

Roteiro Aula Prática



WEB ANALYTICS

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: WEB ANALYTICS

Unidade: 2 - TÉCNICAS DE COLETA DE DADOS

Seção: 4 - Web scraping- Coleta de dados automatizada

OBJETIVOS

Definição dos objetivos da aula prática:

- Compreender o conceito de Web Analytics e sua importância para a análise de dados online.
- Explorar a técnica de Web Scraping como uma ferramenta para coleta automatizada de dados.
- Entender o funcionamento de Data Marketplaces e sua aplicação no contexto da análise de dados.
- Conhecer os perigos associados à raspagem de dados e as práticas éticas envolvidas.

SOLUÇÃO DIGITAL

- BeautifulSoup em Python (Ferramenta)

Beautiful Soup é um pacote Python para analisar documentos HTML e XML. Ele cria uma árvore de análise para páginas analisadas que podem ser usadas para extrair dados de HTML, o que é útil para web scraping.

PROCEDIMENTO/ATIVIDADE nº 1 (Digital)

ATIVIDADE PROPOSTA:

Implementando Aplicações com Web Scraping

Nesta atividade, você explorará o Web Scraping para coletar dados automaticamente de uma página da web. Além disso, entenderá o papel dos Data Marketplaces e os desafios éticos associados à raspagem de dados.

PROCEDIMENTOS PARA A REALIZAÇÃO DA ATIVIDADE:

Passo 1: Compreendendo o Web Scraping

- Pesquise e identifique uma página da web com dados públicos.

- Use uma biblioteca de Web Scraping (como BeautifulSoup em Python) para extrair informações relevantes dessa página.

- Exemplo:

Suponha que você queira extrair informações de preços de produtos de um site de comércio eletrônico.

Dica de execução:

Instalação do BeautifulSoup e Requests

Certifique-se de ter o Python instalado em seu sistema. Abra um terminal ou prompt de comando e execute os seguintes comandos:

```
pip install beautifulsoup4
```

```
pip install requests
```

Importante: Estes comandos instalarão o BeautifulSoup4 e a biblioteca Requests, que é frequentemente utilizada em conjunto com o BeautifulSoup para fazer requisições HTTP.

Escrever o Código de Web Scraping

Seguindo a premissa do exemplo acima, você está interessado em extrair preços de produtos de um site fictício chamado "example.com". Substitua isso pelo URL real do site que você deseja raspar.

```

import requests
from bs4 import BeautifulSoup

# URL do site alvo
url = 'https://www.example.com/produtos'

# Enviar uma solicitação GET para o site
response = requests.get(url)

# Verificar se a solicitação foi bem-sucedida (código 200)
if response.status_code == 200:
    # Analisar o conteúdo HTML da página usando BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')

    # Encontrar os elementos HTML que contêm informações sobre os preços
    # (Isso pode variar dependendo da estrutura específica do site)
    prices = soup.find_all('span', class_='price')

    # Imprimir os preços encontrados
    for price in prices:
        print(price.text)
else:
    print(f'Erro na solicitação: {response.status_code}')

```

Notas Importantes:

- Estrutura HTML: A estrutura HTML de cada site é única. Você precisará inspecionar o código-fonte do site de destino para identificar os seletores corretos para extrair os dados desejados.
- Headers da Solicitação: Alguns sites podem bloquear solicitações de bots. Em casos como esse, você pode adicionar cabeçalhos à solicitação para simular um navegador. Consulte a documentação do site para obter informações sobre cabeçalhos de solicitação.
- Ética e Legalidade: Certifique-se de ler e entender os termos de serviço do site. A raspagem de dados sem permissão pode violar os termos do site e ser ilegal.
- Lembre-se de que a raspagem de dados deve ser feita de maneira ética e respeitando os direitos do site. Este exemplo é apenas para fins educacionais.

Passo 2: Data Marketplaces

- Explore um Data Marketplace (por exemplo, Kaggle) em busca de conjuntos de dados relevantes para seu campo de interesse.
- Baixe um conjunto de dados e analise sua estrutura.

- Exemplo:

Escolha um conjunto de dados relacionado à análise de marketing online.

Passo 3: Perigos da Raspagem de Dados

- Pesquise e identifique casos em que a raspagem de dados resultou em questões éticas ou legais.
- Reflita sobre a importância da ética na coleta e uso de dados.

CHECKLIST:

- Concluiu a atividade de Web Scraping.
- Explorou um Data Marketplace e analisou um conjunto de dados.
- Compreendeu os perigos éticos associados à raspagem de dados.

RESULTADOS

Resultados de Aprendizagem:

Ao final desta atividade, espera-se que o aluno tenha desenvolvido habilidades práticas em Web Scraping, compreendido o funcionamento de Data Marketplaces e esteja ciente dos desafios éticos envolvidos na raspagem de dados.