Relatório LAMIA 8
Prática: Web Scraping com Python p/ Ciência de Dados (II)

Kaique Medeiros Lima

Introdução

Esse relatório tende a abordar os principais conteúdos do CARD 8, “Prática: Web Scraping com Python p/ Ciência de Dados (II)”, o qual se consiste em assistir a uma vídeo aula do canal freecodecamp.org sobre Web Scraping em Python, fazendo uso da biblioteca BeautifulSoup4.

Descrição da atividade

Os principais conteúdos apresentados pelo vídeo foram os seguintes:

- **BeautifulSoup4:** É um módulo da biblioteca BS4, serve para a análise e extração de conteúdo de arquivos HTML. Essa biblioteca é muito utilizada nos projetos de Web Scraping, pois é a encarregada de manipular os arquivos.
- **Biblioteca Requests:** Faz o *request* de sites para a manipulação do conteúdo em Python, assim, fazendo com que possamos ter uma conexão com o site. Ela permite essa conexão que, em junção a manipulação do arquivo com a biblioteca BeautifulSoup4, dão vida às técnicas de Web Scraping.
- **find_all(‘tag’, class_=’class_name’):** Retorna uma array contendo todas as ocorrências específicas da tag com o nome da classe, “filtrando” o arquivo HTML. Sendo útil quando precisamos extrair vários elementos da página.
- **find(‘tag’, class_=’class_name’):** Parecido com o find_all(), mas só retorna a primeira das ocorrências.
- **.text**: É usado para extrair o texto contido dentro de uma tag HTML, descartando todas as tags e a sintaxe HTML. Isso simplifica a obtenção de apenas o conteúdo textual, sem as marcas de formatação HTML.
- **requests.get(‘https://example.com’).text**: Faz uma requisição GET para o link especificado e retorna o conteúdo da página em formato de texto. Se retornar o número 200, significa que a conexão foi bem-sucedida e a página foi carregada corretamente. Muito útil para garantir que o conteúdo desejado foi obtido com sucesso antes de prosseguir com a análise.

Além de técnicas de obtenção de valores específicos utilizando for loops, o vídeo também ensina a escrever esses valores diretamente em arquivos de formato .txt.

```python
for index, job in enumerate(jobs): # pega todas as vagas de emprego
        date = job.find('span', class_='sim-posted').text

        if 'few' in date: # se tiver few na data, printa
            empresa = job.find('h3', class_='joblist-comp-name').text.replace(' ', '')
            skill = job.find('span', class_='srp-skills').text.replace('  ', '')
            link = job.header.h2.a['href'] # pega o link da vaga acessando um por um até chegar no href

            if del_skill not in skill: # se o filtro n estiver nas habilidades, printa
                with open(f'texistes/{index}.txt', 'w') as f: # cria um arquivo de texto com o nome do index
                    f.write(f'empresa: {empresa.strip()}\n')
                    f.write(f'habilidades: {skill.strip()}\n')
                    f.write(f'link: {link}\n')

                print(f'arquivo {index} salvo com sucesso')
```

Conclusões

A importância do Web Scraping na área de IA e Machine Learning é a sua extração de dados de páginas da internet, os quais são usados para o aprendizado da máquina. Com a crescente quantidade de dados disponíveis online, a coleta e análise de grandes volumes de informações em tempo real são automatizadas pela biblioteca Requests, combinada com BeautifulSoup4 (BS4), que oferecem ferramentas robustas para a manipulação dos dados. Essa automação não só economiza tempo, mas também permite a adaptação e resposta rápida às mudanças extremamente rápidas dos dias atuais.