Movie
Web Scraper

Lucas Broering dos Santos

Domínio do Problema

O domínio do problema escolhido é o de review de filmes. Envolve a coleta e apresentação de dados relacionados a filmes de dois sites famosos, Metacritic e The Movie Database. Existem algumas particularidades nesse domínio, a diversidade de dados e como os mesmos são apresentados, cada site tem uma metodologia diferente para avaliar os filmes. Existe também a questão que novos filmes são lançados todos os dias, então há atualização constante desses dados.

Entidades do mundo real envolvidas:

- Filmes
- Avaliações
- Usuários
- Plataformas de Avaliação

Domínio do Problema

Público Interessado

- Cinéfilos
- Críticos de Cinema
- Indústria do Entretenimento
- Desenvolvedores

Ao final do trabalho se espera criar um scraping eficiente que navegue por diversas páginas relacionadas a filmes e extraia informações relevantes. Fazendo a limpeza dos dados e armazenando eles em um formato acessível (json).

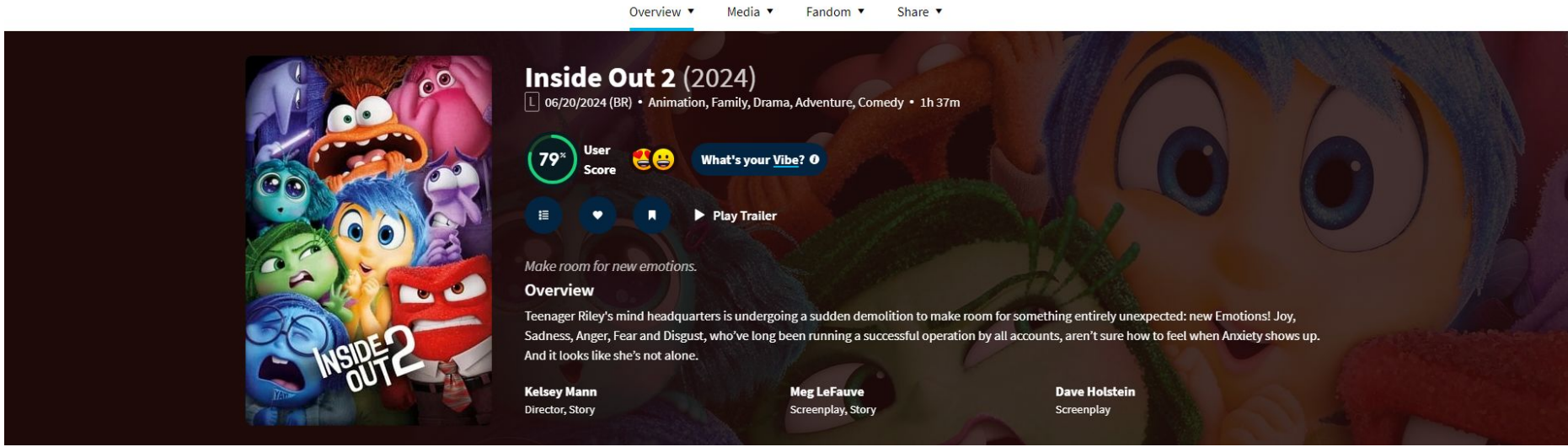
O código faz um trabalho de crawler para percorrer as páginas e extractor para extrair as informações da estrutura html.

Objetivo

Receber páginas html como entrada e gerar um json com as informações limpas e organizadas.

Exemplo dos dados de entrada

Foram utilizadas duas url diferentes no trabalho, do metacritic e The Movie Database (TMDB)

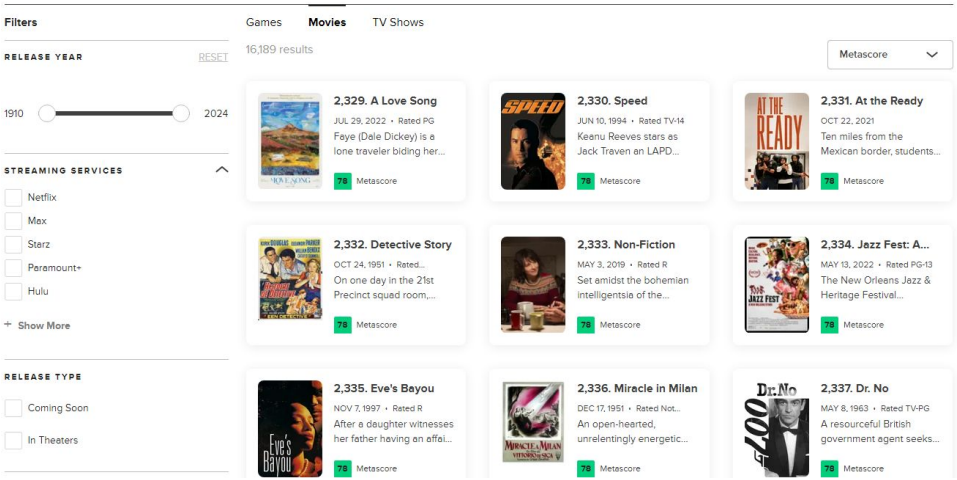


Exemplo de página do The Movie Database (TMDB).

