descrição	link	seção	hora_extração
3	Na véspera de Flamengo e Bangu, Ferj lança nov...	Maracanã passa por processos de higienização e...	https://globoesporte.globo.com/futebol/noticia...	futebol	17-06-202 18:58:17
4	Doutor na terra do Padim Ciço: as memórias do ...	Partida em junho de 1984 marcou a despedida de...	https://globoesporte.globo.com/ce/futebol/noti...	futebol	17-06-2020 18:58:17

Ver anotações

Vamos tornar nossa entrega mais profissional e transformar a solução em uma classe, assim toda vez que for preciso fazer a extração, basta instanciar um objeto e executar o método de extração.

In [29]:

```
from datetime import datetime

import requests
from bs4 import BeautifulSoup
import pandas as pd

class ExtracaoPortal:
    def __init__(self):
        self.portal = None

    def extrair(self, portal):
        self.portal = portal
        texto_string = requests.get('https://globoesporte.globo.com/').text
        hora_extracao = datetime.now().strftime("%d-%m-%Y %H:%M:%S")

        bsp_texto = BeautifulSoup(texto_string, 'html.parser')
        lista_noticias = bsp_texto.find_all('div', attrs={'class':'feed-post-
body'})

        dados = []

        for noticia in lista_noticias:
            manchete = noticia.contents[1].text.replace('"',"")
            link = noticia.find('a').get('href')

            descricao = noticia.contents[2].text
            if not descricao:
                descricao = noticia.find('div', attrs={'class': 'bstn-
related'})
                descricao = descricao.text if descricao else None

            metadados = noticia.find('div', attrs={'class':'feed-post-
metadata'})
            time_delta = metadados.find('span', attrs={'class': 'feed-post-
datetime'})
            secao = metadados.find('span', attrs={'class': 'feed-post-
metadata-section'})

            time_delta = time_delta.text if time_delta else None
            secao = secao.text if secao else None

            dados.append((manchete, descricao, link, secao, hora_extracao,
time_delta))

        df = pd.DataFrame(dados, columns=['manchete', 'descrição', 'link',
'seção', 'hora_extração', 'time_delta'])
        return df
```

In [30]:

```
df = ExtracaoPortal().extrair("https://globoesporte.globo.com/")
df.head()
```

Out[30]:

	manchete	descrição	link	seção	hora_extração
0	VICE SENHORA	Napoli vence Juventus nos pênaltis e leva Copa...	https://globoesporte.globo.com/futebol/futebol...	futebol italiano	17-06-20 18:58:18
1	ESPERA AÍ, LIVERPOOL	Em noite trágica de David Luiz, Manchester Cit...	https://globoesporte.globo.com/futebol/futebol...	futebol inglês	17-06-2020 18:58:18
2	BASTIDORES CONTURBADOS	João Doria só libera volta aos treinos a parti...	https://globoesporte.globo.com/sp/futebol/camp...	campeonato paulista	17-06-2020 18:58:18
3	Na véspera de Flamengo e Bangu, Ferj lança nov...	Maracanã passa por processos de higienização e...	https://globoesporte.globo.com/futebol/noticia...	futebol	17-06-2020 18:58:18
4	Doutor na terra do Padim Ciço: as memórias do ...	Partida em junho de 1984 marcou a despedida de...	https://globoesporte.globo.com/ce/futebol/noti...	futebol	17-06-2020 18:58:18

Ver anotações

DESAFIO DA INTERNET

Ganhar habilidade em programação exige estudo e treino (muito treino). Acesse o endereço https://medium.com/data-hackers/como-fazer-web-scraping-em-python-23c9d465a37f e pratique um pouco mais essa habilidade!