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

Breno Augusto Oliveira Abrantes

Descrição da Atividade

O objetivo principal desta atividade foi aprender a coletar dados da web utilizando web scraping com a biblioteca BeautifulSoup em Python. Durante a prática, diferentes aspectos da estrutura HTML foram analisados, e técnicas para manipular e extrair informações relevantes foram implementadas.

Estrutura HTML e Introdução ao Web Scraping

O primeiro ponto abordado foi a estrutura básica de um documento HTML. A estrutura do HTML é muito útil para realizar web scraping, pois permite identificar e acessar as informações que queremos dentro de uma página.

Os principais componentes a seção <body>, que abriga o conteúdo visível da página, como textos, imagens e links; e também elementos essenciais, como <footer> e <title>, que representam respectivamente o rodapé e o título.

Biblioteca BeautifulSoup e Parsing de HTML

Após a compreensão da estrutura HTML, foi introduzida a biblioteca BeautifulSoup, uma ferramenta poderosa para extrair dados de documentos HTML e XML. Para complementar essa biblioteca, utilizou-se também o lxml, um parser eficiente para analisar e interpretar o HTML, fazendo com que haja uma exibição dos elementos de forma mais organizada.

A instalação de ambas as bibliotecas foi o primeiro passo, possibilitando a manipulação de documentos HTML de maneira eficiente. A BeautifulSoup facilita a busca por elementos específicos, como cabeçalhos, parágrafos e listas, além de permitir o acesso a classes e IDs dentro do HTML.

Manipulação de Arquivos HTML Locais

A segunda etapa envolveu a manipulação de arquivos HTML locais. Isso foi importante para simular ambientes de scraping de maneiras locais, onde as páginas HTML já estão disponíveis na máquina. Ao trabalhar com arquivos locais, é possível testar e refinar técnicas de scraping

em um ambiente estável. Ainda que não seja uma prática tão comum por não estar em contato com um site já disposto na web, é útil para manejar e testar a melhor forma para manutenção daquele HTML.

A utilização da função open() do Python permitiu o carregamento de arquivos HTML para análise, e métodos como o prettify() da BeautifulSoup foram usados para formata-lo de maneira legível.

Scraping de Sites Reais

Na terceira etapa, foi introduzido o conceito de scraping em sites reais, com foco na extração de informações diretamente da web. A biblioteca requests foi utilizada para obter o conteúdo HTML de uma página web, que posteriormente seria analisado e manipulado com a BeautifulSoup. Esse processo foi demonstrado através do scraping de um site de empregos, onde foram extraídas informações sobre vagas de trabalho, como o nome da empresa e as habilidades requeridas para o cargo.

Essa aplicação prática demonstrou como o web scraping pode ser utilizado para automatizar a coleta de grandes volumes de dados de forma eficiente. Tais técnicas são amplamente empregadas em diversas áreas, como coleta de preços de produtos, monitoramento de notícias e análise de tendências de mercado.

Filtragem de Dados e Automação de Tarefas

Outro ponto importante abordado foi a filtragem de dados com base em critérios específicos. No exemplo discutido, a filtragem das vagas de emprego foi realizada com base nas habilidades que o usuário não possuía. Mostrando como o scraping pode ser ajustado para pegar informações relevantes para necessidades específicas, afunilando a coleta de dados de acordo com os interesses do usuário.

No final, foi adicionada uma função para limpar arquivos de texto antigos e gerar novos arquivos com as informações coletadas, destacando a importância da automação no processamento e armazenamento de dados coletados, que poderiam gerar conflito se não fossem apagados.

Conclusões

Usamos Web scraping com Python e BeautifulSoup de uma maneira eficiente para coletar dados de sites. Navegando pela estrutura HTML de uma página e extrair informações

específicas, como vimos com a coleta de vagas de emprego, permitecriar pipelines automáticos de dados úteis, desde análise de preços até monitoramento de informações públicas. Ao longo da atividade, foi possível entender como manipular elementos HTML e estruturar os dados coletados de forma que possam ser utilizados em análises futuras.

https://github.com/brenooabrantes/BT-Machine-Learning-Lamia/tree/main/CARD8

Referências

- https://docs.python-requests.org/en/latest/
- https://docs.python.org/3/library/os.html
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/