Exemplo dos dados de entrada

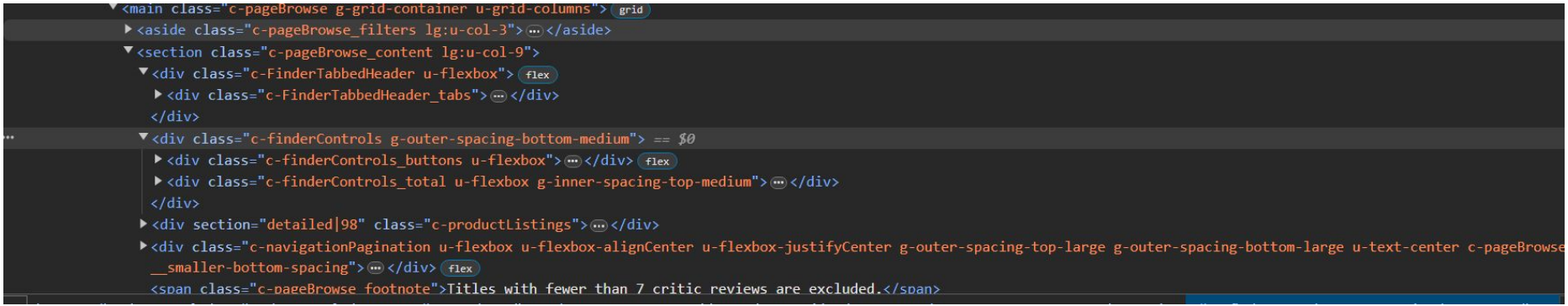
Exemplo da estrutura html da página do The Movie Database (TMDB).

Exemplo dos dados de entrada



Exemplo de página do Metacritic.

Exemplo dos dados de entrada



Exemplo da estrutura html da página do Metacritic.

Exemplo dos dados de saída


```
[
    {
        "ID": "1",
        "TITLE": "Dekalog (1988)",
        "LAUNCH YEAR": "Mar 22, 1996",
        "RATED": "TV-MA",
        "DESCRIPTION": "This masterwork by Krzysztof Kieślowski is one of the twentieth century's greatest achievements in visual storytelling. Orig
        "SCORE": "100/100",
        "URL": "https://www.metacritic.com/browse/movie/?releaseYearMin=1910&releaseYearMax=2024&page=1"
    },
    {
        "ID": "2",
        "TITLE": "The Leopard (re-release)",
        "LAUNCH YEAR": "Aug 13, 2004",
        "RATED": "PG",
        "DESCRIPTION": "Set in Sicily in 1860, Luchino Visconti's spectacular 1963 adaptation of Giuseppe di Lampedusa's international bestseller is
        "SCORE": "100/100",
        "URL": "https://www.metacritic.com/browse/movie/?releaseYearMin=1910&releaseYearMax=2024&page=1"
    },
    {
        "ID": "3",
        "TITLE": "The Godfather",
        "LAUNCH YEAR": "Mar 24, 1972",
        "RATED": "R"
```


Exemplo de saída dos dados extraídos no metacritic.

Exemplo dos dados de saída


```
[
    {
        "ID": "1",
        "TITLE": "Ariel",
        "CERTIFICATION": "K-12",
        "RELEASE": "21/10/1988 (FI)",
        "GENRES": "Drama,Comédia,Romance,Crime",
        "RUNTIME": "1h 13m",
        "DESCRIPTION": "Taisto Kasurinen trabalha numa mina de carvão que passa por graves problemas. Seu pai comete suicídio e ele acaba sendo pres
        "DIRECTOR": "Aki KaurismäkiDirector, Writer",
        "URL": "https://www.themoviedb.org/movie/2"
    },
    {
        "ID": "2",
        "TITLE": "Sombras no Paraíso",
        "CERTIFICATION": "S",
        "RELEASE": "17/10/1986 (FI)",
        "GENRES": "Drama,Comédia,Romance",
        "RUNTIME": "1h 14m",
        "DESCRIPTION": "Depois de perder seu amigo e colega de trabalho para um ataque cardíaco repentino, o solitário catador de lixo Nikander está
        "DIRECTOR": "Aki KaurismäkiDirector, Writer",
        "URL": "https://www.themoviedb.org/movie/3"
    },
```


Exemplo de saída dos dados extraídos no The Movie Database (TMDB).

Descrição sobre o trabalho

```python
def metacritic_scraper(pages):
    response = []

    while page_count < pages:

        url = f'https://www.metacritic.com/browse/movie/?releaseYearMin=1910&releaseYearMax=2024&page={page_count}'

        page = requests.get(url, headers=headers)

        soup = BeautifulSoup(page.text, 'html.parser')

        movies = soup.find_all("div", class_="c-finderProductCard_info u-flexbox-column")

        for movie in movies:
            title = movie.find("h3", class_="c-finderProductCard_titleHeading").get_text(strip=True)

            launch_year_span = movie.find("span", class_="u-text-uppercase")
            launch_year = launch_year_span.get_text(strip=True) if launch_year_span else "N/A"

            rated_span = movie.find("span", class_="u-text-capitalize")
            rated = rated_span.parent.get_text(strip=True) if rated_span else "N/A"

            description_div = movie.find("div", class_="c-finderProductCard_description")
            description = description_div.get_text(strip=True) if description_div else "N/A"

            score_div = movie.find("div", class_="c-siteReviewScore u-flexbox-column u-flexbox-alignCenter u-flexbox-justifyCenter g-text-bold c-siteR
            score = score_div.get_text(strip=True) if score_div else "N/A"
            score_formatted = score + "/100"

            data = {'ID': str(movie_count), 'TITLE': title[2:], 'LAUNCH YEAR': launch_year, 'RATED': rated[5:], 'DESCRIPTION': description, 'SCORE': s

            response.append(data)

            movie_count += 1

        page_count += 1

    print(f'Scraping complete - {movie_count}')

    return response
```

Ao lado está a parte de percorrer as páginas e fazer a extração dos dados significantes da estrutura html.

Descrição sobre o trabalho

```
def json_file(response):
    with open('INE5454 - Tópicos Especiais em Gerência de Dados/metacritic_data.json', 'w', encoding='utf-8') as json_file:
        json.dump(response, json_file, ensure_ascii=False, indent=4)
```

Essa função pega a response e transforma em um arquivo json, já organizado.

Movie Web Scraper

Tecnologias:

Utilizei python e as bibliotecas, BeautifulSoup, requests e json, para extrair o conteúdo da páginas, acessar elas e transformar em arquivo json, respectivamente.

Abordagem, e métodos:

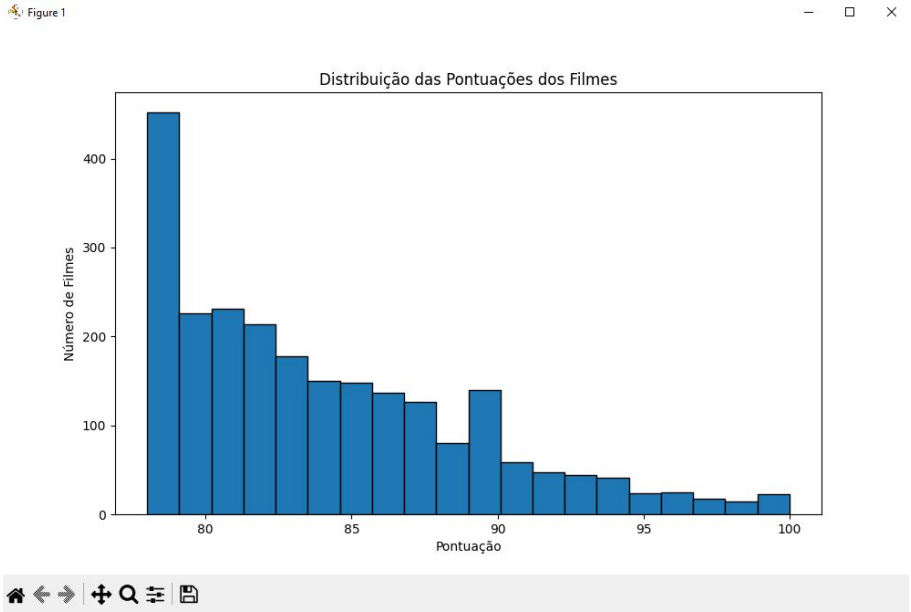
Os dados foram coletados de páginas html. Pessoas da indústria cinematográfica ou desenvolvedores, seriam os principais interessados nos dados extraídos. Dados extraídos em inglês/português em formato html.

Movie Web Scraper

Atributos coletados:

- TITLE - Qualitativo
- LAUNCH YEAR - Qualitativo
- RATED - Qualitativo
- DESCRIPTION - Qualitativo
- SCORE - Qualitativo
- URL - Qualitativo

Análise dos resultados finais



Exemplo de gráfico que pode ser plotado utilizando os dados. Relaciona o número de filmes com a pontuação.

Conclusão

Fazer o trabalho foi bem satisfatório para mim, nunca tinha mexido com web scraping e aprendi bastante. Acho que o mais difícil foi entender a estrutura html das url e fazer a extração corretamente, caminhando pela árvore html. Com certeza que se feito em maior escala, com mais url, seria um ótimo produto para o mercado cinematográfico, serviria fazer análises do que pode dar certo ou nao no mundo do cinema, como analisar tendências de gêneros dos filmes.